

Федеральное государственное учреждение «Федеральный научный центр
Научно-исследовательский институт системных исследований
Российской академии наук»
(ФГУ ФНЦ НИИСИ РАН)

ТРУДЫ НИИСИ РАН

ТОМ 9 №5

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ:
ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ

МОСКВА
2019

Редакционный совет ФГУ ФНЦ НИИСИ РАН:

В.Б. Бетелин (председатель),
Е.П. Велихов, В.А. Галатенко, В.Б. Демидович (отв. секретарь),
Ю.В. Кузнецов (отв. секретарь), Б.В. Крыжановский, А.Г. Кушниренко,
А.Г. Мадера, М.В. Михайлюк, В.Я. Панченко, В.П. Платонов, В.Н. Решетников

Главный редактор журнала:

В.Б. Бетелин

Научный редактор номера:

А.И. Грюнталь

Тематика номера:

Развитие суперкомпьютерных технологий и оптимизация суперкомпьютерных вычислений, моделирование физических процессов в микро- и нанoeлектронике, физические процессы и их моделирование, математическое моделирование и визуализация, математическое моделирование распределенных систем, методические материалы

Журнал публикует оригинальные статьи по следующим областям исследований:

математическое и компьютерное моделирование, обработка изображений, визуализация, системный анализ, методы обработки сигналов, информационная безопасность, информационные технологии, высокопроизводительные вычисления, оптико-нейронные технологии, микро- и нанoeлектроника, математические исследования и вопросы численного анализа, история науки и техники

The topic of the issue:

Development of supercomputer technologies and optimization of supercomputer computations, modeling of physical processes in micro- and nano electronics, physical processes and its modeling, mathematical modeling and visualization, mathematical modeling of distributed systems, guidance materials

The Journal publishes novel articles on the following research areas: mathematical and computer modeling, image processing, visualization, system analysis, signal processing, information security, information technologies, high-performance computing, optical-neural technologies, micro- and nanoelectronics, mathematical researches and problems of numerical analysis, history of science and of technique

Заведующий редакцией: В.Е. Текунов

Издатель: ФГУ ФНЦ НИИСИ РАН,
117218, Москва, Нахимовский проспект 36, к. 1

СОДЕРЖАНИЕ

I. РАЗВИТИЕ СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ОПТИМИЗАЦИЯ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛЕНИЙ

<i>А.А. Рыбаков.</i> Векторизация нахождения пересечения объемной и поверхностной сеток для микропроцессоров с поддержкой AVX-512.....	5
<i>О.С. Аладышев, А.В. Баранов, Д.Ю. Дербышев.</i> Методы и алгоритмы обеспечения прохождения пользовательских заданий с заданным уровнем обслуживания.	15
<i>А.В. Баранов, М.О. Васильев, Е.А. Киселёв, В.Ф. Огарышев.</i> Формирование статистических профилей суперкомпьютерных приложений.....	23
<i>А.В. Баранов, Е.А. Киселёв.</i> Интеграция систем управления заданиями SLURM и СУППЗ.....	29

II. МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ В МИКРО- И НАНОЭЛЕКТРОНИКЕ

<i>Н.В. Масальский.</i> Полуклассическая 3D модель КНИ МОП нанотранзистора с ультратонкой рабочей областью	36
<i>А.А. Воробьев, А.Г. Мадера, А.О. Шаманов.</i> Анализ теплового и вибрационного режимов конструкции электронного модуля с теплоотводами в виде гибкого кондуктивного теплоотвода и единой теплопроводной металлической пластины.....	42

III. ФИЗИЧЕСКИЕ ПРОЦЕССЫ И ИХ МОДЕЛИРОВАНИЕ

<i>И.В. Афанаскин, А.В. Королев, А.А. Глушаков, П.В. Ялов.</i> Математические модели для анализа разработки нефтяных месторождений методом ячеек заводнения при забойном и при пластовом давлении ниже давления насыщения без газовой шапки.....	46
<i>А.А. Колеватов, А.М. Гуацинтов, А.А. Глушаков, А.А. Егоров.</i> Особенности карбонатных трещиноватых нефтенасыщенных коллекторов по данным геофизических исследований скважин.....	54
<i>А.А. Колеватов, Ю.М. Штейнберг, Ю.Б. Чен-лен-сон, А.Г. Дяченко.</i> Классификационные признаки типов нефтенасыщенных коллекторов по результатам гидродинамических исследований.....	59
<i>А.А. Колеватов, Ю.Б. Чен-лен-сон, Ю.М. Штейнберг, И.В. Афанаскин.</i> Интегрированный анализ гидродинамических и геофизических скважинных исследований с целью обнаружения невыработанных зон	63
<i>П.В. Крыганов, И.В. Афанаскин, С.Г. Вольпин.</i> Анализ эффективности применения горизонтальных скважин с помощью гидродинамических исследований.....	71
<i>М.Л. Сидоров, В.А. Пронин, И.В. Афанаскин, А.В. Королев, П.В. Ялов.</i> Разработка параллельного генератора неструктурированных сеток для моделей нефтегазовых месторождений с адаптацией к объектам геологических моделей.....	81

В.А. Юдин, С.Г. Вольпин, И.В. Афанаскин, Н.П. Ефимова. Возможность выявления дизъюнктивных нарушений на нефтяном месторождении по характеру обводнения скважин и данным трассерных исследований.....91

IV. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ВИЗУАЛИЗАЦИЯ

А.В. Мальцев, М.А. Торгашев. Распределенное моделирование глубины резкости при визуализации трехмерных сцен на GPU100
М.В. Михайлюк, И.Н. Чередниченко. Визуализация векторных полей цифровой модели ядра.....105
Е.В. Страшнов, Л.А. Финагин. Определение коллизий аппроксимирующих сфер с рельефом местности.....111

V. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕННЫХ СИСТЕМ

А.Б. Бетелин, И.Б. Егорычев, А.А. Прилипко, Г.А. Прилипко, С.Г. Романюк, Д.В. Самборский. Настройка и оптимизация системы ввода-вывода в среде виртуализации GNU Linux/QEMU/KVM/Libvirt.....119

VI. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

А.А. Асонов, А.Н. Годунов. Подготовка статей для журнала “Труды НИИСИ РАН”.....130

Векторизация нахождения пересечения объемной и поверхностной сеток для микропроцессоров с поддержкой AVX-512

А.А. Рыбаков

МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, rybakov@jssc.ru

Аннотация. Векторизация программного кода является одной из наиболее важных низкоуровневых оптимизаций, позволяющих существенно повысить производительность суперкомпьютерных приложений. Эффект от векторизации кода тем заметнее, чем больше длина вектора. Набор инструкций AVX-512, поддержанный в современных микропроцессорах Intel, предназначен для работы с 512-битными векторами и позволяет проводить векторизацию предикатного кода. В данной работе рассмотрен практический подход к векторизации сложного программного контекста, реализующего нахождение пересечения объемной декартовой и поверхностной неструктурированной расчетных сеток. Эта задача используется в расчетах обтекания тел со сложной геометрией, выполняемых с помощью метода погруженной границы, при этом в качестве объемной декартовой сетки может быть использована как равномерная сетка с одинаковыми ячейками, так и локально-измельчающаяся сетка. Приведено математическое решение поставленной задачи, выполнена ее реализация, а также показано, что векторизация с помощью набора инструкций AVX-512 позволяет достичь ускорения на микропроцессоре Intel Xeon Phi KNL более чем в 6,5 раз по сравнению с не векторизованной версией на вещественных данных одинарной точности.

Ключевые слова. Векторизация, Intel Xeon Phi KNL, AVX-512, объемная декартова сетка, поверхностная неструктурированная сетка, плоский цикл, функции-интринсики.

Введение

При численном решении задач газовой динамики часто приходится сталкиваться с телами, обладающими сложной геометрией. Для таких тел построение согласованной расчетной сетки может быть крайне трудозатратной задачей. Альтернативой в данном случае является использование метода погруженной границы [1]. Данный метод позволяет использовать для расчетов несогласованную сетку и даже простую декартову сетку, что сильно упрощает выполнение расчетов. Единственным тонким моментом метода является выполнение граничных условий на сложной границе, которое достигается путем модификации решаемой системы уравнений. Можно выделить два подхода в методах погруженной границы, различающихся по способу выполнения расчетов на границе: задание граничных условий посредством внешних (или источниковых) членов [2] и методы фиктивных ячеек [3].

Рассмотрим немного подробнее метод погруженной границы с использованием фиктивных ячеек. Пусть в некоторой области пространства расположено тело, ограниченное сложной границей, данная граница, представлена неструктурированной поверхностной расчетной сеткой, ячейки которой являются треугольниками. Пусть

также в охватывающей тело области пространства построена объемная расчетная сетка, ячейки которой могут быть разделены на следующие три основных класса. Внешними ячейками будем называть те ячейки, которые целиком лежат вне тела. Внутренние ячейки лежат целиком внутри тела, все остальные ячейки пересекают границу тела и являются граничными. В методе фиктивных ячеек из граничных ячеек выделяются ячейки, для которых меньшая часть находится вне тела, а большая – внутри тела. Такие ячейки называются фиктивными. На каждой итерации расчетов для фиктивных ячеек требуется выполнить аппроксимацию газодинамических величин (плотность, давление, вектор скорости), чтобы данные фиктивные ячейки могли быть использованы для определения потоков между ними и соседними с ними граничными и внешними ячейками [4]. Таким образом, классификация ячеек объемной сетки является неотъемлемой частью метода погруженной границы. В условиях изменяющейся геометрии обтекаемого тела задача классификации ячеек занимает существенную долю вычислений и должна быть оптимизирована.

Для получения качественного решения вблизи тела со сложной геометрией требуется использовать мелкие объемные

сетки, а вблизи границы тела выполнять также локальное измельчение граничных ячеек. На рис. 1 продемонстрированы варианты построения множества граничных и фиктивных ячеек для тела со сложной геометрией, в которых использованы объемные сетки с разным количеством ячеек. При использовании мелких объемных сеток количество граничных ячеек сильно возрастает и задача их классификации становится критичной по времени выполнения. Основным этапом задачи

классификации ячеек объемной сетки является задача об определении пересечения множества ячеек объемной сетки с множеством ячеек поверхностной сетки. В данной работе в качестве объемной сетки рассматривается декартова сетка, ребра ячеек которой параллельны осям координат.

Для решения задачи классификации ячеек объемной сетки требуется решить частную задачу определения пересечения треугольника и прямоугольного параллелепипеда в пространстве.

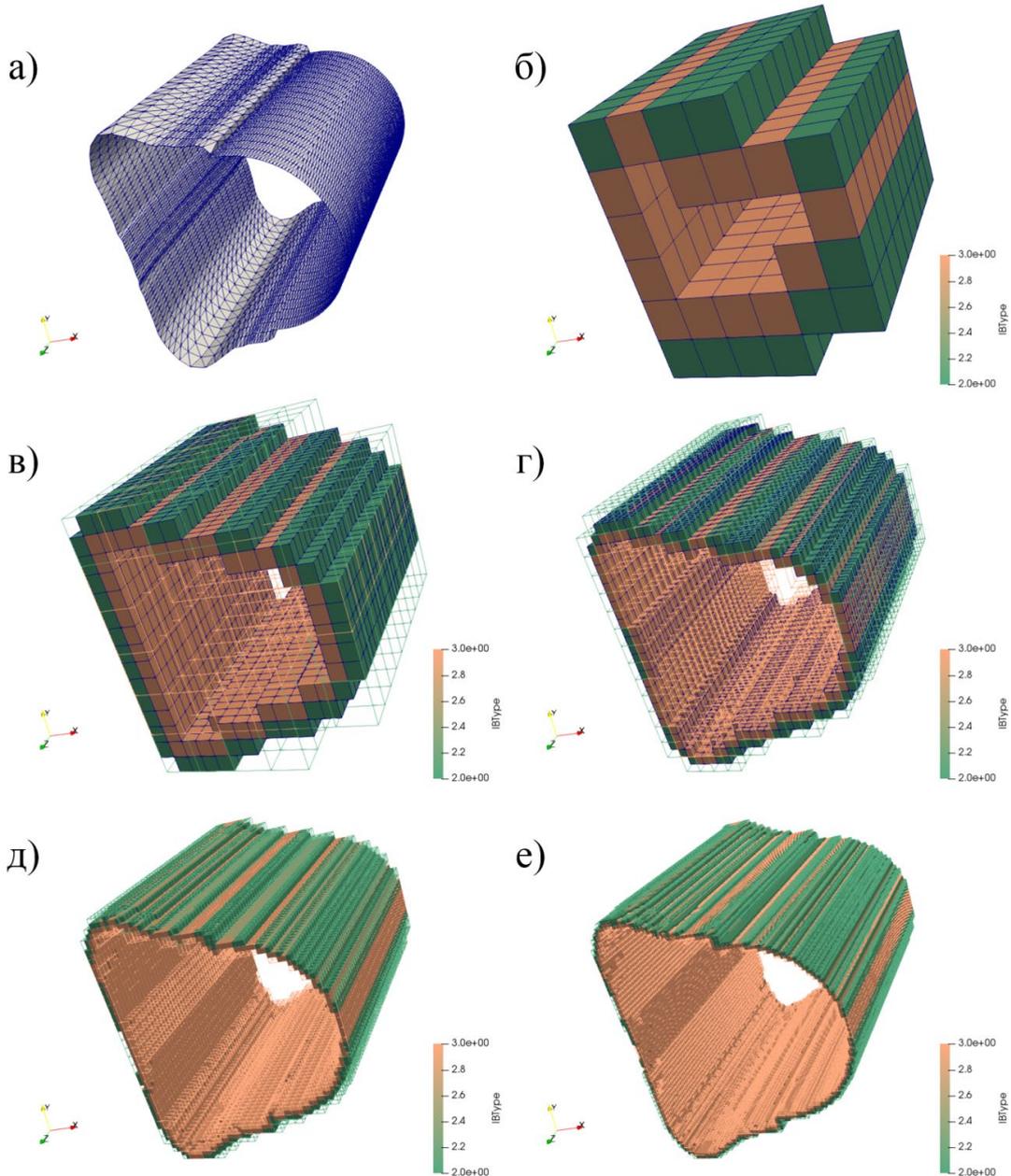


Рисунок 1. Иллюстрация пересечения поверхностной сетки и объемных сеток с разным количеством ячеек.

На рисунках цветами обозначены только граничные и фиктивные ячейки объемной сетки (граничные – зеленые, фиктивные – бежевые). а) Исходная геометрия поверхностной сетки. б) Размер объемной сетки $10 \times 10 \times 10$. в) Размер объемной сетки $25 \times 25 \times 25$. г) Размер объемной сетки $50 \times 50 \times 50$. д) Размер объемной сетки $100 \times 100 \times 100$. е) Размер объемной сетки $150 \times 150 \times 150$.

Задача о пересечении треугольника и прямоугольного параллелепипеда

В данном разделе рассмотрим математическую постановку и метод решения задачи об определении пересечения треугольника и прямоугольного параллелепипеда в пространстве.

Пусть треугольник задан тремя точками: $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$, $C(x_C, y_C, z_C)$. Тогда координаты любой точки $P(x, y, z)$, находящейся внутри треугольника, можно представить следующим образом:

$$\begin{cases} x = x_A + (x_B - x_A)\alpha + (x_C - x_A)\beta \\ y = y_A + (y_B - y_A)\alpha + (y_C - y_A)\beta \\ z = z_A + (z_B - z_A)\alpha + (z_C - z_A)\beta \end{cases} \quad (1)$$

где $\alpha \geq 0$, $\beta \geq 0$, $\alpha + \beta \leq 1$.

Геометрическим местом точек прямоугольного параллелепипеда является множество точек $P(x, y, z)$, координаты которых удовлетворяют следующей системе неравенств:

$$\begin{cases} x_l \leq x \leq x_h \\ y_l \leq y \leq y_h \\ z_l \leq z \leq z_h \end{cases} \quad (2)$$

Для установления факта пересечения треугольника и прямоугольного параллелепипеда нужно определить, имеет ли решение приведенная ниже система неравенств относительно α и β :

$$\begin{cases} x_l \leq x_A + (x_B - x_A)\alpha + (x_C - x_A)\beta \leq x_h \\ y_l \leq y_A + (y_B - y_A)\alpha + (y_C - y_A)\beta \leq y_h \\ z_l \leq z_A + (z_B - z_A)\alpha + (z_C - z_A)\beta \leq z_h \\ \alpha \geq 0 \\ \beta \geq 0 \\ \alpha + \beta \leq 1 \end{cases} \quad (3)$$

Для подобных систем неравенств существует много различных способов решения, описание которых можно найти, например, в [5]. В нашем случае система довольно простая, содержит две переменные, и к ней может быть применен метод свертывания конечных систем линейных неравенств, описанный в [6].

Для решения методом свертывания преобразуем систему неравенств так, чтобы она содержала неравенства только вида

$k_\alpha \alpha + k_\beta \beta + k \leq 0$. После выполнения всех преобразований система неравенств примет следующий вид:

$$\begin{cases} (x_B - x_A)\alpha + (x_C - x_A)\beta + (x_A - x_h) \leq 0 \\ (x_A - x_B)\alpha + (x_A - x_C)\beta + (x_l - x_A) \leq 0 \\ (y_B - y_A)\alpha + (y_C - y_A)\beta + (y_A - y_h) \leq 0 \\ (y_A - y_B)\alpha + (y_A - y_C)\beta + (y_l - y_A) \leq 0 \\ (z_B - z_A)\alpha + (z_C - z_A)\beta + (z_A - z_h) \leq 0 \\ (z_A - z_B)\alpha + (z_A - z_C)\beta + (z_l - z_A) \leq 0 \\ -1 \cdot \alpha + 0 \cdot \beta \leq 0 \\ 0 \cdot \alpha + (-1)\beta \leq 0 \\ \alpha + \beta + (-1) \leq 0 \end{cases} \quad (4)$$

Так как система содержит всего две переменные, то после выполнения одного шага свертывания (или деформации системы) она превратится в систему неравенств относительно одной переменной, проверка разрешимости которой не представляет труда. Будем выполнять деформацию системы с целью исключить из нее переменную α . Для этого составим новую систему, в которую войдут все неравенства исходной системы вида $k_\beta \beta + k \leq 0$, а каждая пара неравенств

$$\begin{cases} k_\alpha^1 \alpha + k_\beta^1 \beta + k^1 \leq 0 \\ k_\alpha^2 \alpha + k_\beta^2 \beta + k^2 \leq 0 \end{cases} \quad (5)$$

где $k_\alpha^1 < 0$, а $k_\alpha^2 > 0$ войдет в деформированную систему в виде

$$(k_\beta^1 k_\alpha^2 - k_\beta^2 k_\alpha^1) \beta + (k^1 k_\alpha^2 - k^2 k_\alpha^1) \leq 0 \quad (6)$$

Так как исходная система содержит 9 уравнений, по крайней мере одно из которых имеет нулевой коэффициент при переменной α , а из оставшихся восемь половина коэффициентов при переменной α неотрицательно, а половина отрицательно, то деформированная система будет содержать не более 17 уравнений.

Оптимизация поиска пересечения сеток

Ячейки объемной сетки являются прямоугольными параллелепипедами. Ячейки поверхностной сетки являются треугольниками. Для того, чтобы найти все ячейки объемной сетки, которые пересекаются с поверхностью, в общем

случае необходимо проверить факт пересечения каждой объемной ячейки с каждой поверхностной ячейкой. В общем случае данная процедура является очень затратной, так как сетки могут содержать большое количество ячеек. Вместо этого для каждой ячейки поверхностной сетки сначала будем определять диапазон ячеек объемной сетки, с которыми в принципе возможно пересечение.

Пусть изначально объемная сетка, описывающая область $[X_l, X_h] \times [Y_l, Y_h] \times [Z_l, Z_h]$, размера $S_x \times S_y \times S_z$, разделена на одинаковые базовые ячейки (которые впоследствии могут быть разделены на более мелкие) размера $s_x \times s_y \times s_z$, где

$$\begin{aligned} n_x &= \frac{S_x}{s_x} \\ n_y &= \frac{S_y}{s_y} \\ n_z &= \frac{S_z}{s_z} \end{aligned} \quad (7)$$

(n_x – количество базовых ячеек по направлению x , аналогично для направлений y и z). Таким образом, объемная сетка представлена трехмерным массивом базовых ячеек, координаты которых ($x_i, x_h, y_l, y_h, z_l, z_h$) могут быть вычислены по индексам (i, j, k) в данном трехмерном массиве:

$$\begin{aligned} x_l(i) &= X_l + is_x \\ x_h(i) &= X_h + (i+1)s_x \\ y_l(j) &= Y_l + js_y \\ y_h(j) &= Y_h + (j+1)s_y \\ z_l(k) &= Z_l + ks_z \\ z_h(k) &= Z_h + (k+1)s_z \end{aligned} \quad (8)$$

Для треугольника, являющегося ячейкой поверхностной сетки, вершинами которого являются точки $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$, $C(x_C, y_C, z_C)$, можно найти охватывающий прямоугольный параллелепипед, координаты которого равны

$$\begin{aligned} \tilde{x}_l &= \min(x_A, x_B, x_C) \\ \tilde{x}_h &= \max(x_A, x_B, x_C) \\ \tilde{y}_l &= \min(y_A, y_B, y_C) \\ \tilde{y}_h &= \max(y_A, y_B, y_C) \\ \tilde{z}_l &= \min(z_A, z_B, z_C) \\ \tilde{z}_h &= \max(z_A, z_B, z_C) \end{aligned} \quad (9)$$

Если треугольник пересекает некоторую объемную ячейку, то его охватывающий

прямоугольный параллелепипед также пересекает эту ячейку. То есть для определения пересечения поверхностной ячейки со всеми ячейками объемной сетки достаточно проверить только те ячейки, с которыми пересекается охватывающий прямоугольный параллелепипед рассматриваемого треугольника. Так как координаты анализируемых параллелепипедов могут быть записаны явно, то можно вычислить диапазоны индексов базовых ячеек, которые требуется проверить на вопрос пересечения с треугольником. Факт пересечения по координате x двух отрезков $[x_l(i), x_h(i)]$ и $[\tilde{x}_l, \tilde{x}_h]$ можно записать в виде системы из двух неравенств

$$\begin{cases} x_l(i) \leq \tilde{x}_h \\ x_h(i) \geq \tilde{x}_l \end{cases} \quad (10)$$

Преобразовав данную систему, а также выполнив аналогичные действия для координат по y и z , получим итоговые диапазоны индексов базовых ячеек

$$\begin{cases} \frac{\tilde{x}_l - X_l}{s_x} - 1 \leq i \leq \frac{\tilde{x}_h - X_l}{s_x} \\ \frac{\tilde{y}_l - Y_l}{s_y} - 1 \leq j \leq \frac{\tilde{y}_h - Y_l}{s_y} \\ \frac{\tilde{z}_l - Z_l}{s_z} - 1 \leq k \leq \frac{\tilde{z}_h - Z_l}{s_z} \end{cases} \quad (11)$$

Данный диапазон индексов содержит малую долю всех ячеек объемной сетки. В данном случае не требуется проводить анализ ускорения от этого преобразования, так как при его отсутствии поиск пересечения сеток просто нежизнеспособен.

Заметим, что при измельчении объемной сетки алгоритм поиска пересечения изменяется не сильно, так как диапазоны индексов базовых ячеек не меняются, однако кроме базовых ячеек обязательными для проверки становятся также все их дочерние ячейки.

Набор инструкций AVX-512

Современные микропроцессоры Intel начиная с Intel Xeon Phi KNL [7] и Intel Xeon Skylake поддерживают набор инструкций AVX-512 – 512-битное расширение AVX-инструкций. Однако основное отличие набора инструкций AVX-512 от AVX не в длине вектора, благодаря которому пиковая

производительность микропроцессора возрастает вдвое. Набор инструкций AVX-512 поддерживает выборочное применение операций к элементам векторов, которое обеспечивается с помощью специальных масок. Маски являются объединением битов, каждый из которых управляет записью результатов векторных операций в регистр назначения. Благодаря использованию масочных операций стало возможным векторизовать широкий спектр программного кода, записанного в предикатном виде. Предикатный режим исполнения аппаратно поддержан, например, в архитектуре «Эльбрус» [8], и с его помощью возможно проведение множества оптимизаций, позволяющих генерировать параллельный на уровне отдельных инструкций код. Помимо наличия предикатных команд набор инструкций AVX-512 обладает другими особенностями, позволяющими создавать эффективный параллельный код. Это операции множественного обращения в память с произвольными смещениями от базового адреса, комбинированные операции (позволяющие за одну операцию производить поэлементное вычисление выражений вида $\pm a \cdot b \pm c$), огромное разнообразие операций перестановки элементов векторов (perm, shuf и другие), реализация поэлементных логических функций от трех переменных и многие другие.

Для упрощения использования инструкций из набора AVX-512 для компилятора icc реализована библиотека функций-интринсиков [9, 10], которые определены в заголовочном файле `immintrin.h`. Данные функции являются обертками над инструкциями AVX-512, оперируют встроенными типами векторных данных (`_m512`, `_m512i`, `_m512d` и другими), и могут быть использованы в программах, написанных на языке C. Таким образом, предоставлена возможность напрямую использовать инструкции AVX-512 в коде, не прибегая к ассемблерным вставкам.

Появление набора инструкций AVX-512 вызвало большой интерес со стороны исследователей и разработчиков программного обеспечения, в настоящее время ведутся работы по их использованию для оптимизации приложений из разных научных областей [11]. Можно отметить, например, работы по векторизации ядра программного кода LAMMPS [12], операций

с разреженными матрицами [13] и матрицами специального вида [14]. В работе [15] описан подход векторизации гнезда циклов на примере построения множества Мандельброта. Можно встретить работы, в которых описывается применение векторизации с использованием набора инструкций AVX-512 для других приложений из широкого круга, начиная от задач ядерной физики [16], и численного решения уравнений мелкой воды [17,18] и заканчивая задачами дискретной математики, включая векторизацию задач сортировки [19] или генератора псевдослучайных чисел [20].

В реальных приложениях компилятор зачастую не способен определить пригодный для векторизации контекст или отказывается от векторизации ввиду низкой ожидаемой эффективности. Причиной тому может служить невозможность доказать отсутствие зависимостей в коде, обилие условий и команд передачи управления, использование невыровненных данных. Однако небольшое участие программиста может привести к векторизации сложного контекста, используя предикатное представление кода и замену скалярных операций векторными. В следующем разделе описано применение данного подхода для векторизации задачи пересечения расчетных сеток.

Векторизация вычислений

Рассмотрим реализацию функции `tri_box_intersect(xa ya, za, xb, yb, zb, xc, yc, zc, xl, xh, yl, yh, zl, zh) → int`, анализирующую наличие пересечения треугольника и прямоугольного параллелепипеда. Функция возвращает 1, если пересечение есть, и 0, если пересечения нет. Логика работы функции следующая. Сначала коэффициенты системы неравенств (4) заносятся в двумерный массив коэффициентов `b[vec][3]`, где `vec` (`basic equations count`) – количество исходных неравенств системы (в нашем случае 9). Затем выполняется один шаг свертывания системы с одновременным поиском множества решения для переменной β . Перед началом свертывания множество допустимых значений для переменной β принимается в виде отрезка $[0, 1]$ ($lo = 0, hi = 1$). По мере свертывания системы неравенств (4) происходит сокращение данного множества решений. Если на каком-то этапе свертывания множество решений обращается в пустое ($lo > hi$), то функция

заканчивает работу и возвращает 0. Если после выполнения всех действий свертывания множество решений осталось ненулевым, то это означает наличие

пересечения, и функция возвращает 1. Реализация свертывания системы уравнений с помощью метода из [6] представлена на рис. 2.

```

01 for (i = 0; i < bec; i++)
02 {
03     bi0 = b[i][0];
04
05     if (bi0 == 0.0)
06     {
07         if (!upgrade(b[i][1], b[i][2], &lo, &hi))
08         {
09             return 0;
10         }
11     }
12     else
13     {
14         for (j = i + 1; j < bec; j++)
15         {
16             if (bi0 * b[j][0] < 0.0)
17             {
18                 f0 = bi0 * b[j][1] - b[j][0] * b[i][1];
19                 f1 = bi0 * b[j][2] - b[j][0] * b[i][2];
20
21                 if (bi0 < 0.0)
22                 {
23                     f0 = -f0;
24                     f1 = -f1;
25                 }
26
27                 if (!upgrade(f0, f1, &lo, &hi))
28                 {
29                     return 0;
30                 }
31             }
32         }
33     }
34 }
35
36 return 1;

```

Рисунок 2. Исходная реализация свертывания системы линейных неравенств для определения пересечения треугольника и прямоугольного параллелепипеда.

Получившийся код можно охарактеризовать как имеющий сложное управление, уровень вложенности конструкций в нем достигает 5 (for-if-for-if-if), к тому же участок содержит 3 выхода из функции.

Функция `upgrade`, которая вызывается из кода на рис. 2, предназначена для обновления текущего множества допустимых значений для переменной β с учетом нового полученного ограничения вида $k_\beta \beta + k \leq 0$, коэффициенты которого передаются в первом и втором параметрах.

Текущее множество решений является отрезком с границами, хранящимися в переменных `lo` и `hi`, и в зависимости от знака коэффициента k_β одна из этих границ внутри вызова функции `upgrade` может измениться (граница `lo` может увеличиться, либо граница `hi` может уменьшиться).

Если после обновления множества решений оно оказывается пустым (нижняя граница становится больше верхней), то

функция `upgrade` возвращает 0, в противном случае она возвращает 1.

После рассмотрения реализации функции `tri_box_intersect` можно перейти к векторизации вычислений. Для анализа пересечения двух сеток необходимо вызывать функцию `tri_box_intersect` многократно с разными наборами входных параметров. Для оптимизации этого процесса реализуем функцию `tri_box_intersect_16`, объединяющую внутри себя обработку 16 вызовов функции `tri_box_intersect` (используется объединение 16 вызовов, так как это совпадает с количеством элементов вещественных данных одинарной точности в одном векторе `_m512`, `VEC_WIDTH = 16`). В первом приближении реализация функции `tri_box_intersect_16` представлена на рис. 3. Конечно в реальной задаче требуется обрабатывать миллионы вызовов, однако они могут быть разбиты на группы по 16 и обработаны с помощью функции `tri_box_intersect_16`, поэтому остановимся подробнее на векторизации данной функции.

```

01 void tri_box_intersect_16(float *xa, float *ya, float *za,
02                          float *xb, float *yb, float *zb,
03                          float *xc, float *yc, float *zc,
04                          float *xl, float *xh,
05                          float *yl, float *yh,
06                          float *zl, float *zh,
07                          int *r)
08 {
09     for (int i = 0; i < 16; i++)
10     {
11         r[i] = tri_box_intersect(xa[i], ya[i], za[i],
12                                xb[i], yb[i], zb[i],
13                                xc[i], yc[i], zc[i],
14                                xl[i], xh[i], yl[i], yh[i], zl[i], zh[i]);
15     }
16 }

```

Рисунок 3. Исходная реализация функции, объединяющей 16 вызовов функции `tri_box_intersect`.

Для векторизации функции `tri_box_intersect_16` следует выполнить подстановку тела функции `tri_box_intersect` в место ее вызова. После выполнения данного преобразования получаем программный контекст, содержащий сложное управление и в частности гнездо из трех вложенных циклов, не считая условий. Данный контекст не может быть векторизован компилятором автоматически, поэтому будем проводить его векторизацию в ручном режиме с использованием функций-интринсиков, которые позволяют напрямую использовать инструкции AVX-512 в синтаксисе языка программирования C без использования ассемблера.

Заметим, что между итерациями внешнего цикла (рис. 3, строка 09) отсутствуют зависимости, то есть цикл является плоским, и его итерации могут быть выполнены в любом порядке, в том числе и одновременно. Такие циклы поддаются векторизации путем перевода тела в предикатное представление и заменой скалярных инструкций на векторные. Наиболее тонким местом при векторизации тела плоского цикла являются условия, то есть наличие конструкций `if-else` [21]. Альтернативные ветви таких конструкций должны быть объединены в предикатном коде под противоположными предикатами. Наличие большого количества условных операторов в исходном коде порождает множество инструкций под нулевыми предикатами в результирующем коде, что негативно сказывается на производительности. Для уменьшения количества условных операторов, можно использовать математические тождества, заменяющие условные конструкции на команды, имеющие векторные аналоги в наборе инструкций AVX-512. Для данных целей хорошо подходят такие векторные команды как `abs`, `min`, `max`, `blend`, `avg` и

другие. Например в рассматриваемом коде вычисление значений `f0` и `f1` в строках 18-25 на рис. 2 с учетом условия $b[i][0] * b[j][0] < 0$ может быть заменено на следующее:

$$\begin{aligned}
 f0 &= \text{fabs}(b[i][0]) * b[j][1] \\
 &\quad + \text{fabs}(b[j][0]) * b[i][1] \\
 f1 &= \text{fabs}(b[i][0]) * b[j][2] \\
 &\quad + \text{fabs}(b[j][0]) * b[i][2]
 \end{aligned}$$

Похожие трудности вызывает условие `if` в строке 05 на рис 2. Данное условие имеет альтернативную ветку, содержащую цикл. Слияние данных двух ветвей снижает производительность результирующего кода, поэтому в данном случае выгодно применить расщепление внешнего цикла по конструкции `if-else` (в [22] данное преобразование фигурирует под названием `loop distribution`). При этом образуются два гнезда циклов, каждое из которых может быть векторизовано независимо. Заметим, что описанное преобразование в общем смысле не является эквивалентным, так как обе ветки условия `if-else`, а значит и тела образовавшихся циклов содержат выходы из функции, выполнение расщепления цикла может изменить условие, провоцирующее выход из функции. По этой причине компилятор не способен выполнить данное преобразование автоматически. Однако с точки зрения результата функции данное преобразование корректно, поэтому мы его и применяем.

Отметим еще один крайне положительный момент в рассматриваемом программном контексте. Условием выхода из вложенных циклов является достижение индуктивной переменной значения `bes`. Данное значение является константой, а значит не зависит от номера итерации, поэтому в векторный код это условие может быть перенесено без изменений. В случае зависимости условия выхода из цикла от

номера итерации само условие должно быть также векторизовано, что может привести к потерям производительности, причины которых описаны в [23] на примере векторизации сортировки Шелла. Однако в рассматриваемом коде проблем

векторизации циклов с нерегулярным количеством итераций нет. На рис. 4 приведен получившийся предикатный код для функции `tri_box_intersect_16`, а также схематично показана его трансформации в векторный аналог.

<pre>float b[bec][3][VEC_WIDTH]; <инициализация b>; for (w = 0; w < VEC_WIDTH; w++) r[w] = 1; for (w = 0; w < VEC_WIDTH; w++) { lo = 0.0; hi = 1.0; for (i = 0; i < bec; i++) { upgrade(b[i][0][w] == 0.0, b[i][1][w], b[i][2][w], &lo, &hi); } for (i = 0; i < bec; i++) { bi0 = b[i][0][w]; abi0 = fabs(bi0); for (j = i + 1; j < bec; j++) { bj0 = b[j][0][w]; abj0 = fabs(bj0); upgrade(bi0 * bj0 < 0.0, abi0 * b[j][1][w] + abj0 * b[i][1][w], abi0 * b[j][2][w] + abj0 * b[i][2][w], &lo, &hi); if (lo > hi) break; } if (lo > hi) break; } if (lo > hi) r[w] = 0; } </pre>	<div style="border: 1px solid gray; padding: 2px; display: inline-block;">последовательный код</div>	<pre>__m512 b[bec][3]; <инициализация b>; __mm512_store_epi32(r, __mm512_set1_epi32(1)); __m512 lo = z0; __m512 hi = z1; for (i = 0; i < bec; i++) { upgrade(__mm512_cmpeq_ps_mask(b[i][0], z0), b[i][1], b[i][2], &lo, &hi); if (!__mm512_cmpl_t_ps_mask(lo, hi)) break; } for (i = 0; i < bec; i++) { bi0 = b[i][0]; abi0 = ABS(bi0); for (j = i + 1; j < bec; j++) { bj0 = b[j][0]; abj0 = ABS(bj0); upgrade(__mm512_cmpl_t_ps_mask(MUL(bi0, bj0), z0), FMADD(abi0, b[j][1], MUL(abj0, b[i][1])), FMADD(abi0, b[j][2], MUL(abj0, b[i][2])), &lo, &hi); if (!__mm512_cmpl_t_ps_mask(lo, hi)) break; } if (!__mm512_cmpl_t_ps_mask(lo, hi)) break; } __mm512_mask_store_epi32(r, __mm512_cmpl_t_ps_mask(hi, lo), __mm512_set1_epi32(0)); </pre>	<div style="border: 1px solid gray; padding: 2px; display: inline-block;">векторный код</div>
--	--	---	---

Рисунок 4. Схема перевода последовательного кода в векторный аналог для ядра функции `tri_box_intersect_16`.

Из рис. 4 видно, что правильно составленный предикатный код (на рисунке слева) может быть довольно просто переведен в векторный аналог. Для этого должны соблюдаться некоторые простые требования. Во-первых, необходимо удалить все `else` ветки в условных операторах. Этого можно добиться, например, путем расщепления оператора `if-else` на два противоположных условия. После этого любое условие `if` легко трансформируется в предикат на весь блок кода, находящийся под условием. Во-вторых, вызовы функции

не должны находиться под предикатами. Вместо этого в нашем случае скалярное условие вызова функции `upgrade` трансформировалось в аргумент данной функции, и после этого код легко поддается векторизации. В остальном все вещественные скалярные инструкции были просто заменены на векторные аналоги. Также была выполнена векторизация функции `upgrade` с помощью слияния всех веток выполнения под их предикатами, результирующий код данной функции представлен на рис. 5.

```
01 void upgrade(__mmask16 m, __m512 f0, __m512 f1,
02             __m512 *lo, __m512 *hi)
03 {
04     __mmask16 c_f0z = __mm512_cmpeq_ps_mask(f0, z0);
05     __mmask16 c_f0n = __mm512_cmpl_t_ps_mask(f0, z0);
06     __mmask16 c_f0p = ~(c_f0z | c_f0n);
07     __mmask16 c_f1p = __mm512_cmpl_t_ps_mask(z0, f1);
08     __m512 k = __mm512_mask_div_ps(k, ~(m & c_f0z), f1, f0);
09
10     k = SUB(z0, k);
11     *lo = __mm512_mask_add_ps(*lo, m & c_f0z & c_f1p, *hi, z1);
12     *hi = __mm512_mask_min_ps(*hi, m & c_f0p, *hi, k);
13     *lo = __mm512_mask_max_ps(*lo, m & c_f0n, *lo, k);
14 }

```

Рисунок 5. Векторная реализация функции `upgrade` с пропигированным условием вызова внутри функции

Заключение

В работе был выполнен анализ задачи об определении пересечения объемной декартовой сетки и поверхностной неструктурированной сетки, состоящей из треугольников. Эффективное решение данной задачи напрямую влияет на производительность расчетных кодов обтекания тел со сложной и изменяющейся геометрией. Была сформулирована математическая постановка задачи, предложены пути оптимизации ее реализации и выполнен анализ векторизации кода для микропроцессоров с поддержкой набора инструкций AVX-512. В результате оптимизации была реализована векторная функция определения попарного пересечения 16 треугольников и 16 прямоугольных параллелепипедов. Полученные программные коды были протестированы на суперкомпьютере МВС-10П, находящемся в МСЦ РАН, на сегменте, оборудованном микропроцессорами Intel

Xeon Phi 7290 Knights Landing. Векторизованный программный код продемонстрировал ускорение 6,7 раз по сравнению со скалярной версией, скомпилированной компилятором icc с полным набором оптимизаций. Полученные результаты подтверждают пригодность описанного подхода к векторизации программного кода.

Работа выполнена в рамках проекта «Исследование с использованием суперкомпьютера физических особенностей нестационарных турбулентных течений и газодинамического управления этими течениями в элементах силовой установки гиперзвукового летательного аппарата» по программе Фундаментальных исследований президиума РАН «Фундаментальные основы прорывных технологий в интересах национальной безопасности». В работе был использован суперкомпьютер МВС-10П, находящийся в МСЦ-РАН.

Vectorization of finding the intersection of volume grid and surface grid for microprocessors with AVX-512 support

A.A. Rybakov

Abstract. Vectorization of program code is one of the most important low-level optimizations that can significantly improve the performance of supercomputer applications. The effect of the vectorization of the code is the more noticeable, the greater the length of the vector. The AVX-512 instruction set supported in modern Intel microprocessors is designed to work with 512-bit vectors and allows vectorize predicated code. In this paper, we consider a practical approach to vectorizing a complex software context that implements the intersection of volume cartesian and surface unstructured computational grids. This problem is used in the calculations of the flow around bodies with complex geometry, performed using the immersed boundary method, while a uniform cartesian grid with cells of the same size and a local-refinement grid can be used as a cartesian volume grid. A mathematical solution of the problem is given, its implementation is carried out, and it is also shown that vectorization using the AVX-512 instruction set allows acceleration of the code on the Intel Xeon Phi KNL microprocessor to be more than 6.5 times compared to the non-vectorized version with single precision float data.

Keywords. Vectorization, Intel Xeon Phi KNL, AVX-512, volume cartesian grid, surface unstructured grid, flat loop, intrinsic functions.

Литература

1. И.В. Абалакин, Н.С. Жданова, Т.К. Козубская. Метод погруженных границ для численного моделирования невязких сжимаемых течений. // Журнал вычислительной математики и математической физики, 2018, Т. 58, № 9, 1462–1471.
2. R. Mittal, G. Iaccarino. Immersed boundary methods. // Annual. Rev. Fluid Mech., 2005, V. 37, P. 239–261.
3. Y.-H. Tseng, J.H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. // Journal of Computational Physics, 2003, V. 192, P. 593–623.

4. В.В. Винников, Д.Л. Ревизников. Метод погруженной границы для расчета сверхзвукового обтекания затупленных тел на прямоугольных сетках. // Труды МАИ, 2007, № 27.
5. В.И. Зоркальцев, М.А. Киселева. Системы линейных неравенств. // Учебное пособие, Иркутск, Издательство Иркутского государственного университета, 2007.
6. С.Н. Черников. Свертывание конечных систем линейных неравенств. // Доклады АН СССР, 1963, Т. 152, № 5, 1075–1078.
7. J. Jeffers, J. Reinders, A. Sodani. Intel Xeon Phi processor high performance programming. Knights Landing edition. // Morgan Kaufmann, 2016.
8. В.Ю. Волконский, С.К. Окунев. Предикатное представление как основа оптимизации программы для архитектур с явно выраженной параллельностью. // Информационные технологии, 2003, № 4, С. 36–45.
9. Intel C++ compiler 16.0 user and reference guide. // Intel Corporation, 2016.
10. Intel Intrinsics Guide. <https://software.intel.com/sites/landingpage/IntrinsicsGuide> // Дата обращения 15.10.2019.
11. В.М. Shabanov, А.А. Rybakov, S.S. Shumilin. Vectorization of High-performance Scientific Calculations Using AVX-512 Instruction Set. // Lobachevskii Journal of Mathematics, 2019, V. 40, № 5, P. 580–598.
12. W. McDoniel, M. Hohnerbach, R. Canales. et al. LAMMPS' PPPM Long-Range Solver for the Second Generation Xeon Phi. // J.M. Kunkel et al. (Eds.): ISC High Performance 2017, LNCS, 2017, V. 10266, P. 61–78.
13. T. Malas, T. Kurth, J. Deslippe. Optimization of the Sparse Matrix-Vector Products of an IDR Krylov Iterative Solver in EMGeo for the Intel KNL Manycore Processor. // M. Taufer et al. (Eds.): ISC High Performance Workshops 2016, LNCS, 2016, V. 9945. P. 378–389.
14. Л.А. Бендерский, А.А. Рыбаков, С.С. Шумилин. Векторизация перемножения малоразмерных матриц специального вида с использованием инструкций AVX-512. // Международный научный журнал «Современные информационные технологии и ИТ-образование», 2018, Т. 14, № 3, С. 594–602.
15. O. Krzikalla, F. Wende, M. Hohnerbach. Dynamic SIMD Vector Lane Scheduling. // M. Taufer et al. (Eds.): ISC High Performance Workshops 2016, LNCS, 2016, V. 9945, P. 354–365.
16. B. Cook, P. Maris, M. Shao. High Performance Optimizations for Nuclear Physics Code MFDn on KNL. // M. Taufer et al. (Eds.): ISC High Performance Workshops 2016, LNCS, 2016, V. 9945, P. 366–377.
17. M. Bader, A. Breuer, W. Holtz, S. Rettenberger. Vectorization of an augmented Riemann solver for the shallow water equations. // Proceedings of the 2014 International Conference on High Performance Computing and Simulation, HPCS 2014, 2014, P. 193–201.
18. C.R. Ferreira, K.T. Mandli, M. Bader. Vectorization of Riemann solvers for the single- and multi-layer shallow water equations. // Proceedings of the 2018 International Conference on High Performance Computing and Simulation, HPCS 2018, 2018, P. 415–422.
19. V. Bramas. Fast Sorting Algorithms using AVX-512 on Intel Knights Landing. // International Journal of Advanced Computer Science and Applications, 2017, V. 8, № 10, P. 337–344.
20. М.С. Гуськова, Л.Ю. Бараш, Л.Н. Щур. Применение AVX512-векторизации для увеличения производительности генератора псевдослучайных чисел. // Труды ИСП РАН, 2018, Т. 30, № 1, С. 115–126.
21. А.А. Рыбаков, С.С. Шумилин. Векторизация сильно разветвленного управления с помощью инструкций AVX-512. // Труды НИИСИ РАН, 2018, Т. 8, № 4, С. 114–126.
22. R. Allen, K. Kennedy. Optimizing Compilers for Modern Architectures. // Morgan Kaufmann, 2001.
23. А.А. Рыбаков, П.Н. Телегин, Б.М. Шабанов. Проблемы векторизации гнезд циклов с использованием инструкций AVX-512. // Электронный научный журнал: Программные продукты, системы и алгоритмы, 2018, № 3, С. 1–11.

Методы и алгоритмы обеспечения прохождения пользовательских заданий с заданным уровнем обслуживания

О.С. Аладышев¹, А.В. Баранов², Д.Ю. Дербышев³

¹ Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, Oleg.Aladyhev@jscc.ru

² Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, antbar@mail.ru

³ Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, kirikusan@gmail.com

Аннотация: В статье рассматривается задача организации обработки привилегированных заданий в научном суперкомпьютерном центре коллективного пользования. В отличие от обычных заданий, привилегированные имеют особые требования, которые определяются отдельным соглашением с пользователем как уровень обслуживания. В статье проанализирован мировой опыт организации обработки заданий с заданным уровнем обслуживания. На основе практики Межведомственного суперкомпьютерного центра РАН (МСЦ РАН) выявлена необходимость в обеспечении привилегированных пользователей возможностями запуска высокоприоритетных заданий за гарантированный срок и предварительного резервирования ресурсов суперкомпьютера путём создания временных окон. Созданные временные окна могут быть использованы для гарантированного старта задания в точно определённое пользователем время. В статье рассмотрена реализация предложенных подходов для отечественной Системы управления прохождением параллельных заданий (СУППЗ), применяемой в МСЦ РАН.

Ключевые слова: суперкомпьютер, система управления заданиями, планирование заданий, соглашение об уровне обслуживания, СУППЗ, SLA, QoS

1. Введение

Суперкомпьютеры, в силу своей уникальности и дороговизны, используются коллективно. Одной из главных тенденций развития суперкомпьютерных технологий является расширение как спектра решаемых задач, так и круга пользователей. У каждого пользователя есть свой, порою уникальный, интерес к суперкомпьютерным ресурсам. Удовлетворение этих интересов и потребностей представляет собой одну из главных задач управления суперкомпьютером как системой коллективного пользования.

Суперкомпьютерные расчёты пользователи оформляют в виде вычислительных заданий, включающих расчётные программы и исходные данные, требования к характеру и объёму запрашиваемых ресурсов. Специальные программные системы управления заданиями (СУЗ) [1] ведут очередь заданий, выделяют ресурсы и запускают на них пользовательские задания, контролируют их выполнение, высвобождают ресурсы после завершения заданий и предоставляют пользователю результаты расчётов.

Суперкомпьютерные ресурсы состоят из многоядерных мультипроцессорных вычислительных модулей (ВМ), объединяемых между собой высокоскоростной коммуникационной средой. Из всей памяти ВМ образуется решающее поле [2]. В требованиях пользовательского задания указываются, как минимум, число необходимых для расчетов ВМ и время, на которое ВМ будут предоставлены заданию. Особенность эксплуатации практически всех современных суперкомпьютеров состоит в том, что ВМ является атомарной единицей планирования ресурсов в СУЗ и не разделяется между различными заданиями.

При наличии большого числа пользователей время пребывания задания в очереди может достигать нескольких часов и даже дней. Для ряда пользователей, чьи задания носят срочный характер, длительное ожидание в очереди неприемлемо. Простое повышение приоритета таких заданий часто оказывается недостаточным – даже находясь первым в очереди, высокоприоритетное задание будет ожидать завершения ранее запущенных заданий. Кроме этого, в системе одновременно может оказаться несколько

высокоприоритетных заданий, которым неизбежно придётся ожидать друг друга. Для таких случаев в СУЗ предусматривается механизм урегулирования конфликтов между высокоприоритетными заданиями, обычно называемый «уровнем обслуживания» или «соглашением об уровне обслуживания» (Service Level Agreement – SLA). Разработка таких механизмов для отечественной Системы Управления Проходом Параллельных Заданий (СУППЗ), применяемой в качестве СУЗ в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН), является целью настоящей работы.

2. Организация приоритетного обслуживания заданий в СУЗ

Среди современных СУЗ лидирующие позиции занимают такие системы как PBS [3], SLURM [4], LSF [5], Moab [6]. Все эти системы обладают развитыми механизмами планирования. Наибольшее распространение, благодаря своим качествам и открытому исходному коду, получила система SLURM. Сравнение некоторых СУЗ с точки зрения организации круглосуточных суперкомпьютерных расчётов в режиме коллективного доступа приведено в работе [7].

Основным методом приоритетного обслуживания практически во всех современных СУЗ является механизм динамического справедливого распределения (fair share) [8]. В SLURM по умолчанию задействован алгоритм т.н. «дерева справедливости» (fair tree) [9]. Для определения приоритетов в механизме используется упорядоченное дерево, образуемое на основе иерархии учётных записей пользователей. Например, пользователь может быть членом одной проектной группы, несколько таких групп могут составлять один проект, несколько проектов могут принадлежать одной организации и т.д.

Предполагается, что администратор СУЗ распределяет квоты на использование ресурсов на верхнем уровне иерархии, например, среди организаций. Полученные организациями квоты распределяются между проектами администраторами соответствующих организаций. Далее администраторы проектов делят доставшиеся им квоты между группами и т.д. Приоритет конечного пользователя складывается из его квоты и квот его иерархических «предков» в дереве справедливости. Важным моментом при этом является то, что квоты определяют только

базовую часть приоритета. Окончательное значение приоритета формируется с учётом того, сколько ресурсов (например, в ядро-минутах) пользовательские задания уже использовали. Основная идея здесь состоит в «справедливом» распределении ресурсов – чем больше пользователь потребил, тем ниже его приоритет.

Потребление ресурсов в прошлом разбивается на ряд учётных периодов, и чем более удалён по времени учётный период, тем меньшее влияние он оказывает на итоговое значение приоритета нового задания. В SLURM для этого применяется формула «полураспада», в которой вес более раннего учётного периода равен половине веса последующего за ним периода, т.е. вес учётных периодов уменьшается в геометрической прогрессии по мере их отодвигания в прошлое. Период «полураспада» в SLURM задаётся администратором СУЗ.

В таких СУЗ, как коммерческие продукты LSF от компании IBM и Moab от компании Adaptive Computing, присутствуют механизмы поддержки различных уровней обслуживания. Предполагается, что пользователь может заключить с суперкомпьютерным центром отдельное соглашение SLA, и СУЗ будет сконфигурирована так, чтобы требования соглашения соблюдались автоматически.

В Moab для соглашений об уровне обслуживания реализован механизм обеспечения заданного качества обслуживания (Quality of Service – QoS) [10]. Механизм QoS имеет множество возможностей для эффективного разделения вычислительных ресурсов между пользователями. С механизмом QoS можно производить немедленный запуск задания с вытеснением остальных заданий, назначение приоритета, подбор ВМ заданной конфигурации, запрет обратного заполнения (backfill) при обработке приоритетных заданий и др. Среди этих возможностей наиболее важное значение имеют обработка заданий к заданному сроку (deadline) и предварительное резервирование ресурсов.

Для отдельного задания или группы заданий сроки выполнения могут быть указаны специально. Для таких заданий сначала оценивается возможность соблюдения этих сроков. Если такая возможность есть, Moab добавляет задание в список срочных и выделяет ресурсы, гарантируя, что все принятые срочные задания будут обработаны не позднее запрошенных сроков. При этом планировщик оптимизирует время использования ресурсов, стараясь выполнить все задания в кратчайшие сроки.

Для выполнения задания пользователя из решающего поля в определённый диапазон времени резервируется некоторое подмножество ВМ. Для каждого резервирования определяются состав резервируемых ресурсов, временной диапазон и список пользователей и групп, имеющих право на использование выделяемых ресурсов (так называемый Access Control List – ACL) в обозначенном временном диапазоне. Причём, если пользователь потребовал выполнить задание не раньше и не позже определённого временного диапазона, то алгоритмы обратного заполнения не будут применяться в процессе резервирования. Тем самым Moab обеспечивает необходимые сроки выполнения заданий и гарантии доступности ресурсов для пользователей, входящих в обозначенный ACL. Moab также поддерживает как регулярно повторяющиеся резервирования, так и создание динамических одноразовых резервирований.

Возможность определения крайних сроков (deadline) для обработки задания обеспечивается и в СУЗ LSF [11]. Так же, как и в Moab, эта возможность реализуется за счёт предварительного резервирования ресурсов в определённый период времени. В LSF это называется «временным окном». Кроме резервирования в определённый период, LSF поддерживает уровни обслуживания, обеспечивающие обработку определённого числа заданий в течение заданного времени или к заданному сроку.

3. Приоритетное обслуживание заданий в СУППЗ

В МСЦ РАН на протяжении двух десятилетий в качестве СУЗ используется Система управления прохождением параллельных заданий (СУППЗ) [12], сравнение которой с известными СУЗ приводится в работах [7, 13].

Правила планирования заданий в СУППЗ задаются при помощи расписания, которое представляет собой последовательность сменяющих друг друга режимов планирования. Планирование заданий в каждый момент времени производится в соответствии с параметрами текущего режима. В системе может быть несколько расписаний, переключение между которыми осуществляется по директиве оператора или системного программиста, при этом переключение режимов внутри каждого расписания осуществляется автоматически.

Одним из параметров режима планирования является шкала определения приоритетов. Приоритеты напрямую зависят от суммарного

времени счёта пользователя за учётный период, который задаётся системным программистом одинаковым для всех режимов текущего расписания.

Например, если шкала имеет следующий вид

(120, 300, 600, 1200, 0)

то это означает, что наивысшим приоритетом (очередь 0) будут обладать задания пользователей, которые за учётный период считали менее 120 минут, чуть меньшим приоритетом (очередь 1) – тех, кто считал менее 300 минут, ещё меньшим (очередь 2) – тех, кто считал менее 600 минут и т.д. Низшим для текущего режима (очередь 4) приоритетом будут обладать задания пользователей, считавших более 1200 минут.

При суммировании времени счёта заданий пользователя за учётный период, время каждого задания умножается на специальный коэффициент – цену задания. Цена задания определяется системным программистом для каждого пользователя при составлении расписания. Чем больше пользователь насчитал, тем ниже будет приоритет его следующих заданий. Приоритет будет выше у тех заданий, владельцы которых затратили меньше «время» для счёта (время, умноженное на цену). Тем самым СУППЗ также обеспечивает принцип динамического справедливого распределения.

Планировщик СУППЗ пытается выделить ресурсы из числа свободных сначала для заданий из очереди 0, потом – из очереди 1 и т.д. Внутри одной очереди ресурсы выделяются в порядке (от меньшего к большему) номеров в списке заданий. Номер очереди и номер задания в очереди образуют составной индикатор порядка выделения ресурсов. Если свободных ресурсов для задания нет, определяется момент времени, когда нужные ресурсы освободятся, и устанавливается время запуска задания. Никакое менее приоритетное задание не может занять ресурсы так, чтобы это отодвинуло запуск более приоритетного задания.

Планировщик может запускать задания, стоящие в очереди позже, если это не повлияет на время старта задания, стоящего в очереди ранее (т.е. применяется упоминавшийся алгоритм обратного заполнения). Такое возможно, например, в случае если для задания А, стоящего в очереди ранее, недостаточно ресурсов, и при этом задание Б, стоящее в очереди позже, успеет завершиться до момента, когда освободится достаточное количество ресурсов для запуска задания А.

Как было сказано выше, приоритет пользователя в очереди СУППЗ задаётся

параметром «цена задания». При достаточно низкой цене «время счёта» пользователя за учётный период практически перестаёт расти, и все задания пользователя будут попадать в самую приоритетную очередь. Однако, постановка в начало очереди не гарантирует запуска задания в определённый срок. Во-первых, приоритеты в СУППЗ относительные, т.е. высокоприоритетное задание не может вытеснить с выполнения никакие запущенные задания, даже ниже приоритетом. Во-вторых, высокоприоритетных заданий может поступить в очередь сравнительно много, и им придётся ждать друг друга.

Предшествующая версия СУППЗ не позволяла задавать уровни обслуживания заданий, что являлось одним из существенных недостатков системы. В настоящей работе с учётом рассмотренного выше мирового опыта авторами предложены механизмы, с помощью которых этот недостаток преодолевается.

4. Уровни обслуживания заданий в СУППЗ

Многолетняя практика МСЦ РАН в организации высокопроизводительных вычислений в режиме коллективного пользования выявила необходимость в обеспечении высокоприоритетных пользователей двумя возможностями. Первая – гарантированный запуск задания не позднее определённого срока, вторая – наличие определённого объёма свободных ресурсов к заданному пользователем времени (резервирование ресурсов на заданное время). Последнее подразумевает запуск привилегированных заданий в заранее выделенных временных окнах.

Конкретные значения по гарантированным срокам запуска заданий, числу этих заданий, объёму и времени резервируемых ресурсов пользователи готовы оговаривать в SLA. В соглашениях также отражается, что в ситуации, когда привилегированный запрос на гарантированный запуск заданий или на резервирование ресурсов не может быть удовлетворён в силу высокой загрузки суперкомпьютера, пользователь извещается об этом при постановке задания в очередь. В этом случае соглашение SLA оставляет за пользователем выбор: либо перенести сроки запуска задания или резервирования ресурсов, либо поставить задание в очередь с высоким приоритетом, но без гарантии сроков запуска, либо отказаться от запроса. Рассмотрим особенности реализации указанных дополнительных режимов обслуживания.

Гарантированный запуск заданий за требуемое время, в случае отсутствия свободных ресурсов в текущий момент, позволяет запускать задание в расчёте на то, что требуемые ресурсы освободятся в течение указанного в соглашении SLA времени. Если ресурсы доступны в момент запуска задания, то этот режим отличается от обычного старта задания только повышенным приоритетом запускаемого задания.

Временным окном называются ресурсы (процессорные ядра или ВМ суперкомпьютера), заранее зарезервированные на определённое время под запуск задания. Стартом окна называется время начала резервирования ресурсов, задержкой – максимальное время, доступное пользователем для запуска задания после старта окна, а размером – максимальная длительность задания, запускаемого на зарезервированных ресурсах. Если задание не было запущено за время задержки со старта временного окна, то выделенные ресурсы автоматически освобождаются. В другом случае излишние ресурсы освобождаются в момент старта задания, а занятые заданием – по его завершении. Временные окна поддерживают запуск только одного задания.

Оба указанных режима старта заданий имеют повышенный относительно обычных заданий приоритет. Различные приоритеты для временных окон и заданий, гарантированно запускаемых за требуемое время, не поддерживаются. Это означает, что при успешном выполнении команды на запуск задания или на создание временного окна никакие дальнейшие действия других пользователей не могут отменить, отложить или иначе повлиять на уже выполненную команду.

Анализируя запущенные в текущий момент задания, можно оценить время занятости вычислительных ресурсов в самом худшем случае, когда задания завершатся не раньше назначенного им срока, при условии отсутствия запусков новых заданий. График занятости ресурсов от времени в этом случае представляет собой невозрастающую функцию с ограниченной областью определения. Так как все задания в очереди имеют более низкий приоритет, чем привилегированные задания, то их можно не учитывать при гарантированном запуске заданий за требуемое время и создании временных окон.

На рисунке 1 приведён пример такой нагрузки, отображённой на графике в виде закрашенной области. В начальный момент времени, в 15:00, загрузка суперкомпьютера уже запущенными заданиями близка к 100%. Постепенно загрузка снижается, и при

отсутствии запусков новых заданий из очереди к 17:00 все ресурсы суперкомпьютера освободятся.

На рисунке 2 показан процесс гарантированного запуска задания за требуемое время. Текущая загрузка отображена так же, как и на предыдущем рисунке, новое задание обозначается новой закрашенной областью другого цвета. При тех же самых начальных условиях, пользователь желает запустить задание, на которое не хватает свободных ресурсов. Допустим, прописанное в соглашении SLA гарантированное время запуска заданий этого пользователя составляет 90 минут. Видно, что достаточные для запуска нового задания ресурсы освободятся к 15:33. Это позволяет СУППЗ добавить новое задание в очередь и

гарантированно поставить его на выполнение в течение оговорённого в соглашении времени. Задание должно гарантированно запуститься через 33 минуты, поскольку имеет высший приоритет, и вновь поступившие задания не смогут отодвинуть его старт. Однако, если бы задание из примера требовало, к примеру, все ресурсы суперкомпьютера, то за оговорённые в соглашении 90 мин. эти ресурсы, скорее всего, не освободились бы. СУППЗ в этом случае извещает пользователя о невозможности выполнения задания, что также должно регулироваться в SLA, например, вариантом выплаты неустойки.

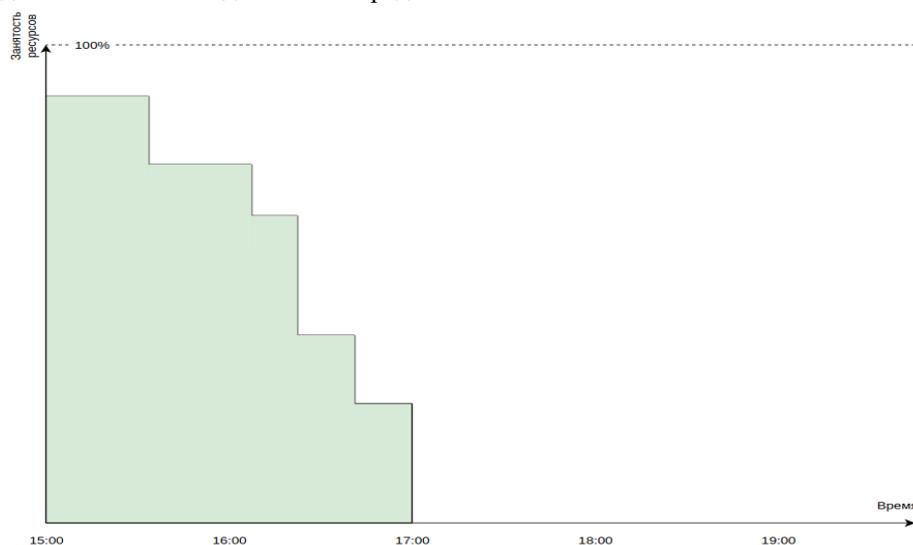


Рис. 1. График загрузки суперкомпьютера запуском заданиями

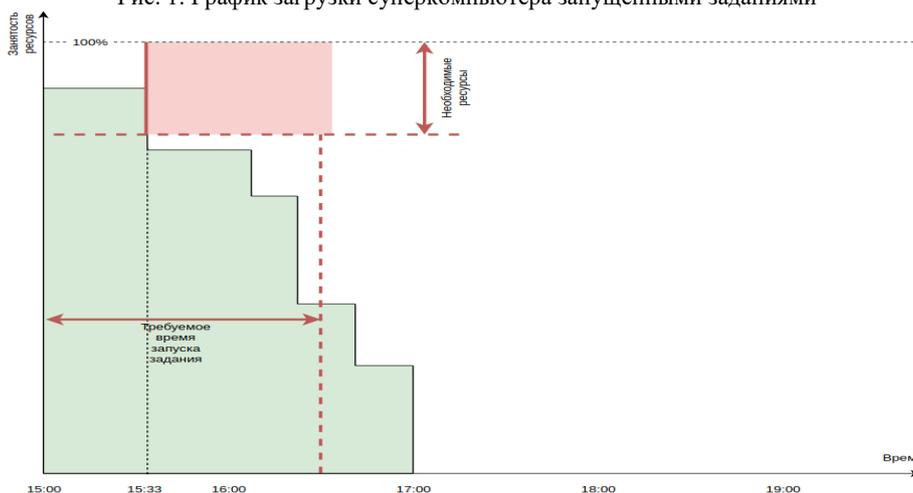


Рис. 2. Гарантированный запуск задания за заданное время

На рисунке 3 показана процедура создания временного окна. Пусть после добавления в очередь задания из предыдущего примера пользователь запросил создание временного

окна. Старт окна назначен на 17:45, задержка равна 15 минутам, а размер окна – 60 минутам. На рисунке окно обозначено новой закрашенной областью. Так как к указанному

времени все выполняющиеся задания, а также добавленные в очередь привилегированные задания гарантированно закончат свою работу, то временное окно может быть успешно создано. Пользователь сможет запустить одно задание в период с 17:45 по 18:00 длительностью не более 60 минут, при условии,

что требуемое для его выполнения количество ресурсов не превышает указанное при создании окна. При старте задания оставшиеся ресурсы окна будут освобождены, а время их полного освобождения скорректировано согласно длительности нового задания.

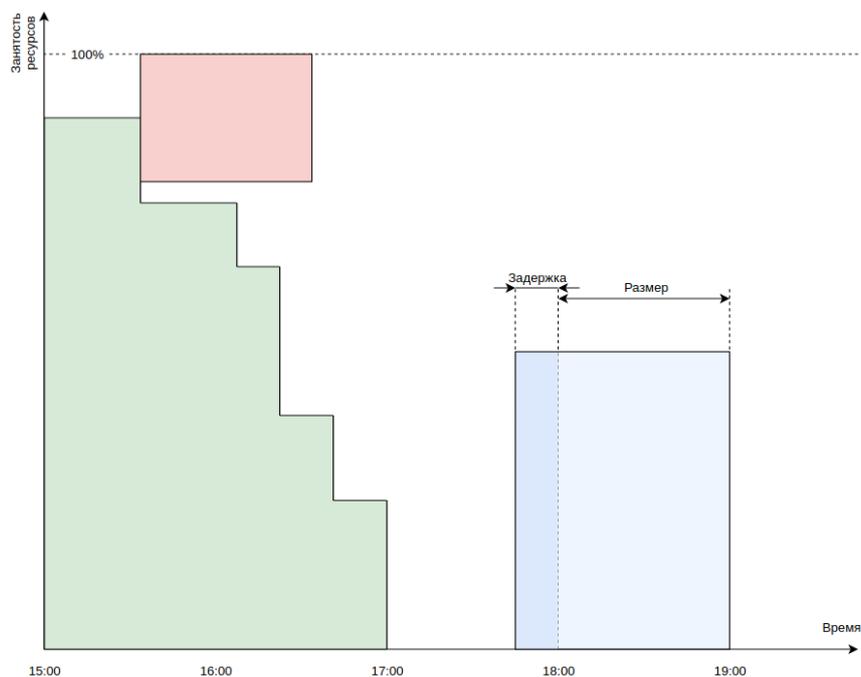


Рис. 3. Создание временного окна

5. Детали реализации уровней обслуживания заданий в СУППЗ

Предполагается, что соглашение об уровне обслуживания будет подписываться пользователем вместе с договором на оказание услуг по высокопроизводительным вычислениям. Соглашение действует в отношении научного проекта, в выполнении которого может быть задействована группа пользователей. Подписанное соглашение отражается в специальных конфигурационных файлах СУППЗ и получает уникальный специальный номер – т.н. номер SLA. Для того чтобы воспользоваться соглашением, пользователь в командах СУППЗ должен указать соответствующий номер SLA.

В конфигурации СУППЗ относительно каждого соглашения об уровне обслуживания по номеру SLA указывается следующая информация:

- имя системной группы, соответствующей проекту, в отношении которого действует соглашение SLA;
- гарантированное максимальное время запуска заданий проекта;
- максимальное число процессорных ядер, которые могут быть выделены для заданий с гарантированным временем старта;
- максимальное число заданий с гарантированным временем старта, которые могут быть одновременно поставлены в очередь или запущены пользователями группы проекта;
- максимально допустимая задержка временного окна (максимальное время, доступное для запуска задания после старта окна), создаваемого пользователями группы;
- максимальная длительность запускаемого во временном окне задания;
- максимальное число процессорных ядер, выделяемых для временного окна;
- максимальное число одновременно создаваемых пользователями группы временных окон.

Для того чтобы назначить крайний срок запуска задания при его постановке в очередь командой `mpirun`, пользователь должен указать ключи `-SLA` и `-ws`:

```
-SLA <номер_SLA> -ws <время>
```

Флаг `-SLA` задает номер SLA, флаг `-ws` – максимально допустимое время ожидания запуска для нового задания.

Во время выполнения команды постановки SLA-задания в очередь происходит блокировка запуска новых заданий из очереди для проверки возможности запуска задания за указанное пользователем время. Проверка производится специально разработанной утилитой `scheduled_start`, анализирующей состояние очереди и состав выполняющихся заданий. При возможности запуска задание добавляется в очередь, иначе пользователь извещается о невозможности выполнения SLA-запроса.

Отметим, что утилита `scheduled_start` доступна пользователям для получения информации о возможности запуска за гарантированное время, но при этом не происходит блокировки запуска новых заданий, и возможна ситуация «гонки» между проверкой возможности запуска SLA-задания и запуском иных заданий.

Временные окна реализованы путем изменения расписания СУППЗ. При создании окна расписание перестраивается с уменьшением с момента старта окна числа доступных процессорных ядер для остальных заданий на время, равное длительности окна. Создание окна осуществляется командой `mreserve`:

```
mreserve -start <START> -delay <DELAY> -length <LENGTH> -SLA <SLA> -proc <PROC>
```

Команда проверяет наличие для группы пользователей с соглашением SLA возможности выделить ко времени `<START>` `<PROC>` процессорных ядер на время `<LENGTH+DELAY>` и принадлежность пользователя к SLA-группе. При возможности выделения этих ресурсов и соответствии запроса ограничениям, указанным в соглашении SLA, производится резервирование ресурсов путём рассмотренного выше изменения расписания СУППЗ. Пользователю при этом сообщается уникальный идентификатор созданного окна, по которому в дальнейшем можно использовать или освободить зарезервированные ресурсы.

Удаление окон производится командой `mrelease`:

```
mrelease -id <ID>
```

Команда удаляет временное окно с идентификатором `<ID>`.

Запуск задания в созданном окне осуществляется указанием в команде `mpirun` флагов `-SLA` и `-win`:

```
-SLA <NUMBER> -win <ID>
```

Флаг `-SLA` указывает номер `<NUMBER>` соглашения SLA, а флаг `-win` – идентификатор `<ID>` существующего временного окна, в котором необходимо запустить задание.

Во время выполнения команды происходит блокировка запуска новых заданий из очереди для предотвращения «гонки». Задание может быть запущено на зарезервированном окне ресурсов, если требуемое число ядер и время выполнения задания не превышают соответствующих параметров временного окна. В случае успеха команды `mpirun` задание запускается в указанном окне. При невозможности запуска зарезервированные ресурсы не освобождаются, а при успешном запуске – освобождаются неиспользованные при старте задания ресурсы.

6. Заключение

В представленной работе авторами решена задача адаптации известных методов и алгоритмов обеспечения заданного уровня обслуживания суперкомпьютерных заданий для отечественной Системы управления прохождением параллельных заданий (СУППЗ), применяемой в МСЦ РАН для обработки пользовательских суперкомпьютерных заданий. Уровни обслуживания предложено задавать через соглашения об уровне сервиса (SLA), в которых определяются параметры запуска привилегированных заданий за гарантированное время и параметры резервирования ресурсов суперкомпьютера. Предварительное резервирование осуществляется за счёт создания т.н. временных окон, в рамках которых возможен старт заданий в точно определённое пользователем время. Некоторые механизмы уже опробованы в рамках пилотного научного проекта. Ближайшей перспективой является полное внедрение разработанных алгоритмов и инструментальных средств в практику Центра коллективного пользования вычислительными ресурсами МСЦ РАН – филиала ФГУ ФНЦ НИИСИ РАН.

Публикация выполнена в рамках программы Президиума РАН по теме (проекту) «Исследование методов и алгоритмов функционирования систем планирования и управления заданиями сверхвысокопроизводительных вычислений для повышения эффективности работы современных суперкомпьютерных центров».

Methods and algorithms for processing supercomputer jobs with service level agreement

O.S. Aladyshv, A.V. Baranov, D.Yu. Derbyshev

Abstract: The article considers the problem of the privileged jobs processing in a scientific supercomputer center. The article analyzes the world experience in the job processing with a given service level. A need has been determined by practice of the Joint Supercomputer Center of the Russian Academy of Sciences (JSCC RAS) that privileged users should be able to run high priority jobs in the guaranteed period and to pre-reserve supercomputer resources by creating time windows or resource reservations. Created time windows can be used to guarantee the job start at the time precisely defined by the user. The article discusses the implementation of the proposed approach for the job management system used in the JSCC RAS.

Keywords: HPC, supercomputer, job scheduling, workload manager, service level agreement, SLA, QoS

Литература

1. A. Reuther et al.: Scalable system scheduling for HPC and big data. “Journal of Parallel and Distributed Computing”, V. 111 (2018), 76–92. DOI: 10.1016/j.jpdc.2017.06.009
2. О.С. Аладышев, Е.А. Киселёв, Г.И. Савин, П.Н. Телегин, Б.М. Шабанов. Влияние характеристик внешней памяти суперкомпьютерных комплексов на выполнение параллельных программ. «Системы и средства информатики», Т. 24 (2014), № 4, 111–123. DOI: 10.14357/08696527140407
3. R.L. Henderson. Job scheduling under the Portable Batch System. “Lecture Notes in Computer Science”, V. 949 (1995), 279–294. DOI: 10.1007/3-540-60153-8_34
4. A.B. Yoo, M.A. Jette, M. Grondona. SLURM: Simple Linux Utility for Resource Management. “Lecture Notes in Computer Science”, V. 2862 (2003), 44–60. DOI: 10.1007/10968987_3
5. IBM Spectrum LSF overview. https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lfsf_foundations/chap_lsf_overview_foundations.html (дата обращения 04.10.2019)
6. Moab HPC Suite Enterprise Edition. <http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-suite-enterprise-edition> (дата обращения 04.10.2019)
7. А.В. Баранов, А.В. Киселёв, В.В. Старичков, Р.П. Ионин, Д.С. Ляховец. Сравнение систем пакетной обработки с точки зрения организации промышленного счета. «Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции», Россия, Новороссийск, 17–22 сентября 2012, 506–508. <http://agora.guru.ru/abrau2012/pdf/506.pdf> (дата обращения: 10.10.2019).
8. Fair Tree Fairshare Algorithm. https://slurm.schedmd.com/fair_tree.html (дата обращения: 04.10.2019).
9. Classic Fairshare Algorithm. https://slurm.schedmd.com/classic_fair_share.html (дата обращения: 04.10.2019).
10. Moab Workload Manager. Administrator Guide 9.0.1. Released: March 2016; Revised: May 16, 2016. <http://docs.adaptivecomputing.com/9-0-1/MWM/Moab-9.0.1.pdf> (дата обращения: 04.10.2019).
11. Using goal-oriented SLA scheduling. https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lfsf_admin/goal_oriented_sla_sched.html (дата обращения: 04.10.2019).
12. Система управления прохождением параллельных заданий. Руководство программиста (пользователя), (2016). <http://www.jscc.ru/wp-content/uploads/2017/06/SUPPZ-user-guide-2016.pdf> (дата обращения: 04.10.2019).
13. А.В. Баранов, Д.С. Ляховец. Сравнение качества планирования заданий в системах пакетной обработки SLURM и СУППЗ. «Научный сервис в сети Интернет: все грани параллелизма: Труды Международной суперкомпьютерной конференции», Россия, Новороссийск, 23–28 сентября 2013, 410–414. <http://agora.guru.ru/abrau2013/pdf/410.pdf> (дата обращения: 04.10.2019).

Формирование статистических профилей суперкомпьютерных приложений

А.В. Баранов¹, М.О. Васильев², Е.А. Киселёв³, В.Ф. Огарышев⁴

¹ Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, antbar@mail.ru

² Московский физико-технический институт (государственный университет), Москва, Россия, vmo1996@yandex.ru

³ Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, kiselev@jssc.ru

⁴ Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ogaryshev@jssc.ru

Аннотация: Статья посвящена задаче расширения базы данных статистической информации системы управления заданиями путем использования данных системы мониторинга суперкомпьютерного центра коллективного пользования. Авторами определены представляющие с их точки зрения интерес параметры базы данных системы мониторинга, позволяющие решить ряд научно-технических задач классификации пользовательских приложений и оптимизации выделения ресурсов в соответствии с классом приложения. В статье приводятся структура и алгоритмы работы разработанного программного средства «Профиль-М», рассматриваются его возможности и отмечаются результаты опытной эксплуатации на суперкомпьютере МВС-10П, установленном в Межведомственном суперкомпьютерном центре Российской академии наук – филиале Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (МСЦ РАН).

Ключевые слова: высокопроизводительные вычисления, суперкомпьютеры, планирование заданий, статистика работы суперкомпьютера, система мониторинга, база данных статистики

1. Введение

Современные суперкомпьютеры, как правило, работают в режиме коллективного пользования. Множество пользователей оформляют свои расчеты в виде вычислительных заданий и направляют их в специальные программные системы управления заданиями (СУЗ) [1]. Последние формируют очередь из заданий пользователей, ведут учет ресурсов суперкомпьютера и накапливают статистическую информацию об обработанных заданиях и потребленных ими вычислительных ресурсах.

Для оптимизации управления заданиями пользователей важным является знание статистических профилей суперкомпьютерных приложений, содержащих информацию об объемах и характере использования потребленных приложениями вычислительных ресурсов.

Одной из систем управления заданиями является отечественная разработка – система управления прохождением параллельных заданий (СУППЗ) [2], применяемая в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН) в течение двух десятилетий. СУППЗ в процессе своей работы накапливает не все сведения из состава

статистического профиля суперкомпьютерного приложения. Не сохраняются сведения о характере использования выделенных заданиям ресурсов (процент использования процессора, памяти, коммуникационной среды). Эти сведения накапливаются в базах данных систем мониторинга без привязки к выполненным заданиям. Для формирования полного статистического профиля суперкомпьютерного приложения необходимо организовать выборку информации из системы мониторинга в привязке к заданиям и поместить эту информацию в базу данных системы управления заданиями. Разработка такой программной системы является целью настоящей работы.

2. Задача формирования статистических профилей

В составе суперкомпьютерной вычислительной системы обычно выделяют решающее поле и управляющую ЭВМ. Решающее поле состоит из объединенных высокоскоростной сетью вычислительных узлов (ВУ), на которых непосредственно производятся вычисления. Через управляющую ЭВМ посредством СУЗ осуществляется ведение очереди вычислительных заданий и последующее распределение ресурсов между

ними. Для каждого задания, прошедшего через очередь, СУЗ выделяет и конфигурирует требуемое подмножество свободных узлов решающего поля и инициирует на них процесс вычислений. При выполнении задания СУЗ осуществляет его контроль, завершая задание по запросу пользователя или по истечении заказанного времени обработки. После завершения задания СУЗ освобождает выделенные ресурсы.

Для оптимизации управления заданиями пользователей важным является знание статистических профилей суперкомпьютерных приложений. В состав статистического профиля суперкомпьютерного приложения входит информация о числе и названиях вычислительных узлов, выделенных заданию, количестве заказанного времени вычислений и реальном времени выполнения приложения. Отметим, что заказанное и реальное времена выполнения приложения могут значительно различаться, и эта разница образует существенную составляющую статистического профиля. В профиль также включается характер использования вычислительных ресурсов, а именно, процент использования процессора, оперативной памяти и сети передачи данных и/или значения характеристик использования других ресурсов.

Главной целью формирования профилей приложения является оптимизация планирования заданий. С использованием статистических профилей можно решить ряд научно-технических задач, например:

1) Классификация приложений пользователей по профилям. Классифицировать можно как по времени выполнения, так и по количеству используемых ресурсов.

2) Оптимизация выделения ресурсов в соответствии с классом приложения. К примеру, заданиям, использующим менее 15% процессорного времени, можно было бы выделять менее мощные узлы. Классификация всех приложений по профилям позволит для каждого класса подобрать оптимальную для его приложений аппаратно-программную конфигурацию суперкомпьютера.

3) Комбинирование различных заданий для выполнения на одном ресурсе. Если профили двух разных заданий показывают, что на протяжении вычислений они использовали менее 50 процентов процессора, то их выполнение можно спланировать на один вычислительный узел.

3. Подсистемы биллинга и мониторинга

Необходимой составляющей любой СУЗ является биллинговая подсистема, накапливающая статистическую информацию о выполненных заданиях и потребленных пользователями суперкомпьютерных ресурсах. В СУППЗ за сбор и хранение статистической информации о работе пользовательских заданий на суперкомпьютерных установках отвечают подсистема «Статистика» [3] и программное средство «Кростат» [4].

СУППЗ в процессе своей работы накапливает не все сведения из состава статистического профиля суперкомпьютерного приложения. Как уже отмечалось, не сохраняются сведения о характере использования выделенных заданием ресурсов (процент использования процессора, памяти, коммуникационной среды). Эти сведения накапливаются в базах данных систем мониторинга без привязки к выполненным заданиям.

Системы мониторинга суперкомпьютерных комплексов – важная часть их программного обеспечения, призванная обеспечить максимально эффективное и бесперебойное функционирование, а в случае возникновения нештатных ситуаций – сохранность оборудования и оповещение персонала. Система мониторинга в непрерывной работе осуществляет сбор информации об использовании аппаратных ресурсов и сохраняет сведения в собственную базу данных. При необходимости администратор в любой момент может ознакомиться с собранной статистикой за определенный период. Среди распространенных систем мониторинга следует отметить следующие: Ganglia[5], Zabbix[6], Nagios[7].

В качестве примера рассмотрим программное средство Ganglia – масштабируемую распределенную систему мониторинга кластеров параллельных и распределенных вычислений и облачных систем с иерархической структурой. Ganglia позволяет наблюдать статистику и историю (загруженность процессоров, сети) вычислений в реальном времени для ВУ решающего поля суперкомпьютера. Система Ganglia состоит из 3 основных компонентов:

1. Монитор – приложение, которое устанавливается на каждом ВУ и выполняет сбор параметров производительности узла (список параметров задается пользователем). Собранные данные отсылаются серверу сбора статистики.

2. Сервер сбора статистики – приложение, устанавливаемое на управляющей ЭВМ. Сервер принимает данные от мониторов и сохраняет в локальную базу данных информацию о параметрах производительности каждого ВУ за определенный период. Сервер может предоставлять информацию о загрузке узлов по сетевому запросу.

3. Веб-интерфейс используется для графического отображения данных, собираемых сервером мониторинга.

Все собранные данные по умолчанию хранятся в циклической базе данных. Собранные данные перезаписываются через равные интервалы времени, что позволяет сохранять постоянный размер базы. Текущие сведения будут максимально подробными, в то время как сведения недельной или месячной давности будут агрегированы и сокращены.

Система мониторинга сохраняет информацию об использовании ресурсов без привязки к заданиям, которым эти ресурсы были выделены. Для формирования полного статистического профиля суперкомпьютерного приложения необходимо организовать выборку информации из системы мониторинга в привязке к заданиям и поместить эту информацию в базу данных биллинговой подсистемы СУЗ.

4. Модульная система формирования статистических профилей приложений «Профиль-М»

Разные системы мониторинга по-разному сохраняют собранные статистические данные о

работе вычислительных узлов в свои базы данных. Из этого следует необходимость разработки отдельного модуля сбора информации для каждой системы мониторинга. При этом должен быть единый формат сохранения данных для всех модулей сбора.

Кроме того, для хранения полученной из систем мониторинга статистической информации об использовании пользовательским заданием вычислительных ресурсов необходимо модифицировать структуру базы данных хранения статистики СУППЗ, а также разработать единый модуль загрузки собранной информации в модифицированную базу данных.

В результате авторами была разработана система «Профиль-М», позволяющая формировать файлы профилей суперкомпьютерных приложений и записывать информацию с этих файлов в модифицированную базу данных СУППЗ. Общая схема системы «Профиль-М» приведена на рисунке 1.

Модули сбора получают необходимые сведения о работе суперкомпьютерных приложений из баз данных систем мониторинга, на выходе формируя файлы профилей приложений в определенном формате. Модуль загрузки собранной информации считывает сформированные файлы и заносит собранные данные в модифицированную базу данных статистики СУППЗ.

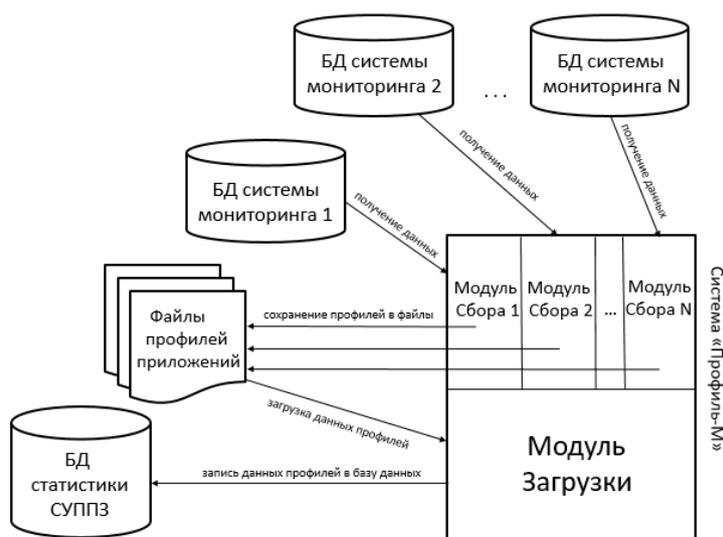


Рис. 1. Схема работы системы «Профиль-М»

Сохранение информации в базу данных должно вестись по ряду характеристик использования приложением выделенных ему ресурсов (процент использования процессорного времени, оперативной памяти, интенсивность сетевых обменов). Для списка характеристик необходима отдельная таблица-справочник. Сохранение всех значений характеристик, полученных из систем мониторинга (в которых сохраняемая информация циклически обновляется) значительно увеличит как объем базы данных статистики, так и время поиска в ней. По этой причине возникает необходимость в агрегировании полученной из систем мониторинга информации. Хранение только максимального, минимального и среднего значений по каждой характеристике для каждого выделенного заданию вычислительного узла позволяет достаточно точно определить характер выполняемого задания и, соответственно, отличать профили приложений друг от друга.

Исходя из вышесказанного, структура базы данных статистики СУППЗ была расширена необходимыми таблицами для хранения дополнительной информации о приложениях пользователей. Связь с существующими таблицами базы данных обеспечивается наличием внешнего ключа на таблицу, хранящую информацию о запусках заданий. В структуру базы данных были добавлены таблицы для хранения имен вычислительных узлов, для метаданных собираемых системами мониторинга метрик и для значений этих метрик для каждого ВУ и для каждого задания.

Как уже отмечалось, модульность системы «Профиль-М» обуславливается существованием различных систем мониторинга, для каждой из которых необходима разработка своего модуля сбора информации. На этапе записи собранных сведений в модифицированную базу данных производятся одинаковые действия, независимые от типа и числа модулей сбора. Следовательно, требуется единый формат сохранения полученной из разных систем мониторинга информации. В качестве базового был выбран формат json, как широко распространенный и часто используемый в качестве стандарта де-факто при обмене данными между информационными подсистемами. Так как у каждого задания есть уникальный системный идентификатор, то его можно использовать в качестве названия файла, но поскольку запусков у задания может быть больше одного, после уникального идентификатора было решено использовать

через дефис автоматически инкрементируемый номер запуска.

Модулю загрузки в БД необходимо определять, занесены ли сведения из определенного файла в модифицированную базу данных. Авторами было принято решение вести журнал загрузок, в который помещаются имена промежуточных файлов и пометки, произведена ли загрузка информации из конкретного файла в базу данных СУППЗ.

В каждом промежуточном файле соответствующими модулями сбора сохраняется статистическая информация – название суперкомпьютера, имена выделенных заданию вычислительных узлов, время старта и время завершения задания, максимальное, минимальное и среднее значения использования процессора, оперативной памяти и сети передачи информации для каждого вычислительного узла, а также дата и время, когда эти значения были достигнуты.

5. Порядок работы с системой «Профиль-М»

Система «Профиль-М» может содержать в своем составе один или несколько модулей сбора. Каждый модуль соответствует отдельной системе мониторинга. Все модули сбора сохраняют полученную от своих систем мониторинга информацию в файлы единого формата, рассмотренного выше.

Вызов модуля сбора осуществляется автоматически при завершении задания. В этот момент на управляющей ЭВМ суперкомпьютера СУППЗ выполняет специальный командный файл, в котором системный программист определяет действия, необходимые для выполнения по завершении задания [8]. Перед вызовом модуля сбора системы «Профиль-М» определяются тип модуля сбора, идентификатор завершившегося задания, время его старта и завершения, список узлов, на которых это задание выполнялось. Эта информация передается в качестве параметров вызываемому модулю сбора.

Вызванный модуль сбора обращается к соответствующей системе мониторинга и по списку узлов получает информацию по использованию этих узлов (процессорное время, оперативная память, объем переданных по сети данных) за время выполнения задания. Далее происходит определение максимального, минимального и среднего значения каждой характеристики за период счета задания. Для каждого узла модуль сбора записывает полученные данные в файл формата json с уникальным именем, соответствующим системному идентификатору задания и номеру

его запуска. Эти файлы накапливаются в специально заданном каталоге.

Программный модуль загрузки данных, как правило, вызывается оператором после занесения в базу данных статистики информации из журналов СУППЗ о выполненных заданиях. Модуль загрузки последовательно считывает сформированные модулями сбора файлы и записывает статистические сведения из них в базу данных. Для каждого файла по его имени в базе данных определяется задание, к которому будут привязаны сохраненные в файле значения статистических метрик. После этого из файла считываются имена вычислительных узлов, выделенных заданию. Если узла с некоторым именем в базе данных нет, то он добавляется в таблицу узлов. Затем из файла для каждого узла считываются максимальные, минимальные и средние значения собранных метрик и записываются в базу данных с привязкой к конкретному заданию.

После успешной загрузки данных из очередного файла в журнале загрузки системы «Профиль-М» проставляется соответствующая отметка, после чего осуществляется переход к следующему незагруженному файлу.

6. Опытная эксплуатация системы «Профиль-М» на суперкомпьютере МВС-10П

Опытная эксплуатация системы «Профиль-М» производилась на суперкомпьютере МВС-10П (раздел «Торнадо») [9], установленном в МСЦ РАН. МВС-10П состоит из 207 вычислительных узлов, построенных на 8-ядерных процессорах Intel Xeon E5-2690 и 61-ядерных ускорителях Intel Xeon Phi 7110X. Пиковая производительность МВС-10П составляет 523,8 ТФлопс, суммарный объем оперативной памяти – 13 ТБ. Коммуникационная сеть функционирует на базе Infiniband FDR.

На МВС-10П установлены система управления прохождением параллельных заданий (СУППЗ), система мониторинга Ganglia, система управления базами данных PostgreSQL. Исходя из этого, для опытного образца системы «Профиль-М» был разработан модуль сбора информации из системы мониторинга Ganglia.

При попытках запуска модулей системы «Профиль-М», реализованных на языке Python, возникли конфликты версий системной библиотеки, отвечающей за работу СУБД PostgreSQL, причем как при использовании стандартных установленных библиотек, так и при применении виртуальной среды Python.

Конфликты удалось разрешить с помощью средства создания автономных исполняемых файлов `sx_Freeze` [10] из исходных файлов Python. Приложение `sx_Freeze` создает каталог, в который помещает все зависимости Python-модуля, а также добавляет в этот каталог исполняемый файл.

Опытная эксплуатация системы «Профиль-М» проводилась в течение апреля-мая 2019 года. За этот период было успешно сформировано более 5000 статистических профилей суперкомпьютерных приложений, при этом в базу данных статистики было внесено более 100 тыс. записей о профилях заданий. После формирования профилей приложений стало возможно выполнение статистических запросов вида «Выбрать пользователей, приложения которых расходуют в среднем более 90% процессорного времени».

7. Заключение

Для обеспечения возможности формирования и сохранения статистических профилей суперкомпьютерных приложений в отечественной системе управления прохождением параллельных заданий (СУППЗ) авторами решены следующие задачи.

Разработана программная система «Профиль-М», состоящая из модулей сбора данных для формирования профиля суперкомпьютерного приложения и модуля загрузки собранной статистической информации в базу данных статистики СУППЗ. Модули сбора получают информацию об объемах использованных заданием ресурсов из баз данных систем мониторинга и сохраняют ее в промежуточные файлы-профили суперкомпьютерных приложений. Модуль загрузки в БД, извлекая информацию из промежуточных файлов, заносит ее в БД статистики СУППЗ с модифицированной авторами структурой. Модификация структуры БД заключалась в добавлении новых таблиц для хранения статистической информации полученной из баз данных систем мониторинга.

Разработанная система «Профиль-М» прошла опытную эксплуатацию на действующем в режиме коллективного пользования под управлением СУППЗ суперкомпьютере МВС-10П, установленном в МСЦ РАН.

Перспективными являются дальнейшая разработка модулей сбора системы «Профиль-М» для различных систем мониторинга и внедрение системы на остальные суперкомпьютеры МСЦ РАН.

Публикация выполнена в рамках государственного задания по проведению

фундаментальных научных исследований по числительных комплексов, в том числе теме «Разработка архитектур, системных гетерогенных суперкомпьютеров и нетрадиционных архитектур микропроцессоров» и методов для создания и циионных архитектур микропроцессоров» использования высокопроизводительных вы- № 0065-2019-0016.

The composing statistical profiles for supercomputer applications

A.V. Baranov, M.O. Vasiliev, E.A. Kiselev, V.F. Ogaryshev

Abstract: The article discusses expanding the job management system statistics database using the monitoring systems data of a supercomputer center on the example of the Joint Supercomputer Center of the Russian Academy of Sciences (JSCC RAS). The authors identified, from their point of view, the monitoring system parameters allowing solving a number of scientific and technical problems, the classification of supercomputer user applications and the resource allocation optimization in accordance with the application class. The article presents the structure and algorithms of the developed software called "Profile-M", considers its capabilities and notes the results of trial operation "Profile-M" software in the MVS-10P supercomputer installed at the JSCC RAS.

Keywords: high-performance computing, supercomputers, job scheduling, supercomputer operation statistics, monitoring system, statistics database

Литература

1. A. Reuther et al.: Scalable system scheduling for HPC and big data. "Journal of Parallel and Distributed Computing", V. 111 (2018), 76–92. DOI: 10.1016/j.jpdc.2017.06.009
2. СУППЗ. <http://suppz.jssc.ru/> (дата обращения: 23.04.2019).
3. А.В. Баранов, С.А. Дбар. Статистика Системы управления прохождением параллельных заданий. Свидетельство о государственной регистрации базы данных № 2016621669 от 15 декабря 2016 года.
4. О.С. Аладышев, А.В. Баранов, Е.А. Киселёв, Р.И. Гришин. Система сбора и обработки статистики «МСЦ-КроСтат». Свидетельство о государственной регистрации программы для ЭВМ № 2016663606 от 13 декабря 2016 года.
5. M. Massie, B. Chun, D. Culler. The Ganglia Distributed Monitoring System: Design Implementation and Experience. "Parallel Computing", V. 30, I. 7 (2004), 817–840. DOI: 10.1016/j.parco.2004.04.001
6. Zabbix Documentation. <https://www.zabbix.com/manuals> (дата обращения 07.03.2019).
7. Nagios 3 Enterprise Network Monitoring, (2008). DOI: 10.1016/B978-1-59749-267-6.X0001-0
8. Система управления прохождением параллельных заданий. Руководство системного программиста (администратора). М.: ИПМ им. М.В. Келдыша РАН, МСЦ РАН, (2016).
9. Межведомственный суперкомпьютерный центр Российской Академии наук. МВС-10П Торнадо. <http://www.jssc.ru/resources/hpc/#item85> (дата обращения 18.05.2019).
10. Создание запускаемых файлов из скриптов на языке Python с помощью cx_Freeze. <https://jenuay.net/Programming/Cxfreeze> (дата обращения 31.07.2019)

Интеграция систем управления заданиями SLURM и СУППЗ

А.В. Баранов¹, Е.А. Киселёв²

¹ Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, antbar@mail.ru

² Межведомственный суперкомпьютерный центр РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, kiselev@jsgc.ru

Аннотация: В статье рассматривается задача интеграции распространенной системы управления заданиями SLURM и отечественной Системы управления прохождением параллельных заданий (СУППЗ), применяемой в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН). В интегрированной системе функции планирования заданий и ведения их очереди возлагаются на СУППЗ, а функции управления суперкомпьютерными ресурсами – на SLURM. Принцип интеграции состоит в следующем. Очередь SLURM постоянно поддерживается пустой, и прошедшие через очередь СУППЗ задания передаются и немедленно запускаются в окружении SLURM. В ходе исследования были выявлены различия в подходах к управлению заданиями и ресурсами в SLURM и СУППЗ. Выявленные различия были компенсированы авторами в процессе осуществления интеграции. Для предотвращения несанкционированного доступа пользователей к SLURM в обход СУППЗ авторами была разработана специальная программная подсистема контроля доступа. В результате работы была сохранена исторически сложившаяся экосистема МСЦ РАН, основу которой составляет СУППЗ, и в то же время стал доступным весь спектр возможностей управления суперкомпьютерным оборудованием, реализованный в SLURM.

Ключевые слова: суперкомпьютер, система управления заданиями, планирование заданий, SLURM, СУППЗ, Intel Xeon Phi, Intel Knights Landing

1. Введение

Основу современных суперкомпьютеров составляет решающее поле из объединенных высокоскоростной сетью вычислительных узлов (ВУ). Вычислительный узел, как правило, представляет собой высокопроизводительный многоядерный сервер, архитектура которого варьируется в зависимости от специфики и назначения суперкомпьютера: может быть универсальной (типа x86 или Power), гибридной (в состав ВУ включены ускорители), либо специализированной (например, на базе ПЛИС).

В силу своей уникальности и дороговизны суперкомпьютеры являются вычислительными системами коллективного пользования. Суперкомпьютерные расчеты пользователи оформляют в виде вычислительных заданий, включающих, помимо расчетной программы, требования к характеру и объему запрашиваемых ресурсов, а также нередко исходные данные. Специальные программные системы управления заданиями (СУЗ) [1] осуществляют распределение пользовательских заданий по ВУ суперкомпьютера.

Все функции СУЗ можно условно разделить

на две группы. Первую группу составляют функции управления заданиями: прием входного потока различных заданий от разных пользователей, ведение их очереди, построение расписания запусков. Вторую группу составляют функции управления ресурсами: выделение ресурсов для прошедших очередь заданий, конфигурация выделенных ресурсов, запуск заданий, контроль их выполнения, освобождение ресурсов после завершения заданий.

Нетрудно видеть, что функции управления заданиями одинаковы для любых типов суперкомпьютеров и не зависят от архитектуры ВУ. Функции управления ресурсами, наоборот, во многом определяются архитектурой и комплектацией ВУ. По этой причине функции управления ресурсами часто выносятся в отдельную подсистему командных сценариев, в которые администратор суперкомпьютера может вносить изменения, отражающие специфику вычислительной системы. Подобные подсистемы сценариев имеют в своем составе практически все современные СУЗ, в том числе широко распространенная система SLURM [2] и отечественная Система управления прохождением параллельных

заданий (СУППЗ) [3], применяемая в качестве СУЗ в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН).

Особенностью SLURM, в силу ее широкого распространения, является значительный объем программного обеспечения управления ресурсами (командных сценариев), разработанного производителями ВУ и системными интеграторами для поставляемого ими суперкомпьютерного оборудования. Со своей стороны значительно более специфичная СУППЗ за два десятилетия прочно вошла в практику обслуживания пользователей в МСЦ РАН и является основой исторически сложившейся экосистемы центра. По этой причине в МСЦ РАН, начиная с суперкомпьютера МВС-10П (2013 г.) [4], практикуется соВУестное функционирование СУППЗ и SLURM. На СУППЗ возлагаются функции управления заданиями, а часть функций управления ресурсами осуществляется путем применения готовых командных сценариев из состава SLURM.

Однако, при появлении в 2017 году ВУ на базе микропроцессоров Intel Xeon Phi KNL [5] ситуация изменилась. Архитектура KNL предполагает несколько альтернативных режимов работы, поддержку которых компания Intel осуществила не в виде командных сценариев подсистемы управления ресурсами, а внедрила непосредственно в ядро SLURM [6]. В результате управление ВУ на базе процессоров KNL без применения SLURM стало невозможным. Для сохранения сложившейся экосистемы МСЦ РАН стало необходимо осуществить интеграцию СУППЗ и SLURM. Решению этой задачи посвящена настоящая работа.

2. Сравнение СУППЗ и SLURM

СУППЗ и SLURM демонстрируют примерно одинаковые показатели качества [7] и схожие функции управления заданиями. Планирование заданий в обеих системах основано на схеме «справедливого» распределения ресурсов суперкомпьютера, которое обеспечивается за счет механизма динамических приоритетов пользователей. Суть динамических «справедливых» приоритетов заключается в следующем. За некоторый учетный период (в СУППЗ – один период, в SLURM – несколько периодов с разным весом) суммируется потребленный заданиями пользователя объем суперкомпьютерных ресурсов. Чем больше ресурсов потребил пользователь за учетный период, тем ниже его приоритет, при понижении приоритета задания пользователя начинают выполняться реже, потребление

ресурсов уменьшается, приоритет увеличивается – и так циклически повторяется процесс изменения приоритета. Такой механизм позволяет автоматически организовать саморегулирующуюся «справедливую» систему и делает невозможным захват большого объема ресурсов суперкомпьютера одним пользователем на длительное время.

В обеих системах задания планируются строго в соответствии со своим приоритетом, и никакое низкоприоритетное задание не может отложить старт высокоприоритетного. При этом для повышения утилизации ресурсов в обеих системах применяется т.н. алгоритм обратного заполнения (backfill). СУЗ позволяет запускать задания, стоящие в очереди позже (и, возможно, ниже приоритетом), если это не повлияет на время запуска заданий, стоящих в очереди ранее. Такое возможно, например, в случае, если для задания А, стоящего в очереди ранее, недостаточно ресурсов, и при этом задание Б, стоящее в очереди позже, успеет завершиться до момента, когда освободится достаточное количество ресурсов для запуска задания А.

Обе системы предоставляют пользователю возможность контролировать свое задание на всех стадиях выполнения: постановки в очередь, наблюдения в очереди с прогнозом времени начала выполнения, удаления из очереди, просмотра стандартного вывода, снятия задания с выполнения.

Отличительной особенностью планирования заданий в СУППЗ является наличие двух особых категорий заданий – отладочных и фоновых. Отладочные – это короткие по времени задания, которые запускаются исключительно в целях отладки на малом числе процессорных ядер. Фоновые – это задания с большим временем счета, которые могут прерываться системой. Для фонового задания пользователь должен явно указать квант – минимальное время счета фонового задания, в течение которого задание прерывать нельзя. Существование отладочных заданий и некоторого количества зарезервированных под них ВУ позволяет пользователям провести проверку работоспособности своего задания, не тратя длительное время в очереди. Фоновые задания эффективно используют время простоя ВУ, выполняются в течение длительного времени, не занимая при этом ресурсы ВС, если они нужны для расчета других заданий.

В SLURM задания не делятся на различные категории, но при этом существует возможность создания отладочной секции суперкомпьютера, на которой будут выполняться короткие по времени задания. Фоновые задания в SLURM не

поддерживаются.

Другим существенным отличием СУППЗ и SLURM является подход к запуску MPI-программ. СУППЗ для этого предлагает пользователю специальную команду `mpirun`, по синтаксису максимально приближенную к аналогичной команде, поставляемой различными реализациями MPI. Команда `mpirun` в СУППЗ по набору заданных параметров автоматически формирует паспорт задания, содержащий в том числе командный сценарий (также формируемый автоматически) инициализации MPI-программы на выделенных после прохождения задания через очередь ресурсов суперкомпьютера. При смене суперкомпьютера или реализации MPI команда `mpirun` корректируется системным программистом СУППЗ для обеспечения формирования адекватного командного сценария инициализации MPI-приложения. Внешний интерфейс и набор ключей команды `mpirun` при этом остается неизменным для пользователя, что, несомненно, является удобным.

В SLURM для инициализации MPI-программы пользователь должен разработать собственный командный сценарий. SLURM при этом предоставляет документированный набор переменных окружения, которыми можно воспользоваться в подготавливаемом сценарии. Значительная часть реализаций MPI содержит в своем составе готовые шаблоны сценариев инициализации MPI-программ для SLURM, что существенно упрощает работу пользователя.

Еще одним существенным различием двух рассматриваемых систем являются разные подходы к резервированию ресурсов в режиме IaaS. В СУППЗ для этого используется специальная команда `getnodes`, которая формирует и ставит в очередь «пустое» задание. После прохождения такого задания через очередь, ему выделяются требуемые ресурсы в виде изолированного мини-кластера, к которому предоставляется доступ пользователю. Последний после запуска задания может и должен совершить вход на предоставленные ресурсы и по своему усмотрению организовать на них вычислительный процесс. В SLURM для этой цели служит команда `salloc`, отличие которой от `getnodes` заключается в том, что `salloc` блокируется на время нахождения задания в очереди, а после прохождения задания через очередь автоматически предоставляет пользователю командную оболочку на выделенных ресурсах. Далее организация вычислительного процесса также полностью возлагается на усмотрение пользователя.

Если в области функций управления заданиями СУППЗ и SLURM можно наблюдать некоторый паритет систем, то в области управления ресурсами SLURM явно опережает отечественную систему. Причиной этого является широкое распространение SLURM, обеспечившее мощную поддержку как со стороны производителей оборудования, так и со стороны независимых разработчиков. Многие реализации MPI, например, от компании Intel, рассчитывают на запуск MPI-программ именно в окружении SLURM.

Многолетняя практика обслуживания пользователей в МСЦ РАН показала, что при всей схожести SLURM и СУППЗ, исторически сложившийся порядок работы с последней пользователей и администраторов определяет их предпочтения в пользу СУППЗ, несмотря на ее недостатки в области функций управления ресурсами. Пользователи разработали множество командных сценариев, автоматизирующих их работу с СУППЗ. Совокупность отношений пользователей, администраторов и СУППЗ вместе со стеком разработанного вспомогательного ПО составили уникальную рабочую и инструментальную среду, фактически сложившуюся за несколько лет в стабильную экосистему центра.

3. Принцип интеграции СУППЗ и SLURM

Как уже упоминалось, до определенного момента удавалось в СУППЗ применять функции управления ресурсами, предоставляемые SLURM. С появлением в МСЦ РАН суперкомпьютерной установки МВС-10П МП2 KNL [8] с вычислительными узлами на базе Intel Xeon Phi 7290 (KNL) такое разделение функций стало затруднительным, поскольку поддержка архитектуры Knights Landing компанией Intel была внедрена непосредственно в ядро SLURM. Суперкомпьютер МВС-10П МП2 KNL был предоставлен пользователям под управлением SLURM, но оказался слабо востребованным по сравнению с системами на базе СУППЗ. По этой причине было принято решение об интеграции СУППЗ и SLURM.

Основной принцип интеграции, представленный на рисунке 1, состоит в следующем. Все функции управления заданиями, включая командный интерфейс с пользователями и ведение очереди заданий, возлагаются на СУППЗ. Прошедшее через очередь СУППЗ задание помещается в SLURM, очередь которой всегда искусственно поддерживается пустой. При передаче задания

СУППЗ сообщает SLURM список ВУ (узлов суперкомпьютера), на которых следует запустить задание.

Поскольку очередь SLURM пуста, переданное задание немедленно поступает на выполнение на указанных в списке узлах. При этом за тем, чтобы один узел не выделялся одновременно более чем одному заданию, отвечает СУППЗ. Она же предоставляет пользователям привычный командный интерфейс.

SLURM при такой схеме непрозрачным для СУППЗ образом выполняет все функции управления ресурсами суперкомпьютера, независимо от того, реализованы они в системе командных сценариев или в ядре SLURM. При этом пользовательскому заданию становится доступным окружение SLURM, пользуясь которым можно осуществлять желаемую настройку суперкомпьютерного оборудования в полном соответствии с документацией и

рекомендациями производителя. Другими словами, при такой схеме интеграции получается оптимальное сочетание функций управления: СУППЗ обеспечивает привычные интерфейс и порядок работы с очередью заданий, SLURM позволяет использовать все богатство возможностей высокопроизводительной программно-аппаратной платформы.

Отдельно следует отметить, что предлагаемый подход позволяет в полной мере продолжать использовать подсистему сбора и обработки статистики СУППЗ. Эта подсистема разрабатывается и развивается в МСЦ РАН [9] на протяжении двух десятилетий и повседневно применяется администраторами суперкомпьютеров и руководством МСЦ РАН для анализа эффективности функционирования суперкомпьютерного оборудования.

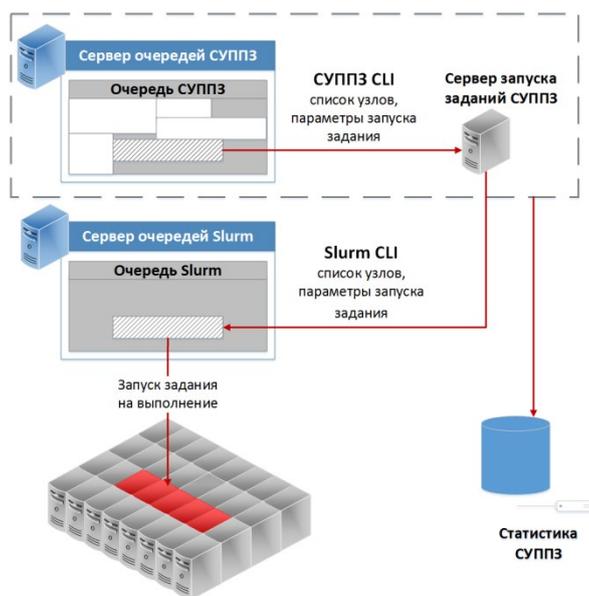


Рис. 1. Принцип интеграции СУППЗ и SLURM

4. Практическая реализация интеграции СУППЗ и SLURM

При практической реализации рассмотренного принципа интеграции (рис. 2) авторами был решен ряд научно-технических задач. Во-первых, был разработан механизм предотвращения несанкционированного доступа пользователей к SLURM в обход очереди СУППЗ. Во-вторых, в SLURM обеспечена поддержка фоновых заданий

СУППЗ. В-третьих, компенсировано различие в подходах к резервированию ресурсов. В-четвертых, осуществлена поддержка командных сценариев SLURM для обеспечения полноты функций управления ресурсами.

Для каждого поступившего на выполнение задания формируется специальный файл – паспорт задания СУППЗ. В паспорте перечисляются различные параметры запуска задания: максимальное время выполнения задания, количество запрашиваемых

процессорных ядер, тип используемых ресурсов, имя вычислительной системы и др. Одной из формируемых секций паспорта задания СУППЗ является секция [Batch]. Эта секция содержит код командного сценария (runmvs.bat) инициализации задания. Сценарий runmvs.bat будет запущен на первом из выделенных заданию вычислительных узлов суперкомпьютера. Для запуска runmvs.bat используется специальный командный файл /common/runmvs/sbin/runtask. Сценарий runmvs.bat инициирует запуск задания через SLURM на выделенных СУППЗ вычислительных узлах.

Содержанием сценария runmvs.bat определяется характер запуска задания, который осуществляется командами СУППЗ [3] mpirun, getnodes или mbatch. Команда mpirun отвечает за «штатный» запуск задания в виде MPI-программы и транслируется в сценарии runmvs.bat в команду SLURM srun [10].

Команда getnodes резервирует ресурсы, предоставляя их пользователю в режиме IaaS. В сценарии runmvs.bat getnodes транслируется в запуск команды SLURM salloc [11] специального вида, который позволяет выделить требуемые узлы суперкомпьютера на заданное время в полном соответствии с синopsisом команды getnodes. Команда mbatch позволяет запускать в виде задания сценарии SLURM и транслируется в runmvs.bat в аналогичную команду SLURM sbatch [12].

Взаимодействие СУППЗ и SLURM организуется через файл /usr/runmvs/.job, который создается на управляющей машине (head) от имени суперпользователя root командным файлом /usr/runmvs/runtask.slurm в момент выделения заданию вычислительных узлов. В файле /usr/runmvs/.job содержится набор параметров для запуска задания через SLURM и осуществления функций контроля запускаемых заданий.

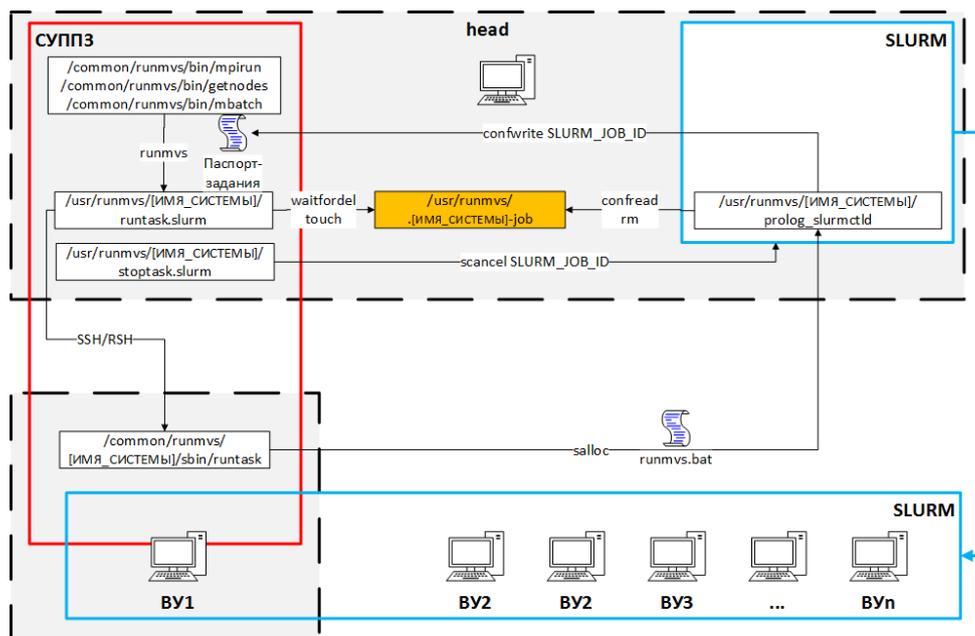


Рис. 2. Схема запуска заданий СУППЗ через SLURM

К числу этих параметров относятся:

- имя пользователя, выполняющего запуск задания через СУППЗ;
- идентификатор задания СУППЗ;
- уникальное хэш-значение файла-паспорта задания.

Хэш генерируется СУППЗ и записывается как в файл .job, так и в паспорт задания.

Поскольку значение хэша заранее неизвестно, и файл .job недоступен обычным пользователям, то проверка соответствия параметров паспорта задания их значениям, указанным в файле .job, обеспечит защиту от несанкционированного доступа пользователей к SLURM в обход очереди СУППЗ. Проверка осуществляется непосредственно перед запуском задания в

SLURM и выполняется сценарием `/usr/runmvs/prolog_slurmctld`, включенным в систему программ пролога и эпилога SLURM [13]. Программы пролога запускаются SLURM на управляющей машине и выделенных узлах перед стартом задания, а программы эпилога – после завершения задания. Сценарий пролога `prolog_slurmctld` сравнивает значения из файла `/usr/runmvs/.job` со значениями в файле-паспорте задания СУППЗ. Если хотя бы одно значение не совпадает, запуск задания отменяется. Не прошедшее через очередь СУППЗ задание не сможет обеспечить правильные значения параметров в файле `.job`, запуск такого задания в SLURM будет запрещен, что предотвратит доступ к SLURM в обход очереди СУППЗ.

Если все значения совпадают, то SLURM выполняет запуск задания на выполнение, удаляет файл `.job` и добавляет запись с идентификатором задания (`SLURM_JOB_ID`) в паспорт задания СУППЗ. Данный идентификатор используется для завершения по команде СУППЗ запущенного в SLURM задания. За корректное завершение задания отвечает сценарий `/usr/runmvs/stoptask.slurm`, применяемый в том числе для принудительного прерывания фоновых заданий СУППЗ в SLURM по истечении квантов времени.

Если СУППЗ управляет несколькими независимыми вычислительными системами, то для каждой из них создается отдельный раздел SLURM и комплект командных файлов `runtask.slurm`, `stoptask.slurm`, `prolog_slurmctld` и `runtask`.

5. Заключение

В представленной работе авторами решена задача интеграции систем управления суперкомпьютерными заданиями и ресурсами SLURM и СУППЗ. Поступающее в интегрированную систему задание проходит через очередь СУППЗ, которая осуществляет планирование заданий. После прохождения очереди СУППЗ задание передается в SLURM, где немедленно запускается в среде и окружении, предоставляемыми SLURM. Интегрированная система сочетает порядок обслуживания пользователей, принятый в СУППЗ и составляющий экосистему МСЦ РАН, и возможности управления суперкомпьютерными ресурсами, включенные производителями оборудования в функции SLURM. Разработанная авторами интегрированная система внедрена в Центре коллективного пользования вычислительными ресурсами МСЦ РАН – филиала ФГУ ФНЦ НИИСИ РАН и обслуживает пользовательские задания на суперкомпьютере МВС-10П МП2 KNL, установленном в МСЦ РАН.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований по теме «Разработка архитектур, системных решений и методов для создания и использования высокопроизводительных вычислительных комплексов, в том числе гетерогенных суперкомпьютеров и нетрадиционных архитектур микропроцессоров» № 0065-2019-0016.

Integration of Job Management Systems SLURM and SUPPZ

A.V. Baranov, E.A. Kiselev

Abstract: The paper discusses the integration problem of the widespread job management system SLURM and the domestic system called SUPPZ used at the Joint Supercomputer Center of the Russian Academy of Sciences (JSCC RAS). In the integrated system, job queue and job scheduling functions are assigned to SUPPZ, and supercomputer resource management functions are assigned to SLURM. The SLURM queue constantly maintained empty. Jobs passed through the SUPPZ queue are transferred to SLURM. In this case jobs start immediately in the SLURM environment. The study revealed differences in the approaches to managing jobs and resources in SLURM and SUPPZ. The revealed differences were compensated by the authors during the integration process. To prevent unauthorized user access to SLURM bypassing the SUPPZ, the authors developed a special software access control subsystem. As a result of the work, the historically formed JSCC RAS ecosystem based on the SUPPZ was preserved, and at the same time, the whole range of supercomputer equipment management capabilities implemented in SLURM became available.

Keywords: HPC, supercomputer, job scheduling, workload manager, SLURM, SUPPZ, Intel Xeon Phi, Intel Knights Landing

Литература

1. A. Reuther et al.: Scalable system scheduling for HPC and big data. “Journal of Parallel and Distributed Computing”, V. 111 (2018), 76–92. DOI: 10.1016/j.jpdc.2017.06.009
2. A.V. Yoo, M.A. Jette, M. Grondona. SLURM: Simple Linux Utility for Resource Management. “Lecture Notes in Computer Science”, V. 2862 (2003), 44–60. DOI: 10.1007/10968987_3
3. Система управления прохождением параллельных заданий. Руководство программиста (пользователя), (2016). <http://www.jscc.ru/wp-content/uploads/2017/06/SUPPZ-user-guide-2016.pdf> (дата обращения: 04.10.2019).
4. Уникальная научная установка «Суперкомпьютер MBC-10П». <http://www.jscc.ru/ckp/mvs-10p/> (дата обращения: 04.10.2019).
5. A. Sodani. Knights landing (KNL): 2nd Generation Intel® Xeon Phi processor. “2015 IEEE Hot Chips 27 Symposium (HCS)”, Cupertino, CA, (2015), 1–24. DOI: 10.1109/HOTCHIPS.2015.7477467
6. Intel Knights Landing (KNL). User and Administrator Guide. https://slurm.schedmd.com/intel_knl.html (дата обращения: 04.10.2019).
7. А.В. Баранов, Д.С. Ляховец. Сравнение качества планирования заданий в системах пакетной обработки SLURM и СУППЗ. «Научный сервис в сети Интернет: все грани параллелизма: Труды Международной суперкомпьютерной конференции», Россия, Новороссийск, 23–28 сентября 2013, 410–414. <http://agora.guru.ru/abrau2013/pdf/410.pdf> (дата обращения: 04.10.2019).
8. Вычислительные ресурсы МСЦ РАН. MBC-10П MPI2 KNL. <http://www.jscc.ru/resources/hpc/#item1463> (дата обращения: 04.10.2019).
9. А.В. Баранов, Е.А. Киселёв, Е.С. Кормилицин, В.Ф. Огарышев, П.Н. Телегин. Модернизация подсистемы сбора и обработки статистики центра коллективного пользования вычислительными ресурсами МСЦ РАН. «Труды НИИСИ РАН», Т. 8 (2018), № 4, 136–144. DOI: 10.25682/NIISI.2018.4.0016
10. SLURM commands – srun. <https://slurm.schedmd.com/srun.html> (дата обращения: 04.10.2019).
11. SLURM commands – salloc. <https://slurm.schedmd.com/salloc.html> (дата обращения: 04.10.2019).
12. SLURM commands – sbatch. <https://slurm.schedmd.com/sbatch.html> (дата обращения: 04.10.2019).
13. Prolog and Epilog Guide. https://slurm.schedmd.com/prolog_epilog.html (дата обращения: 04.10.2019).

Полуклассическая 3D модель КНИ МОП нанотранзистора с ультратонкой рабочей областью

Н.В. Масальский

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, volkov@niisi.ras.ru

Аннотация. Обсуждается трехмерная модель наноразмерного КНИ МОП транзистора на основе самосогласованного решения 3D уравнение Пуассона и Шредингера, в которой проявляются преобладающие коротко-канальные и квантово-механические эффекты, ограничивающие масштабирование топологии транзистора. Данные уравнения решаются методом разделения переменных. Численно проанализирован с учетом дрейфового-диффузионного переноса заряда ток в прототипе транзистора с ультра тонкой низколегированной рабочей областью. Показано, что есть возможность частичной компенсации коротко-канальных и квантово-механических эффектов.

Ключевые слова: КНИ КМОП нанотранзистор, асимметричная структура, 3D уравнение Пуассона, 2D уравнение Шредингера, метод разделения переменных

1. Введение

Усовершенствование кремниевых полевых нанотранзисторов, связанное с прямым масштабированием его топологии, для достижения высокого быстродействия и повышения степени интеграции, сталкивается с принципиальными физическими ограничениями [1]. Увеличивается напряженность электрического поля и уменьшается количество электронов, участвующих в переносе заряда. Проявляются ранее не учитываемые физические эффекты - квантово-механические эффекты, которые оказывают существенное влияние, когда размерность транзистора становится сопоставимой с длиной волны де Бройля для электрона [2]. Это происходит из-за того что, длина свободного пробега носителей существенно зависит от толщины пленки кремния и резко снижается по мере ее уменьшения [3, 4]. Это приводит к уменьшению плотности ионизированной примеси в этой же пленке. Пока кремниевая пленка достаточно «толстая», чтобы поддержать 3D транспорт, можно рассматривать «классический случай». Критическая «толщина» (правильнее сказать «эффективный объем») кремниевой рабочей области, при которой «отменяются» постулаты классической физики, оказывается фундаментальной фактором [5, 6]. Топология транзистора начинает сильно влиять на его электрические характеристики. Поэтому конструктивно-технологические

особенности прибора могут фундаментальным образом изменить требования к его модели вплоть до перехода к трехмерному квантово-механическому представлению [4]. Одной из самых сложных задач является переход из 2D- в 3D- состояние (и наоборот) и его имплементация в математические формы и вычислительные алгоритмы.

Перспективной стратегией разработки моделей наноразмерных МОП-структур является комбинирование уравнений Пуассона, Больцмана, Шредингера с разными пространственными представлениями при условии самосогласованности их решений и учета максимально возможных механизмов рассеяния [7].

В данной работе рассматривается трехмерная модель для наноразмерного КНИ МОП транзистора на основе решения 3D уравнение Пуассона и Шредингера, в которой проявляются преобладающие коротко-канальные и квантово-механические эффекты, ограничивающие масштабирование топологии транзистора. В работе уравнения решаются методом разделения переменных. В этом случае появляется возможность получить квазианалитические решения и достоверно оценить возможность линеаризации дискретных уравнений исходной модели. Этот подход позволяет существенно повысить вычислительную эффективность алгоритмов, так как исключается необходимость большого числа расчётов.

2. 3D уравнение Пуассона

Рис. 1 иллюстрирует трехмерное представление анализируемой приборной структуры, где на кремниевой пластине (Si) расположен слой оксида кремния (SiO₂), на верхней грани которого помещена транзисторная структура. Координатная ось вдоль рабочей области (C) – ось y, по глубине – ось x, ось z направлена по ширине рабочей области. Внутренние края истока (S) и стока (D) расположены в y=0 и y=L_g, соответственно, где, L_g – длина рабочей области. Фронтальная (верхняя) и обратная (нижняя) поверхности Si-SiO₂ расположены в x=0 и x=t_s, где t_s – толщина рабочей области. t_f и t_b – толщины подзатворного оксида фронтального и обратного затворов, соответственно. При этом t_f << t_b – мы рассматриваем асимметричный случай. U_{gf} и U_{gb} – напряжение на фронтальном и обратном затворах, соответственно. Поверхности Si-SiO₂ боковых граней канала стены расположены в z=0 и z=W, где W – ширина рабочей области. Площади L_SxW_S = W_DxL_D занимаемые истоком и стоком одинаковые.

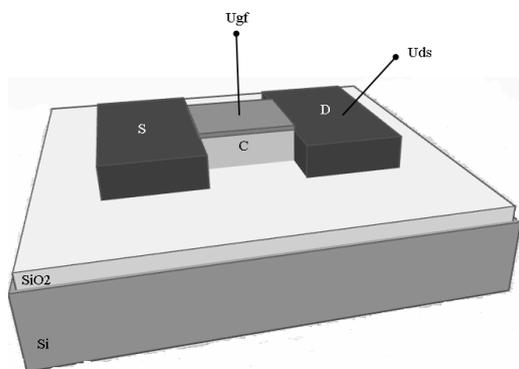


Рис. 1. Эскизное 3D представление транзистора

Для того чтобы найти распределение потенциала необходимо решить уравнение Пуассона с соответствующими граничными условиями, что реализуется при помощи метода разделения переменных [7]. Тогда 3D уравнение Пуассона в области рабочей области полностью обедненного КНИ МОП

транзистора запишем в хорошо известном виде:

$$\Delta\varphi = \frac{qN_A}{\epsilon_S}, \quad (1)$$

где N_A – концентрация легирования, и $\varphi(x, y, z)$ является потенциалом в определенной точке (x, y, z) в рабочей области.

Чтобы решить уравнение (1), оно разделено на 1D уравнение Пуассона

$$\frac{d^2\varphi_l(x)}{dx^2} = \frac{qN_A}{\epsilon_S}, \quad (2)$$

2- и 3 D уравнения Лапласа в виде:

$$\frac{d^2\varphi_s(x, y)}{dx^2} + \frac{d^2\varphi(x, y)}{dy^2} = 0 \quad (3)$$

$$\frac{d^2\varphi_v(x, y, z)}{dx^2} + \frac{d^2\varphi_v(x, y, z)}{dy^2} + \quad (4)$$

$$\frac{d^2\varphi_v(x, y, z)}{dz^2} = 0$$

где общее решение для потенциала представляет суперпозицию решений трех представленных выше уравнений. Тогда справедливо следующее выражение [8, 9]:

$$\varphi(x, y, z) = \varphi_l(x) + \varphi_s(x, y) + \varphi_v(x, y, z) \quad (5)$$

Решение для $\varphi_l(x)$. Для 1D уравнения Пуассона (2) граничные условия запишем в виде:

- для фронтальной границы

$$\varphi_l(x) - \frac{t_f}{\epsilon_{ox}} \epsilon_S \frac{d\varphi_l}{dx} \Big|_{x=0} = U_{gf} - U_{FBf}$$

- для обратной границы

$$\varphi_l(t_s) + \frac{t_b}{\epsilon_{ox}} \epsilon_S \frac{d\varphi_l}{dx} \Big|_{x=t_s} = U_{gb} - U_{FBb},$$

где $\epsilon_S, \epsilon_{ox}$ – диэлектрическая проницаемость рабочей области и подзатворного диэлектрика фронтального и обратного затворов, соответственно, U_{FBf}, U_{FBb} – напряжения плоских зон фронтального и обратного затворов. Тогда решение (2) можно записать так:

$$\varphi_l(x) = \varphi_l(t_s) + E_{sb}(t_s - x) + \frac{qN_A}{2\varepsilon_s}(t_s - x)^2, \quad (6)$$

$$\text{где } E_{sb} = -\left. \frac{d\varphi_l}{dx} \right|_{x=t_s}$$

Решение для $\varphi_s(x, y)$. Для 2D уравнения Лапласа (3) граничные условия запишем в виде:

$$\varphi_s(0, y) - \frac{t_f}{\varepsilon_{ox}} \varepsilon_s \frac{d\varphi_s}{dx} \Big|_{x=0} = 0$$

$$\varphi_s(t_s, y) + \frac{t_b}{\varepsilon_{ox}} \varepsilon_s \frac{d\varphi_s}{dx} \Big|_{x=t_s} = 0,$$

$$\varphi_s(x, 0) = U_{bi} - \varphi_l(x)$$

$$\varphi_s(x, L_g) = U_{bi} + U_{ds} - \varphi_l(x)$$

где u_{bi} – контактная разность потенциалов со стороны стока и истока.

Для сформулированных граничных условий аналитическое решение 2D уравнения Лапласа (3) можно представить в следующем виде:

$$\varphi_s(x, y) = \left[\frac{1}{\sinh(\gamma L_g)} \right] [\Phi_1^{(2)} \sinh(\gamma y) + \Phi_2^{(2)} \sinh(\gamma(L_g - y))], \quad (7)$$

$$[\sin(\gamma x) + \frac{\varepsilon_s}{\varepsilon_{ox}} (\gamma t_f \cos(\gamma x))]$$

где $\gamma = \frac{t_f}{t_b}$, $\Phi_1^{(2)}$, $\Phi_2^{(2)}$ – нормировочные

функции, которые рассмотрены в [10].

Решение для $\varphi_v(x, y, z)$. Для 3D уравнения Лапласа (4) граничные условия запишем в виде:

$$\varphi_v(0, y, z) - \frac{t_f}{\varepsilon_{ox}} \varepsilon_s \frac{d\varphi_v}{dx} \Big|_{x=0} = 0$$

$$\varphi_v(t_s, y) + \frac{t_b}{\varepsilon_{ox}} \varepsilon_s \frac{d\varphi_v}{dx} \Big|_{x=t_s} = 0,$$

$$\varphi_v(x, 0, z) = 0$$

$$\varphi_v(x, L_g, z) = 0$$

$$\varphi_v(x, y, 0) - \frac{t_n}{\varepsilon_{ox}} \varepsilon_s \frac{d\varphi_v}{dz} \Big|_{z=0} =$$

$$U_{gf} - U_{FBf} - \varphi_l(x) - \varphi_s(x, y)$$

$$\varphi_v(x, y, 0) + \frac{t_n}{\varepsilon_{ox}} \varepsilon_s \frac{d\varphi_v}{dz} \Big|_{z=W} =$$

$$U_{gf} - U_{FBf} - \varphi_l(x) - \varphi_s(x, y)$$

где t_n – толщина бокового изолирующего слоя.

Решение 3D уравнения Лапласа запишем так.

$$\varphi_v(x, y, z) =$$

$$\Phi_1^{(3)} [\sinh(\chi(W - z)) + \sinh(\chi z)] \left[\frac{\sin(\alpha(y - L_g))}{\cos(\alpha L_g)} \right], \quad (8)$$

$$[\sin(\beta x) + \frac{\varepsilon_s}{\varepsilon_{ox}} (\beta t_f \cos(\beta x))]$$

где $\Phi_1^{(3)}$ – нормировочная функция [10], α, β, χ – подгоночные параметры, где $\alpha^2 + \beta^2 = \chi^2$

3. Уравнение Шредингера

Ток транзистора рассчитывается исходя из подхода, основанного на дискретном представлении уравнения Шредингера [5]. С целью учета влияния механизмов рассеяния на ток стока использовалась модифицированное выражение Тсу–Есаки:

$$I_{ds} = \frac{2q}{h} \int_0^{E_f} dE |t(E, U_{gf})|^2 D(E) \quad (9)$$

$$[f(E) - f(E + U_{ds})]$$

где I_{DS} – ток стока, t – коэффициент прохождения электронной волны, определяемый с помощью матрицы рассеяния [11], $f(E)$ – функция распределения Ферми–Дирака, h – постоянная Планка, E_f – энергия Ферми, $D(E)$ – коэффициент, характеризующий затухание электронной волны вследствие рассеяния.

Решение уравнения Шредингера в каждом сечении рабочей области раскладывается в суперпозицию конечного поперечных мод, включая и затухающие моды [12]. Тогда волновую функцию $\psi(x, y, z)$ можно представить в виде суммы

всех поперечных мод $\phi_i(x, z)$ с соответствующими амплитудами:

$$\psi(x, y, z) = \sum_{i=1}^{\infty} c_i(y) \phi_i(x, z), \quad (10)$$

где функции $\phi_i(x, z)$ удовлетворяют 2D уравнению Шредингера -

$$-\left[\frac{\partial^2}{2m_x \partial x^2} - \frac{\partial^2}{2m_z \partial z^2} \right] \phi_i(x, z) = \varepsilon_i \phi_i(x, z)$$

здесь m_x , m_y , m_z – эффективные массы электрона в соответствующих направлениях.

В данное представление включены только конечное число мод и оно связывает амплитуды со стороны истока с амплитудами на стоке. В приближении прямоугольной квантовой ямы, решения этого уравнения хорошо известны [5, 6]:

$$\phi_i(x, z) = \frac{2}{\sqrt{W t_s}} \sin \frac{\pi n z}{W} \sin \frac{\pi m x}{t_s} \quad \text{и}$$

$$\varepsilon_i = \left(\frac{\pi \hbar}{\sqrt{2}} \right)^2 \left(\frac{n^2}{m_z W^2} + \frac{m^2}{m_x t_s^2} \right).$$

Опуская несложные математические выкладки, получаем уравнения для амплитуд $c_i(y)$

$$-\frac{\partial^2 c_i(y)}{2m_y \partial y^2} + \sum_j U_{ij}(y) c_j(y) = (E - \varepsilon_i) c_i(y), \quad (11)$$

$$(E - \varepsilon_i) c_i(y)$$

где

$$U_{ij}(y) = \int dx \int dz \phi_i(x, z) U(x, y, z) \phi_j(x, z)$$

- матричный элемент потенциала.

Граничные условия для амплитуды $c_i(y)$ на левой (со стороны истока) и правой (со стороны стока) границах в виде плоских бегущих волн справа налево и слева направо [5]. Они задаются так, что только со стороны истока падает волна i -той моды с единичной амплитудой [12]. После их подстановки получаем систему линейных уравнений, решениями которой являются амплитуды c_i , а затем вычисляются коэффициенты t . Они необходимы в (9) при вычислении тока I_{ds} . Новое распределение зарядов по объему рабочей области необходимо для корректировки самосогласованного потенциала [6]. Эти итерации продолжаются до достижения заданной точности вычислений.

4. Общий ток транзистора

Транспорт носителей в анализируемой асимметричной структуре с широкой ультра тонкой рабочей областью в модельном представлении является суперпозицией дрейфовой и баллистической компонент [13]. Такие структуры отличаются возможностью подавления ККЭ за счет КМЭ. Для них существенно 3D рассмотрение. При 2D и 3D подходах для одного и того же уровня Ферми общая численность носителей будет разной из-за разной плотности состояний [6]. Этот механизм влияет на заряд носителей в инверсионном слое. Таким образом, требуется большее напряжение на затворе для того, чтобы заполнить 2-D инверсионный слой, чтобы иметь то же число носителей как соответствующая 3-D система. Этим и достигается равновесие порогового напряжения транзистора- эффект roll-off за который отвечают ККЭ компенсируется эффектом roll-on за который отвечают КМЭ.

Общий ток транзистора будет представлять суммой вкладов баллистической и дрейфовой компонент. Тогда плотность тока можно представить так:

$$j(y) = q(n_b(y)v(y) + n_s(y)\mu E_y(y)), \quad (12)$$

где $n_b(y)$, $n_s(y)$ - баллистическая и дрейфовая концентрация носителей, соответственно, $v(y)$ - баллистическая компонента скорости, $E_y(y)$ - напряженность поля, μ - подвижность носителей.

Для моделирования дрейфовой компоненты тока транзистора использовались 2D аналитическое выражение для потенциала, видоизмененное выражение для скорости насыщения носителей и модель подвижности Lombardi [14].

Для тестовых расчетов вольт-амперных характеристик (ВАХ) прототипа транзистора n-типа выбраны следующие топологические параметры: $L_g=20$ нм, $W=50$ нм, $t_{Si}=3.8$ нм, $t_f=1$ нм, $t_b=20$ нм, $N_A=1 \times 10^{15} \text{см}^{-3}$, $L_S=L_D=50$ нм, $W_S=W_D=120$ нм, $N_{SD}=2 \times 10^{20} \text{см}^{-3}$. Следует отметить, что эффективная длина рабочей области L_{eff} сопоставима длиной свободного пробега электрона в объемном кремнии. Области

стока и истока также задействованы в расчетах. В данном случае они рассматриваются в качестве накопителя свободных носителей заряда. Результаты расчетов – зависимости $I_{ds}(U_{ds})$ и $I_{ds}(U_{gf})$ при $U_{gb}=0$ показаны на рис. 2.

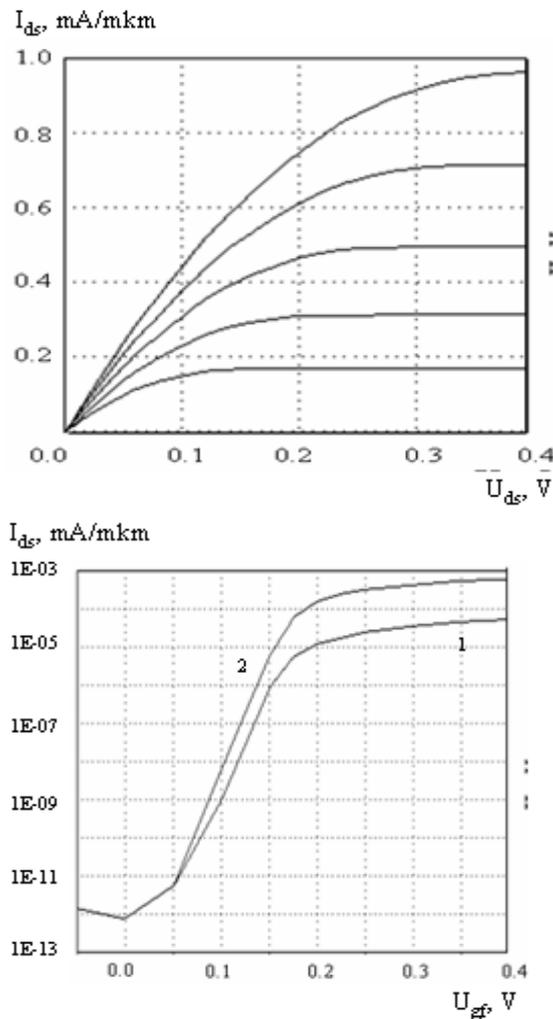


Рис. 2. Результаты моделирования:
 - верхний ВАХ $I_{ds}(U_{ds})$ при $U_{gf}=0.2; 0.25;$
 $0.3; 0.35; 0.4$ В;
 - нижний ВАХ $I_{ds}(U_{gf})$ при 1- $U_{ds}=0.05$; 2 -
 $U_{ds}=0.4$ В

Полученные данные позволяют сделать вывод о высоком уровне тока открытого транзистора при отличительно низких управляющих напряжениях при том, что ККЭ существенно подавляются и обеспечивается контроль подпорогового тока в состоянии OFF. Следует отметить, что сопоставление полученных данных с результатами расчетов по классической баллистической модели (КБМ) [5, 6]

показывает, что уровень тока нашей модели превышает примерно в два раза ток по КБМ при одинаковых условиях. Пороговые напряжения для случая трех моделей следующие 35 мВ – наша модель, 120 мВ – КБМ и -85 мВ дрейфово-диффузионная модель (ДДМ) [15]. Отличительной особенностью является достаточно низкое значение напряжения U_{ds} (~ 0.4 В), при котором линейный участок зависимости переходит в область насыщения. При этом отношение токов I_{on}/I_{off} составляет более 7 порядков величины. Эти два параметра в рамках ДДМ принимают значения 0.82 В и 4, соответственно.

Однако такая ситуация для ультра тонких слабо легированных кремниевых транзисторных наноструктур реализуется только при корректном учете зонной структуры, значимых механизмов рассеяния (на фононах, кулоновское, ударная ионизация) и сильно нелокальных свойств транспорта [7, 10]. Очень важно, что во-первых, количественное взаимное подавление ККЭ и КМЭ достигается в результате неустойчивого равновесия между многими эффектами, во-вторых, даже небольшое отклонение отдельной аппроксимации способно изменить электрофизические характеристики транзистора в несколько раз.

Работа выполнена в рамках Государственного задания по проведению фундаментальных научных исследований по теме (проекту) 0065-2019-0001.

Заключение

Разработана трехмерная модель двух затворного наноразмерного КНИ МОП транзистора на основе самосогласованного решения 3D уравнений Пуассона и Шредингера и уравнения непрерывности, которые связаны между собой распределением потенциала в рабочей области транзистора. Особенностью модели является учет доминирующих в рассматриваемых транзисторных структурах коротко-канальных и квантово-механических эффектов, которые ограничивают масштабирование топологии транзистора. Разработан метод решения 3D уравнений Пуассона и Шредингера при помощи разделения переменных. Численно проанализирован прототип транзистора с ультра тонкой низколегированной рабочей областью. Показано, что есть возможность частичной компенсации коротко-канальные и квантово-механические эффектов.

Semiclassical 3D model SOI MOSFET nanotransistor with ultra thin channel

N. Masalsky

Abstract. A three-dimensional model of a nanoscale SOI MOSFET based on a self-consistent solution of the 3D Poisson and Schrödinger equation is discussed, in which the prevailing short-channel and quantum mechanical effects limiting the scaling of the transistor topology are manifested. These equations are solved by separating the variables. The current in the prototype transistor with an ultra-thin low-doping channel area is numerically analyzed taking into account the drift-diffusion charge transfer. It is shown that there is a possibility of partial compensation of short-channel and quantum-mechanical effects.

Keywords: SOI CMOS nanotransistors, asymmetric structure, 3D Poisson equation, 2D Schrodinger equation, variable separation method

Литература

1. Y. Taur, T. Ning. Fundamentals of modern VLSI devices. Cambridge, U.K, Cambridge University Press, 1998.
2. S. Datta. Electronic Transport in mesoscopic systems. Cambridge, U.K, Cambridge University Press, 1995.
3. J.H. Davies. The physics of low dimensional semiconductors. New York, Plenum, 1998.
4. M. Gilbert, D. Ferry. Efficient quantum three-dimensional modeling of fully depleted ballistic silicon-on-insulator metal-oxide-semiconductor field-effect-transistors. «J. Appl. Phys». V. 95(2004), 7954-7960.
5. J.-H. Rhew, Z. Ren, M. S. Lundstrom. A numerical study of ballistic transport in a nanoscale MOSFET. «Solid-State Electronics», V. 46(2002), 1899-1906.
6. J. Rhew. Physics and simulation of quasi-ballistic transport in nanoscale transistor. A Thesis of PhD, Purdue University, Desember 2003.
7. Н.А. Агафонов, Н.В. Масальский, В. В. Осипов, А.В. Родителей. Проблемы моделирования переноса в квазипланарных КНИ КМОП нанотранзисторах. «Труды НИИСИ РАН», Т. 8(2018), 139-147.
8. P. Mani Manoj, K . Pandey. Analytical solution of 3D Poisson equation using separation of variable method. «International Journal of Application or Innovation in Engineering & Management», V. 2(2013), 571-576.
9. S. Swatski, S. Khuvis, M. Gobbert. A comparison of solving the Poisson equation using several numerical methods in Matlab and Octave on the Cluster maya. Department of Mathematics and Statistics, University of Maryland, Baltimore County. 2014. Technical Report HPCF-2014-10, URL:www.umbc.edu/hpcf
10. P. Mani Manoj, K . Pandey. Surface potential and threshold voltage model of fully depleted narrow channel SOI MOSFET using analytical solution of 3D Poisson's equation. «J. of nano- and electronic physics», V. 7(2015), 2002-2007.
11. A. Amara, O. Rozeau. Planar double-gate transistor. Springer, 2009.
12. A. Orlikovsky, V. Vyurkov, V. Lukichev, I. Semenikhin, A. Khomyakov. All quantum simulation of ultrathin SOI MOSFET in nanoscaled semiconductor-on-insulator structures and devices. Springer. 2007.
13. F. Liu, F. He, Y. Fu, J. Hi, W. Bian, Y. Song, X. Zhang, M. Chan. Generic carrier-based core model for undoped four-terminal double-gate MOSFETs valid for symmetric, asymmetric, and independent-gate-operation modes. «IEEE Trans. Electron Devices», V. 55(2008), 816-826.
14. S. Cristoloveanu, N. Rodriguez, F. Gamiz. Why the universal mobility is not. "IEEE Trans. Electron Devices", V. 57(2010), 1327-1333.
15. F. Schwierz, H. Wong, J. Liou. Nanometer CMOS. Singapore, Pan Stanford Publishing, 2010.

Анализ теплового и вибрационного режимов конструкции электронного модуля с теплоотводами в виде гибкого кондуктивного теплоотвода и единой теплопроводной металлической пластины

А.А. Воробьев¹, А.Г. Мадера², А.О. Шаманов³

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, vorobiev@cs.niisi.ras.ru

² ФГУ ФНЦ НИИСИ РАН, Москва, Россия, omm@niisi.ras.ru

³ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, formal432@niisi.ras.ru

Аннотация: В статье рассматриваются результаты компьютерного моделирования тепловых и вибрационных режимов конструкции электронного модуля с различными вариантами конструкций теплоотвода от микросхем, установленных на печатной плате. В качестве вариантов теплоотводов рассматриваются две конструкции, а именно, в виде гибкого кондуктивного теплоотвода и традиционной конструкции в виде единой для всех охлаждаемых микросхем, теплопроводной металлической пластины. Результаты моделирования показывают, что гибкий кондуктивный теплоотвод по сравнению с единой теплопроводной металлической пластиной, обладает существенно более высокими теплоотводящими свойствами и при этом обеспечивает индивидуальное прижатие каждой охлаждаемой микросхемы к теплоотводу и достаточную для практики устойчивость к вибрационным воздействиям.

Ключевые слова: микросхема, теплоотвод, электронный модуль, печатная плата, гибкий кондуктивный теплоотвод, металлическая пластина-теплоотвод, тепловой режим, вибрации, контактное сопротивление

1. Введение

Анализ теплового режима конструкции гибкого кондуктивного теплоотвода (ГКТ) [1, 2] показал существенно большую его эффективность при отводе тепла от мощных микросхем (МС) процессора в сравнении с традиционными теплоотводами, выполненными в виде единой теплопроводной металлической пластины (ЕТМП). В работе [1, 2] было показано, также, что одним из основных достоинств ГКТ является его конструктивная гибкость, позволяющая обеспечивать индивидуальный и эффективный тепловой контакт между различными тепловыделяющими элементами (МС), расположенными на одной печатной плате, и теплоотводящими / теплопоглощающими (теплоотводы, охлаждательные элементы) элементами конструкции электронного модуля. В то время как традиционные теплоотводы в виде ЕТМП не позволяют обеспечить низкие значения контактных тепловых сопротивлений между ЕТМП и всеми тепловыделяющими элементами (МС), размещенными на одной печатной плате в электронном модуле. Указанное обстоятельство, как показано в работах [3, 4, 5], обуславливается

статистическим технологическим разбросом как при изготовлении, так и монтаже МС на печатной плате, и приводит к случайному разбросу толщины зазоров между различными МС и ЕТМП неустраимому в принципе, в том числе и с использованием высокотеплопроводных паст и компаундов. Последнее понятно принимая во внимание тот факт, что статистический технологический разброс изготовления МС приводит к статистическому разбросу параметров конструкции корпуса МС (высоты корпуса, контактирующих с теплоотводами площади, неплоскостности, волнистости), а статистический технологический разброс при монтаже различных МС на единой печатной плате, – разбросу зазоров как между корпусами МС и печатной платой, так и корпусами МС и поверхностью ЕТМП, причем последняя обладает собственным разбросом неплоскостности и волнистости, присущих различным экземплярам ЕТМП. Таким образом, конструкция ЕТМП не в состоянии обеспечить эффективный тепловой контакт одновременно со всеми тепловыделяющими МС. Проблема качественного теплового контакта конструкции теплоотвода в виде ЕТМП со всеми МС установленными в

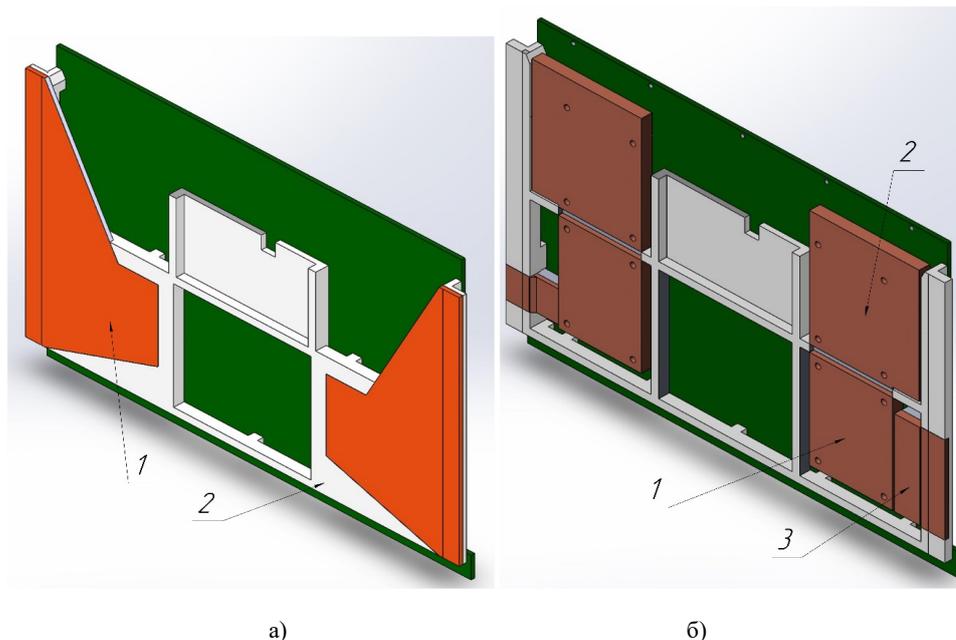


Рис. 1. Конструкция теплоотвода в электронном модуле в виде (а) единой теплопроводной металлической пластины (ЕТМП) и (б) гибких кондуктивных теплоотводов (ГКТ).
Обозначения (а): 1 – конструкция теплоотвода в виде ЕТМП, 2 – конструктивная рамка; (б): 1 – интерфейс источника МС процессора, 2 – интерфейс стока конструкции ГКТ, 3 – интерфейс стока двух МС

электронном модуле одновременно не решена до сих пор. Не решает эту проблему и конструкция ЕТМП, разделенная на две части, одна из которых (1, рис. 1, а) призвана отвести тепло от четырех МС одновременно, размещенных в пределах конструктивной рамки (2, рис. 1, а), а другая – «сбросить» тепло от МС к клиновому механизму. Тем не менее проблема теплового контакта между тремя МС, расположенным под теплоотводом ЕТМП и клиновым механизмом остается и в этом случае.

В настоящей статье в дополнение к анализу тепловых режимов конструкции ГКТ [2] рассматривается анализ конструкции ГКТ на воздействие вибрации. Анализ проводится применительно к конкретной конструкции электронного модуля, включающего восемь МС по четыре с каждой стороны (МС процессора и три МС питания №1, №2, №3), симметрично установленных на печатной плате (рис. 1, б). Теплоотвод от корпусов МС осуществляется с помощью ГКТ, один из которых предназначен для отвода тепла от МС процессора, другой ГКТ – от трех менее мощных МС питания. Интерфейсы источника ГКТ установлены на корпусе мощной МС процессора (1, рис. 1, б) и на корпусах трех менее мощных МС питания (3, рис. 1, б), а интерфейс стока ГКТ (2, рис. 1, б) служит для отвода тепла от МС процессора и посредством гибких звеньев связывает интерфейс источника мощной МС процессора с клиновым механизмом. В качестве реперного

сравнения тепловых и вибрационных режимов электронного модуля с применением и без применения ГКТ, использовалась конструкция отвода тепла от МС в виде ЕТМП, приведенного на рис. 1, а.

Компьютерное моделирование теплового и вибрационного режимов электронного модуля для обеих конструкций теплоотвода (рис. 1, а и б) проводилось по программному комплексу SolidWorks с конструктивными параметрами ГКТ принятыми в работе [2].

2. Результаты моделирования тепловых режимов

Результаты моделирования температуры корпуса МС процессора и МС питания для обеих конструкций теплоотвода (рис. 1, а и б), приведены в табл. 1.

Таблица 1. Температуры корпусов МС с различными конструкциями теплоотводов

Тепловыделяющие элементы	Температуры (°С) МС для теплоотвода в виде	
	ЕТМП	ГКТ
МС процессора	90,25	87,05
МС №1	81,44	76,9
МС №2	81,46	76,9
МС №3	81,4	76,5

Для моделирования вибрационных воздействий на устойчивость конструкции

электронного модуля в обоих исполнениях (рис. 1, а и б).

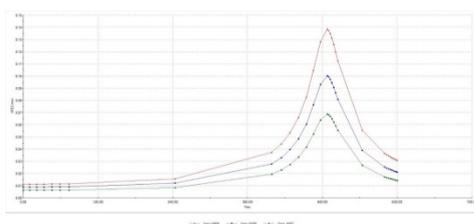
3. Результаты моделирования вибрационных режимов

Наиболее критическим режимом электронного модуля с точки зрения неустойчивости к воздействию синусоидальной вибрации является воздействие ускорения по нормали (оси z) к печатной плате, нежели в ее продольных направлениях. Поэтому расчёт воздействия вибрации проводился именно для наиболее критического случая, то есть для вибрационных воздействий по оси z . Параметры воздействия составляли: амплитуда ускорения $5g$, m/c^2 ; диапазон частот 1 – 500 Гц. Закрепление электронного модуля и теплоотвод от МС, установленных на печатной плате, осуществлялось с двух сторон, по противоположным краям платы с помощью клиновых механизмов. Механические параметры многослойной печатной платы

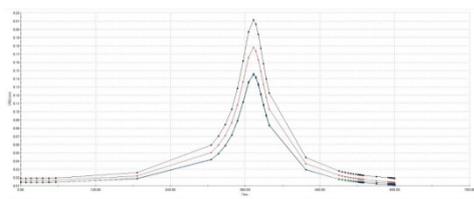
укрепленной рамкой конструкции электронного модуля (в)

(E_x , E_y , E_z – модули упругости материала платы по направлениям осей координат x, y, z ; ν – коэффициент Пуассона, ρ – плотность) рассчитывались с использованием методик усреднения характеристик анизотропных материалов, учитывающих объемное содержание разнородных материалов в различных слоях печатной платы – медных проводящих дорожек, заземлений, питания, диэлектрика, – и составили следующие значения: $E_x = 60$, $E_y = 59$, $E_z = 48$, ГПа; $\nu = 25\%$, $\rho = 1719$ кг/м³.

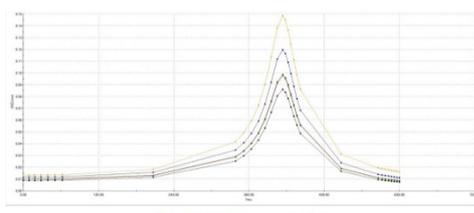
Компьютерное моделирование перемещений различных узлов конструкции электронного модуля с теплоотводами в виде ЕТМП и ГКТ в зависимости от частоты вибраций приведены на рис. 2а,б. На рис. 2,в приведены эпюры перемещений для доработанной конструкции электронного модуля с теплоотводом в виде ГКТ. Доработка электронного модуля с ГКТ была направлена на усиление ее жесткости, путем установки дополнительной планки, служащей еще одним ребром жесткости конструкции в направлениях осей x, y, z . Из сравнения зависимостей на рис. 2,а,б следует, что частота резонанса конструкции электронного модуля с теплоотводом в виде ЕТМП составляет 400 Гц, а с теплоотводом в виде ГКТ наступает раньше и составляет примерно 315 Гц. Однако после доработки конструкции электронного модуля в направлении увеличения ее жесткости (рис. 3),



а)



б)



в)

Рис. 2. Значения перемещений трех наиболее критических узлов электронного модуля в зависимости от частоты вибраций для конструкции теплоотвода в виде ЕТМП (а), в виде ГКТ (б) и для

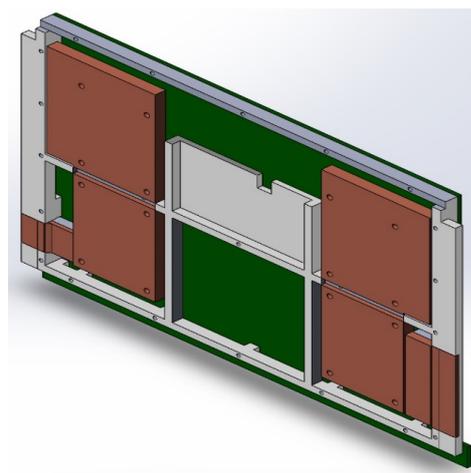


Рис. 3. Доработанная конструкция электронного модуля с гибкими кондуктивными теплоотводами

резонансная частота сдвинулась вправо и составили примерно 400 Гц, как и для конструкции с теплоотводом в виде ЕТМП

(рис. 2, в). При этом моделирование эпюр напряжений в наиболее критических точках конструкции электронного модуля показывает, что максимальные напряжения в обоих видах теплоотводов (ЕТМП и ГКТ) практически совпадают.

Полученные результаты позволяют заключить, что незначительная доработка несущей конструкции электронного модуля путем установки дополнительной планки в качестве ребра жесткости, позволила уменьшить максимальную амплитуду отклонений печатной платы с ГКТ, практически

сравнив ее с таковой в случае конструкции и ЕТМП.

Таким образом, результаты компьютерного моделирования конструкции электронного модуля с различными теплоотвода (ЕТМП и ГКТ) показали, что отвод тепла от мощных элементов (МС), установленных на печатной плате с применением ГКТ [2] существенно превышает таковой в случае использования традиционных конструкций теплоотводов в виде ЕТМП. При этом вибрационная устойчивость конструкций с ГКТ и ЕТМП практически совпадают.

Analysis of the thermal and vibrational modes of electronic module with flexible conductive heatsink and single heat-conducting metal plate

A.A. Vorobyev, A.G. Madera, A.O. Shamaev

Abstract: The article discusses the results of computer simulation of thermal and vibrational modes of the electronic module with various designs of heat removal from microcircuits installed on a printed circuit board. Two designs were considered as options for heat sinks, namely, in the form of a flexible conductive heat sink and a traditional design in the form of a metal plate common to all cooled microcircuits. The simulation results showed that a flexible conductive heat sink, compared with a single metal plate, has significantly higher heat sink properties, while ensuring that each cooled microcircuit is individually pressed to the heat sink, and also provides sufficient vibration resistance for practice.

Keywords: integrated circuit, heatsink, electronic module, printed-circuit board, flexible conductive heatsink, metal plate-heatsink, temperature mode, vibrations, contact thermal resistance

Литература

1. Устройство охлаждения и отвода тепла от компонентов электронных систем // Патент на полезную модель, № 117056, Заявка № 2011149886, Зарегистрировано в Гос. реестре полезных моделей РФ 10 июня 2012 / А.Г. Мадера, В.И. Гречишников, М.Ю. Гнитиев.
2. А.А. Воробьев, А.Г. Мадера. Компьютерное моделирование гибкого кондуктивного теплоотвода для охлаждения мощных микросхем. «Труды НИИСИ РАН», Т. 9 (2019), № 2, 36–39.
3. А.Г. Мадера. Концепция математического и компьютерного моделирования тепловых процессов в электронных системах. «Программные продукты и системы», Т. 112 (2015), № 4, 79–86.
4. A.G. Madera, P.I. Kandalov. Mathematical modeling of the interval stochastic thermal processes in technical systems at the interval indeterminacy of the determinative parameters. “Computer Research and Modeling”, Vol. 8 (2016), No. 3, 501–520.
5. А.Г. Мадера, П.И. Кандалов. Анализ интервально стохастических температурных полей технических систем. «Программные продукты и системы», Т.108 (2014), № 4, 41–45.

Математические модели для анализа разработки нефтяных месторождений методом ячеек заводнения при забойном и при пластовом давлении ниже давления насыщения без газовой шапки

И.В. Афанаскин¹, А.В. Королев², А.А. Глушаков³, П.В. Ялов⁴

¹ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ivan@afanaskin.ru;

²ФГУ ФНЦ НИИСИ РАН, Москва, Россия, alexandre.korolev@mail.ru;

³ФГУ ФНЦ НИИСИ РАН, Москва, Россия, stormwww@yandex.ru;

⁴ФГУ ФНЦ НИИСИ РАН, Москва, Россия, petryalov@gmail.com

Аннотация: Приведены математические модели для анализа разработки нефтяных месторождений методом ячеек заводнения при забойном и при пластовом давлении ниже давления насыщения без газовой шапки. Модели протестированы на данных реальных месторождений. Получено хорошее совмещение расчетных и замеренных кривых.

Ключевые слова: анализ разработки, ячейки заводнения, математическая модель

1. Введение

При разработке нефтяных месторождений много внимания уделяется анализу изменения показателей, характеризующих этот процесс: забойное и пластовое давления, дебиты нефти, обводненность, газовый фактор и др. Кроме того, много внимания уделяется краткосрочному и среднесрочному планированию, важную роль в котором занимает прогнозирование указанных выше показателей. При этом применяются либо очень сложные цифровые гидродинамические модели, либо совсем простые аналитические или эмпирические зависимости (материальный баланс, характеристики вытеснения и пр.). Причем последние применяются чаще первых ввиду своей простоты и высокой скорости получения результатов. Использование подробных цифровых моделей ограничивается не только трудоемкостью их использования, но и проблемами с качеством и количеством исходных данных для их построения. Поэтому актуальным является разработка промежуточного звена моделей. Одним из вариантов такого звена являются ячейки заводнения, которые описывают поведение участка пласта, который эксплуатируется несколькими скважинами. Особенно сложным для анализа является случай работы скважин при падении забойного давления ниже давления насыщения, а тем более при

пластовом давлении ниже давления насыщения, поскольку для них простые модели практически отсутствуют. Поэтому данная работа посвящена математическим моделям для анализа разработки нефтяных месторождений методом ячеек заводнения при забойном и при пластовом давлении ниже давления насыщения без газовой шапки.

2. Забойное давление ниже давления насыщения

Рассмотрим работу скважин при забойном давлении ниже давления насыщения нефти газом, а пластовом давлении – выше давления насыщения нефти газом. Тогда газ будет выделяться из нефти только в прискважинной зоне пласта. В этом случае процессы в основной части пласта можно рассматривать на двухфазном уровне – нефть и вода. При этом наличие свободного газа в прискважинной зоне учитывать в модели скважины. В этом случае для ячейки заводнения систему уравнений сохранения объемов нефти и воды в стандартных условиях по аналогии с [1, 4, 6, 7] можно записать в виде:

$$Fh \frac{d}{dt} \left(\frac{mS_o}{B_o} \right) = - \sum_{i=1}^N q_{o,i}, \quad (1)$$

$$Fh \frac{d}{dt} \left(\frac{mS_w}{B_w} \right) = - \sum_{i=1}^N q_{w,i} + \sum_{j=1}^M q_{iw,j} + q_a, \quad (2)$$

где F – площадь ячейки заводнения, h – эффективная нефтенасыщенная толщина, m – пористость, S_o – нефтенасыщенность, B_o – объемный коэффициент, $q_{o,i}$ – дебит нефти i -ой добывающей скважины, N – количество добывающих скважин в ячейке, S_w – водонасыщенность, B_w – объемный коэффициент воды, $q_{w,i}$ – дебит воды i -ой добывающей скважины, $q_{iw,j}$ – расход закачиваемой воды j -ой нагнетательной скважины, M – количество нагнетательных скважин в ячейке, q_a – приток воды в ячейку из законтурной области.

Система уравнений (1)-(2) замыкается следующими дополнительными соотношениями для насыщенностей и объемных коэффициентов (в предположении слабой сжимаемости породы и флюидов):

$$S_o + S_w = 1, \quad (3)$$

$$m = m_0 [1 + C_r (P - P_0)], \quad (4)$$

$$B_\alpha = B_{\alpha,0} [1 - C_\alpha (P - P_0)], \quad \alpha = o, w, \quad (5)$$

где m_0 и $B_{\alpha,0}$, $\alpha = o, w$ – пористость и объемные коэффициенты при начальном пластовом давлении P_0 , C_r и C_α , $\alpha = o, w$ – сжимаемость породы и флюидов.

Используя подход аналогичный изложенному в [3], уравнения (1)-(2) с учетом соотношений (3)-(5) можно преобразовать к виду ($S \equiv S_w$):

$$\begin{aligned} Vm_0 [C_r + (C_w - C_o)S + C_o] \frac{dP}{dt} = \\ = -B_{o0} \sum_{i=1}^N q_{o,i} + B_{w0} \left(q_a + \sum_{j=1}^M q_{iw,j} - \sum_{i=1}^N q_{w,i} \right) \end{aligned} \quad (6)$$

$$\begin{aligned} Vm_0 \left[\frac{dS}{dt} + (C_r + C_w)S \frac{dP}{dt} \right] = \\ = B_{w0} \left(q_a + \sum_{j=1}^M q_{iw,j} - \sum_{i=1}^N q_{w,i} \right) \end{aligned} \quad (7)$$

где уравнение (6) является линейным обыкновенным дифференциальным уравнением для давления, а уравнение (7) – линейным обыкновенным дифференциальным уравнением для насыщенности. Уравнения (6) и (7) легко решаются с помощью метода Эйлера. При этом на каждом шаге по времени сначала решается уравнение для давления (6), а потом уравнение для насыщенности (7). Для

обеспечения устойчивости решения необходимо ограничивать максимально возможное изменение давления и насыщенности за один шаг по времени, как это делается в [1].

В качестве начальных условий задается пластовое давление и водонасыщенность.

Работа нагнетательных скважин моделируется с помощью стандартной формулы Дюпюи. Добывающие скважины моделируются с помощью модифицированной формулы Фетковича для дебита жидкости Q_{li} [2, 5]:

$$q_{l,i} = K_{0,i} (P - P_{w,i}), \quad P_{w,i} \leq P_S^*, \quad (8)$$

$$q_{l,i} = K_{0,i} (P - P_S^*) + J_i (P_S^{*2} - P_{w,i}^2)^{n_i}, \quad P_{w,i} < P_S^*, \quad (9)$$

где K_0 – коэффициент продуктивности скважины по жидкости при двухфазной фильтрации нефти и воды, P_w – забойное давление, $P_S^* \leq P_S$, P_S – давление насыщения нефти газом, P_S^* – давление искривления индикаторной диаграммы, J_i и n_i – параметры формулы Фетковича, определяемые с помощью гидродинамических исследований скважин на установившихся (псевдоустановившихся) режимах.

Доказано, что для каждого нефтяного пласта существует зависимость между параметрами формулы Фетковича (J_i и n_i) и коэффициентом продуктивности скважины по жидкости при двухфазной фильтрации K_0 [2, 5]:

$$\ln(J_i) = \ln(a) + b \ln(K_{0,i}), \quad (10)$$

$$n_i = c - d K_{0,i}, \quad (11)$$

где параметры a , b , c и d – индивидуальны для каждого пласта, не меняются со временем и определяются с помощью вышеупомянутых исследований. Таким образом, для определения параметров J_i и n_i не нужно исследовать все добывающие скважины. Достаточно построить зависимости (10) и (11).

Коэффициент продуктивности скважины по жидкости при двухфазной фильтрации нефти и воды определяется как:

$$K_{0,i} = \frac{2\pi k_i h_i}{\ln\left(\frac{R_c}{r_{w,i}}\right) + Skin_i(t)} \left(\frac{k_{ro,i}(S)}{\mu_o B_o(P)} + \frac{k_{rw,i}(S)}{\mu_w B_w(P)} \right), \quad (12)$$

где k – абсолютная проницаемость, k_{ra} – относительная фазовая проницаемость для фазы $\alpha = o, w$, $Skin$ – скин-фактор, R_c – радиус контура питания, r_w – радиус скважины.

Дебиты нефти и воды определяются как:

$$q_{\alpha,i} = \lambda_{\alpha,i}^* q_{l,i}, \quad \alpha = o, w, \quad (13)$$

где $\lambda_{\alpha,i}^*$ - отношение подвижности фазы $\alpha=o, w$ и жидкости для скважины i в данный момент времени.

Рассмотрим пример использования математической модели для анализа разработки нефтяных месторождений методом ячеек заводнения при забойном давлении ниже давления насыщения. Месторождение находится в Северо-Кавказской нефтегазоносной провинции. Залежь нефти замкнута, литологически экранирована. Эксплуатируется единственной скважиной № 4, рис. 1. Коллектор терригенный.

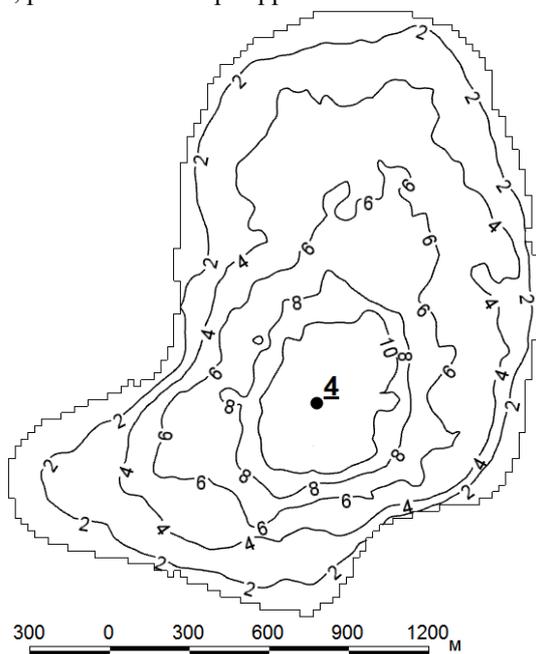


Рис. 1. Карта эффективных нефтенасыщенных толщин для первого примера

Начальное пластовое давление аномально высокое – 463.4 бар на глубине 2872 м. Объемный коэффициент нефти – 1.21 м³/м³. Вязкость нефти – 0.175 мПа·с. Сжимаемость нефти – 2.4·10⁻⁴ 1/бар. Давление насыщения нефти газом – 293.4 бар. Растворимость газа в нефти – 333.2 м³/м³. Объемный коэффициент воды – 1.017 м³/м³. Вязкость воды – 0.24 мПа·с. Сжимаемость воды – 3.3·10⁻⁵ 1/бар. Сжимаемость породы – 4.3·10⁻⁵ 1/бар. Проницаемость пласта – 22.3 мД. Пористость – 0.28 д.ед. Средняя эффективная нефтенасыщенная толщина – 4.9 м. Начальная водонасыщенность – 0.3 д.ед. На рис. 2 приведены исходные относительные фазовые проницаемости в системе нефть-вода. На рис. 3 приведена индикаторная диаграмма (зависимость дебита от забойного давления) скв. № 4, полученная после введения ее в

эксплуатацию. Параметры формулы Фетковича, аппроксимирующей данную диаграмму, следующие: $P=463.4$ бар, $P_S^*=283.4$ бар, $K_0=2.322$ м³/сут/бар, $J=0.035$, $n=0.708$. Результаты моделирования приведены на рис. 4. В качестве управляющего параметра на скважине задавался дебит нефти. Расчет проведен до момента падения пластового давления до давления насыщения. Поскольку начальная водонасыщенность равна насыщенности связанной водой, приток воды извне отсутствует (залежь замкнута), а пластовое давление падает не достаточно для изменения насыщенности за счет деформации пор, то дебит воды равен нулю. Получено хорошее совмещение расчетных и замеренных кривых.

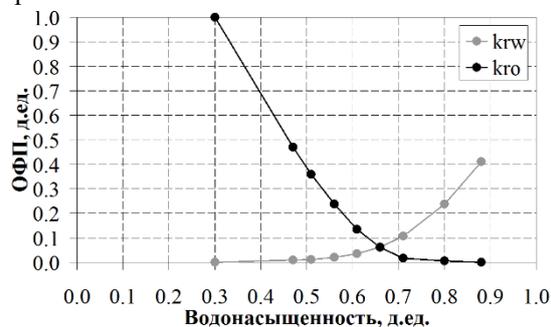


Рис. 2. Относительные фазовые проницаемости (ОФП) по нефти (kro) и воде (krw)

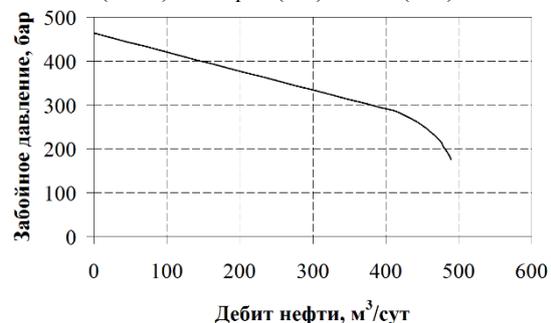


Рис. 3. Индикаторная диаграмма скв. № 4

3. Пластовое давление ниже давления насыщения

Рассмотрим работу скважин в случае, когда не только забойное, но и пластовое давление ниже давления насыщения нефти газом. При этом будет предполагать отсутствие газовой шапки. Тогда в пласте возможно одновременное существование трех фаз: нефти, газа и воды. Считаем, что газ нерастворим в воде. Тогда уравнения сохранения объёмов нефти, газа и воды в стандартных условиях по аналогии с [1, 4, 6, 7] можно записать в виде:

$$Fh \frac{d}{dt} \left(\frac{mS_o}{B_o} \right) = - \sum_{i=1}^N q_{o,i}, \quad (14)$$

$$Fh \frac{d}{dt} \left(\frac{mS_w}{B_w} \right) = - \sum_{i=1}^N q_{w,i} + \sum_{j=1}^M q_{iw,j} + q_a, \quad (15)$$

$$Fh \frac{d}{dt} \left(\frac{mS_g}{B_g} + \frac{mS_o R_s}{B_o} \right) = - \sum_{i=1}^N q_{fg,i} - R_s \sum_{i=1}^N q_{o,i}, \quad (16)$$

где дополнительно к обозначениям, введенным в разделе 1 данной работы, S_g – газонасыщенность, B_g – объемный коэффициент газа, R_s – растворимость газа в нефти, q_{fg} – дебит свободного газа.

Система уравнений (14)-(16) дополняется следующим замыкающими соотношениями (в

предположении слабой сжимаемости породы и воды):

$$S_o + S_g + S_w = 1, \quad (17)$$

$$m = m_0 [1 + C_r (P - P_0)], \quad (18)$$

$$B_w = B_{w,0} [1 - C_w (P - P_0)], \quad \mu_w = \text{const}, \quad (19)$$

$$B_\alpha = B_\alpha(P), \quad \mu_\alpha = \mu_\alpha(P), \quad \alpha = o, g, \quad (20)$$

$$R_s = R_s(P), \quad (21)$$

где зависимости $B_\alpha(P)$, $\mu_\alpha(P)$, $\alpha = o, g$ и $R_s(P)$ определяются по результатам PVT-исследований глубинных проб нефти и газа.

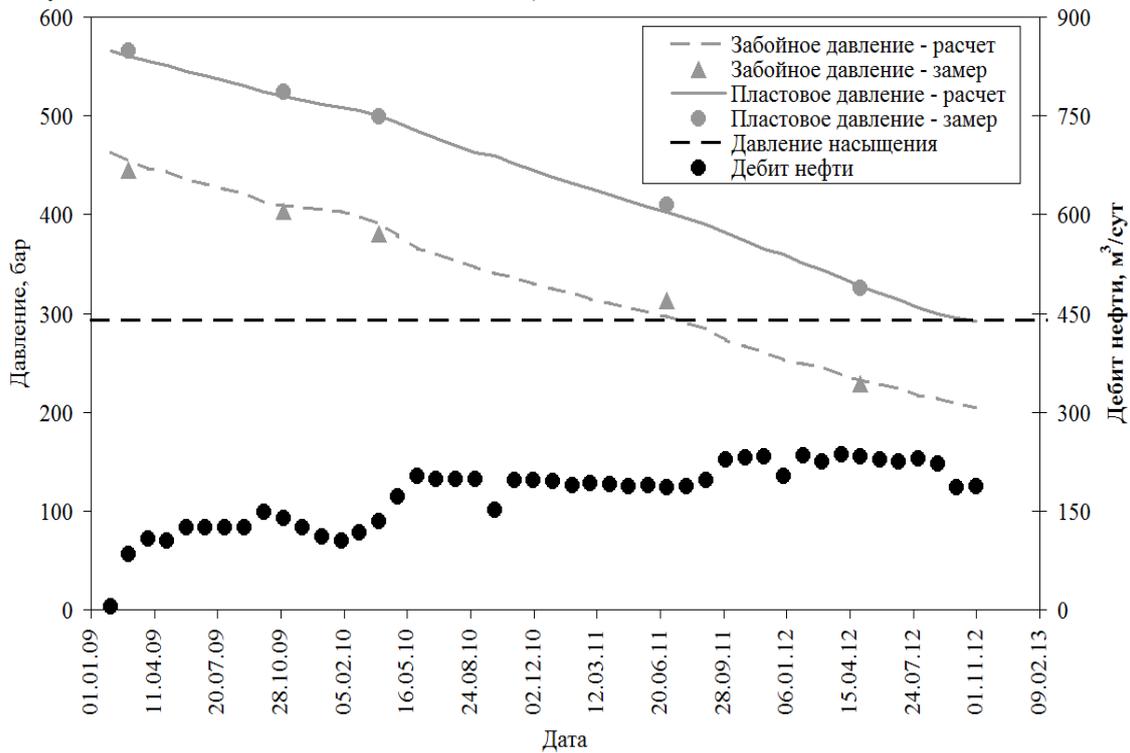


Рис. 4. Результаты расчетов для задачи с забойным давлением ниже давления насыщения

По аналогии с [4], комбинируя уравнения системы (14)-(16), с учетом соотношений (17)-(21) можно получить следующее нелинейное обыкновенное дифференциальное уравнение для давления:

$$\begin{aligned} & V(C_{op} + AC_{wp} + BC_{gp} + BC_{dgp}) \frac{dP}{dt} + \\ & + A \left(\sum_{i=1}^N q_{w,i} - \sum_{j=1}^M q_{iw,j} - q_a \right) + \\ & + B \sum_{i=1}^N q_{fg,i} + (1 + BR_s) \sum_{i=1}^N q_{o,i} = 0 \end{aligned} \quad (22)$$

где

$$B = \frac{C_{os}}{C_{gs} - C_{dgs}}, \quad A = B \frac{C_{gs}}{C_{ws}}, \quad C_{os} = \frac{m}{B_\alpha},$$

$$C_{ap} = \left(\frac{1}{B_\alpha} \frac{dm}{dP} - \frac{m}{B_\alpha^2} \frac{dB_\alpha}{dP} \right) S_\alpha, \quad \alpha = o, w, g,$$

$$C_{dgp} = C_{op} R_s + \frac{mS_o}{B_o} \frac{dR_s}{dP}, \quad C_{dgs} = \frac{mR_s}{B_o}. \quad (23)$$

Обозначим V_{oSC} и V_{gSC} – удельные объемы нефти и газа:

$$V_{oSC} = \frac{mS_o}{B_o}, \quad V_{gSC} = \frac{mS_g}{B_g} + \frac{mS_o R_s}{B_o}. \quad (24)$$

После аппроксимации производных в уравнении (22) с первым порядком точности получается нелинейное алгебраическое уравнение относительно давления на новом временном шаге, которое решается методом Ньютона.

Вспомним, что одним из замыкающих соотношений является зависимость (21) растворимости газа в нефти от давления $R_s=R_s(P)$, полученная по результатам лабораторных исследований. Отметим, что $R_s(P)$ - это максимальное содержание газа, растворимого в нефти при данном давлении. Возможны два случая:

1. Нефть недонасыщена газом. Фактическое содержание газа в нефти при данном давлении меньше максимально возможного $R_s(P)$. Тогда свободного газа в пласте нет.
2. Нефть полностью насыщена газом. Фактическое количество газа в пласте (относительно количества нефти) больше, чем может раствориться в нефти при данном давлении $R_s(P)$. Тогда в пласте присутствует газовая фаза.

Тогда можно использовать следующий способ вычисления насыщенностей нефтью и газом.

Если выполняется условие:

$$\frac{V_{gSC}}{V_{oSC}} \leq R_s(P), \quad (25)$$

то нефть недонасыщена газом, следовательно свободный газ в пласте отсутствует. Тогда:

$$S_o = 1 - S_w, \quad S_g = 0 \quad \text{и} \quad R_s = \frac{V_{gSC}}{V_{oSC}}, \quad (26)$$

где R_s - фактическое количество газа, растворенного в нефти.

Если выполняется условие:

$$\frac{V_{gSC}}{V_{oSC}} > R_s(P), \quad (27)$$

то нефть полностью насыщена газом и в пласте присутствует газовая фаза. Тогда:

$$S_g = \frac{B_g}{m} [V_{gSC} - V_{oSC} R_s(P)],$$

$$S_o = 1 - S_w - S_g \quad \text{и} \quad R_s = R_s(P) \quad (28)$$

Водонасыщенность S_w определяется из линейного обыкновенного дифференциального уравнения (15). Удельные объемы нефти и газа V_{oSC} и V_{gSC} определяются из линейных обыкновенных дифференциальных уравнений (14) и (16) соответственно.

Для обеспечения устойчивости решения необходимо ограничивать максимально возможное изменение давления и насыщенностей за один шаг по времени, как это делается в [1].

В качестве начальных условий задается пластовое давление, насыщенности двух фаз и количество растворенного в нефти газа, если в пласте на начальный момент времени отсутствует свободный газ.

Работа скважин моделируется с помощью уравнения Дюпюи:

$$q_{\alpha,i} = \frac{2\pi k_i h_i}{\ln\left(\frac{R_c}{r_{wi}}\right) + Skin_i(t)} \cdot \frac{k_{r\alpha,i}(S)}{\mu_\alpha(P) B_\alpha(P)} (P - P_{wi}),$$

$$\alpha = o, w, fg,$$

$$q_{g,i} = q_{fg,i} + q_{o,i} R_s. \quad (29)$$

Относительные фазовые проницаемости задаются, как:

$$k_{ro} = k_{ro}(S_g, S_w), \quad k_{rg} = k_{rg}(S_g),$$

$$k_{rw} = k_{rw}(S_w). \quad (30)$$

Рассмотрим пример использования математической модели для анализа разработки нефтяных месторождений методом ячеек заводнения при пластовом давлении ниже давления насыщения без газовой шапки. Месторождение находится в Северо-Кавказской нефтегазоносной провинции. Залежь нефти замкнута, литологически экранирована. Эксплуатируется двумя скважинами №№ 3 и 11, рис. 5. Коллектор терригенный.

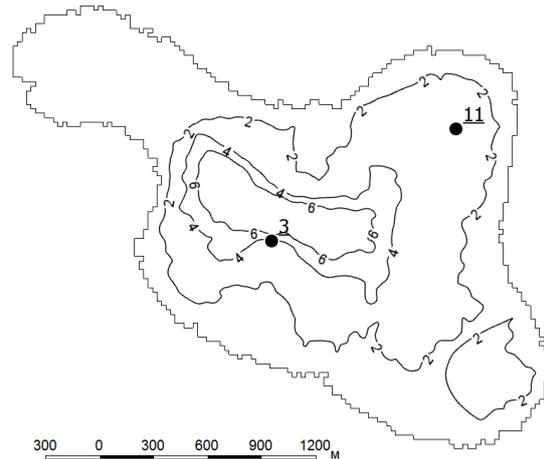


Рис. 5. Карта эффективных нефтенасыщенных толщин для второго примера

Начальное пластовое давление аномально высокое – 445.9 бар на глубине 2848 м. Объемный коэффициент воды – 1.015 м³/м³. Вязкость воды – 0.23 мПа·с. Сжимаемость воды – 3.4·10⁻⁵ 1/бар. Сжимаемость породы –

$5.35 \cdot 10^{-4}$ 1/бар. Проницаемость пласта - 68 мД (в районе скв. № 3 – 30.8 мД, в районе скв. № 11 – 160.8 мД). Пористость – 0.24 д.ед. (в районе скв. № 3 – 0.24 д.ед., в районе скв. № 11 – 0.26 д.ед.). Средняя эффективная нефтенасыщенная толщина – 2.4 м. (в районе скв. № 3 – 4.3 м., в районе скв. № 11 – 2.3 м.). Начальная водонасыщенность – 0.3 д.ед., свободного газа в пласте нет, начальная растворимость газа в нефти – $450 \text{ м}^3/\text{м}^3$. На рис. 6 и 7 приведены зависимости PVT-параметров от давления. На рис. 8 и 9 приведены исходные относительные фазовые проницаемости в системах нефть-вода и нефть-газ. Относительная фазовая проницаемость по нефти при двухфазной фильтрации вычисляется по второй модели Стоуна.

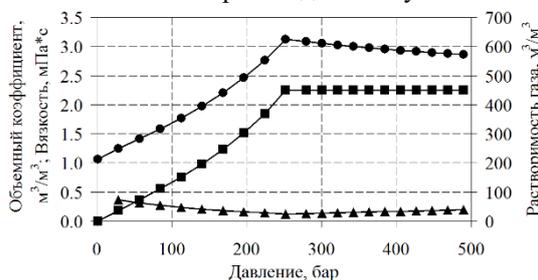


Рис. 6. Зависимости PVT-параметров нефти от давления

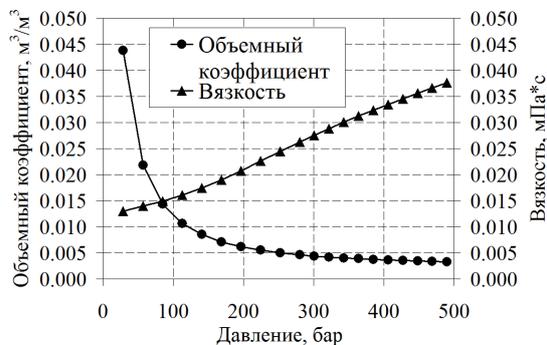


Рис. 7. Зависимости PVT-параметров газа от давления

Результаты моделирования приведены на рис. 10. В качестве управляющего параметра на скважинах задавался дебит нефти. Поскольку начальная водонасыщенность равна насыщенности связанной водой, приток воды извне отсутствует (залежь замкнута), а пластовое давление падает не достаточно для изменения насыщенности за счет деформации пор, то дебит воды равен нулю. Получено хорошее совмещение расчетных и замеренных кривых.

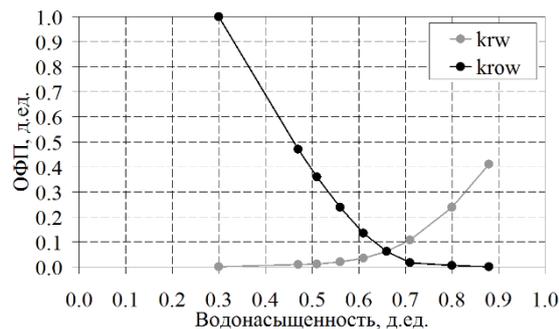


Рис. 8. Относительные фазовые проницаемости (ОФП) по нефти в присутствии воды (krow) и воде (krw)

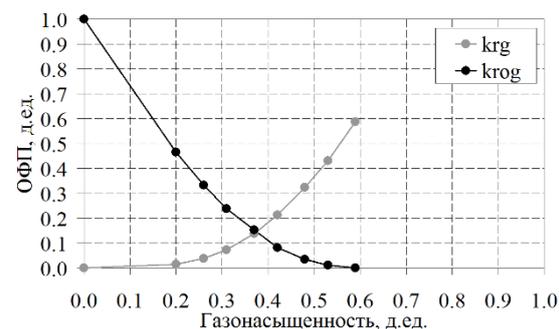


Рис. 9. Относительные фазовые проницаемости (ОФП) по нефти в присутствии газа (krog) и газу (krg)

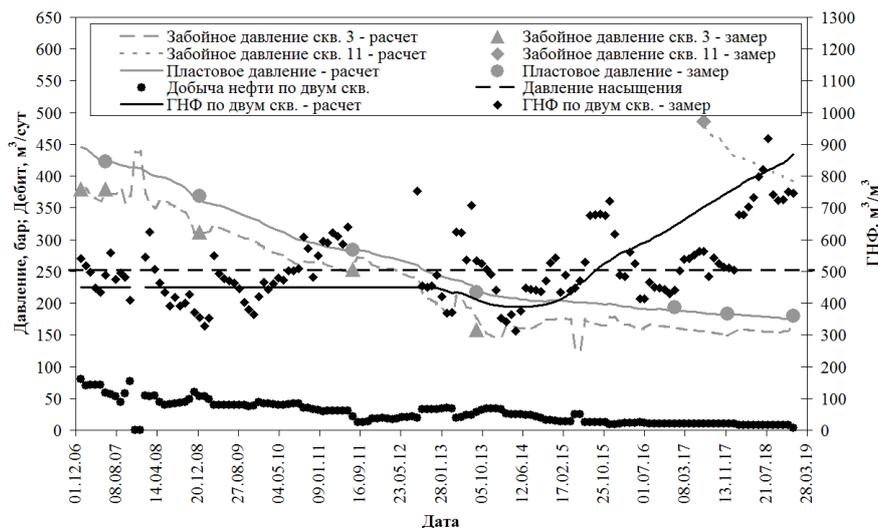


Рис. 10. Результаты расчетов для задачи с пластовым давлением

ниже давления насыщения без газовой шапки

4. Заключение

Приведена математическая модель для анализа разработки нефтяных месторождений методом ячеек заводнения при пластовом давлении выше давления насыщения и забойном давлении ниже давления насыщения. Модель протестирована на данных реального месторождения. Получено хорошее совмещение расчетных и замеренных кривых.

Приведена математическая модель для анализа разработки нефтяных месторождений

методом ячеек заводнения при пластовом давлении ниже давления насыщения без газовой шапки. Модель протестирована на данных реального месторождения. Получено хорошее совмещение расчетных и замеренных кривых.

Работа выполнена при поддержке гранта РФФИ 18-07-00677 А.

Mathematical Models for the Analysis of Oil Fields Production by the Method of Waterflood Cells at Reservoir and Bottom-Hole Pressures Lower than Bubble Point Pressure (Without Gas Cap)

I.V. Afanaskin, A.V. Korolev, A.A. Glushakov, P.V. Jalov

Abstract. Mathematical models for the analysis of oil fields production by the method of waterflood cells at reservoir and bottom-hole pressures lower than bubble point pressure (without gas cap) are described. The models were tested using real field data. Good agreement between numerical and observed data was achieved.

Keywords: production analysis, flooding cells, mathematical model

Литература

1. Х. Азиз, Э. Сеттари. Математическое моделирование пластовых систем. М.-Ижевск, Институт компьютерных исследований, 2004.
2. Н.Н. Диева, С.Г. Вольпин, Д.А. Корнаева и др. Повышение информативности исследований скважин, работающих при забойном давлении ниже давления насыщения, методом установившихся отборов. «Бурение и нефть», 2014, № 1, 41-43.
3. Р.М. Кац, Е.Р. Волгин, И.В. Афанаскин Численное моделирование двухфазной фильтрации нефти и воды. «Труды НИИСИ РАН», Т. 4 (2014), №2, 141-148.
4. Р.Д. Каневская. Математическое моделирование гидродинамических процессов разработки месторождений углеводородов. М.-Ижевск, Институт компьютерных исследований, 2002.
5. Д.А. Корнаева. Совершенствование методов ГДИ скважин, работающих при забойном давлении ниже давления насыщения. «Инженерная практика», 2013, № 9, 14-16.
6. Г.Б. Кричлоу. Современная разработка нефтяных месторождений. М., Недра, 1979.
7. Т. Эртекин, Дж. Абу-Кассем, Г. Кинг. Основы прикладного моделирования пластов. М.-Ижевск, Институт компьютерных исследований, 2012.

Особенности карбонатных трещиноватых нефтенасыщенных коллекторов по данным геофизических исследований скважин

А.А. Колеватов¹, А.М. Гиацинтов², А.А. Глушаков³, А.А. Егоров⁴

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, akolevatov@niisi.ras.ru

² ФГУ ФНЦ НИИСИ РАН, Москва, Россия, giatsintov@niisi.ras.ru

³ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, stormwww@yandex.ru

⁴ ООО «КБ АССА», Сургут, Россия, eaafit@gmail.com

Аннотация: Статья содержит результаты анализа петрофизических и геофизических данных по скважинам, вскрывшим карбонатные коллектора. Рассматриваются особенности строения коллекторов с целью обнаружения дополнительных идентификационных признаков.

Ключевые слова: карбонатные трещиноватые коллектора, геофизические исследования скважин (ГИС), промысловые исследования скважин, реконструкция фильтрационных характеристик

1. Введение

Информация, необходимая для прогнозирования типов карбонатных трещиноватых коллекторов комплексирована из разных источников. Одним из таких источников являются геофизические исследования скважин (ГИС) в открытом стволе. В отличие от гидродинамических исследований (ГДИ) скважин ГИС открытого ствола преимущественно дают информацию о петрофизических свойствах пород: плотность, пористость (пустотность), глинистость, песчаность, нефтенасыщенность, радиоактивность и др. В зависимости от типа пород, их минералогического состава и насыщения, результаты измерений скважинными приборами будут иметь отличия, достаточные для однозначной идентификации разных типов пород, вскрытых скважинами. При комплексировании результатов исследований несколькими методами возможна оценка фильтрационных характеристик отдельных участков пласта. Для случая с терригенными коллекторами оценка фильтрационных характеристик [1] возможна с большей точностью, нежели для карбонатных коллекторов, имеющих значительные отличия в структуре пустотного пространства [2].

Далее в статье будут рассмотрены особенности ГИС для карбонатных трещиноватых коллекторов с целью идентификации их типов и возможного

прогнозирования фильтрационно-емкостных свойств.

2. Описание объекта исследования

Исследуемый объект представляет собой классическое рифовое тело, сложенное из остатков раковин морских моллюсков, обитавших в поздне-девонское время. В последствии в результате преобразования рифового тела под воздействием вторичных процессов [3, с.155] рифовое тело было в некоторой степени метаморфизовано: кальцит частично заместился доломитом, изменилась пустотная составляющая в структуре пласта-коллектора, геологические процессы привели к растрескиванию в той или иной степени разных частей объекта.

Результатами исследования объекта на стадии разведки стали два набора данных о зависимости проницаемости от пористости (рис. 1): 1) по керновому материалу и 2) по результатам гидродинамических исследований. Оба набора данных взаимосвязаны и взаимодополняющие в некоторой степени. В частности данные ГДИ подтвердили зависимость проницаемости от пористости для низкопроницаемого коллектора (40 мД и менее). Но для коллекторов с проницаемостью более 40мД наблюдаются значительные расхождения, связанные, в том числе, со строением пустотного пространства.

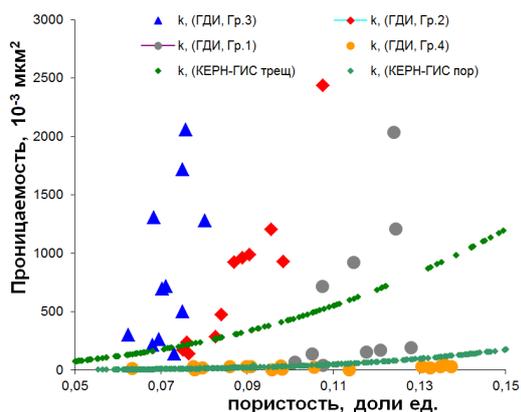


Рисунок 1. Результаты интерпретации ГДИ скважин в сопоставлении с зависимостями проницаемости от пористости, построенными только по данным керна.

Т.е. для таких коллекторов стандартный набор скважинных ГИС не дает полной информации о взаимосвязи фильтрационно-емкостных и петрофизических свойств, которые определяются посредством специальных измерений.

Далее будет рассмотрен подход к возможному комплексированию методов ГИС и ГДИ с целью идентификации признаков, необходимых для однозначного разделения коллекторов на разные типы и прогнозирования их свойств в новых скважинах по аналогии с имеющимися данными.

3. Методы скважинных исследований нефтенасыщенных коллекторов.

Методы скважинных ГИС включают исследования открытого ствола, а также промысловые геофизические исследования (ПГИ). Результатами ГИС являются объемные безразмерные петрофизические характеристики пород вскрытых скважиной, например, открытая пористость и начальная водонасыщенность. Результатами ПГИ являются данные о фильтрации «на стенке» скважины. Также, следует отметить, что «глубинность» скважинных методов исследований и, как правило, не превышает первые метры вглубь пласта. Но, даже с такими ограничениями, возможно получение полезной информации. Например, результатами ПГИ с целью исследования профиля притока является определение положения работающей толщины скважины,

а также границы раздела сред нефть-вода. Однако, по результатам ПГИ с целью определения «профиля притока» для исследуемых коллекторов были выявлены следующие особенности – положение границ толщин, дававших приток в скважину, менялось в зависимости от дебита, с которым работала скважина в момент исследований (рис. 2). Аналогичные результаты были получены и для других скважин. Т.е. приведенные результаты подтверждают наличие фильтрации пластовых флюидов не только через поровые каналы, но и через трещины, раскрытость которых меняется в зависимости от разницы давлений между скважиной и пластом.

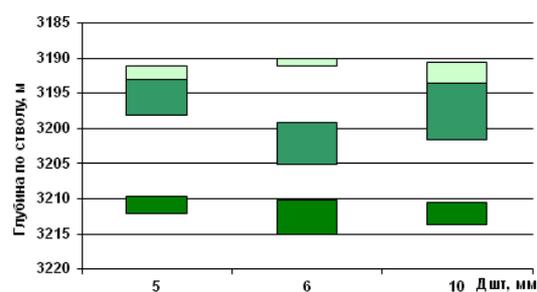


Рисунок 2. Работающие толщины согласно данным исследований по определению профилей притока. Скважина XZ03.

Другими словами, изменение границ продуктивных толщин в зависимости от дебита скважин следует учитывать при определении набора фильтрационно-емкостных характеристик и искать способы учета проводимости трещин, пронизывающих продуктивные интервалы, вскрытые скважинами.

Еще одним значительным источником ошибки при построении зависимостей проницаемости от пористости является увязка только данных по керну с данными ГИС. Объем керна даже при выносе более 60% не отражает реальной пространственной картины каналов фильтрации флюидов в карбонатных трещиноватых коллекторах, т.к. объем керна микроскопически мал по сравнению даже с зоной дренирования отдельной скважины.

Нередки случаи извлечения керна с низкой пористостью (2-3%) и низкой проницаемостью (<10 мД) из скважин, имеющих дебит несколько сотен кубических метров в сутки. Как правило, это объясняется связью скважины с продуктивными зонами пласта через систему трещин. Но говорить в этом случае об адекватности петрофизической зависимости

проницаемости от пористости по керну необоснованно.

Кроме приведенных выше данных о строении пласта коллектора с точки зрения геологии существует такой источник, как анализ разработки отдельных пластов и скважин. Заключается он в оценке начальных геологических и остаточных извлекаемых запасов посредством материального баланса [4,5,6]. В материалах, опубликованных по итогам 1-го года работ, приводится пример оценки извлекаемых запасов и определение режима дренирования залежи [7]. Дополнительно, было установлено: карбонатные трещиноватые коллектора могут содержать гораздо больше запасов, чем величина, рассчитанная объемным методом; ошибки в величинах запасов преимущественно связаны с погрешностями ГИС открытого ствола, что приводит к неверной оценке насыщенных толщин и объема пустот, заполненного нефтью.

4. Реконструкция фильтрационных характеристик нефтенасыщенного пласта-коллектора.

В рамках выполнения исследований по проекту проводилась реконструкция диаграмм относительных фазовых проницаемостей (ОФП) (рис. 3). Данные диаграммы обычно строятся на основании керновых данных. Но, как уже было отмечено выше, для карбонатных трещиноватых нефтенасыщенных коллекторов построение таких диаграмм изначально несет в себе ошибки, связанные с особенностями структуры пустотного пространства карбонатных коллекторов.

Необходимость реконструкции фазовых диаграмм возникла в связи с тем, что в зависимости от минералогического состава и структуры пустотного пространства диаграммы ОФП могут иметь разную форму, разное положение точки пересечения таких кривых, а также разную форму кривой фракционального потока. В случае обнаружения существенных различий между формами диаграмм возможен анализ различий каротажных диаграмм открытого ствола с целью обнаружения существенных отличий. Обнаруженные отличия каротажных диаграмм далее могут использоваться как идентификационные

признаки разных типов карбонатных коллекторов.

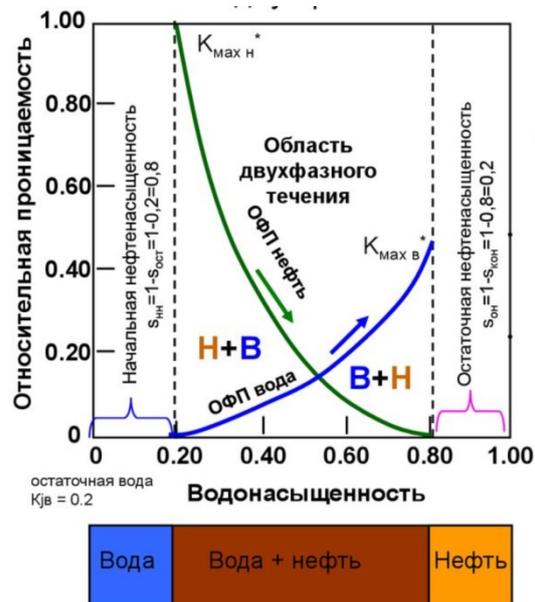


Рисунок 3. Типичный вид кривых ОФП в двухфазной системе нефть-вода.

Реконструкция диаграмм ОФП проводилась с использованием материального баланса: 1) Определялся режим дренирования скважины на основе истории данных по дебитам, давлениям и притока из законтурной области 2) Определялись «истинные» геологические и извлекаемые запасы с целью расчета изменений насыщенности за анализируемый период от начала добычи 3) Производился расчет диаграмм ОФП на основе методик Corey и Ershagi [8].

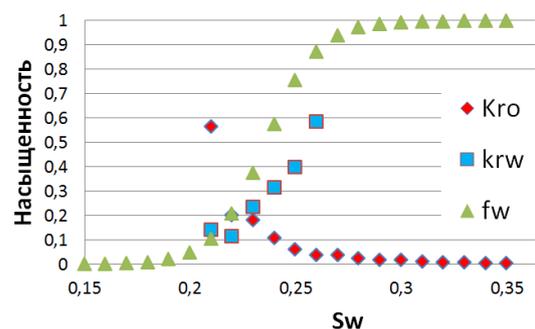


Рисунок 4. Реконструированные кривые ОФП в двух-фазной системе нефть-вода по скв X21.

Реконструкция диаграмм ОФП (рис. 4-9) показала схожий вид и положение кривых ОФП и кривой фракционального потока для более чем 15 скважин, данные по которым удалось проанализировать.

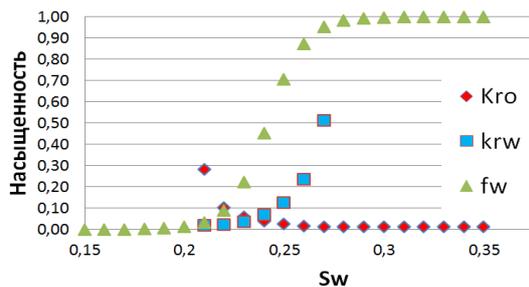


Рисунок 5. Реконструированные кривые ОФП в двух-фазной системе нефть-вода по скв X24.

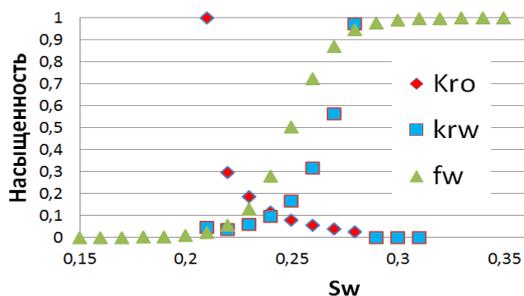


Рисунок 6. Реконструированные кривые ОФП в двух-фазной системе нефть-вода по скв X30.

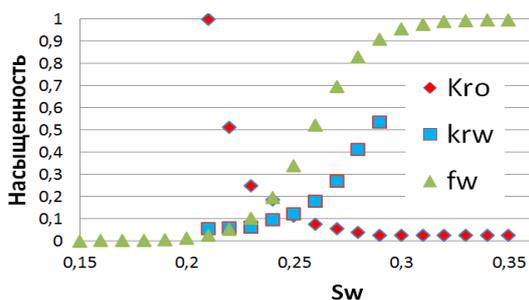


Рисунок 7. Реконструированные кривые ОФП в двух-фазной системе нефть-вода по скв X21.

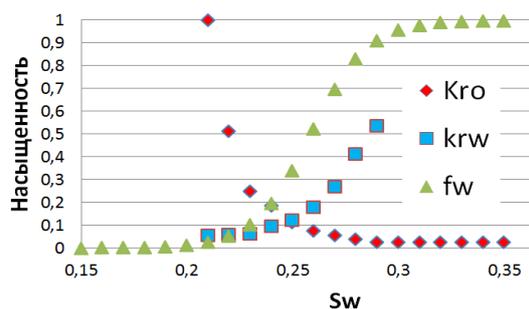


Рисунок 8. Реконструированные кривые ОФП в двух-фазной системе нефть-вода по скв YY04.

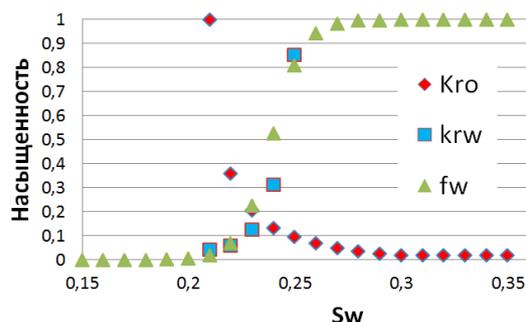


Рисунок 9. Реконструированные кривые ОФП в двух-фазной системе нефть-вода по скв YY08.

5. Заключение

1. Рассмотрены особенности скважинных ГИС с точки зрения применимости для идентификации разных типов карбонатных коллекторов. Показаны некоторые причины неточности результатов таких исследований.

2. Рассмотрены петрофизические и гидродинамические особенности карбонатных трещиноватых коллекторов, влияющие на результаты скважинных методов исследования.

3. Рассмотрено влияние неточности результатов ГИС на информацию, касающуюся фильтрационно-емкостных свойств.

4. Рассмотрены результаты реконструкции диаграмм ОФП карбонатных трещиноватых коллекторов в разных скважинах с целью возможной идентификации различных петрофизических характеристик. Однозначных различий в настоящий момент не установлено, что может свидетельствовать об однотипном типе коллектора для большей части объема продуктивного пласта.

«Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №18-07-00673 А».

Carbonate fractured oil saturated reservoir properties based on open-hole logging survey

A.A. Kolevator, A.M. Giatzintov, A.A. Glushakov, A.A. Egorov

Abstract: Article overviews petrophysical and geophysical survey data results for wells, which penetrated carbonate fractured reservoir. Reservoir structure properties analyzed to derive additional identification markers.

Keywords: carbonate fractured reservoirs, open-hole logging, production logging survey, filtration properties reconstruction

Литература

1. R. Baouche, K. Baddari Prediction of permeability and porosity from well log data using the nonparametric regression with multivariate analysis and neural network, Hassi R'Mel Field, Algeria. Egyptian Journal of Petroleum Volume 26, Issue 3, September 2017, Pages 763-778
2. Т.Д. Голф-Рахт Основы нефтепромысловой геологии и разработки трещиноватых коллекторов. М.: Недра, 1986.
3. В.Н. Киркинская, Е.М. Смехов. Карбонатные породы — коллекторы нефти и газа.— Л.: Недра, 1981.— 255 с.
4. L.P. Dake. Natural Water Influx. Ch.9. 1978. ISBN: 978-0-444-41667-4.
5. Victor T Bui, Total E & P; Okotie Sylvester, Stanley Buduka. A Time Function Havlena and Odeh MBE Straight Line Equation. SPE-178408-MS., Energia Limited;, Federal University of Petroleum Resources
6. L.P. Dake. The practice of reservoir engineering. Elsevier, 1994.
7. А.А. Колеватов, Ю.М. Штейнберг, Ю.Б. Чен-лен-сон, А.А. Егоров, А.М. Гиацинтов, А.Г. Дяченко Разработка методики определения типа карбонатного нефтенасыщенного коллектора. Журнал Труды НИИСИ РАН Том 8 № 4. с. 25-35 М.: Изд-во ФГУ ФНЦ НИИСИ РАН, 2018.
8. А.А.Колеватов, И.В. Афанаскин, А.Г. Дяченко, Д.В. Солопов, А.К. Пономарев Реконструкция диаграмм относительных фазовых проницаемостей с целью уточнения гидродинамической модели нефтяного месторождения // Актуальные проблемы нефти и газа, № 3 (22), 2018.

Классификационные признаки типов нефтенасыщенных коллекторов по результатам гидродинамических исследований

А.А. Колеватов¹, Ю.Б. Чен-лен-сон², Ю.М. Штейнберг³, А.Г. Дяченко⁴

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, akolevatov@niisi.ras.ru

² ФГУ ФНЦ НИИСИ РАН, Москва, Россия, jchenlenson@niisi.ras.ru

³ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, yush@niisi.ras.ru

⁴ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, adyachenko@niisi.ras.ru

Аннотация: В статье рассматриваются результаты гидродинамических исследований карбонатных трещиноватых нефтенасыщенных коллекторов с точки зрения идентификации классификационных признаков, которые могут быть использованы в дальнейшем для прогноза свойств.

Ключевые слова: карбонатные трещиноватые коллектора, классификационные признаки нефтенасыщенных коллекторов, гидродинамические исследования скважин

1. Введение

Для разработки методики прогнозирования типов карбонатных трещиноватых коллекторов необходим набор граничных параметров, с использованием которых возможна дальнейшая идентификация типов пласта коллектора при минимальном наборе исходных данных. Каждый метод исследования имеет ряд своих преимуществ, но не является универсальным для решения задач связанных с проектированием или анализом разработки.

Результаты гидродинамических исследований (ГДИ) преимущественно задействуются при анализе разработки, но также, как и другие методы исследования скважин и пластов, не являются полностью достаточными для полной реконструкции информации о строении пласта-коллектора.

По итогам предыдущих работ [1] были получены данные, в первом приближении указывающие на значительные отличия свойств отдельных участков нефтенасыщенного пласта-коллектора, вскрытого добывающими скважинами.

Далее будет приведена информация об отличительных признаках отдельных участков нефтенасыщенного пласта-коллектора, выявленных по результатам ГДИ скважин анализируемого объекта.

2. Описание объекта исследования

Результаты интерпретации ГДИ и их сопоставление с зависимостями керн-ГИС (рис. 1), которые строились при подготовке проектного документа по исследуемому объекту, показали в первом приближении разделение значений проницаемости на несколько групп по объему пустотного пространства.

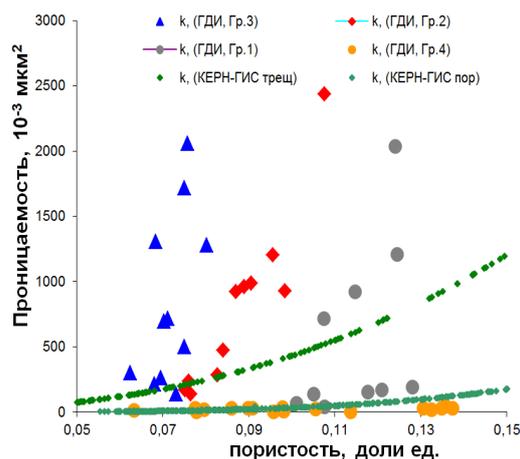


Рисунок 1. Результаты интерпретации ГДИ скважин в сопоставлении с зависимостями проницаемости от пористости, построенными только по данным керна.

Условно, выделенные группы значений проницаемости от пористости локализовались в следующих пределах:

- 1) При значениях пористости от 5 до 8% проницаемости достигали 2000 мД.
- 2) При значениях пористости 8 до 10% проницаемости достигали 2500 мД.
- 3) При значениях пористости 10 до 13% проницаемости достигали 2000 мД.
- 4) 4-я группа значений проницаемости по данным ГДИ практически совпала с зависимостью, построенной по данным керн. Абсолютные значения проницаемости в этом случае не превышали 40 мД, но пористость колебалась в широких пределах от 5 до 13%.

Описанное выше разделение нефтенасыщенных пластов коллекторов на группы по значениям проницаемости и пористости, отчасти, согласуется с классификацией карбонатных трещиноватых коллекторов (Таблица 1), представленной в работе В.Д. Викторина [2]. Отличия состоят в том, что в упомянутой работе указываются нижние пределы параметров пористости и проницаемости без указания максимумов.

Таблица 1. Обобщенная классификация типов карбонатных трещиноватых коллекторов [2]

Коллекторы		Порова я проницаемость, мкм ²	Радиус фильтрации, мкм	Радиус пор и каверн, мкм	Нижний предел открытой пористости, %
Группа	Тип, подтип				
I, высокопроницаемые (высокопродуктивные)	Каверновый, каверново-поровый, поровый	>0.1	14-20 и выше	20-500	10-15
II, среднепроницаемые (среднепродуктивные)	Трещино-поровый, трещино-каверновый	0.1-0.01	14-5	140-500	8-10
III, низкопроницаемые (низкопродуктивные)	Трещино-каверновый, трещино-поровый	0.01-0.001	5-3	300-500	4-8
IV, непроницаемые (потенциально-продуктивные)	Трещино-каверново-поровый	<0.001	3-2 и ниже	500	2-4

Так, практически совпали пределы открытой пористости для коллекторов с пористостью более 5% и проницаемостями более 40 мД. Для коллекторов с проницаемостью менее 40 мД параметр пористости может достигать 13-15%, но в данном случае поры будут иметь слабую связь. То есть, этот низкопроницаемый тип коллектора будет преимущественно содержать заполненные нефтью каверны, слабо связанные между собой. Низкая проницаемость такого коллектора объясняется тем, что такой коллектор с

момента формирования почти не подвергался вторичным процессам и не метаморфизован, как три других.

3. Компоненты проницаемости выделенных групп пластов коллекторов.

Относительно величин поровой (матричной) и трещинной составляющих в проницаемости коллекторов мнения исследователей расходятся. Учитывая структуру пустотного пространства карбонатных трещиноватых коллекторов в величине проницаемости всегда будет сочетание проницаемостей разной природы – проницаемости пор и проницаемости трещин. Это связано с тем, что после образования карбонатного трещиноватого коллектора очень часто имеют место процессы вторичного преобразования. Другими словами, вследствие тектонических движений и выщелачивания происходит растрескивание и переотложение карбонатного скелета в виде доломитов. Степень метаморфизации и определяет в дальнейшем величину поровой и трещинной составляющей. Кроме того, как таковое разделение этих составляющих крайне затруднено без специальных исследований кернов в шлифах. Дело в том, что трещины при некоторых нагрузках могут иметь микроскопические размеры, сопоставимые с размером пор и межпоровых каналов. Одно из вероятных разделений трещинной и поровой составляющей в величине проницаемости приведено в таблице 2 [3].

Таблица 2. Соотношение поровой и трещинной составляющих в проницаемости разных типов карбонатных трещиноватых коллекторов.

№ группы	1	2	3	4
$K_{общ}$, 10^{-3} мкм ²	40-2000	40-3000	40-7000	<40
$K_{пор\ ср. зв}$, %	10-13	8-10	5-8	4-15
$K_{матр.}$, % от $K_{общ}$	25-50	15-20	5-10	5-10
$K_{трещ.}$, % от $K_{общ}$	50-75	80-85	90-95	90-95

4. «Связь» скважин с пластом

Одним из наиболее важных результатов ГДИ скважин является определение величины скин-фактора. Этот безразмерный параметр [4] характеризует степень совершенства «связи» скважины с пластом.

При вскрытии скважиной нефтенасыщенного пласта-коллектора в зависимости от условий бурения имеет место повреждение прискважинной зоны пласта из-за проникновения в окружающее поровое пространство фильтрата бурового раствора. Но повреждение пласта имеет место не для всех коллекторов одинаково. Так, для карбонатных коллекторов степень повреждения может быть гораздо ниже за счет наличия трещинной составляющей в проницаемости пласта, которая способствует более легкому очищению призабойной зоны пласта от остатков бурового раствора.

В ходе интерпретации данных ГДИ скважин исследуемого объекта была построена диаграмма распределения значений скин-фактора и проницаемости (рис. 2). Как можно отметить, для значений проницаемости менее 40 мД значения скин-факторов локализовались, преимущественно в зоне менее -3.5 единиц. Это говорит о том, что связь скважин с пластом для этих участков осуществляется, преимущественно, через систему трещин и близка к «идеальной» [5]. Для проницаемостей со значениями более 40 мД значения скин-факторов распределились как в зоне -3.5, так и в области положительных значений. Это говорит о том, что для отдельных скважин может иметь место связь с пластом не только через систему трещин разного размера, но и через систему пор либо трещинных каналов с низкой проницаемостью.

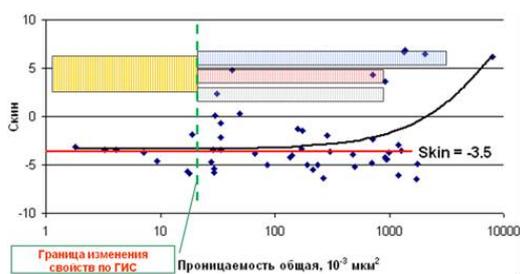


Рисунок 2. Распределение значений скин-фактора от проницаемости для разных типов коллекторов.

5. Зона дренирования и извлекаемые запасы

Описанные в предыдущих разделах статьи особенности разных видов карбонатных трещиноватых коллекторов, как правило, обозначают тот факт, что на самом деле скважина, вскрывшая пласт в той или иной зоне может дренировать гораздо большую область, чем предполагают инженеры-проектировщики и недрополь-

зователи. Отдельные трещины могут иметь гораздо большее распространение по пласту, чем зона дренирования отдельной скважины.

Очень часто для карбонатных трещиноватых коллекторов свойственно наличие системы трещин, связывающих несколько пластов разного геологического времени в единый гидродинамический объект. В этом случае даже не вскрытые скважиной пласты могут дренироваться через другие пласты посредством системы трещин. Это явление получило название «двойная проницаемость».

Интерпретационная модель двойной проницаемости [6] (рис. 3) по результатам ГДИ скважин (рис. 4, 5) изучаемого объекта зарегистрирована для 50% скважин, в которых удалось провести качественные ГДИ. По остальным скважинам, вероятно, имело место искажение зарегистрированного давления, связанное с неустранимыми на момент исследований технологическими особенностями.

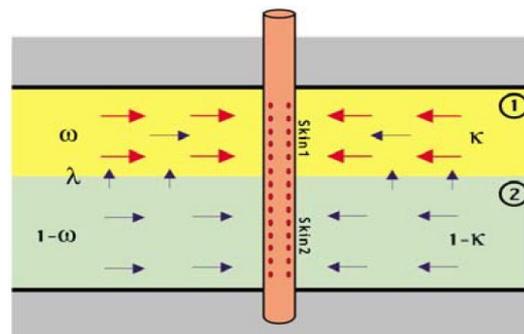


Рисунок 3. Схема дренирования нефтенасыщенного пласта коллектора посредством «двойной проницаемости».

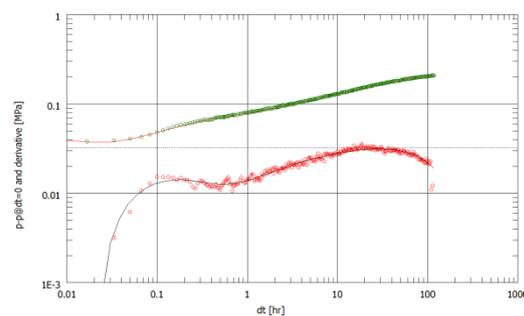


Рисунок 4. Диагностический график в билогарифмических координатах проинтерпретированный методом наилучшего совмещения с помощью интерпретационной модели «двойной проницаемости». (скв Х4)

Наличие в нефтенасыщенном пласте коллекторе системы взаимосвязанных трещин разного размера и, следовательно,

взаимного влияния скважин в процессе добычи было подтверждено ГДИ методом гидропрослушивания, которые подтвердили гидродинамическую связь на расстоянии нескольких километров.

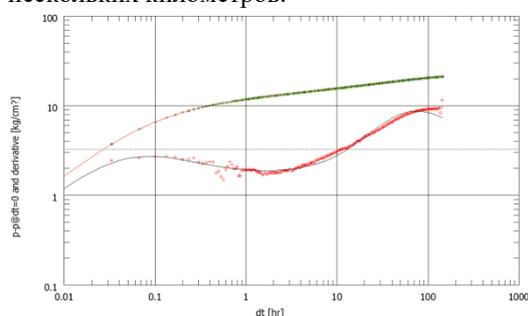


Рисунок 5. Диагностический график в билигарифмических координатах проинтерпретированный методом наилучшего совмещения с помощью интерпретационной модели «двойной проницаемости» (Скв X01).

6. Заключение.

1. Описанные особенности нескольких типов нефтенасыщенных коллекторов связаны с различиями структуры пустотного пространства, являющиеся причиной преобладания того или иного механизма фильтрации флюидов.

2. Особенности структуры пустотного пространства каждого из типов пласта коллектора выражаются, в том числе, в степени «совершенства связи» скважин с продуктивным пластом.

3. Преобладающий механизм фильтрации флюидов на большей части площади исследуемого объекта – «двойная проницаемость», обозначающая взаимосвязь даже удаленных друг от друга участков посредством системы субвертикальных трещин, которые объединяют эти части в единый гидродинамический объект.

«Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №18-07-00673 А».

Oil saturated reservoir's classification features after welltest results

A.A. Kolevatov, Y.B. Chen-len-son, Y.M. Steinberg, A.G. Dyachenko

Abstract: Article overviews carbonate fractured reservoirs welltest results, applied for classification features identification for further reservoir properties forecasting.

Keywords: carbonate fractured reservoirs, oil-saturated reservoir classification features, well testing,

Литература

1. А.А. Колеватов, Ю.М. Штейнберг, Ю.Б. Чен-лен-сон, А.А. Егоров, А.М. Гиацинтов, А.Г. Дяченко. Разработка методики определения типа карбонатного нефтенасыщенного коллектора. Журнал Труды НИИСИ РАН Том 8 № 4. с. 25-35 М.: Изд-во ФГУ ФНЦ НИИСИ РАН, 2018.
2. В.Д. Викторин, «Влияние особенностей карбонатных коллекторов на эффективность разработки нефтяных залежей», Недра, 1988.
3. Л.Б. Берман, В.С. Нейман, М.Д. Каргер и др. Промысловая геофизика при ускоренной разведке газовых месторождений. М.: Недра, 1987.
4. Д. Бурде Интерпретация результатов исследований скважин. Материалы лекций. Petroleum Engineering and Related Management Training Gubkin Academy. Москва, 1994. 109 с.
5. A.C. Gringarten Interpretation of Tests in Fissured and MultiLayered Reservoirs with Double-Porosity Behavior. Theory and Practice/Pet.Tech. – 1984. – April. – P. 549-564.
6. Т.А. Деева, М.Р. Камартинов, Т.Е. Кулагина, П.В. Мангазеев Гидродинамические исследования скважин: анализ и интерпретация данных – Томск 2009.

Интегрированный анализ гидродинамических и геофизических скважинных исследований с целью обнаружения невыработанных зон

А.А. Колеватов¹, Ю.Б. Чен-лен-сон², Ю.М. Штейнберг³, И.В. Афанаскин⁴

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, akolevatov@niisi.ras.ru

² ФГУ ФНЦ НИИСИ РАН, Москва, Россия, jchenlenson@niisi.ras.ru

³ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, yush@niisi.ras.ru

⁴ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, iafanaskin@niisi.ras.ru

Аннотация. В работе представлены выводы и обобщение результатов интегрированного анализа истории работы отдельных добывающих скважин, вскрывших карбонатный трещиноватый нефтенасыщенный коллектор. Анализ позволил сделать вывод о несоответствии исходных данных итоговым величинам, относящимся к так называемым подсчетным параметрам, используемым для оценки геологических и дренируемых запасов. Рассмотрены возможные источники ошибки, приводящие к значительному занижению такого подсчетного параметра, как пористость.

Ключевые слова: карбонатный трещиноватый нефтенасыщенный коллектор, гидродинамические исследования, подсчетные параметры.

1. Введение

Одним из значительных источников углеводородов на территории РФ являются запасы нефти в карбонатных трещиноватых коллекторах. Для оценки выработки запасов из таких коллекторов одним из необходимых действий является анализ разработки с целью корректировки плана мероприятий по нефтеизвлечению. При этом в ходе анализа разработки нередко выявляется взаимное несоответствие исходных и итоговых параметров, описывающих дренирование запасов в исследуемой зоне:

- ✓ занижение либо завышение пористости и/или нефтенасыщенных толщин;
- ✓ несоответствие поступающих (при наличии нагнетания) и отбираемых объемов жидкости в исследуемой зоне.

Неточности в величинах подсчетных параметров и неверная оценка запасов, как правило, приводит к ошибкам в сценариях управления текущей разработкой объекта. Основным методом оценки состояния объекта является анализ разработки с применением, в том числе, метода материального баланса. Материальный баланс [1] позволяет произвести оценку энергетического состояния залежи, как в истории, так и на прогноз; оценить геологические и извлекаемые начальные и остаточные запасы углеводородов; а также выявить несоответствия в исходных и

итоговых параметрах, описывающих изучаемый объект.

Данная статья рассматривает, в том числе, ошибки в подсчетных параметрах на примере карбонатного трещиноватого нефтенасыщенного коллектора (далее - *коллектор*). Результаты геофизических исследований скважин (ГИС), вскрывших такой коллектор, могут иметь некоторый разброс определяемых свойств объекта, связанный, прежде всего, с особенностями строения коллектора такого типа.

2. Исходные данные для интегрированного анализа

Интегрированный анализ использует весь набор данных, включаемых в геолого-физическую характеристику объекта, а также данные гидродинамических исследований скважин (ГДИ), историю дебитов и давлений. Результаты гидродинамических исследований скважин представляют собой набор данных, полученных при эксплуатации скважины с заданными дебитами и соответствующими измерениями забойного давления высокоточными манометрами. Подходы к интерпретации таких исследований представлены во многих известных работах, например [2-7], и имеют целью определение параметров пластов и скважин. В рассматриваемом случае для материального баланса важно не только пластовое давление и проницаемость, но и интерпретационная модель, описывающая процесс фильтрации флюида к скважине. Например, в случае с диагностированием модели двойной

проницаемости для оценки запасов важно учитывать всю нефтенасыщенную толщину, а не только ту, из которой имеет место непосредственный приток в скважину.

Важное значение имеют данные PVT-свойств углеводородов и их изменение в процессе разработки пласта и эксплуатации отдельных скважин. Необходима информация по изменению объемного коэффициента содержащихся в пласте углеводородов по мере изменения пластового давления. Эти параметры являются ключевыми при расчете материального баланса, т.к. в процессе анализа производится ряд вычислений, позволяющих сделать заключение, об имеющемся объеме геологических и извлекаемых запасов, степени поддержания пластовой энергии притоком из законтурной области, а также об эффективности мероприятий по поддержанию пластовой энергии (давления) при их наличии. Все эти расчеты оперируют изменением объема флюидов в пласте в процессе разработки. Из-за больших значений этих объемов даже небольшие ошибки в объемных коэффициентах могут привести к существенной погрешности в вычислениях.

3. Концепция материального баланса

Точкой отправления для расчетов методом материального баланса является закон сохранения объема флюидов в пластовых условиях (при пластовом давлении и температуре). В практических примерах, представленных далее, будет использован упрощенный подход для объектов, эксплуатируемых при пластовом давлении выше давления насыщения. Это позволит уменьшить расчетную часть за счет исключения перераспределения давления при выделении попутного нефтяного газа.

Уравнение материального баланса при частичном вытеснении нефти водой, необходимо применять в общем виде. В этом случае член, выражающий приток воды, является по существу переменной величиной (наличие или отсутствие нагнетания, реакция законтурной области на изменение давления в зоне отбора продуктивного пласта). За исключением случаев, когда рост поступления воды в пласт задерживается отбором у водонефтяного контакта, слагаемое уравнения, описывающее приток воды, представляет собой непрерывно

растущую функцию. Распространенное допущение на «среднее» постоянное значение для величины притока не правомерно. Один из подходов к оценке активности законтурной области с целью определения остаточных извлекаемых запасов при известном изменении пластового давления во времени заключается в вычислении объемов притока воды как функции от изменения пластового давления. Тогда колебания значений, описывающих объем притока, отражают процесс внедрения/оттока воды в продуктивный пласт.

Одно из приближенных решений этой задачи заключается в том, что общее поступление воды должно выражаться функцией изменения давления в процессе разработки месторождения. Для этого случая широко используется методика Van Everdingen & Hurst [8]. В основе прогнозирования пластового давления лежит уравнение материального баланса [1]:

$$N_p B_o = NB_{oi} \Delta P C_e + W_e + (W_i - W_p) B_w,$$

где:

N_p - накопленная добыча нефти в m^3 в поверхностных условиях;

B_o - текущий объемный коэффициент нефти (в случае отсутствия данных вычисляется по корреляциям), m^3/m^3 ;

N - балансовые запасы в поверхностных условиях в m^3 ;

B_{oi} - начальный объемный коэффициент нефти, m^3/m^3 ;

ΔP - снижение пластового давления от начального, ат;

C_e - эффективная сжимаемость, $1/ат$;

W_e - накопленный приток воды из аквифера (законтурной области) в пластовых условиях в m^3 ;

W_i - накопленная закачка воды в поверхностных условиях, m^3 ;

W_p - накопленная добыча воды в поверхностных условиях, m^3 ;

B_w - текущий объемный коэффициент воды (может быть вычислен по корреляциям, также может быть использован начальный, так как незначительно изменяется при снижении давления).

Эффективная сжимаемость системы пласт-флюид рассчитывается по формуле:

$$C_e = \frac{C_o S_{oi} + C_w S_{wc} + C_f}{1 - S_{wc}},$$

где:

C_o, C_w, C_f - сжимаемости нефти, воды и породы соответственно;

S_{oi}, S_{wc} - начальная нефтенасыщенность, начальная водонасыщенность.

Основная сложность состоит в расчёте накопленного притока воды из законтурной области (**аквифера**) W_e необходимого для совмещения истории замеренного давления с расчетным для некоторой принятой величины запасов. Кроме того, оценка притока воды из законтурной области позволяет сделать прогноз пластового давления исходя из предположения определенных уровней добычи жидкости. Существует ряд методов, позволяющих провести такие расчёты для неустановившегося (Van Everdingen & Hurst, Carter & Tracy), установившегося (Schilthuis) и псевдоустановившегося (Fetkovich) режимов.

Наиболее точной, хотя и более сложной в реализации считается методика Van Everdingen & Hurst. Накопленный приток из аквифера рассчитывается по уравнению:

$$W_e = 6,278 f \phi h \bar{c} r_o^2 \Delta P W_D,$$

где:

f - безразмерный угол, характеризующий степень охвата залежи аквифером, если аквифер окружает пласт со всех сторон, то $f=1$;

ϕ - пористость аквифера, д/ед;

h - толщина аквифера, м;

\bar{c} - полная сжимаемость аквифера, равна сумме сжимаемостей воды и породы;

r_o - радиус резервуара, м;

ΔP - снижение пластового давления, ат;

W_D - безразмерный приток из аквифера.

Безразмерный приток W_D определяется по палеткам или таблицам и зависит от безразмерного радиуса аквифера ($r_{eD} = r_e / r_o$) и безразмерного времени (здесь время t в годах):

$$t_D = \frac{3,052kt}{\phi \mu \bar{c} r_o^2},$$

где:

k - проницаемость аквифера, мД;

μ - вязкость воды, сП.

После достижения максимума зависимость W_D от t_D представляет собой горизонтальную прямую линию. Это максимальное значение безразмерного притока можно рассчитать по формуле:

$$W_{D(\max)} = 0.5 \cdot (r_{eD}^2 - 1)$$

Следует отметить, что расчет притока по методике Van Everdingen & Hurst является необходимым условием для (а) **оценки запасов** по участку (или месторождению в целом) и (б) **определения режима работы** участка (или месторождения в целом).

Для **оценки запасов** в комплексе с методикой Van Everdingen & Hurst используется методика Navlena&Odeh [9], которая позволяет удостовериться в корректности рассчитанных параметров работы аквифера и оценить запасы интересующего объекта. Принцип методики Navlena&Odeh заключается в том, что в период добычи при пластовом давлении выше давления насыщения по мере снижения пластового давления происходит практически «идеальное» расширение нефти, оставшейся в пласте. Т.е. «работают» именно начальные геологические запасы участвующие в дренировании на самом деле, а не те запасы, которые были изначально оценены объемным или другими методами. Аналогичным образом почти «линейно» (пропорционально) по мере снижения пластового давления в нефтенасыщенной части пласта происходит и изменение свойств пластовой нефти, а именно, объемного коэффициента, вязкости, сжимаемости и плотности. Наиболее важным для оценки запасов и установления механизма работы нефтенасыщенной залежи является наличие регулярных и корректных замеров пластового давления (методом восстановления давления либо по данным о работе скважин на режимах) и объемного коэффициента по данным глубинных проб нефти.

Также критичной для корректной оценки запасов является точность замеров сжимаемостей воды, нефти и породы.

Основная формула, математически описывающая принцип объемного расширения геологических работающих начальных запасов нефти:

$$F = N [E_0 + E_{fw}] + W_e,$$

где:

- F - объем добычи воды и нефти за i -й период;

- N - начальные геологические работающие запасы нефти (искомые НГЗ);

- E_0 и E_{fw} - коэффициенты, описывающие объемное расширение нефти, воды и породы по мере снижения пластового давления через изменение объемных коэффициентов и сжимаемости воды и вмещающей породы, рассчитываемые для каждого периода отдельно;

- W_e - величина водопритока из законтурной области, определяемая для каждого периода отдельно (описательная часть приведена выше);

- $E_0 = B_o - B_{oi}$, где B_o и B_{oi} - текущий и начальный объемные коэффициенты.

Текущий объемный коэффициент в идеальном случае должен определяться на периодической основе по глубинным пробам нефти;

$$- E_{fw} = B_{oi} \left[\frac{c_w S_{wi} + cf}{1 - S_{wi}} \right] \Delta p,$$

условно постоянная величина, описывающая расширение содержащейся в коллекторе воды и изменение порового объема пропорционально разнице давлений от начального пластового давления до текущего пластового давления.

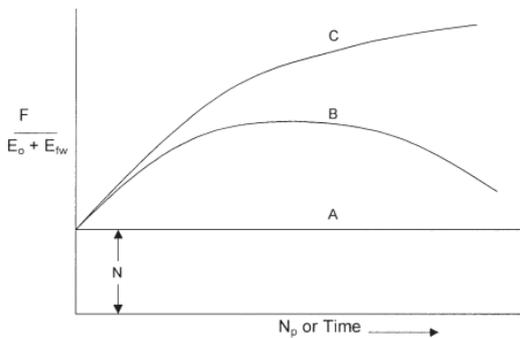


Рис. 1. Диаграмма классификации режима работы нефтенасыщенной залежи и оценки работающих геологических запасов нефти (НГЗ) [10] (Изменение объема работающих геологических запасов в процессе добычи воды и нефти из пласта).

Дейк в своей работе 1994 года приводит следующее описание диаграммы (рис. 1), применяемой для идентификации режима работы залежи:

- горизонтальная линия А свойственна объектам, работающим на режиме истощения, а именно за счет расширения породы, связанной воды и нефти (без притока из законтурной области);

- кривая В свойственна объектам, у которых продуктивность законтурной области ограничена и её энергия истощается по мере добычи нефти;

- кривая С свойственна объектам, у которых законтурная область по отношению к объему запасов нефти многократно больше и «действует» как аквифер бесконечной по времени энергии. Т.е. его продуктивность

не зависит от стадии разработки месторождения.

Рисунок 2 иллюстрирует диаграмму, позволяющую оценить корректность расчетного водопритока. Отклонение от линии 45 градусов вверх или вниз (искривление) будет указывать на ошибку в расчетах уровня активности аквифера, связанную, с недостаточным или, наоборот, излишним объемом водопритока.

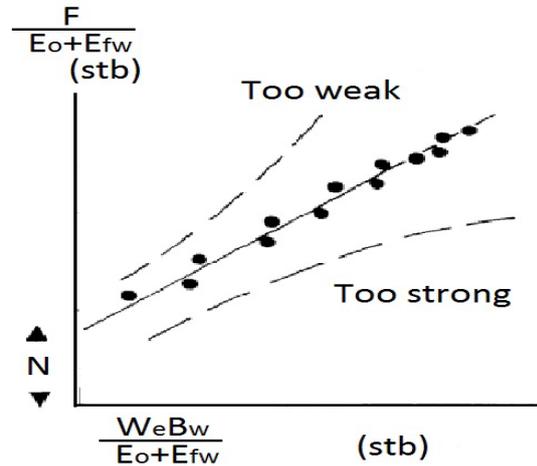


Рис. 2. Диаграмма проверки корректности оценки притока из законтурной области (Havlena&Odeh). [1]

4. Прогноз динамики давления и оценка запасов

Одним из результатов расчетов материального баланса по методике Van Everdingen & Hurst является проверка и уточнение подсчетных параметров, приведенных в геолого-физической характеристике (ГФХ) и таблице результатов интерпретации геофизических исследований скважин (РИГИС), т.к. именно эти параметры используются при расчете материального баланса с целью оценки начальных извлекаемых запасов (НИЗ) и остаточных извлекаемых запасов (ОИЗ).

Анализ ГДИ, в том числе, данных о давлениях и дебитах при работе скважин на режимах был необходим для оценки изменения пластового давления во времени, т.к. при оценке запасов по методике Van Everdingen & Hurst участвуют две «взаимоисключающие» величины – данные о пластовом давлении и активность законтурной области /влияние нагнетания. Т.е. активность законтурной области возможно оценить только при наличии корректных данных о пластовом давлении.

Следующим важным фактором является учет геологического строения продуктивного пласта в плане непроницаемых границ (литологических или тектонических), т.к. расстояние до непроницаемых границ может определять радиус зоны дренирования отдельных скважин.

Основным исследуемым объектом в рамках данной статьи является карбонатный

трещиноватый нефтенасыщенный коллектор девонского возраста. Количество исследованных скважин – 27.

Подход к применению материального баланса будет продемонстрирован на примере одной такой скважины, находящейся в системе разработки в сетке с шагом 500 метров (рис. 3). Расчет запасов и прогноз динамики пластового давления посредством материального баланса скважины №XX01 произведен с (а) подсчетными параметрами из ГФХ и (б) параметрами, полученными по результатам анализа ГДИ и ГИС. Пересматривались преимущественно пористость и эффективная нефтенасыщенная толщина ($H_{эфнн}$).

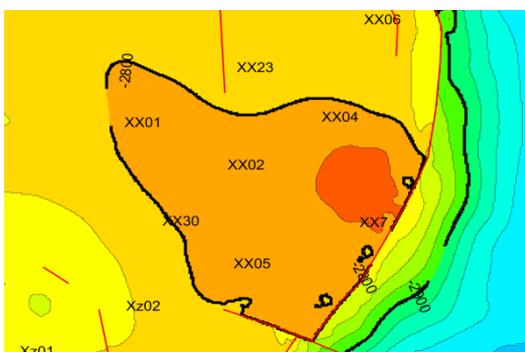


Рис.3 Выкопировка из структурной карты с указанием взаимного расположения исследовавшихся скважин. Масштаб 1:25 000.

Для расчета запасов в соответствии с параметрами, утвержденными в ГФХ, использовались следующие значения величин:

- радиус контура питания, м	500
- пористость по залежи, доли ед.	0.114
- утвержденный КИН, доли ед.	0.35
- $H_{эфнн}$, м	19.4
- рассчитанные НГЗ, тыс. т	700.3
- рассчитанные НИЗ, тыс. т	245.1

Для расчета запасов в соответствии с параметрами, полученными после анализа материалов ГИС и ГДИ использовались следующие значения величин:

- радиус контура питания, м	500
- пористость по ГИС, доли ед.	0.126
- утвержденный КИН ¹ , дес	0.35
- $H_{эфнн}$, м	13.0
- рассчитанные НГЗ, тыс. т	835.3
- рассчитанные НИЗ, тыс. т	292.3

¹ – коэффициент извлечения нефти

По истории добычи скв №XX01 на 01.01.2016 добыла 59.7 тыс. т. нефти, что составляет 9% от рассчитанных НГЗ по ГФХ, и 7% по данным ГДИ и ГИС в районе скважины.

Дальнейший анализ показал, что величины НГЗ и НИЗ, рассчитанные по данным ГФХ, вероятно, оценены некорректно. На рисунках 4 и 5 представлены графики расчетных пластовых давлений. На рисунке 4 представлен график, как если бы скважина не имела поддержки пластового давления и запасы соответствовали бы рассчитанным – добыча прекратилась бы в 2012 г. На рисунке 5 представлен прогноз пластового давления относительно истории добычи с учетом фактических данных и предположениям о наличии поддержки пластового давления и большей величины запасов. Очевидно, что в оценке запасов, исходя из начальных параметров, имеется неточность. Данные о необходимой корректировке будут приведены далее.

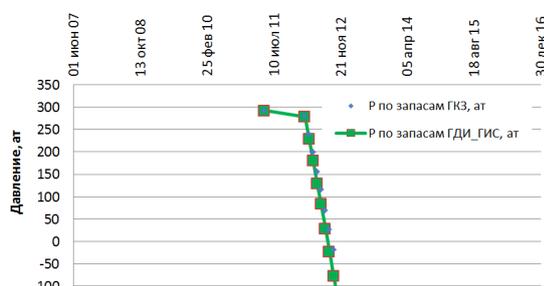


Рис.4 Расчетное пластовое давление в районе скв. XX01 исходя из запасов, рассчитанных по параметрам пласта по ГФХ.

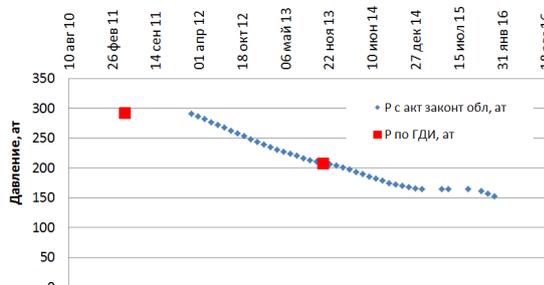


Рис.5 Расчетное пластовое давление в районе скв. XX01 исходя из запасов, рассчитанных с учетом предположения о поддержке пластового давления.

Таким образом, можно утверждать, что пласт вокруг скважины №XX01 имеет поддержку давления из законтурной области и/или содержит большой объем запасов.

Расчет исторического давления и оценка влияния законтурной области проводились в следующем порядке:

1. На каждый момент времени рассчитывается накопленная добыча нефти и воды.

2. На каждый момент времени рассчитывается изменение давления от накопленной добычи по двум вариантам (по параметрам, утвержденным по ГФХ и по

параметрам полученным при анализе ГИС и ГДИ).

3. На каждый момент времени по формулам, приведенным в разделе 2 настоящей статьи, производится расчет притока воды из законтурной области с учетом снижения давления от добычи и исторического пластового давления, полученного по результатам ГДИ скважин. Приток «регулируется» размером водоносной области и отношением её размеров к размерам анализируемого участка (итерационный расчет). Таким образом, если имеет место однозначность в размерах зоны дренирования запасов и в их объеме, то становится возможным оценить степень влияния законтурной области и рассчитать прогноз изменения пластового давления.

Расчет изменения пластового давления с использованием материального баланса проводился параллельно для обоих вариантов оценки запасов с вариантами как для замкнутого пласта, так и для влияния законтурной области.

Следующим этапом проводилась оценка работающих геологических запасов (НГЗ) с применением методики графического анализа Havlena&Odeh, описание которой приводилось выше. Первым этапом производилось определение режима работы залежи согласно рис. №6. Для скв №XX01 дугообразная возрастающая линия свидетельствует о работе залежи с преимущественной поддержкой из законтурной области либо за счет нагнетания. Следовательно, история и прогноз пластового давления в районе скважины должны обязательно производиться с учетом притока воды, но никак не для режима истощения.

Кроме проверки наличия водопритока и поддержки добычи нефти за счет вытеснения её водой материальный баланс позволяет оценить объем геологических запасов, находящихся в зоне дренирования конкретной скважины посредством расчета комплексных параметров [11] в сопоставлении с накопленной добычей жидкости на разные моменты времени (рис. 7).

Число, описывающее наклон линии тренда, обозначает величину геологических запасов и составляет 3526.5 тыс. т нефти для скважины №XX01. Как можно отметить, это число в 3.5 раза больше геологических запасов, посчитанных объемным методом, исходя из радиуса зоны дренирования для

скважины №XX01 и параметра пористости. Другими словами, исследуемая скважина, вероятнее всего, имеет много больший (чем 500 метров, согласно сетке разбуривания) радиус контура питания и гораздо большее значение параметра пористости. И тому и другому предположению есть подтверждения, связанные с особенностями строения карбонатного трещиноватого нефтенасыщенного коллектора, а также с неточностями при определении подсчетного параметра пористости, причины которых будут объяснены ниже.

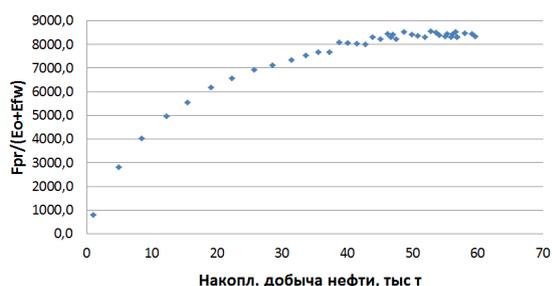


Рис.6 Диаграмма определения режима дренирования пласта в районе скв. №XX01.

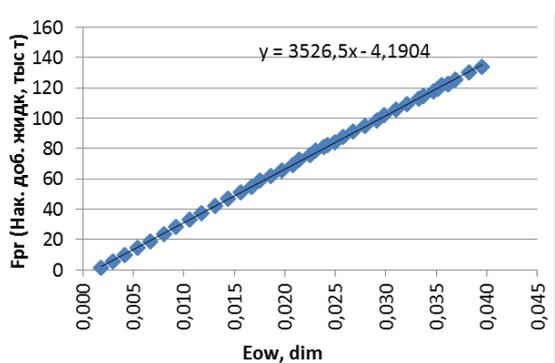


Рис.7 Диаграмма определения величины начальных геологических запасов в районе скв № XX01.

Аналогичные заключения были получены по большинству скважин рассматриваемого объекта, обработанных описанным выше образом. Также были сделаны выводы о несоответствии величин запасов, рассчитанных объемным методом и методом материального баланса. Причины несоответствий будут подробно рассмотрены в следующем разделе.

5. Об оценке объема пустотности карбонатных трещиноватых коллекторов

Подход к определению пустотности рассматриваемого коллектора, сложившийся в современной практике ГИС открытого ствола, преимущественно классический [12-15]. Т.е. используются три ключевых метода для

определения пористости: нейтронный, плотностной и акустический. Межзерновая пористость определяется по среднему значению между этих трех методов. При завышенном значении пористости по акустическому каротажу делается вывод о наличии каверновой и трещинной составляющих [12-15]. Но данные выводы носят преимущественно качественный характер и имеют место при наличии соответствующей квалификации интерпретаторов.

Ограничения ГИС открытого ствола при определении кавернозности. Нейтронный и плотностной методы практически не регистрируют разницы для межзерновой и кавернозной составляющей, но поскольку объем каверн на единицу объема породы значительно меньше объема пор, то и общая пористость исследуемой породы будет казаться ниже.

Акустический каротаж имеет особенность в том, что продольная волна реагирует на трещиноватость. Вторая особенность в том, что акустическая волна «огихает» слабосообщающиеся с поровым объемом каверны, и при интерпретации этот объем также «выпадает». Значительные различия в показаниях пористости по акустическому каротажу и нейтронному каротажу относят к трещинно-кавернозной составляющей, но по причине, описанной выше этот объем пустотности сильно занижен относительно реально существующего (рис. 8) [13].

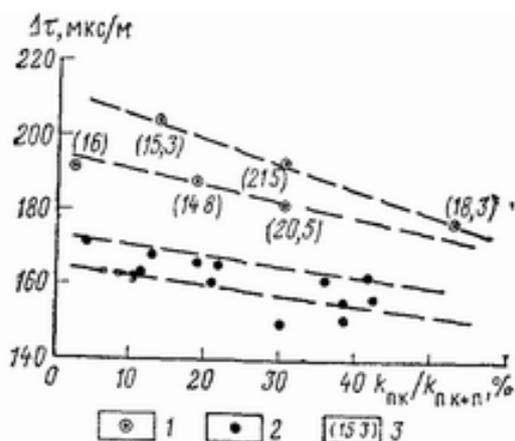


Рис.8 Сопоставление интервального времени пробега упругой волны $\Delta\tau$ с долей кавернозности $k_{пк}/k_{пк+п}$ образцов пород. 1 — образцы с диаметром каверн $d_{кав}$ более 5 мм, 2 — образцы мелкокавернозные $d_{кав} < 2-3$ мм, 3 — полный коэффициент пористости образца $k_{пк+п}$ в % [13].

На результатах интерпретации преимущественно видна только трещинная составляющая, а объем кавернозной, как правило остается неизвестен и определяется только косвенно, например, посредством материального баланса.

Волна Лэмба-Стоунли для определения объема кавернозности не подходит, т.к. "работает" только на трещиноватость. Остальные методы ГИС также мало-информативны для определения объема каверн. Так, электрические фокусированные зонды не реагируют на каверны, которые остаются заполненными нефтью даже в зоне проникновения фильтрата бурового раствора и повышают сопротивление породы во время ГИС открытого ствола [16].

Вероятные пути решения проблемы с определением кавернозной составляющей. Вероятно, данную проблему можно частично решить посредством использования импульсного нейтронного каротажа в варианте СО каротажа, т.к. зоне проникновения фильтрата бурового раствора каверны остаются заполненными нефтью из-за плохой сообщаемости с остальным пустотным пространством. Но этот метод не используется в процессе ГИС открытого ствола. Другим вероятным способом решения является использование тщательно увязанного по глубине керна и распиленного на шлифы с целью подсчета объема кавернозной составляющей в общем объеме пустотности с целью определения поправочных коэффициентов для расчета пористости.

6. Заключение

В работе представлен пример интегрированного анализа данных по истории добычи отдельных скважин с последующим выявлением режимов дренирования нефтенасыщенного коллектора. Кроме определения режимов дренирования коллектора по отдельным скважинам обнаружено значительное несоответствие в подсчетных параметрах, определяющих объем дренируемых запасов по исследовавшимся скважинам. Приведены возможные причины несоответствий, а также возможные пути корректировки неточности в исходных данных.

Работа выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН на 2019 год по программе ФНИ государственных академий наук на 2013-2020 годы «Создание методики выявления невыработанных зон на нефтяных месторождениях и подсчета остаточных запасов нефти на основе комплексирования математического моделирования, анализа разработки с исследованиями скважин и пластов», тема (проект) № 0065-2019-0019.

Integrated analysis of well test and open-hole logging data for undeveloped zones identification

A.A. Kolevatov, Yu. B. Chen-len-son, Yu.M. Steinberg, I.V. Afanaskin

Abstract. Article describes conclusions of wells production history integrated analysis conducted for carbonate fractured oil saturated reservoir. Analysis results identified initial and calculated data inconsistency, which refers to reservoir volumetric parameters, that applied to geological and recoverable reserves estimation. Volumetric data inconsistency sources were investigated to identify potential error of porosity parameter.

Key words: carbonate fractured oil saturated reservoir, well test, volumetric parameters.

Литература

1. L.P. Dake. Material balance applied to oil reservoirs. Ch.3. 1978. ISBN: 978-0-444-41667-4.
2. Д. Бурде Интерпретация результатов исследований скважин. Материалы лекций. Petroleum Engineering and Related Management Training Gubkin Academy. Москва, 1994. 109 с.
3. Т.А. Деева, М.Р. Камардинов, Т.Е. Кулагина и др. Гидродинамические исследования скважин: анализ и интерпретация данных. Томск: ЦППС НД ТПУ, 2009. – 243 с.
4. Л.Г. Кульпин, Ю.А. Мясников Гидродинамические методы исследования нефтегазоводонасыщенных пластов. М.: Недра, 1974. – 200 с.
5. Оливье Узе, Дидье Витура, Оле Фьяре. Анализ динамических потоков. Теория и практика интерпретации данных ГДИС и анализа добычи, а также использование данных стационарных глубинных манометров. Carpa Engineering, 2009. – 359 с.
6. Bourdet D. et al. A New Set of Type Curves Simplifies Well Test Analysis // World Oil. 1983. May. P. 95–106.
7. D. Bourdet Well Test Analysis: The Use of Advanced Interpretation Models. 2002. - 436 p.
8. L.P. Dake. Natural Water Influx. Ch.9. 1978. ISBN: 978-0-444-41667-4.
9. Victor T Biu, Total E & P; Okotie Sylvester, Stanley Buduka. A Time Function Havlena and Odeh MBE Straight Line Equation. SPE-178408-MS., Energia Limited, Federal University of Petroleum Resources
10. L.P. Dake. The practice of reservoir engineering. Elsevier, 1994.
11. Mark P. Walsh. Effect of Pressure Uncertainty on Material-Balance Plots., SPE, Petroleum Recovery Research Institute. SPE 56691. 1990.
12. М.Г. Латышова, В.Г. Мартынов, Т.Ф. Соколова Практическое руководство по интерпретации данных ГИС. Недра, Москва, 2007 г., 327 стр., УДК: 550.83.001.2:553.8(075.8), ISBN: 978-5-8365-0299-7
13. Г.М. Золоева Изучение карбонатных коллекторов методами промысловой геофизики Недра, Москва, 1977 г., 150 стр.
14. В.М. Добрынин, Б.Ю. Вендельштейн, Д.А. Кожевников – Петрофизика. Нефть и газ, Москва, 2004 г., 368 стр., УДК: 550.832-П (075), ISBN: 5-7246-0295-4
15. А.С. Некрасов Геолого-геофизические исследования карбонатных коллекторов нефтяных месторождений. Издательство Пермского университета, Пермь, 2006 г., 422 стр., УДК: 553.982.2:550.83, ISBN: 5-7944-0725-5.
16. Б.Ю. Вендельштейн, Р.А. Резванов Геофизические методы определения параметров нефтегазовых коллекторов М.: Недра, 1978. -318 с.

Анализ эффективности применения горизонтальных скважин с помощью гидродинамических исследований

П.В. Крыганов¹, И.В. Афанаскин², С.Г. Вольпин³

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, kryganov@mail.ru;

² ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ivan@afanaskin.ru;

³ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, sergvolpin@gmail.com

Аннотация: В работе изучены характеристики эффективности работы горизонтальных скважин с помощью гидродинамических исследований методом установившихся отборов, позволяющие оценить их преимущества по сравнению с вертикальными скважинами. Проведен анализ особенностей интерпретации гидродинамических исследований методом восстановления давления, связанных с отсутствием достоверной информации об эффективной длине горизонтального ствола скважины.

Ключевые слова: горизонтальная скважина, гидродинамические исследования

1. Введение

Использование горизонтальных скважин для разработки нефтяных месторождений России в большинстве случаев уже несколько десятилетий подтверждает их преимущество перед вертикальными скважинами. Их эффективность доказана временем из-за вовлечения все большего количества трудноизвлекаемых запасов в общую добычу нефти в стране. В особенности это касается тех случаев, в которых горизонтальные скважины служат практически единственным возможным способом извлечения нефти, а именно при:

- разработке коллекторов с пониженной проницаемостью, с повышенной вязкостью нефти и пластов с небольшой эффективной толщиной — во всех этих случаях использование вертикальных скважин неминуемо приводит к низкому коэффициенту продуктивности (по сравнению с горизонтальными скважинами);
- разработке тонких нефтяных оторочек — во избежание подтягивания воды и/или газа, что неминуемо бы имело место при использовании вертикальных скважин, работающих при повышенных депрессиях.

Однако горизонтальные скважины при всех их вышеперечисленных преимуществах значительно дороже при бурении, при эксплуатации и при различных типах исследований и вопрос целесообразности их применения всегда актуален по отношению к вертикальным скважинам. Данная работа

направлена на изучение сравнительных характеристик эффективности применения горизонтальных скважин с помощью гидродинамических исследований [1] методом установившихся отборов и на анализ особенностей интерпретации гидродинамических исследований методом восстановления давления, связанных с отсутствием достоверной информации об эффективной длине горизонтального ствола скважины.

2. Установившийся приток жидкости к горизонтальному стволу скважины в анизотропном пласте

В работе [2] приведена формула для расчета дебита горизонтальной скважины, полученная Ренардом и Дюпой в результате точного решения задачи об установившемся притоке несжимаемой жидкости к горизонтальной скважине в бесконечном недеформируемом пласте ограниченной мощности с учетом вертикальной анизотропии проницаемости:

$$q = \frac{2\pi k_h h \Delta P}{\mu \left[\operatorname{Arch}\left(\frac{2a}{L}\right) + \left(\frac{\beta h}{L}\right) \ln\left(\frac{h}{2\pi r_w^*}\right) \right]}$$

(1)

где q - дебит скважины в поверхностных условиях; h - толщина пласта; μ - вязкость жидкости; ΔP - депрессия на пласт;

$a = l \left[0.5 + \sqrt{0.25 + (R_k / l)^4} \right]$; $\beta = \sqrt{k_h / k_v}$; L - длина горизонтальной части ствола скважины; $r_w^* = [(1 + \beta) / (2\beta)] r_w$; k_v - вертикальная проницаемость; k_h - горизонтальная проницаемость, R_k - радиус контура питания.

Также в работе [2] приведена приближенная формула Джоши для решения той же задачи:

$$q = \frac{2\pi k_h h \Delta P}{\mu \left[\ln \left(\frac{a + \sqrt{a^2 - 0,25L^2}}{0,5L} \right) + \left(\frac{\beta h}{L} \right) \ln \left(\frac{\beta h}{2\pi r_w} \right) \right]} \quad (2)$$

Для сравнения дебита горизонтальной скважины с дебитом вертикальной использована формула Дююи, описывающая плоскорадиальный установившийся приток несжимаемой жидкости к совершенной по степени и характеру вскрытия вертикальной скважине в бесконечном недеформируемом пласте ограниченной мощности [4]:

$$q = \frac{2\pi k_h h}{\mu} \frac{\Delta P}{\left(\ln \frac{R_k}{r_c} \right)}, \quad (3)$$

3. Анализ эффективности горизонтальных скважин при установившемся притоке жидкости в анизотропном пласте

Расчеты согласно (2) и (3) позволили определить зависимость отношения коэффициента продуктивности горизонтальной скважины к коэффициенту продуктивности вертикальной скважины (K_{zc}/K_{ac}) от отношения длины горизонтального ствола (рассмотрена длина от 100 м до 1200 м) к фиксированной эффективной толщине (10 м) при 6-ти разных значениях коэффициента анизотропии k_v/k_h (1, 0.5, 0.05, 0.01, 0.005, 0.001) для 2-х значений радиуса контура питания $R_k=250$ м (рис. 1) и

$R_k=150$ м (рис. 2). В данной работе для расчетов был использован общепринятый коэффициент анизотропии k_v/k_h вместо k_h/k_v , присутствующий в формуле (1). Значения радиусов контура питания выбраны таким образом, поскольку $R_k=250$ м и $R_k=150$ м соответствуют наибольшему (500 м) и наименьшему (300 м) расстоянию между рядами скважин, которые чаще всего встречаются на практике.

Из рис. 1, 2 виден непрерывный рост кривых K_{zc}/K_{ac} с ростом длины горизонтального ствола, что говорит о большей эффективности (почти в 20 раз – рис. 1, почти в 10 раз – рис. 2) горизонтальных скважин с более длинными горизонтальными стволами. Следует отметить, что такая завышенная эффективность может объясняться тем, что при расчетах по формуле (2) не учитывалось влияние трения в горизонтальном стволе скважины, которое может быть существенной при длине в несколько сот метров.

Искривление линий тока вблизи ствола скважины приводит к увеличению фильтрационного сопротивления, поскольку фильтрационное сопротивление зависит от длины горизонтального ствола [3]. И поэтому даже в изотропном пласте (при значении $1/\beta^2=1$) на рис. 1, 2 видны такие L/h , при которых эффективность горизонтальной скважины соизмерима с вертикальной скважиной.

На обоих рис. 1, 2 видно некоторое количество точек со значениями $K_{zc}/K_{ac}<1$, а значит с меньшей эффективностью горизонтальных скважин по сравнению с вертикальными. Это также объясняется увеличением фильтрационного сопротивления при искривлении линий тока вблизи ствола скважины и ростом этого эффекта при увеличении анизотропии.

Эти графики могут быть использованы для прогноза продуктивности горизонтальных скважин на любом месторождении.

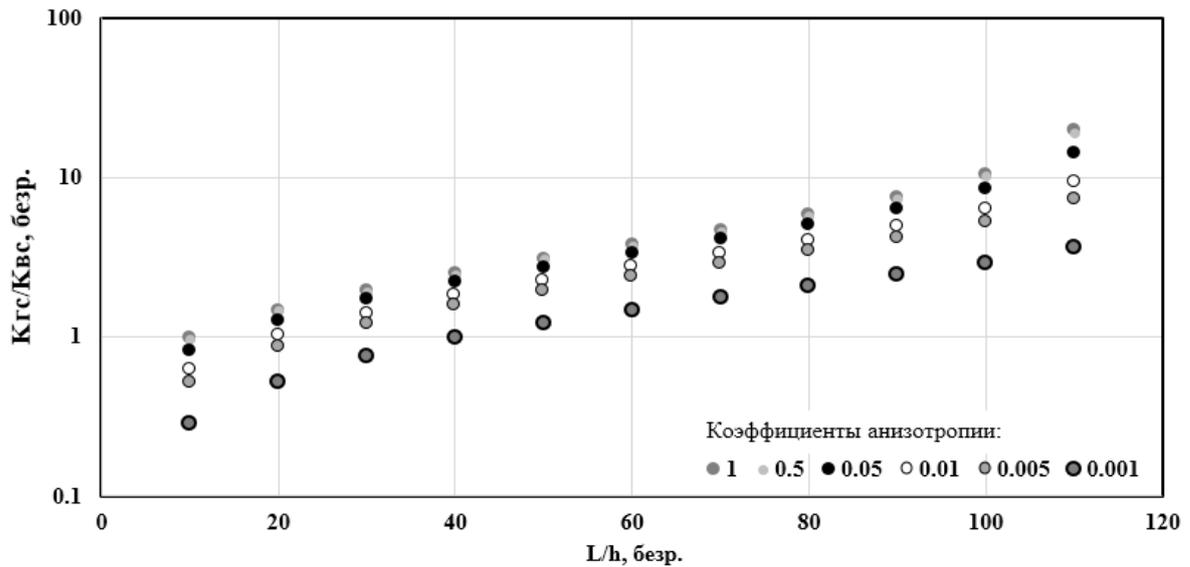


Рис. 1. Зависимость K_{zc}/K_{bc} от L/h при разных значениях анизотропии, $R_k = 250$ м

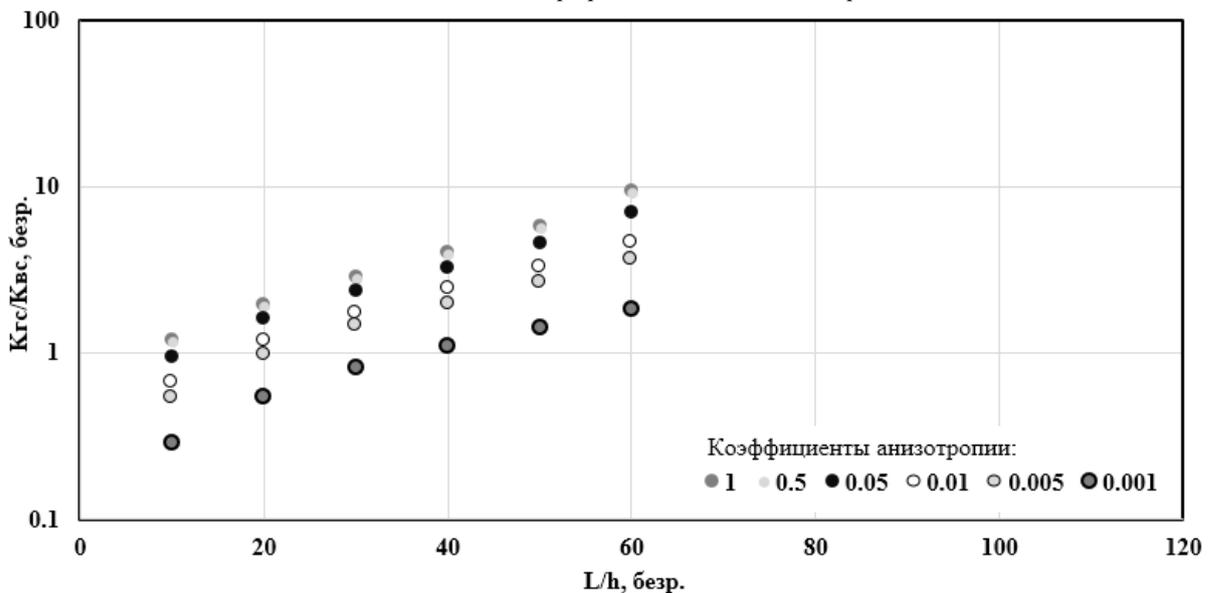


Рис. 2. Зависимость K_{zc}/K_{bc} от L/h при разных значениях анизотропии, $R_k = 150$ м

4. Неустановившийся приток жидкости к горизонтальному стволу скважины в анизотропном пласте

Исходное трехмерное уравнение пьезопроводности, которое позволяет описать изменение давления после смены режима работы горизонтальной скважины, записывается как:

$$(4) \quad \frac{\mu\beta^*}{k_y} \frac{dP}{dt} = \frac{k_x}{k_y} \frac{d^2P}{dx^2} + \frac{d^2P}{dy^2} + \frac{k_z}{k_y} \frac{d^2P}{dz^2},$$

где k_x , k_y , k_z - проницаемость пласта в направлении x , y , z ; μ - вязкость, β^* - упругость системы пласт-флюид.

Для того, чтобы найти аналитическое решение выражения (4), необходимо его упростить или подключить дополнительные условия. Изучение литературы показало, что описание неустановившегося потока к горизонтальной скважине в большинстве опубликованных методиках сводится к применению решений для 2 потоков - плоскорадиального и прямолинейно-параллельного [6, 7, 9-11, 13-24].

Рассмотрим обобщенные решения уравнения нестационарной фильтрации в анизотропном пласте во время работы горизонтальной скважины с переменным дебитом.

В работе [24] предложено подробное решение дифференциального уравнения, описывающего неустановившуюся фильтрацию жидкости к наклонной скважине с постоянным дебитом, сделано обобщение для случая переменного дебита горизонтальных скважин, а также приведено решение для распределения потока по длине скважины. В полярных координатах уравнение (4) имеет вид:

$$\frac{1}{\aleph} \frac{\partial p}{\partial t} = \frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} + \frac{1}{r^2} \frac{\partial^2 p}{\partial \theta^2} + \frac{k_z}{k} \frac{\partial^2 p}{\partial z^2}. \quad (5)$$

где \aleph - коэффициент пьезопроводности.

При этом было допущено [5], что:

1. Пласт анизотропный бесконечный по горизонтали с толщиной h , с пористостью m , с радиальной проницаемостью k и вертикальной проницаемостью k_z , без зависимости от времени и давления.

2. Жидкость слабосжимаема, постоянной вязкости μ и сжимаемости $\beta_{ж}$.

3. Скважина наклонная с углом наклона α , с длиной a эксплуатируемая с постоянным дебитом q , распределенным равномерно по длине и радиусом r_c . Связь продуктивного интервала и толщины пласта записывается, как:

$$a \leq \frac{h}{\cos \alpha}; \quad \frac{a}{2} \leq \frac{z_c}{\cos \alpha}; \quad \frac{a}{2} \leq \frac{h - z_c}{\cos \alpha}.$$

4. Кровля и подошва пласта непроницаемы, т.е.

$$\left. \frac{\partial p}{\partial z} \right|_{z=h} = 0, \quad \left. \frac{\partial p}{\partial z} \right|_{z=0} = 0.$$

5. Первоначальное пластовое давление равно p_0 и не изменяется при $r \rightarrow \infty$, т.е.

$$\lim_{r \rightarrow \infty} p = p_0$$

Задача решалась с использованием уравнения мгновенного источника-стока

$$\Delta p = \frac{q\mu}{8\pi^{3/2}k\sqrt{\aleph_z}t^{3/2}} \exp\left(-\frac{r^2}{4\aleph t} - \frac{z^2}{4\aleph_z t}\right) \quad (6)$$

Изменение давления, связанное с работой источника наклонного и с равномерным дебитом в анизотропном пласте в точке (r, θ, z) равно:

$$\Delta p = \frac{q\mu}{8\pi^{3/2}ka\sqrt{\aleph} \cos \alpha} \times \int_0^t \frac{1}{\tau^{3/2}} \sum_{k=-\infty}^{\infty} \int_{\frac{a}{2} \cos \alpha}^{\frac{a}{2} \cos \alpha} \exp\left[-\frac{r^2 + z_p^2 \operatorname{tg}^2 \alpha - 2rz_p \operatorname{tg} \alpha \cos \theta}{4\aleph \tau} - \frac{(z + 2kh - z_c - z_p)^2}{4\aleph_z \tau}\right] +$$

$$+ \exp\left[-\frac{r^2 + z_p^2 \operatorname{tg}^2 \alpha + 2rz_p \operatorname{tg} \alpha \cos \theta}{4\aleph \tau} -$$

$$\frac{(z + 2kh + z_c - z_p)^2}{4\aleph_z \tau}\right] dz_p d\tau \quad (7)$$

Прделав ряд операций по замене переменных и вводу обозначений, получено [5] безразмерное уравнение (7), которое по форме оказывается соответствующим уравнению изотропного пласта:

$$\Delta p = \frac{h^*}{\sqrt{\pi a^* \cos \alpha_{np}}} \int_0^{st/r_c^2} \frac{\exp\left(-\frac{r^{*2}}{4\tau'}\right)}{(\tau')^{3/2}} \times \sum_{k=-\infty}^{\infty} \int_{\frac{a^*}{2} \cos \alpha_{np}}^{\frac{a^*}{2} \cos \alpha_{np}} \left\{ \exp\left[-\frac{z^{*2} \operatorname{tg} \alpha_{np} - 2r^* z_0^* \operatorname{tg} \alpha_{np} \cos \theta + (z^* + 2kh^* - z_c^* - z_0^*)^2}{4\tau'}\right] + \exp\left[-\frac{z^{*2} \operatorname{tg} \alpha_{np} + 2r^* z_0^* \operatorname{tg} \alpha_{np} \cos \theta + (z^* + 2kh^* + z_c^* - z_0^*)^2}{4\tau'}\right] dz_0^* d\tau' \right. \quad (8)$$

Из уравнения (8) видно, что давление не будет одинаковым по длине стенки скважины, поскольку зависит от угла θ и координаты z_0 . Этого можно избежать путем усреднения давления интегрированием по длине ствола уравнения (8).

При определении распределения давления в высоконаклонных скважинах и на дальнем расстоянии от источника можно применить тета-функции [4]. При этом распределение давления в пласте, связанное работой горизонтальной скважины, определяется путем третьих тета-функций выражением:

$$\Delta p(x, y, z, t) = \frac{q\mu}{4\pi k h a} \int_0^t \frac{\exp\left(-\frac{y^2}{4\aleph t}\right)}{\tau} \left[v_3\left(\frac{z}{2h}, \frac{\pi \aleph_z \tau}{h^2}\right) + v_3\left(\frac{2\delta - z}{2h}, \frac{\pi \aleph_z \tau}{h^2}\right) \right] \int_{-a/2}^{a/2} \exp\left[-\frac{(x - x_0)^2}{4\aleph \tau}\right] dx_0 d\tau \quad (9)$$

Параметры тета-функций $z/2h$ и $(2\delta - z)/2h$ меняются в пределах $0 \div 0,5$. Характер поведения этих функций в зависимости от $\pi \aleph_z t/h^2$ показан на рис. 3. Как видно, для времен $t > h^2/\pi \aleph_z$ сумма тета-функций под интегралом постоянна и равна 2. Это значит, что горизонтальная скважина с момента времени

$t \geq h^2/\pi \kappa_z$ равнозначна вертикальной трещине той же длины.

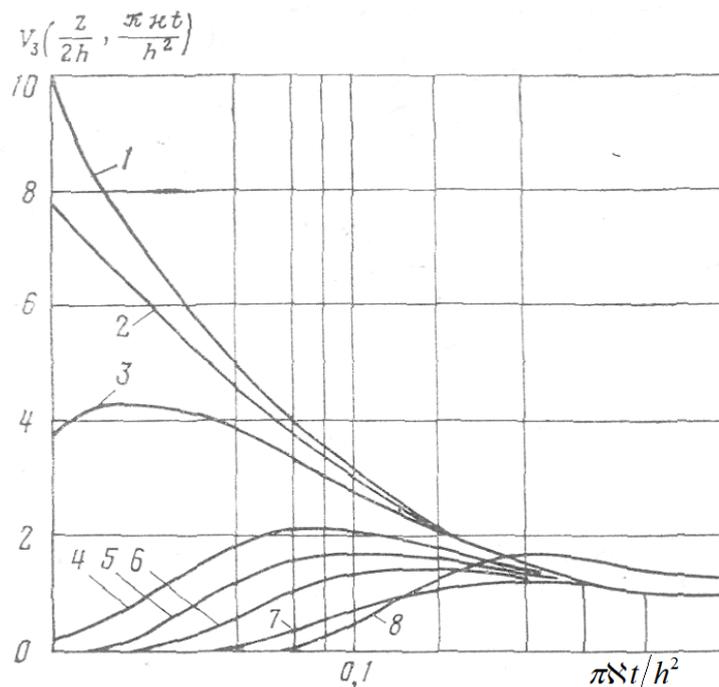


Рис. 3. Зависимость $V_3\left(\frac{z}{2h}, \frac{\pi \delta t}{h^2}\right)$ от $\frac{\pi \delta t}{h^2}$;

1, 2, 3, 4, 5, 6, 7, 8 – $z/2h=0; 0,05; 0,10; 0,20; 0,25; 0,30; 0,40; 0,50$ соответственно [4]

Будем иметь:

$$\Delta p(x, y, z, t) = \frac{q\mu}{8\pi^{3/2}k_z\sqrt{\kappa_z}} \frac{1}{a} \left\langle \int_0^t \frac{\exp\left(-\frac{y^2}{4\kappa_z\tau}\right)}{\tau^{3/2}} \times \right. \\ \times \int_{-a/2}^{a/2} \exp\left[-\frac{(x-x_0)^2}{4\kappa_z\tau}\right] dx_0 \times \\ \times \sum_{k=-\infty}^{\infty} \left\{ \exp\left[-\frac{(z+2kh)^2 \frac{\kappa_z}{N_z}}{4\kappa_z\tau}\right] + \right. \\ \left. + \exp\left[-\frac{(2\delta-z+2kh)^2 \frac{\kappa_z}{N_z}}{4\kappa_z\tau}\right] \right\} d\tau \right\rangle \quad (10)$$

При интегрировании и замене переменных в (10), получаются следующие типовые интегралы:

$$M(u, \beta) = \int_u^{\infty} \frac{\exp(-x)}{x} \operatorname{erf} \beta \sqrt{x} dx.$$

При применении функций $M(u, \beta)$ выражение для изменения давления в пласте, связанного с влиянием работы горизонтальной скважины выглядит, как

$$\Delta p(x, y, z, t) = \frac{q\mu}{8\pi kh} \frac{h\sqrt{\frac{k}{k_z}}}{a} \times \\ \times \left\{ \sum_{k=-\infty}^{\infty} M \left[\frac{y^2 + (z+2kh)^2 \frac{k}{k_z}}{4\kappa_z t}, \frac{\left(x + \frac{a}{2}\right)}{\sqrt{y^2 + (z+2kh)^2 \frac{k}{k_z}}} \right] - \right. \\ - M \left[\frac{y^2 + (z+2kh)^2 \frac{k}{k_z}}{4\kappa_z t}, \frac{\left(x - \frac{a}{2}\right)}{\sqrt{y^2 + (z+2kh)^2 \frac{k}{k_z}}} \right] + \\ + M \left[\frac{y^2 + (2\delta - z + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}, \frac{\left(x + \frac{a}{2}\right)}{\sqrt{y^2 + (2\delta - z + 2kh)^2 \frac{k}{k_z}}} \right] - \\ \left. - M \left[\frac{y^2 + (2\delta - z + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}, \frac{\left(x - \frac{a}{2}\right)}{\sqrt{y^2 + (2\delta - z + 2kh)^2 \frac{k}{k_z}}} \right] \right\} \quad (11)$$

На рис. 4 на момент времени $t_1 = h^2/\kappa_z$ для условий $\frac{x}{h} \sqrt{\frac{k_z}{k}} = \frac{y}{h} \sqrt{\frac{k_z}{k}} = \frac{a}{h} \sqrt{\frac{k_z}{k}}$, $z/h=0$ представлены результаты расчета изменения безразмерного давления $\Delta p^* = \Delta p(x, y, z, t) \delta \pi k h / q \mu$ от отношения $\frac{a}{h} \sqrt{\frac{k_z}{k}}$ при разных δ/h , то есть изменения расположения горизонтальных скважин относительно кровли и подошвы.

Для того, чтобы определить изменение давления на стенке параллельной горизонтальной скважины длиной $b = x_2 - x_1 = a$ необходимо проинтегрировать (10) по x . Так как решение при этом не изменится, распределение давления лишь на стенке скважины равно:

$$\begin{aligned} \Delta p(r_c, t) = & \frac{q\mu}{4\pi k h} \frac{h}{a} \sqrt{\frac{\kappa_z}{\kappa_z}} \times \\ & \times \left\{ \sum_{k=-\infty}^{\infty} M \left[\frac{(r_c + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}, \frac{a}{(r_c + 2kh) \sqrt{\frac{k}{k_z}}} \right] - \right. \\ & - M \left[\frac{(2\delta - r_c + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}, \frac{a}{(2\delta - r_c + 2kh) \sqrt{\frac{k}{k_z}}} \right] + \\ & + \frac{4\sqrt{\kappa_z t}}{a} \left[\operatorname{ierfc} \sqrt{\frac{a^2 + (r_c + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}} + \right. \\ & + \operatorname{ierfc} \sqrt{\frac{a^2 + (2\delta - r_c + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}} - \\ & - \operatorname{ierfc} \sqrt{\frac{(r_c + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}} - \\ & \left. \left. - \operatorname{ierfc} \sqrt{\frac{(2\delta - r_c + 2kh)^2 \frac{k}{k_z}}{4\kappa_z t}} \right] \right\} \end{aligned} \quad (12)$$

Для обобщения полученных решений нестационарной фильтрации в анизотропном пласте при переменном дебите горизонтальных, наклонных, несовершенных скважин делается предположение, что дебит скважины является функцией времени и независимо от сложности его изменения может быть аппроксимирован с помощью полиномов с общим членом $q_n t^n$ (q_n – постоянный коэффициент; n – показатель полинома).

В случае точечного источника [4] и отражения источников-стоков при анизотропии пласта для неустановившегося распределения давления, связанного с влиянием горизонтальной скважины с равномерным дебитом, определяемого степенной функцией времени t^n (при условии $q_n = 1$), запишем выражение:

$$\begin{aligned} \Delta p(x, y, z, t) = & \frac{\mu}{8\pi^{3/2} k \sqrt{\kappa_z}} \frac{1}{a} \times \\ & \times \left\langle \int_0^t \frac{(t-\tau)^n}{\tau^{3/2}} \int_{-a/2}^{a/2} \exp\left[-\frac{(x-x_0)^2}{4\kappa_z \tau}\right] dx_0 \times \right. \\ & \times \exp\left(-\frac{y^2}{4\kappa_z \tau}\right) \times \\ & \times \sum_{k=-\infty}^{\infty} \left\{ \exp\left[-\frac{(z+2kh)^2}{4\kappa_z \tau}\right] + \right. \\ & \left. \left. + \exp\left[-\frac{(2\delta - z + 2kh)^2}{4\kappa_z \tau}\right] \right\} \right\rangle \end{aligned} \quad (13)$$

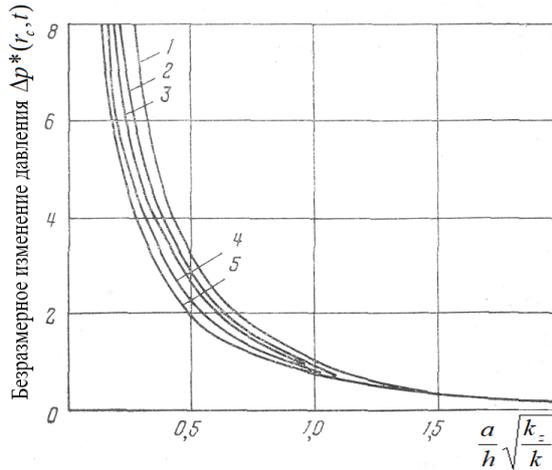


Рис. 4. Зависимость безразмерного давления $\Delta p(r_c, t) = \frac{\Delta p(r_c, t) 8\pi kh}{q\mu}$ от

отношения длины скважины и ее удаления от кровли к толщине пласта

$$\left(\frac{a}{h} \sqrt{\frac{k_z}{k}} u \frac{\delta}{h} \right): 1, 2, 3, 4, 5 - \frac{\delta}{h} = 0,01; 0,05; 0,10; 0,25; 0,50 \text{ соответственно [4]}$$

Для расчета изменения давления на стенке горизонтальной скважины необходимо проинтегрировать (10) по x от $x_1 = -a/2$ до $x_2 = a/2$, при условии $y=0$, $z=r_c$, и после чего усреднить по длине фильтра. Опуская промежуточные преобразования, получено (14)

$$\begin{aligned} \Delta p(r_c, t) = & \frac{\mu}{4\pi kh} \frac{h}{\sqrt{\frac{k_z}{k}}} \frac{1}{a} \times \\ & \times \sum_{k=-\infty}^{\infty} \int_{\frac{\xi_0^2(k)}{4\aleph t}}^{\infty} \frac{\exp(-u) \left(t - \frac{\xi_0^2(k)}{4\aleph u} \right)^n}{u} \operatorname{erf} \frac{a}{\xi_0(k)} \sqrt{u} du + \\ & + \int_{\frac{\eta_0^2(k)}{4\aleph t}}^{\infty} \frac{\exp(-u) \left(t - \frac{\eta_0^2(k)}{4\aleph u} \right)^n}{u} \operatorname{erf} \frac{a}{\eta_0(k)} \sqrt{u} du + \\ & + \frac{2}{\sqrt{\pi}} \frac{\sqrt{a^2 + \xi_0^2(k)}}{a} \int_{\frac{a^2 + \xi_0^2(k)}{4\aleph t}}^{\infty} \frac{\exp(-u) \left(t - \frac{a^2 + \xi_0^2(k)}{4\aleph u} \right)^n}{u\sqrt{u}} du + \\ & + \frac{2}{\sqrt{\pi}} \frac{\sqrt{a^2 + \eta_0^2(k)}}{a} \int_{\frac{a^2 + \eta_0^2(k)}{4\aleph t}}^{\infty} \frac{\exp(-u) \left(t - \frac{a^2 + \eta_0^2(k)}{4\aleph u} \right)^n}{u\sqrt{u}} du - \\ & - \frac{2\xi_0(k)}{a\sqrt{\pi}} \int_{\frac{\xi_0^2(k)}{4\aleph t}}^{\infty} \frac{\exp(-u) \left(t - \frac{\xi_0^2(k)}{4\aleph u} \right)^n}{u\sqrt{u}} du - \\ & - \frac{2\eta_0(k)}{a\sqrt{\pi}} \int_{\frac{\eta_0^2(k)}{4\aleph t}}^{\infty} \frac{\exp(-u) \left(t - \frac{\eta_0^2(k)}{4\aleph u} \right)^n}{u\sqrt{u}} du \end{aligned} \quad (14)$$

где $\xi_0(l) = \kappa_c + 2lрж$ $\eta_0(l) = 2\delta - \kappa_c + 2лрю$

В работе [8] выявлено, что в случае выполнения условия $2k_r w / ka \geq 300 \div 500$ (k_r , w – проницаемость и ширина трещины, соответственно; a – длина трещины), то определяемые решения должны быть с учетом бесконечной проводимости.

5. Анализ изменения гидродинамических параметров при различных заданных значениях эффективной длины горизонтального участка ствола скважины

Кривые восстановления давления (КВД), полученные при исследовании горизонтальных скважин месторождения У, интерпретировались по встроенной в ПО «Saphir» [25] аналитической модели, наиболее близко соответствующей реальным геолого-физическим условиям месторождения У: по модели бесконечного пласта с непроницаемой подошвой и постоянным давлением на кровле (газовой шапкой).

При интерпретации КВД строится диагностический график в билогарифмических координатах, по которому появляется возможность выделять различные режимы фильтрации в пласте [26]. Диагностический график представляет собой семейство двух кривых, построенных по двойной логарифмической шкале: кривой восстановления давления, построенной в координатах изменение давления ΔP от времени после остановки скважины и производной, взятой от ΔP по времени

суперпозиции, рассчитанном с учетом истории работы скважины до остановки). На диагностическом графике можно увидеть различные участки наклона, соответствующие разным режимам течения флюида.

Для горизонтальной скважины в начальный период времени, при отсутствии влияния ствола скважины, производная должна выйти на горизонтальный участок – ранний радиальный режим течения к стволу скважины в вертикальной плоскости. Далее производная должна начать расти до достижения наклона 0.5, характеризующего плоскопараллельное (линейное) течение по пласту к горизонтальному стволу скважины. После чего производная должна выйти на второй горизонтальный участок – поздний радиальный режим течения в горизонтальной плоскости к скважине как к точке.

К сожалению, практически на всех диагностических графиках кривая производной не имеет выраженных участков, однозначно определяющих искомые фильтрационные характеристики (участки радиального, билинейного и псевдорadiального притока). Начальные участки КВД искажаются влиянием продолжающего поступать в скважину свободного газа из газовой шапки, а на дальнейшее поведение давления оказывает большое влияние сама газовая шапка как граница постоянного давления.

Из-за влияния указанных выше факторов, компьютерная программа, осуществляющая нелинейную регрессию для подбора параметров модели, не может дать однозначного достоверного решения, предлагая бесчисленное множество зачастую абсурдных вариантов. С целью освобождения от

неопределенности было решено сократить при интерпретации количество искомых неизвестных величин, закрепив значение «эффективной длины горизонтального участка скважины», приняв его равным фактической длине горизонтального ствола по данным бурения. С таким допущением были обработаны все имеющиеся в распоряжении КВД по горизонтальным скважинам У. Возможные погрешности в оценке фильтрационных характеристик зависят от латеральной неоднородности пласта.

Так, после интерпретации КВД, зарегистрированной в скв. 1, был проведен численный эксперимент с помощью программы Saphir. КВД была переинтерпретирована при различных заданных значениях эффективной длины горизонтального участка ствола скважины в диапазоне от 200 м до фактической (1024 м).

В результате переинтерпретации были получены другие значения гидродинамических параметров, изменение которых представлено на рис. 5. Серым цветом показана зона несовместимости (от 0 до 350 м) фактической и теоретических кривых, что также видно по динамике значений среднеквадратичного отклонения между кривыми. Белым цветом обозначена зона приемлемого совмещения этих кривых, где значения среднеквадратичного отклонения низкие и практически не отличаются. Наилучшее совмещение фактической и теоретических кривых (среднеквадратичное отклонение минимально – 0.144) было получено при длине горизонтального участка ствола скважины 600 м.

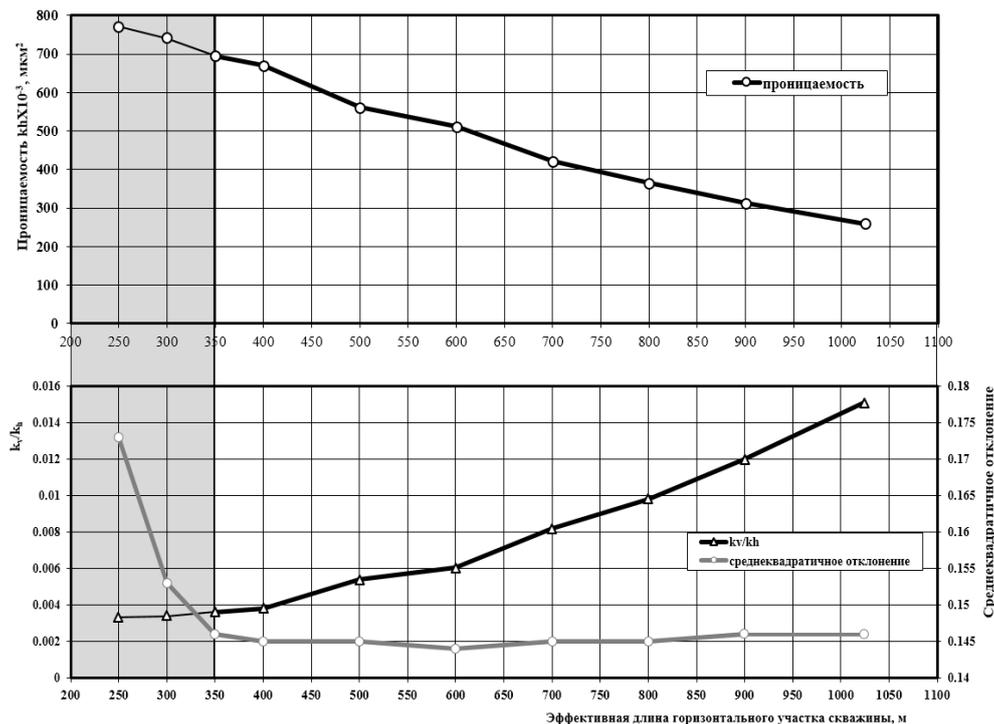


Рис. 5. Изменение расчетных гидродинамических параметров при различных заданных значениях эффективной длины горизонтального участка ствола скважины

6. Заключение

В работе изучены характеристики эффективности работы горизонтальных скважин с помощью гидродинамических исследований методом установившихся отборов, позволяющие оценить их преимущества по сравнению с вертикальными скважинами. При непрерывном росте эффективности горизонтальных скважин по сравнению с вертикальными при увеличении длины горизонтального ствола имеется некоторое количество точек с меньшей или соизмеримой эффективностью. Это объясняется увеличением фильтрационного сопротивления при искривлении линий тока вблизи ствола скважины и влиянием на этот эффект анизотропии пласта по проницаемости.

Эти графики могут быть использованы для прогноза продуктивности горизонтальных скважин на любом месторождении.

Также в работе рассмотрены особенности интерпретации гидродинамических исследований методом восстановления давления, связанные с отсутствием достоверной информации об эффективной длине горизонтального ствола скважины.

Работа выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН на 2019 год по программе Президиума РАН № 2 (I27), тема (проект) № 0065-2019-0111.

Analysis of horizontal wells application efficiency using wells testing

P.V. Kryganov, I.V. Afanaskin, S.G. Volpin

Abstract: The paper studies the characteristics of horizontal wells performance with the help of Wells Testing in applying of pressure transient test which allows to evaluate their advantages compared to vertical wells. Analysis of Wells Testing in applying of pressure build up interpretation features are performed associated with reliable information deficiency about effective length of horizontal shaft.

Keywords: horizontal well, well test

Литература

1. Л.Г. Кульпин, Ю.А. Мясников. Гидродинамические методы исследования нефтегазоводоносных пластов. М., Недра, 1974.
2. R.V. Smith Practical Natural Gas Engineering. PennWell Publishing Co, Tulsa, OK, 1983.
3. Т.А. Деева, М.Р. Камарудинов, Т.Е. Кулагина и др. Гидродинамические исследования скважин: анализ и интерпретация данных. Томск, ЦППС НД ТПУ, 2009.
4. С.Н. Бузинов, И.Д. Умрихин. Исследования пластов и скважин при упругом режиме фильтрации. М., Недра, 1964.
5. В.С. Евченко, Н.П. Захарченко, Я.М. Каган и др. Разработка нефтяных месторождений наклонно-направленными скважинами. М., Недра, 1986.
6. Р.Г. Шагиев. Исследование скважин по КВД. М., Наука, 1998.
7. R. Aguilera, J.S. Artindale, G.M. Cordell, M.C. Ng., G.W. Nicholl, G.A. Runions: Horizontal wells, Gulf Publishing, Houston, 1991.
8. H. Cinco, F. Samaniego, N. Domingues Transient pressure behavior for a well with a finite-conductivity vertical fracture. // Soc. of Petrol. Eng. J. – 1978. – v.18. – Aug., № 4. – p. 253 – 264.
9. С.А. Ehlig-Economides Use of pressure derivative for diagnosing pressure-transient behavior // JPT, 1988, Oct., pp. 1280 - 1282.
10. C.J. Gilbert Pressure transient analysis in horizontal wells in some sole pit area fields U.K. // SPE FE. 1996. May. pp. 101 - 108.
11. P.A. Goode, R. K. M. Thambynaygam. Pressure drawdown and buildup analysis of horizontal wells in anisotropic media // SPE FE. – 1987. – Dec. – p. 683 - 699.
12. A.C. Gringarten, H.J. Ramey. Unsteady – state pressure distributions created by a well with a single horizontal fracture, partial penetration, or restricted entry. // Soc. of Petrol. Eng. J. – 1974. – v.14. – Aug., № 4. – p. 413 – 426.
13. A.C. Gringarten, H.J. Ramey, R. Raghavan Unsteady – state pressure distributions created by a well with a single infinite-conductivity vertical fracture, partial penetration, or restricted entry. // Soc. of Petrol. Eng. J. – 1974. – v.14. – Aug., № 4. – p. 347 – 360.
14. F.J. Kuchuk et al. Pressure transient behavior horizontal wells with and without gas cap or aquifer // SPEFE. 1991. Mar. pp. 86 - 94.
15. F.J. Kuchuk Well testing and interpretation for horizontal wells // JPT. – 1995. – Jan. – pp. 36 - 41.
16. F.J. Kuchuk, P.A. Goode, B.W. Brice et al. Pressure transient analysis and inflow performance for horizontal wells // JPT. 1990. Aug. pp. 974 - 1031.
17. W.J. Lee Characterizing formations with well tests // SPE. – S. A. Holditch and Associates, Inc. – 1997.
18. G.J. Lichtenberger Data acquisition and interpretation of horizontal well pressure transient tests // JPT, 1994. Febr. pp.157 - 162.
19. M. Methnani Simulator helps analyze horizontal- well tests // OGJ, 1999, June, pp. 54 – 57.
20. A.S. Odeh, D.K. Babu Transient flow behavior of horizontal wells: Pressure drawdown and buildup analysis // SPE FE. 1990. Mar. pp. 7 - 15.
21. P.C. Shah, D.K. Gupta, L. Singh, B.G. Deruyck. Field application of a method for interpretation of horizontal-well transient tests // SPE FE, 1994, March, pp. 23-31.
22. J.P. Spivey, W.J. Lee Fundamentals of type curve analysis. // Petroleum Engineer, 1997, Sept., pp. 63 - 70.
23. G. Stewart, Recent Developments in well Test analysis // Petroleum Engineer, 1997, Aug., pp. 47 - 56.
24. G. Stewart, P. Westaway Future developments in well test analysis: horizontal well test interpretation techniques // Petroleum Engineer, 1997, Nov., pp. 77 - 80.
25. У. Оливье, Д. Витура, О. Фьяре. Анализ динамических потоков. Теория и практика интерпретации данных ГДИС и анализа добычи, а также использование данных стационарных глубинных манометров. Карра Engineering, 2009.
26. П.В. Крыганов, С.Г. Вольпин, И.В. Афанаскин, Ю.М. Штейнберг, П.В. Ялов. Особенности математических моделей для интерпретации гидродинамических исследований горизонтальных скважин. Сборник материалов Международной конференции «Математика и информационные технологии в нефтегазовом комплексе», посвященная дню рождения великого русского математика академика П.Л. Чебышёва, Обнинск, 2019

Разработка параллельного генератора неструктурированных сеток для моделей нефтегазовых месторождений с адаптацией к объектам геологических моделей

М.Л. Сидоров¹, В.А. Пронин², И.В. Афанаскин³, А.В. Королев⁴, П.В. Ялов⁵

¹ФГУП «РФЯЦ-ВНИИЭФ», Саров, Россия, gridgrind@yandex.ru;

²ФГУП «РФЯЦ-ВНИИЭФ», Саров, Россия, virtualvi@mail.ru;

³ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ivan@afanaskin.ru;

⁴ФГУ ФНЦ НИИСИ РАН, Москва, Россия, alexandre.korolev@mail.ru;

⁵ФГУ ФНЦ НИИСИ РАН, Москва, Россия, petryalov@gmail.com

Аннотация: Ранее был представлен метод ФСМ для параллельного построения трехмерных неструктурированных сеток, используемых при дискретизации сложных геологических структур при численном моделировании нефтегазовых месторождений. Метод позволяет строить планарные треугольные и многоугольные сетки, проводить экструзию по оси аппликата, строить сетку по методу угловой точки и обобщенному методу угловой точки. Приведенный метод обладает высокой скоростью построения и хорошо зарекомендовал себя при моделировании месторождений. В данной статье приводится усовершенствование метода ФСМ для построения трехмерной сетки с адаптацией к объектам геологических моделей.

Ключевые слова: неструктурированная сетка, экструзия, адаптация сетки, параллельные вычисления, формат Corner-Point

1. Введение

Ранее в работе [1] был представлен метод ФСМ для параллельного построения трехмерной неструктурированной сетки, используемой при дискретизации моделей нефтегазовых месторождений. Данный метод позволяет строить достаточно произвольные сетки в плане, а затем проводить экструзию по оси аппликата для получения финальной трехмерной сетки. В силу того, что построение сетки производится в модели распределенной памяти, финальная сетка представляется в виде наборов сеточных областей с перекрытием на один (или несколько слоев) на границах параллельных областей (параобластей). Общее количество ячеек финальной сетки может доходить до нескольких миллиардов, что позволяет проводить дискретизацию моделей с очень большой территорией. Данный метод обладает высокой скоростью построения, и он хорошо зарекомендовал себя при моделировании месторождений.

При создании параллельного сеточного генератора авторами был проведен анализ методов, используемых в мировой практике [2]-[5], и на их основе синтезирован наиболее пригодный для распараллеливания и обобщения на трехмерный случай алгоритм.

В геологических моделях реальных месторождений достаточно часто могут присутствовать объекты, которые требуют адаптации сетки при дискретизации. В качестве таких объектов могут выступать: скважины сложной структуры и траектории, слоистая структура отложений, поверхности разломов, объемные источники, и т.д. Адаптивность сетки к объектам геологических моделей позволяет получать численное решение более точным в интересующих областях.

В данной статье приводится усовершенствование метода ФСМ для построения трехмерной сетки с адаптацией к объектам геологических моделей. Основное внимание уделено алгоритмам адаптации сетки к вертикальным и горизонтальным скважинам, поверхностям разломов и неоднородностям пластов.

2. Метод построения сетки в области на основе регулярного заполнения узлами (метод RNG)

Распространенным методом построения сеток в гидрогеологических и нефтяных задачах является метод глобального шаблона. Его суть состоит в следующем: на область моделирования накладывается сетка определенного вида, а затем проводится анализ

сеточных элементов. Элементы, попавшие на границу области, либо рассекаются граничной линией, либо удаляются (в этом случае слой граничных ячеек в незанятом приграничном пространстве достраивается). Данный метод в двухэтапной формулировке был реализован авторами в рамках развития сеточного генератора. Результат работы данного метода показан на рис. 1 и 2.

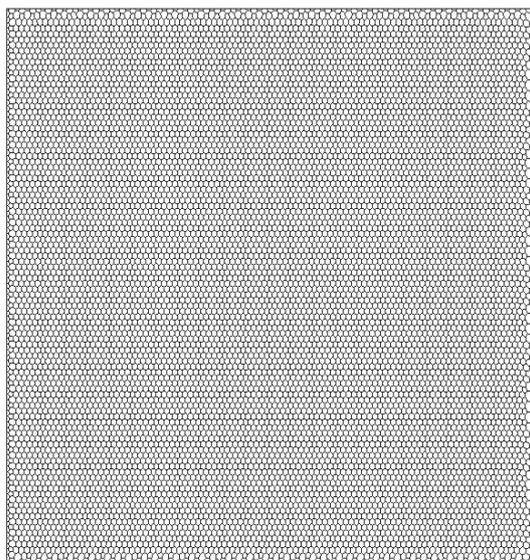


Рис. 1. Сетка, полученная с помощью метода RNG

Вносимый шаблон воздействует на функцию распределения характерного размера: в случае если значение характерного размера в точке меньше максимального размера в шаблоне, его значение становится равным максимальному размеру.

Характерный размер может быть задан пользователем, а может задаваться как минимальная или максимальная длина ребра. Построение точек шаблона может быть реализовано различными методами:

- Матричная расстановка точек для получения сетки внутри области из правильных прямоугольников;
- Матричная расстановка со смещением, чтобы получить сетку из треугольников, близким к правильным;
- Матричная расстановка предыдущих двух видов с операцией вращения на заданный угол и смещения в заданную координату.

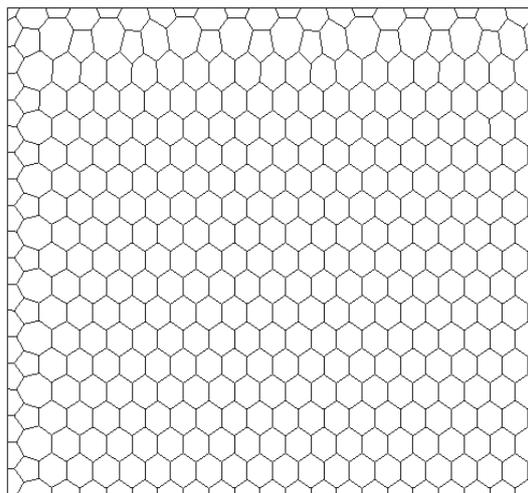


Рис. 2. Фрагмент сетки

3. Условия на функцию характерного размера ячеек

Для того чтобы учесть различные объекты в сетке (с целью улучшения аппроксимации, задания внутренних граничных условий) и воспользоваться существующим генератором, необходимо чтобы эти объекты влияли на функцию определения характерного размера ячеек. В этом случае на этапах расстановки точек и триангуляции образуются плавные переходы характерного размера. Таким образом, получаем дополнительные ограничения в функции характерного размера. Рассмотрим различные вариации ограничений.

3.1 Точки как источники

Пользователь задает точки и радиус в области моделирования, внутри которого функция характерного размера имеет некоторое постоянное значение. В итоге на этапе формирования сетки самих точек – центров окружностей – не появятся, однако они окажут влияние на характерный размер вблизи себя. С помощью данного способа удобно проводить адаптацию на вертикальные скважины (рис. 3-5).

Стоит отметить, что форму области влияния точечного источника можно изменять. Например, вокруг точечного источника можно задать квадратную форму области его влияния.

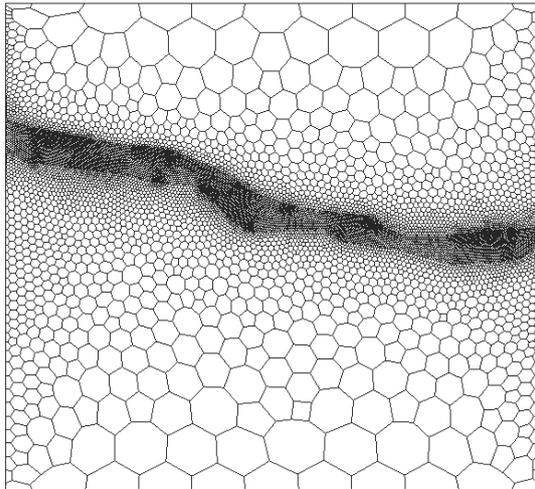


Рис. 3. Сетка со стандартной функцией распределения

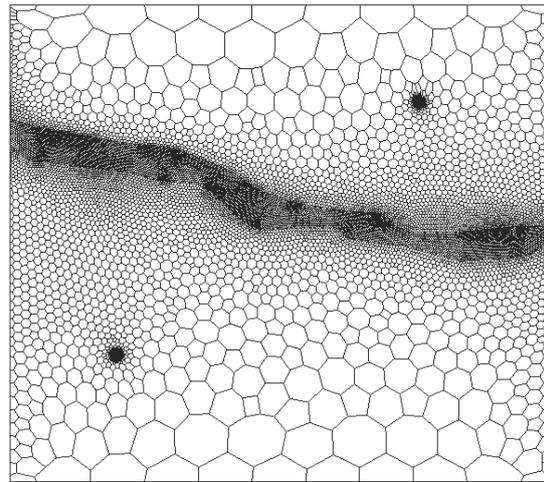


Рис. 4. Сетка с функцией распределения, испытывающей воздействие двух точечных источников

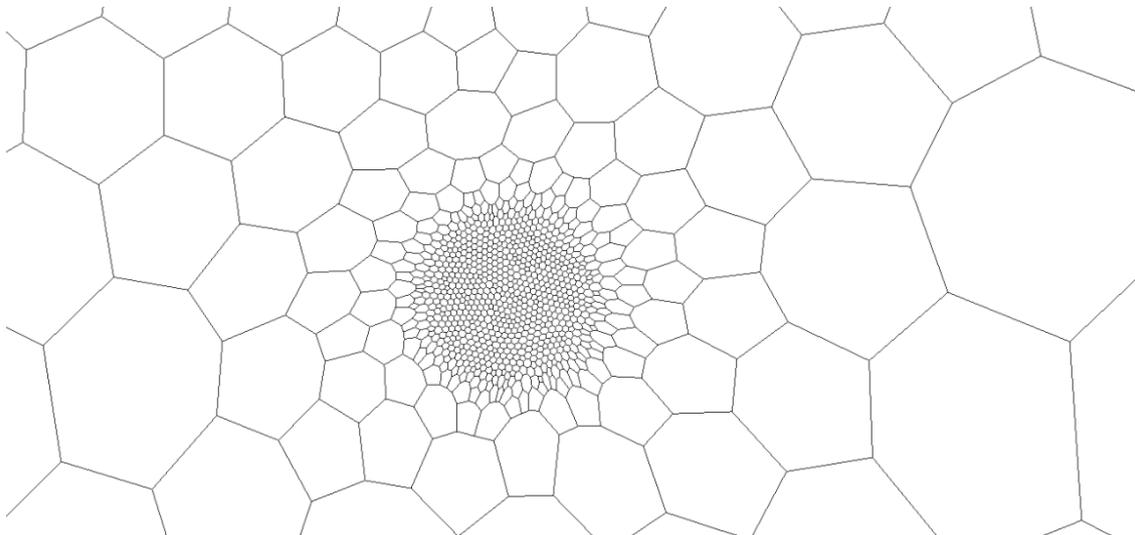


Рис. 5. Фрагмент сетки вблизи точечного источника

3.2. Точки как ограничения

Если пользователю необходимо, чтобы некоторые точки (как объекты сетки) обязательно присутствовали в финальной сетке, необходимо задать соответствующие ограничения. На этапе расстановки точек необходимо задавать условие совместности с существующими ребрами и точками. На рис. 6 приведен пример сетки без дополнительных ограничений, а на рис. 7 – с ограничением в виде двух точек (они присутствуют в финальной сетке и оказывают влияние на вид сетки в их некоторой окрестности).

Добавление избранных точек в сетку необходимо, например, для учета вертикальных скважин.

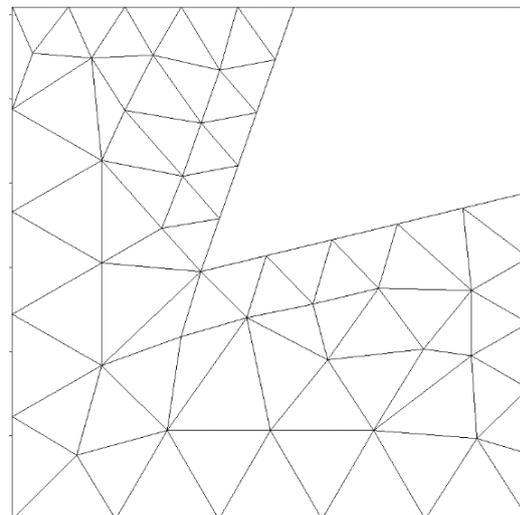


Рис. 6. Триангуляция без дополнительных ограничений

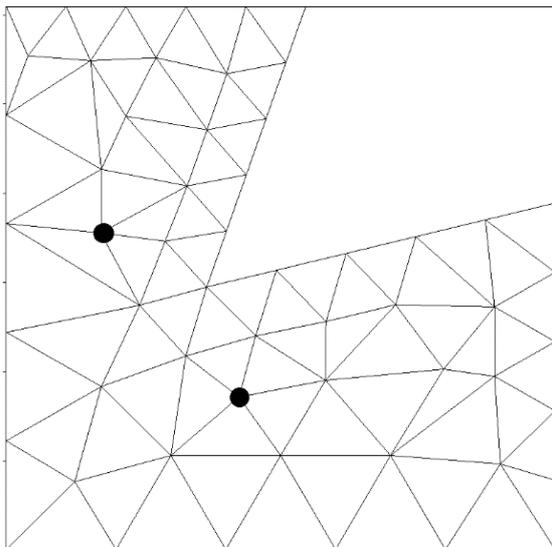


Рис. 7. Триангуляция с дополнительным ограничением в виде 2-х точек

3.3. Точки как ограничения и источники

Этот тип ограничения является объединением двух предыдущих ограничений и позволяет точкам присутствовать в сетке, а также задавать определенный характерный размер ребер вокруг себя.

3.4. Ребра как источники

Пользователь может определить множество ребер-источников, которые будут оказывать

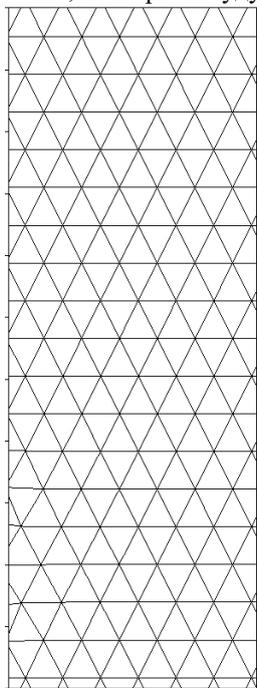


Рис. 8. Пример сетки без ограничений

влияние на функцию характерного размера в содержащих их точках по следующей формуле:

$$\begin{cases} D_i = \frac{1}{2}(L_i + L_{i+1}), i \neq 1, N, \\ D_1 = L_1, D_{N+1} = L_N, \end{cases}$$

где L_i, L_{i+1} – длины ребер, содержащих точку i .

Ребра-источники будут влиять на густоту или разреженность итоговой сетки вокруг себя. Задание линий (как набора ребер), к которым сетка сгущается или разрежается, дает некоторое удобство в контроле над характерным размером ребер: не нужно задавать конкретный размер в каждой точке, достаточно приписать характерный размер на всей линии.

3.5. Ребра как ограничения

В случае необходимости можно задать ребра, которые должны обязательно присутствовать в итоговой сетке. Заданные ребра не будут являться источниками и подвергнутся воздействию функции распределения, которая приведет к разбиению заданного ребра. На рис. 8, 9 приведен пример сетки без ограничений и с ограничением в виде ребра соответственно.

Вставка в сетку избранных ребер необходима, например, для задания горизонтальных скважин, разломов.

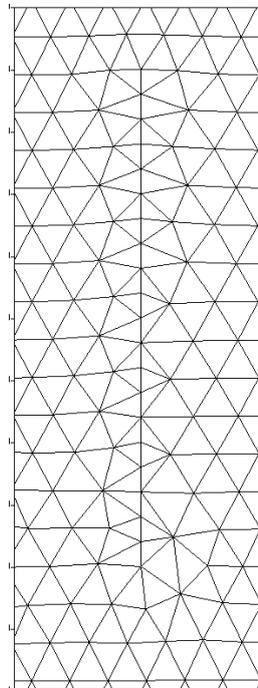


Рис. 9. Пример сетки с ограничением в виде ребра

3.6. Ребра как ограничения и источники

Этот тип ограничения является объединением двух предыдущих ограничений и означает, что ребра могут присутствовать в сетке и задавать определенный характерный размер ребер вокруг себя.

3.7. Локальные шаблоны

Локальный шаблон есть объединение множества ребер и точек, построенных по определенному правилу. Шаблон должен быть корректным, т.е. в нем не должно быть самопересекающихся ребер. Кроме того, различные локальные шаблоны по геометрическому расположению должны быть заданы таким образом, чтобы они не имели пересечения между собой. В качестве локального шаблона может выступать, например, радиальная сетка для вертикальной скважины или регулярная анизотропная для горизонтальных скважин.

Продемонстрируем работу сеточного генератора на основе локальных шаблонов. Для этого зададим квадрат, вершины которого имеют следующие координаты соответственно: $(0,0)$, $(3000,0)$, $(3000,3000)$ и $(0,3000)$. В точках $(500,500)$, $(2500,500)$, $(2500,2500)$, $(500,2500)$ и $(1500,1500)$ заданы вертикальные скважины. Чтобы учесть их в итоговой сетке зададим в этих точках радиальные шаблоны с параметрами: радиус влияния 150, число лучей 15, число разбиений по радиусу 6 (рис. 10). На рис. 11 построена преимущественно четырехугольная сетка путем объединения треугольников сетки на рис. 10.

На рис. 12 показана двойственная сетка (к сетке на рис. 11), а на рис. 13 показан фрагмент двойственной сетки вблизи скважины.

На рис. 14-16 показано применение генератора и возможностей по адаптации к скважинам на примере некоторой реальной задачи с 19 скважинами.

3.8. Дробление в избранных пластах

Для задания подробной детализации в избранных слоях реализовано дробление с учетом номера слоя (рис. 17-18). При этом степень дробления между слоями может различаться в 2, 3 и более раз (рис. 18). Дробимая ячейка делится только в двух направлениях, вертикальные ребра остаются нетронутыми, соответственно число слоев остается тем же.

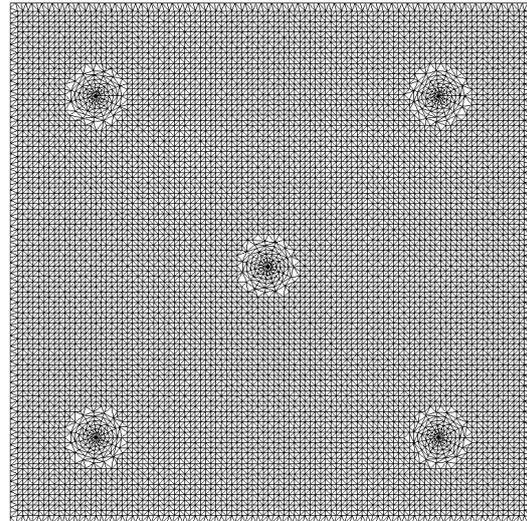


Рис. 10. Триангуляция области с ограничением в виде пяти локальных шаблонов

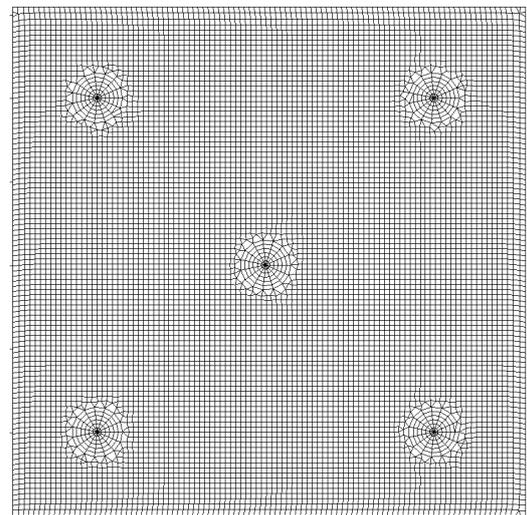


Рис. 11. Вид сетки, полученной путем объединения треугольников в четырехугольники

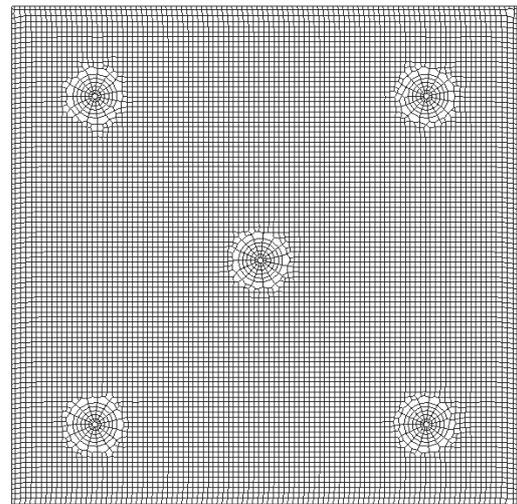


Рис. 12. Двойственная сетка

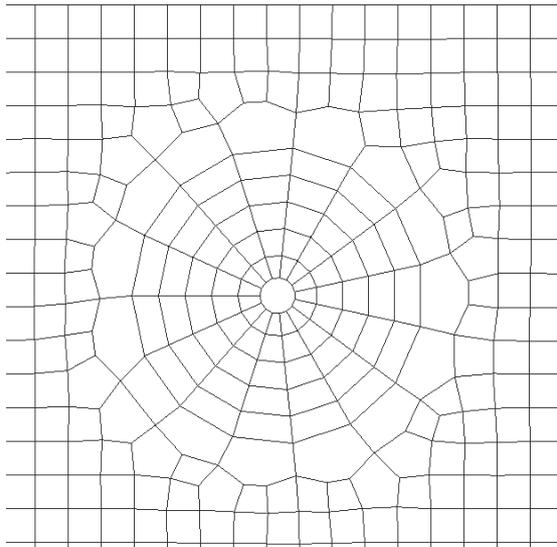


Рис. 13. Фрагмент двойственной сетки вблизи скважины

На рис. 19 приведен пример декомпозиции области на 6 параобластей, а на рис. 20 показан пример одной из параобластей.

4. Метод построения трехмерной сетки формата Corner-Point в параллельном режиме на основе сторонних начальных данных

Для возможности моделирования задач подготовленных в сторонних программных средствах в формате Corner-Point был реализован построитель сеток такого формата, но с учетом специфики работы в параллельном режиме.

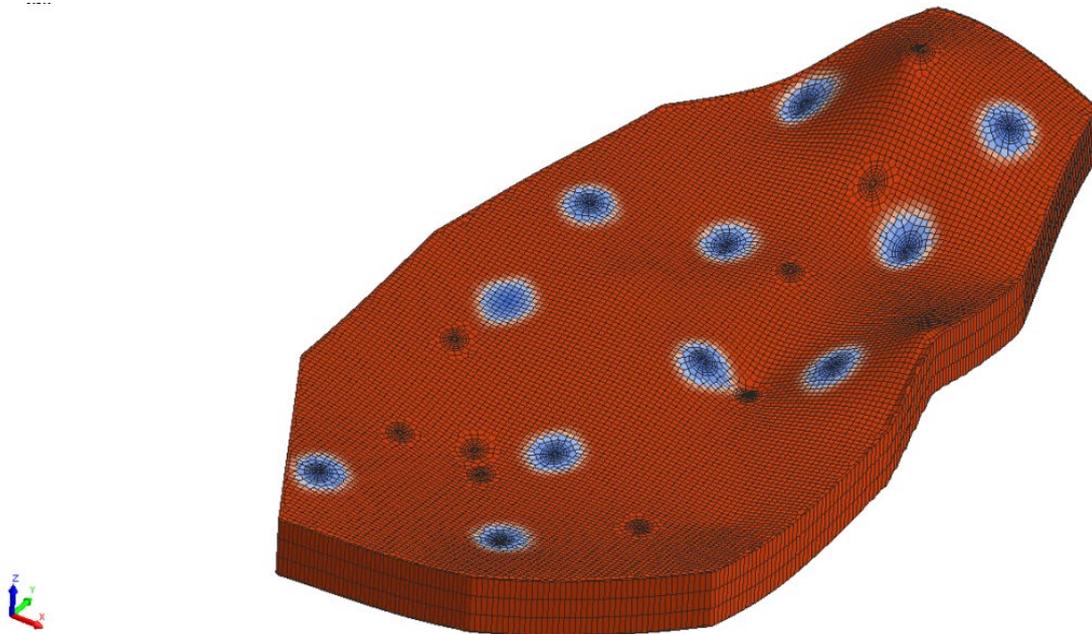


Рис. 14. Вид сетки в задаче с 19 вертикальными скважинами

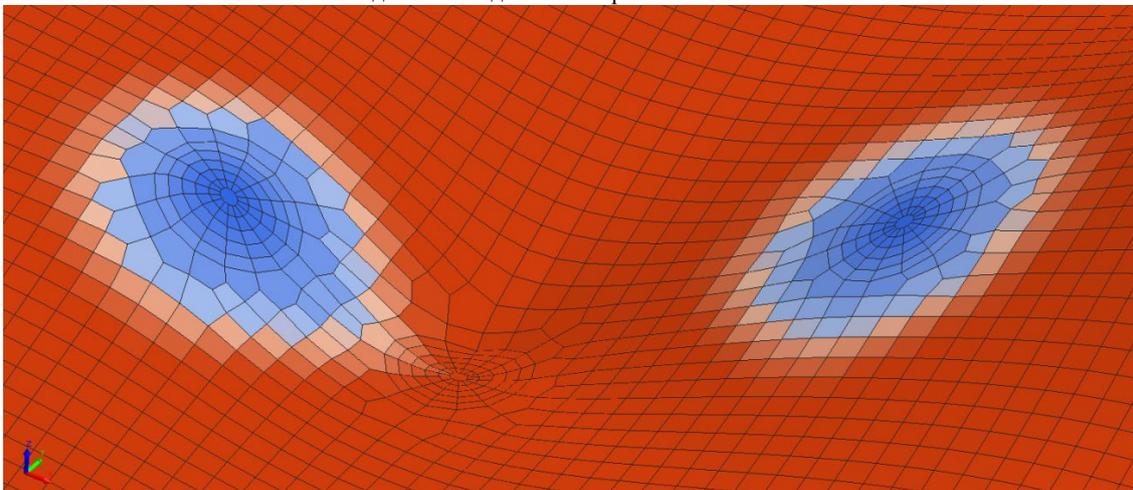


Рис. 15. Фрагмент сетки для задачи с 19 вертикальными скважинами

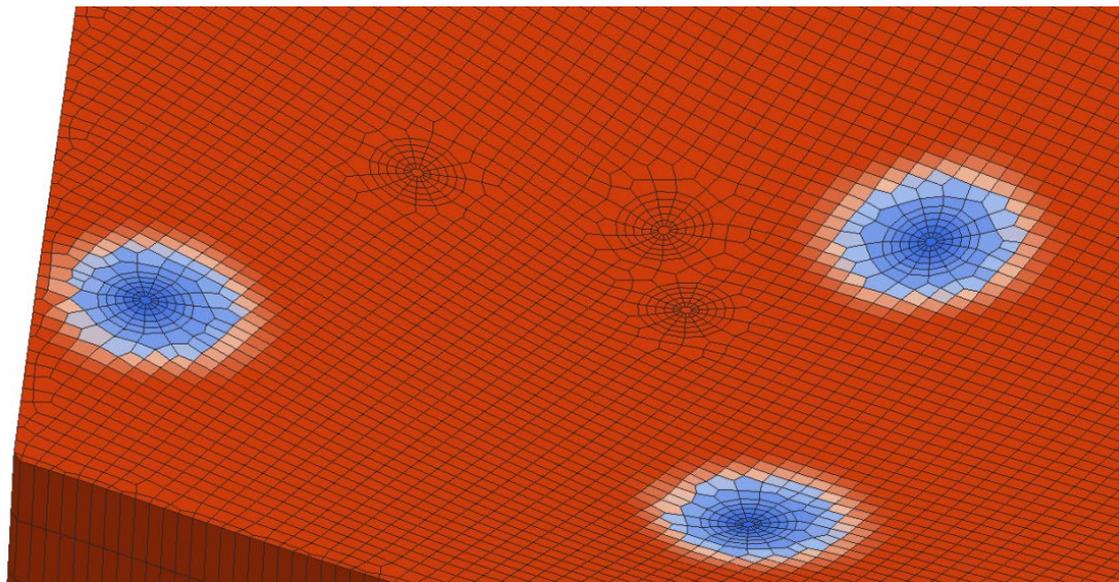


Рис. 16. 3D фрагмент сетки для задачи с 19 вертикальными скважинами

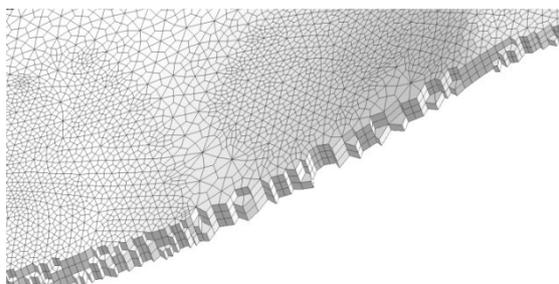


Рис. 17. Пример сетки с дроблением во всех пластах

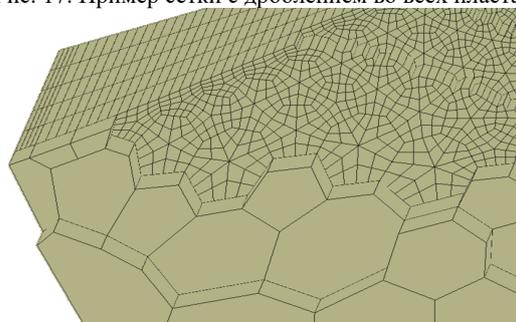


Рис. 18. Пример сетки с дроблением в некоторых пластах

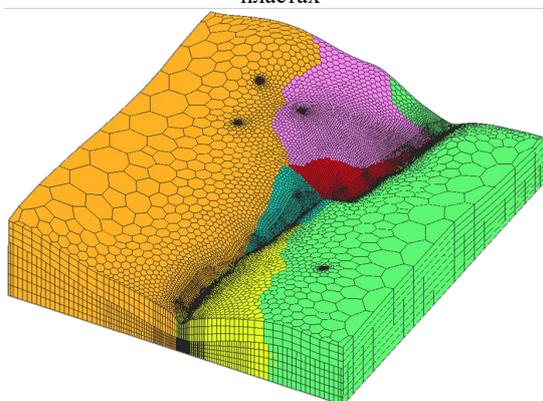


Рис. 19. Пример построения сетки с декомпозицией на 6 параобластей

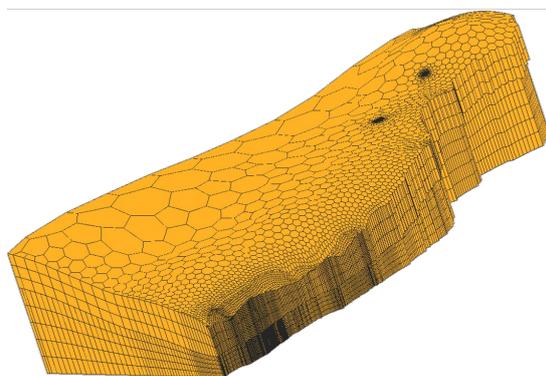


Рис. 20. Пример одной из параобластей сетки

Алгоритм программы построения такой сетки выглядит следующим образом:

1. По заданным габаритам и числу строк-столбцов ячеек (NX , NY) на каждом ядре строится двумерная регулярная сетка.
2. На каждом ядре производится полное чтение массива ACTNUM (активных или иначе счетных ячеек).
3. Производится первичная декомпозиция регулярной сетки на заданное число параобластей (равно числу ядер в задаче). Весом двумерной ячейки сетки выступает число активных ячеек под ней.
4. Каждое ядро выполняет сужение множества ACTNUM – получение множества глобальных номеров только счетных ячеек параобласти и множества глобальных номеров несчетных ячеек (на границе с частями сеток на других ядрах). После этого память, занятая массивом ACTNUM, освобождается.
5. На основании суженного множества номеров из п.3 производится чтение только

- тех частей массивов с координатами узлов COORD и глубинами узлов ZCORN, которые принадлежат будущей сетке на данном ядре.
6. На каждом ядре из полной двумерной регулярной сетки выделяется только та часть, которая после декомпозиции имела признак соответствующей параобласти.
 7. Производится выделение памяти под сеточные структуры (узлы, грани, ячейки) для будущего заполнения.
 8. В цикле по всем граням двумерной сетки производится построение колонн под ними на основании необходимых данных из массива COORD и создание ячеек на основании данных из массива ZCORN.
 9. Производится сшивка между ячейками соседних колонн (после этого возможно образование соседей, у которых К индекс отличается более чем на единицу).
- После работы алгоритма производятся дополнительные действия по заполнению необходимой информации по ячейкам сетки.
- В качестве примера работы алгоритма продемонстрируем реальную модель нефтяного месторождения. В модели 110 млн. ячеек, из них 12 млн. активных. Для построения сетки было использовано 768 ядер. Время построения порядка 4 минут. На рис. 21 разным цветом представлены 768 параобластей. Границы параобластей показаны на рис. 22, 23.

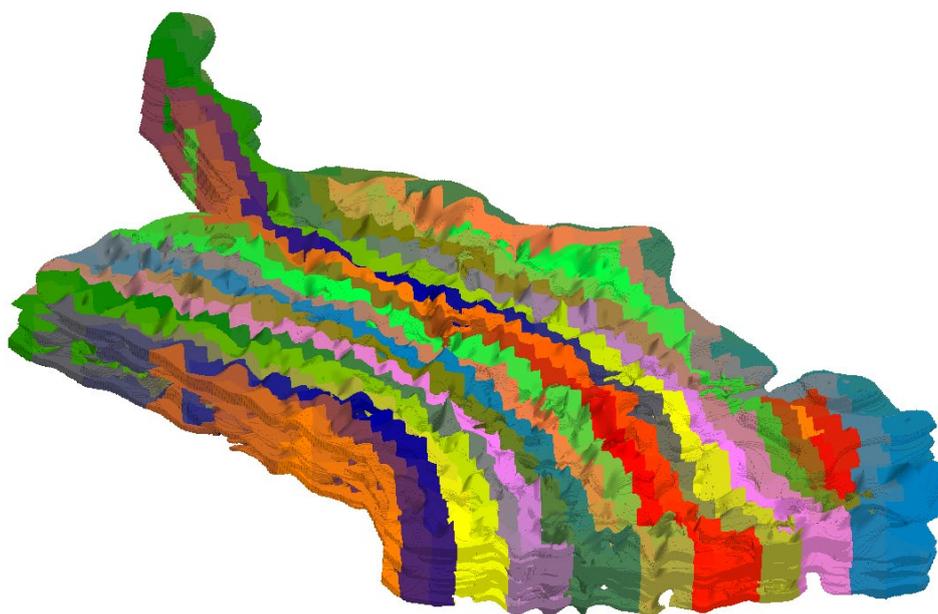


Рис. 21. Декомпозиция задачи на 768 параобластей

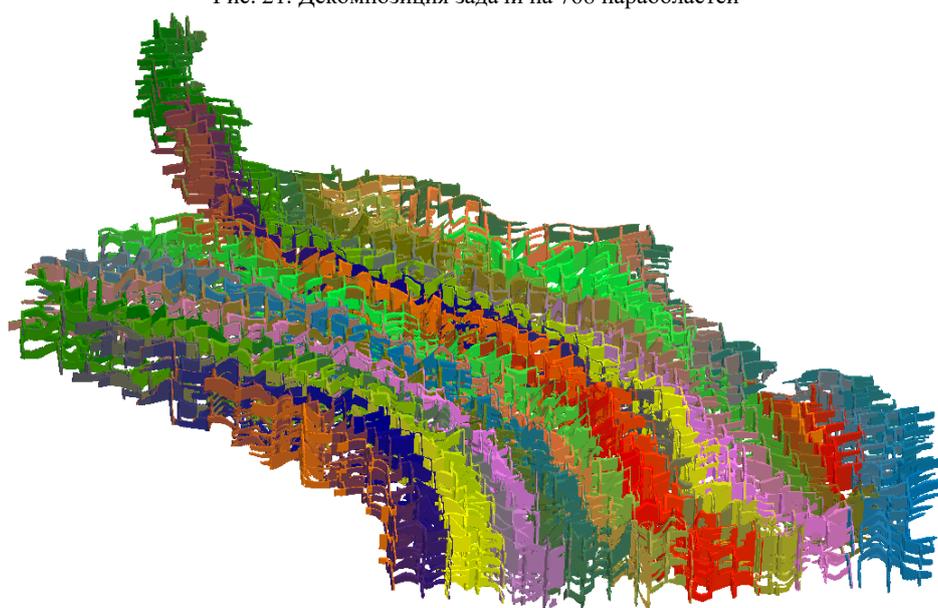


Рис. 22. Границы между параобластями

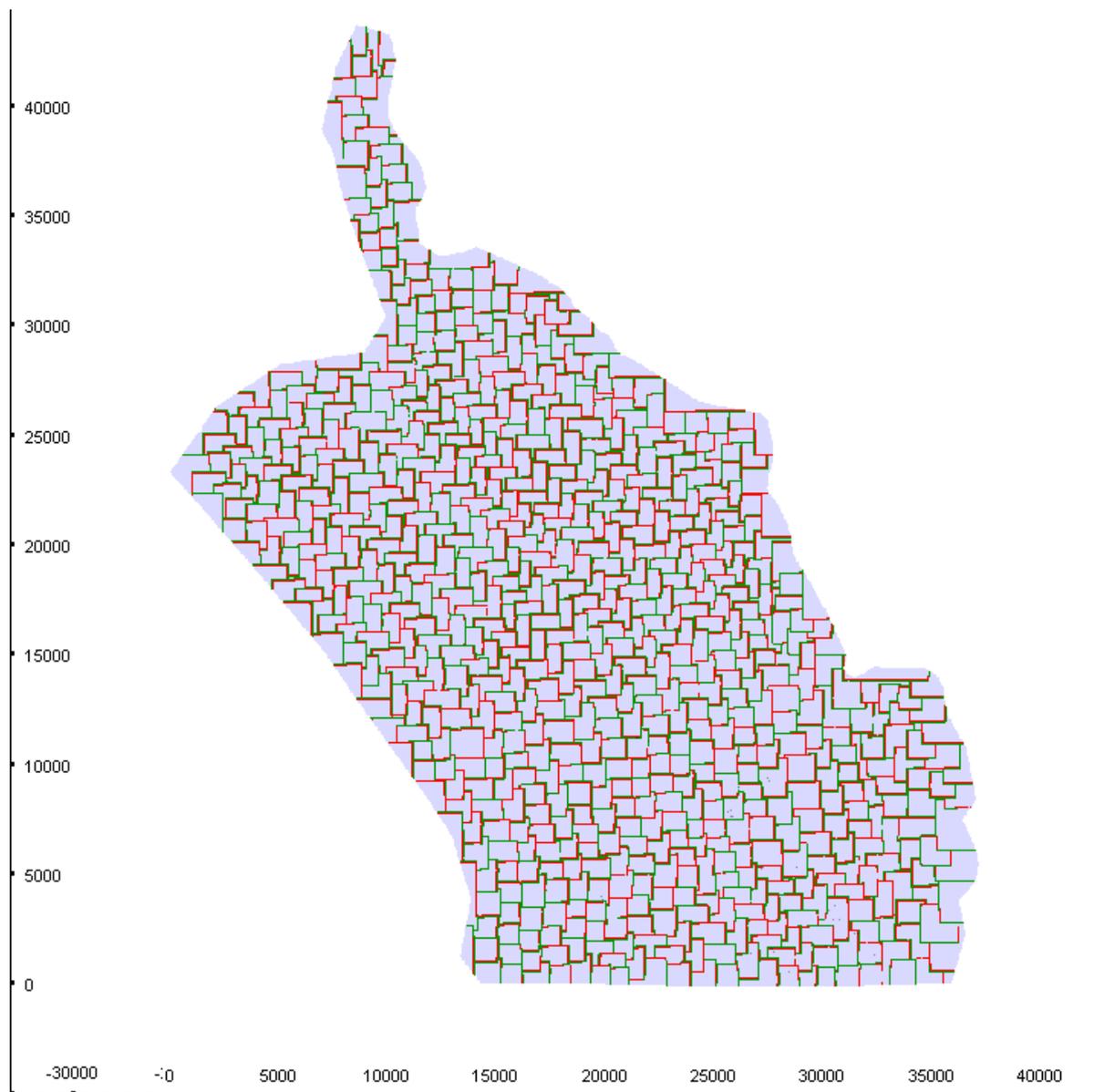


Рис. 23. Границы между параобластями (вид сверху)

5. Заключение

В статье приведены новые возможности параллельного сеточного генератора, использующиеся для решения задач нефтегазодобычи. Приведены способы адаптации сеточных моделей к особенностям геологических моделей. Показаны возможности управления характерным размером сетки, вставки дополнительных наборов узлов и ребер в сетку. Работоспособность разработанного генератора демонстрируется на основе шаблонов, позволяющих строить квазирегулярную сетку внутри области со сшивками вблизи заданных особенностей. Также приведен метод построения трехмерной сетки формата Corner-

Point в параллельном режиме на основе сторонних начальных данных. Работоспособность алгоритма показана на примере реального нефтяного месторождения.

Работа выполнена при поддержке гранта РФФИ 18-07-00671 А.

Unstructured Grid Parallel Generator for Oil and Gas Field Models with Adaptation to Geological Model Objects

M.L. Sidorov, V.A. Pronin, I.V. Afanaskin, A.V. Korolev, P.V. Yalov

Abstract: In previous paper, Frontal Sphere Method was introduced and applied to 3D mesh generation. Generated 3D mesh used for discretization of complex geological formations in numerical modelling of oil-gas reservoirs. In this paper, we present modified frontal sphere method, applied to 3D mesh generation with adaptation to geological model features. Presented method can be used for planar triangular and polygonal grid generation; it can generate 2.5D mesh by extrusion of surface grid in Z dimension; build Corner-Point grid and Generalized Corner-Point grid. All mesh generation stages are performing in parallel mode. Method has high grid generation speed and can be used in reservoir modelling.

Key words: unstructured grid, extrusion, grid adaptation, parallel computations, Corner-Point grid

Литература

1. М.Л. Сидоров, В.А. Пронин, В.Ю. Кузнецов, И.В. Афанаскин, А.В. Королев, П.В. Ялов. Разработка параллельного генератора неструктурированных сеток для моделей нефтегазовых месторождений и тестирование однопроцессорной версии симулятора двухфазной фильтрации нефти и воды, «Нефтепромышленное дело», 2019. № 12.
2. Y. Ito, A.M. Shih, B.K. Soni. Reliable Isotropic Tetrahedral Mesh Generation based on an Advancing front method. Proceedings of the 13th International meshing roundtable, Williamsburg, 2004, 95-105.
3. K. Shimada, A. Yamakawa., T. Itoh. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal. Proceedings of the 6th International Meshing roundtable, 1997, 375-390.
4. X.Y. Li, S.H. Teng, A. Ungor. Biting spheres in 3d. Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, CA, U.S.A, 1999, 85-95.
5. G.L. Miller, D. Talmor, S.H. Teng, N. Walkington. A Delaunay Based Numerical Method for Three Dimensions: generation, formulation and partition. Proceedings of the 27th Annu. ACM Sympos. Theor. Comput., 1995, 683-692.

Возможность выявления дизъюнктивных нарушений на нефтяном месторождении по характеру обводнения скважин и данным трассерных исследований

В.А. Юдин¹, С.Г. Вольпин², И.В. Афанаскин³, Н.П. Ефимова⁴

¹ФГУ ФНЦ НИИСИ РАН, Москва, Россия, yudinval@yandex.ru;

²ФГУ ФНЦ НИИСИ РАН, Москва, Россия, sergvolpin@gmail.com;

³ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ivan@afanaskin.ru;

⁴ФГУ ФНЦ НИИСИ РАН, Москва, Россия, efnatka@gmail.com

Аннотация. Изучена возможность выявления высокопроницаемого нарушения по характеру обводнения скважин и данным трассерного метода. Исследование проведено методом численного моделирования для ячейки пятиточечной системы разработки, работающей в рамках опытно-промышленной эксплуатации в отсутствие окружения (остальные скважины пласта удалены на значительное расстояние и существенного влияния не оказывают). Показано, что: 1. наличие высокопроницаемого нарушения может приводить к оттоку нагнетаемой воды за пределы участка, что вызывает резкое замедление продвижения воды к скважинам, отделённым нарушением от нагнетательной скважины; 2. при некоторых параметрах нарушения по кривым обводнения скважин от времени наличие нарушения может быть выявлено; 3. трассерные исследования существенно дополняют данные по характеру обводнения, особенно, если проводятся неоднократно в процессе заводнения с разными типами трассеров.

Ключевые слова: дизъюнктивные нарушения, трассерные исследования, обводнение скважин.

1. Введение

В настоящее время во многих публикациях сообщается о наличии на нефтяных месторождениях дизъюнктивных нарушений различного ранга. Причём, не только на сложенных карбонатными породами, для которых этот факт давно установлен, но и на терригенных залежах, в частности, на большинстве месторождений Западной Сибири, особенно в юрских отложениях [1, стр. 8–24].

Известно также, что зона локализации нарушений имеет достаточно сложную фильтрационную структуру [1, стр. 34–58]. Внутренняя зона нарушений содержит: различные виды глин; брекчию; катаклазиты, и т.п., формируется как за счёт механических процессов, так и при диагенетических преобразованиях пород. Она может иметь самую различную величину проницаемости – от нуля до нескольких Дарси. Во внешней же зоне нарушения, т.е. в зоне пласта, примыкающей к осевой зоне дизъюнктивного нарушения, формируется трещиноватость разной ориентации, с последовательно убывающей плотностью вторичных трещин в направлении, ортогональном плоскости сместителя. Таким образом, нарушение

представляет собой своеобразную горную породу, имеющую полоскообразный характер, сложную фильтрационную структуру, и гидродинамически связанную с окружающим пластом.

Наличие нарушений заметно сказывается на эффективности разработки, в частности, на величине текущего и конечного коэффициентов нефтеотдачи [1], поэтому их обнаружение имеет большое значение для оптимизации разработки с целью повышения её эффективности и улучшения экономических показателей.

В настоящей работе изучена возможность выявления нарушения по данным анализа характера обводнения скважин и данным трассерного метода. При этом наибольший интерес представляет выявление высокопроницаемого нарушения, поскольку непроницаемые «стенки» достаточно хорошо выявляются по данным гидродинамических исследований скважин (ГДИ) [2].

2. Численная модель

Исследование проведено методом численного моделирования. Для анализа выбрана ячейка пятиточечной площадной системы разработки, от которой все остальные

скважины удалены на значительное расстояние. Также рассмотрена ситуация, когда имеются две добывающих скважины, между которыми расположена одна нагнетательная. Участок пересекает высокопроводящее нарушение, отсекающее одну добывающую скважину от остальных и нагнетательной (обозначенной как I), рис 1. В случае двух добывающих скважин нарушение ориентировано перпендикулярно линии расположения скважин и локализовано между нагнетательной скважиной и одной из добывающих.

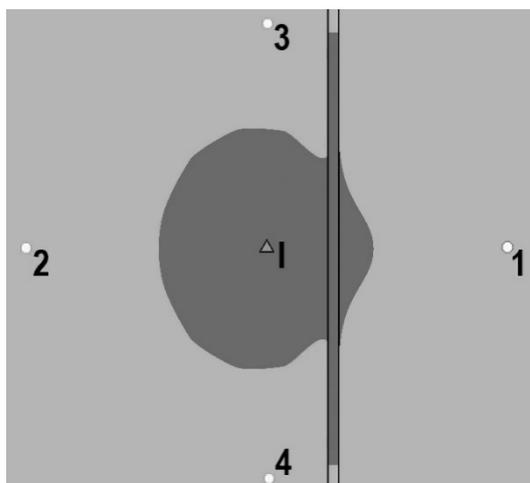


Рис. 1. Распространение воды (темная зона) по площади пласта через 10 месяцев после начала разработки в однородном пласте с проницаемостью 39 мД при наличии высокопроницаемого нарушения с проницаемостью 3900 мД и шириной 25 м, расположенного на расстоянии 125 м от нагнетательной скважины

Такая геометрия соответствует не этапу полного разбуривания месторождения, а этапу опытно-промышленной эксплуатации участка в начале освоения месторождения, что представляет несомненный интерес для возможно раннего обнаружения и определения параметров нарушений.

Для простоты нарушение моделируется одной узкой высокопроницаемой, пористой полосой, полностью гидродинамически связанной с остальной частью пласта.

Для расчёта использована математическая модель двухфазной фильтрации слабосжимаемых несмешивающихся жидкостей (нефти и воды) в упругом пласте в отсутствие гравитационных и капиллярных сил [3] с использованием представления соответствующей системы уравнений, предложенного в [4]. На удалённых внешних границах всего участка задаются условия непротекания.

Полученная система уравнений по времени аппроксимируется в явном виде по насыщенности S и схемой «классики» [5] по давлению P . Подробно вычислительная схема описана в работе [6].

Количество ячеек по каждой из осей в горизонтальной плоскости – 22 шт. Вертикальный размер ячеек 1 м (что позволяет масштабировать задачу на любую толщину пласта), модель однослойная.

Добывающие скважины работают при заданном дебите 5,5 м³/сут (при эффективной толщине пласта 1 м). Нагнетательная скважина работает при постоянном расходе закачиваемой воды, рассчитанном исходя из компенсации отбора закачкой 100%. Скин-факторы скважин равны 0. Все скважины вводятся в эксплуатацию одновременно. Прочие параметры модели указаны в табл. 1.

Таблица 1. Параметры расчётной модели пласта

Название параметра	Величина	Единицы измерения
Глубина кровли	2010	м
Проницаемость пласта	39	мД
Проницаемость нарушения	от 390 до 3900	мД
Пористость пласта	0,15	д.ед.
Пористость нарушения	0,15	д.ед.
Ширина нарушения	от 15 до 50	м
Объёмный коэффициент воды	1,01	м ³ /м ³
Сжимаемость воды	$4,7 \cdot 10^{-5}$	1/бар
Вязкость воды	0,4	мПа·с
Объёмный коэффициент нефти	1,15	м ³ /м ³
Сжимаемость нефти	$1,01 \cdot 10^{-4}$	1/бар
Вязкость нефти	1,3	мПа·с
Сжимаемость породы	$4,7 \cdot 10^{-5}$	1/бар
Растворимость газа в нефти	121	м ³ /м ³
Давление насыщения нефти газом	165	бар
Начальное пластовое давление	201	бар
Начальная водонасыщенность	0,27	д.ед.

Относительная фазовая проницаемость нефти при начальной водонасыщенности ($S_{нач}=0,27$ д.ед.) равна 1 д.ед., и равна 0 д.ед. при максимальной водонасыщенности ($S_{макс}=0,69$ д.ед.). Относительная фазовая проницаемость воды равна 0 д.ед. при $S_{нач}$ и равна 0,24 д.ед. при $S_{макс}$.

В расчётах варьировались проницаемость и ширина нарушения, табл. 1, а также расстояние от нагнетательной скважины до нарушения – от 50 до 375 м.

3. Результаты расчётов

Результаты расчётов показывают, что наличие высокопроницаемого, хотя и узкого, нарушения кардинально меняет характер обводнения скважин, прежде всего, за счёт распространения воды по нарушению к окружающим скважинам и за пределы рассматриваемой ячейки, рис. 1 и 2.

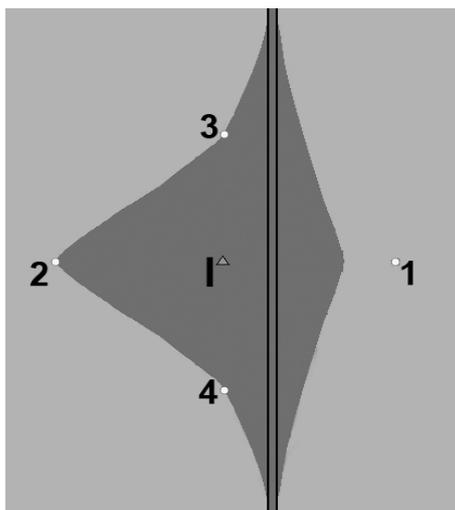


Рис. 2. Распространение воды (темная зона) по площади однородного пласта с проницаемостью 39 мД через 4 года после начала разработки при наличии высокопроницаемого нарушения при его проницаемости 3900 мД и ширине 25 м, расположенного на расстоянии 125 м от нагнетательной скважины

Видно, что за счёт ухода воды по нарушению происходит резкое замедление ее продвижения к добывающей скважине 1, и вода до неё доходит лишь спустя очень значительное время.

Следует отметить, что в такой ситуации и по характеру обводнения скважины 1, и по результатам трассерных исследований, проведённых в этот момент времени, как правило, будет сделано заключение о наличии непроницаемого барьера между нагнетательной скважиной 1 и добывающей скважиной 1. Хотя в действительности нарушение есть, но оно не непроницаемое, а, наоборот, сверхпроницаемое.

Если трассер не обнаружен в добывающей скважине в течение длительного времени, то это ещё не свидетельствует о наличии непроницаемого барьера между нагнетательной и добывающей скважинами; на практике же из этого факта часто делают именно такой вывод.

Полученный качественный расчётный факт, по-видимому, подтверждается и на практике. Например, на рис. 3 из работы [7] приведён результат трассерного исследования, проведённого для обнаружения дизъюнктивных нарушений на Ватьёганском месторождении.

Видно, что при закачке трассера в скв. 9094 отмечено его быстрое распространение вдоль предполагавшегося разлома. При этом, несмотря на высокую проводимость разлома, в скважинах, расположенных по другую сторону от нагнетательной скв. 9094, поступление трассера отмечено не было. Возможно, это связано именно с влиянием высокопроницаемого разлома, как и в расчётном случае, показанном на рис. 1 и рис. 2.

Второй важный вывод из результатов расчёта – появление высокопроницаемого нарушения приводит к резкому различию кривых обводнения различных добывающих скважин от времени, рис. 4 и табл. 2, 3.

При этом характерно, что наличие нарушения приводит к замедлению обводнения не только в «отсечённой» скв. 1, но и в скв. 2, расположенной по ту же сторону от нарушения, что и нагнетательная. Это вполне понятно: вода уходит по нарушению за пределы ячейки разработки, эффективность заводнения снижается, большая часть воды расходуется непроизводительно, поэтому и обводнение скв. 2 наступает позже, чем в однородном пласте.

При этом кривые обводнённости не только различаются между собой, но значительно отличаются от аналогичной кривой для однородного пласта, соответствующей отсутствию нарушения, табл. 2, 3 и рис. 4.

Как различие времён достижения определённого уровня обводнённости в разных скважинах элемента, так и отличие этих времён от времени, характерного для однородной среды могут свидетельствовать о наличии некоторого резкого гидродинамического возмущения в пределах элемента, т.е. указывать на наличие нарушения.

Такие различия вполне заметны и для очень высокой проницаемости нарушения (3900 мД) и для меньшей – 390 мД, табл. 3. Однако, при снижении проницаемости разлома заметно уменьшается различие времён достижения определённого уровня обводнённости. Например, если для проницаемости разлома в 3900 мД и локализации разлома на 50 м от нагнетательной скважины, различие времён достижения уровня обводнения, равного 0,5 %, в скв. 2 и 3 составляет 8 раз, то при 390 мД оно равно всего 3,5 раза. Хотя и это – вполне заметное различие.

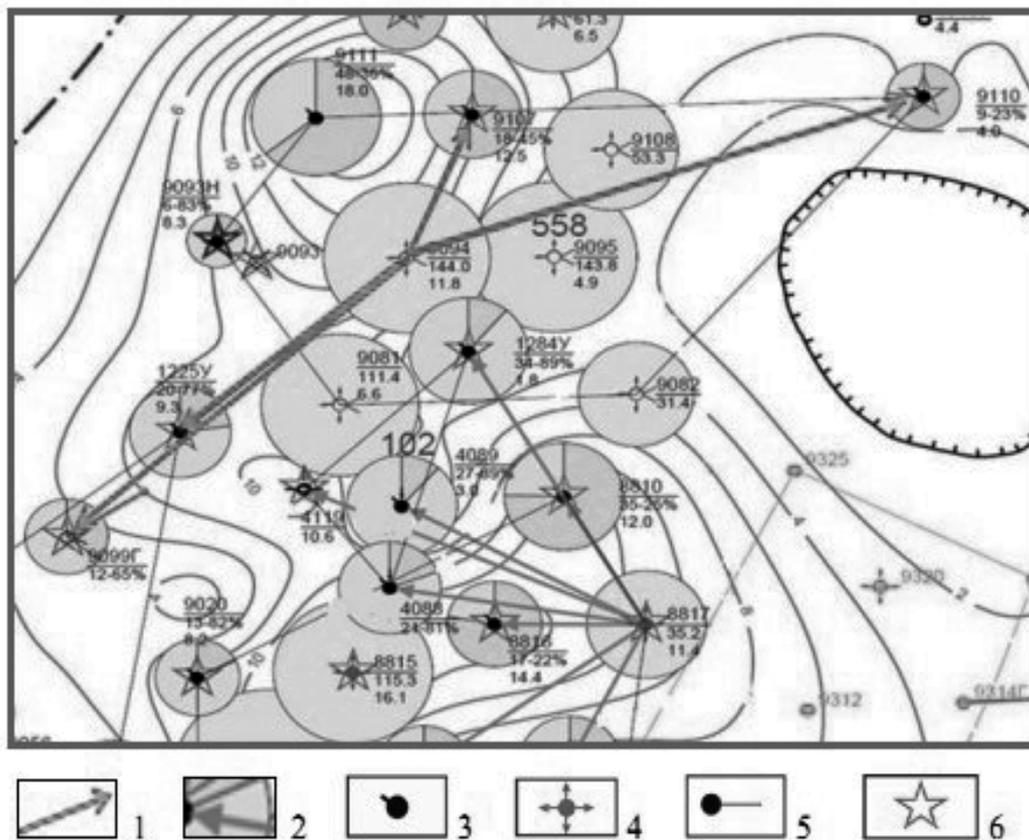


Рис. 3. Схема закачки раствора трассера в нагнетательные скв. 8817, 9094 на карте разработки (пласт ЮВ₁¹ Ватыганского месторождения) [7]; 1 – движение жидкости с трассером по плоскости сместителя разрывного нарушения; 2 – движение трассера по пласту; 3 – нефтяные скважины; 4 – нагнетательные скважины; 5 – нефтяные горизонтальные скважины; 6 – скважины с ГРП

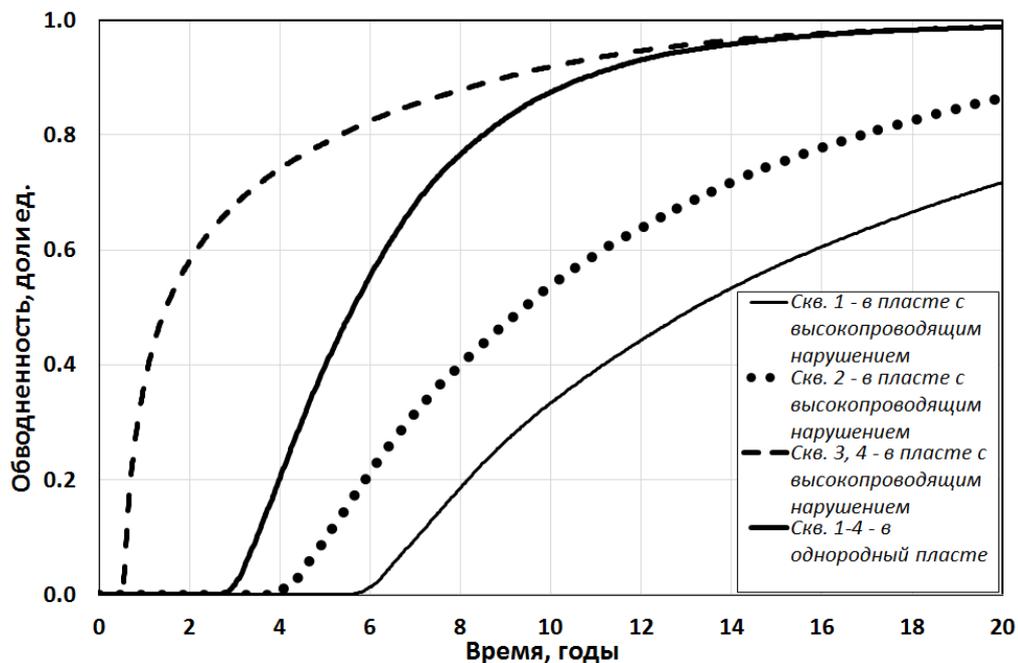


Рис. 4. Кривые изменения обводнённости продукции со временем в скважинах пятиточечного элемента площадной системы разработки в однородном пласте проницаемости 39 мД и при наличии высокопроницаемого нарушения, расположенного на расстоянии 50 м от нагнетательной скважины, при его проницаемости 3900 мД и ширине 25 м

Таблица 2. Времена достижения обводнённости продукции, равной 5 %, в скважинах пятиточечного элемента площадной системы разработки при проницаемости нарушения 3900 мД и его ширине 25 м

Проницаемость пласта, мД	39			
Проницаемость нарушения, мД	3900			
Расстояние до нарушения, м	50	125	250	375
Разница во времени достижения обводнённости продукции, равной 5 %, между скв. 2 и скв. 1 в пласте с высокопроницаемым нарушением, годы	1.9	1.6	0.5	0.0
Разница во времени достижения обводнённости продукции, равной 5 %, между скв. 2 и скв. 3 (скв.4) в пласте с высокопроницаемым нарушением, годы	4.0	1.7	0.2	0.0
Разница во времени достижения обводнённости продукции, равной 5 %, между скв. 1 и скв. 3 (скв. 4) в пласте с высокопроницаемым нарушением, годы	5.9	3.4	0.7	0.0
Время достижения обводнённости продукции, равной 5 %, в скв. 1-4 для однородного пласта, годы	3.1			
Разница во времени достижения обводнённости продукции, равной 5 %, между скв. 2 в случаях: пласта с высокопроницаемым нарушением и однородного пласта, годы	1.5	0.7	0.2	0.0
Разница во времени достижения обводнённости продукции, равной 5%, между скв.1 в случаях: пласта с высокопроницаемым нарушением и однородного пласта, годы	3.4	2.4	0.7	0.0
Разница во времени достижения обводнённости продукции, равной 5%, между скв.3 (скв.4) в случаях: пласта с высокопроницаемым нарушением и однородного пласта, годы	-2.5	-1.0	0.0	0.0

Таблица 3. Времена достижения обводнённости продукции, равной 5%, в скважинах пятиточечного элемента площадной системы разработки в однородном пласте и при наличии высокопроницаемого нарушения различной проницаемости и ширины, расположенного на расстоянии 50 м от нагнетательной скважины

Проницаемость пласта, мД	39					
Проницаемость нарушения, мД	3900			390		
Ширина нарушения, м	15	25	50	15	25	50
Разница во времени достижения обводнённости продукции, равной 5%, между скв.2 и скв.1 в пласте с высокопроницаемым нарушением, годы	1.5	1.9	2.0	0.1	0.2	0.2
Разница во времени достижения обводнённости продукции, равной 5%, между скв.2 и скв.3 (скв.4) в пласте с высокопроницаемым нарушением, годы	4.0	4.0	3.8	2.3	2.6	2.5
Разница во времени достижения обводнённости продукции, равной 5%, между скв.1 и скв.3 (скв.4) в пласте с высокопроницаемым нарушением, годы	5.5	5.9	5.8	2.5	2.8	2.7
Время достижения обводнённости продукции, равной 5%, в скв.1-4 для однородного пласта, годы	3.1					
Разница во времени достижения обводнённости продукции, равной 5%, между скв.2 в пласте с высокопроницаемым нарушением и в однородном пласте, годы	1.0	1.5	1.5	0.2	0.5	0.6
Разница во времени достижения обводнённости продукции, равной 5 %, между скв. 1 в пласте с высокопроницаемым нарушением и в однородном пласте, годы	2.6	3.4	3.5	0.4	0.7	0.7
Разница во времени достижения обводнённости продукции, равной 5 %, между скв. 3 (скв. 4) в пласте с высокопроницаемым нарушением и в однородном пласте, годы	-2.9	-2.5	-2.3	-2.1	-2.1	-2.0

При удалении же нарушения от нагнетательной скважины уменьшается различие во временах достижения определённого уровня обводнённости, характер обводнения приближается к однородной среде. Например, если проницаемость нарушения равна 3900 мД и нарушение локализовано на расстоянии 50 м от нагнетательной скважины, то отношение времён достижения 0,5 %-й обводнённости в скв. 2 и 3 составляет 8 раз, а при удалении на 125 м – всего 1,8 раз; при удалении же на 250 м – всего 1,1 раза, т.е. отличие от однородной среды практически несущественно и вряд ли будет возможно на практике.

Разумеется, такие же выводы следуют и из табл. 2 и 3, в которых жирным шрифтом помечены те параметры нарушения, при которых возможно его выявление по разнице времён достижения определённого уровня обводнённости.

Точно также вероятность выявления нарушения снижается, если уменьшается его ширина (сумма внутренней и внешней его частей). Однако если нарушение отстоит от нагнетательной скважины всего на 50–100 м, то его обнаружение возможно, даже при ширине 15 м и проницаемости всего 390 мД, табл. 3.

В целом, можно сказать, что *если: проницаемость нарушения более чем в 10 раз превосходит проницаемость пласта, оно удалено от нагнетательной скважины на расстояние не более 50–100 м, а его ширина более 15 м, то, при тщательном измерении кривой обводнения скважин от времени, наличие нарушения по этим данным может быть выявлено.*

Разумеется, эти данные могут дать информацию и об ориентировке нарушения. Например, если велико расхождение времён достижения, скажем, 5 %-го уровня обводнения по двум направлениям: по линии скв. 1–2 и по линии скв. 3–4, причём по сравнению с однородной средой по линии скв. 3–4 идёт опережение, а по линии скв. 1–2 происходит замедление времён обводнения – то нарушение ориентировано в направлении скв. 3–4.

Аналогичные результаты получены и для варианта двух добывающих скважин, между которыми расположена нагнетательная скважина, а на некотором удалении от неё, между нагнетательной и добывающей скважиной, располагается нарушение.

Результат замедления в обводнении по сравнению с однородным пластом будет контролироваться тремя безразмерными параметрами: отношением проницаемостей пласта и нарушения, отношением величин двух

расстояний – от нагнетательной скважины до нарушения и между скважинами, безразмерным временем, равным реальному времени, умноженному на темп закачки воды в нагнетательной скважине и делённому на поровый объём рассматриваемого участка. Этот объём, в свою очередь, равен произведению пористости на эффективную толщину пласта и на площадь расчётного участка.

Как показали специальные контрольные расчёты, одинаковое изменение проницаемости пласта, проницаемости нарушения, продуктивности и приёмистости скважин, приводит к пропорциональному изменению масштаба времён достижения определённого уровня обводнённости. Поэтому полученные результаты имеют более общий характер, чем только для рассмотренных численных значений параметров.

Важную дополнительную информацию может дать и применение трассерного метода исследований [8–10]. Одним из его показателей является время прорыва первой порции трассера к добывающей скважине. Рассмотрим, сугубо качественно, как сказывается наличие нарушения на величину времени прорыва трассера.

Применение трассерного метода считается практически реализуемым, если обводнение скважин превышает 5 % [10]. Рассмотрим ситуацию, когда обводнение скв. 3 и скв. 4 равно 5 %. До скв. 1 и 2 ни вода, ни трассер не дойдут. Распространение воды в этом случае показано на рис. 5.

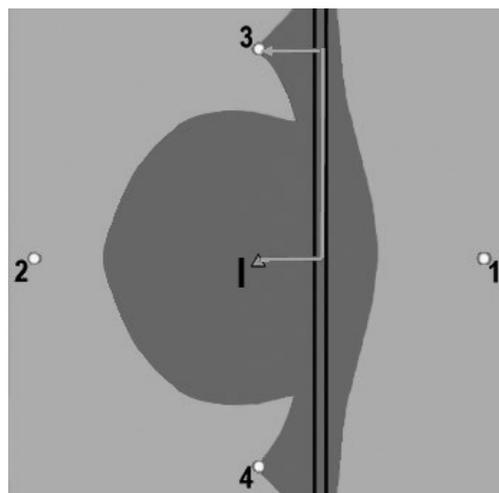
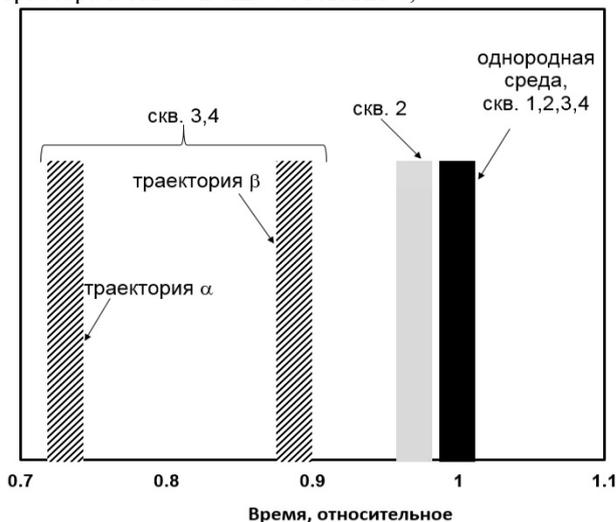


Рис. 5. Примерная линия тока трассера в случае нулевой обводнённости скв. 1 и 2, 5 %-й обводнённости скв. 3 и 4 при наличии высокопроницаемого нарушения проницаемости 3900 мД и ширины 25 м, расположенного на расстоянии 125 м от нагнетательной скважины

Ясно, что при закачке трассера единственным путём его прихода в скв. 3 и скв.4 является показанная на рисунке линия тока через нарушение. Если принять время прорыва первой порции трассера в добывающие скважины в однородном пласте, без нарушения, за 1, то в ситуации, изображённой на рис. 5, время прорыва трассера к скв. 3 и скв. 4 составит 0,79.



В другом случае, когда обводнённость скв.2 будет равна 5 % (а обводнённость скв. 3 и скв.4 равна 34 %), появляется намного большее количество линий тока воды и трассера, например, как показано на рис. 6 – линия α и линия β .

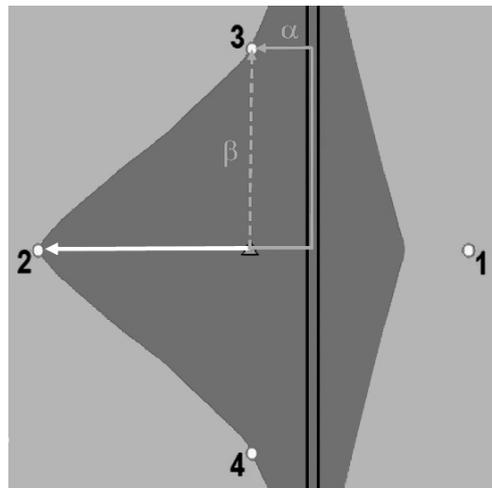


Рис. 6. Время прихода первой порции трассера (слева) в однородном пласте проницаемости 39 мД при наличии высокопроницаемого нарушения (проницаемости 3900 мД и ширины 25 м, расположенного на расстоянии 125 м от нагнетательной скважины) по двум возможным линиям тока, показанным справа; скв. 1 ещё не обводнилась, обводнённость продукции в скв. 2 равна 5 %, а скв. 3 (4) – 34 %

При этом время прихода трассера по линии тока β составит 0,89 относительных единиц, т.е. близко ко времени прорыва в однородной среде, в то время как по линии тока α оно равно 0,73. К скв. 2 в этом случае первая порция трассера подойдёт через относительное время 0,97, т.е. неотличимо от однородной среды.

Следовательно, данные трассерного метода дадут дополнительное подтверждение наличию сильного гидродинамического возмущения, т.е. нарушения.

Однако данные трассерных исследований дают и дополнительную информацию о наличии нарушения и его локализации. Дело в том, что даже в соответствии с полученными простейшими оценками, форма кривой «концентрация трассера – время» для двух рассмотренных моментов обводнения будет различна. Для первого варианта, когда обводняются только скважины 3 и 4, она будет иметь форму сравнительно узкого пика, тогда как во втором варианте эта кривая будет сильно размыта за счёт прихода трассера по более широкому набору линий тока. Это же следует и из теории трассерного метода для случая стационарного течения [11]. При этом набор линий тока будет разный в различные

моменты заводнения, за счёт различий в конфигурации линий тока.

Из этих, достаточно иллюстративных, результатов следует два вывода.

Во-первых, трассерный метод даёт информацию о распределении линий тока, которую данные об обводнённости не содержат. Иными словами, эти два метода обнаружения нарушения и изучения межскважинного расстояния существенно дополняют друг друга [11].

Во-вторых, для обнаружения нарушений на стадии опытно-промышленной эксплуатации, трассерные исследования целесообразно проводить несколько раз, в различные моменты процесса обводнения, используя различные типы трассеров. Тогда информация о степени неоднородности исследуемого участка будет существенно более полной. Определение наличия нарушения и его параметров тогда может быть сделано и количественно, стандартным приёмом нахождения оптимального набора параметров модели, обеспечивающего минимальную величину суммы квадратов отклонения экспериментальных и модельных кривых: обводнения скважин и «концентрация трассера

– время», полученных экспериментальным путём в процессе опытно-промышленной эксплуатации.

4. Заключение

1. Наличие высокопроницаемого нарушения может приводить к оттоку нагнетаемой воды за пределы участка, что вызывает резкое замедление продвижения воды к скважинам, отделённых нарушением от нагнетательной скважины. Отсутствие поступления воды (или трассера) в добывающую скважину в течение продолжительного времени не может однозначно интерпретироваться как наличие непроницаемого барьера между нагнетательной и добывающей скважинами.

2. Наличие нарушения вызывает заметное изменение характера обводнения скважин по сравнению с однородной средой. При этом времена достижения определённого уровня обводнения продукции в разных скважинах заметно различаются, как для разных скважин, так и по сравнению с однородной средой.

3. При благоприятном сочетании различных геометрических и фильтрационных параметров пласта и нарушения возможно

выявление нарушения и его ориентировки по данным об обводнении скважин и трассерных исследований. Благоприятным является случай, когда проницаемость нарушения более чем в 10 раз превосходит проницаемость пласта, оно удалено от нагнетательной скважины на расстояние не более 50–100 м, а его ширина (включая внутреннюю и внешнюю зоны) более 15 м.

4. Трассерные исследования существенно дополняют данные по характеру обводнения, особенно, если проводятся неоднократно в процессе заводнения с разными типами трассеров.

5. Целесообразно проводить подобный анализ данных по обводнению и трассерных исследований на стадии опытно-промышленной эксплуатации для наиболее раннего обнаружения высокопроницаемых дизъюнктивных нарушений.

Работа выполнена при поддержке гранта РФФИ 18-07-00676 А.

Determination of Tectonic Fault in Oil Field Using Well's Flooding Characteristics and Tracer Data

V.A. Yudin, S.G. Volpin, I.V. Afanaskin, N.P. Efimova

Abstract. Possibility of detecting high permeable tectonic fault by using well's flooding characteristics and tracer data is investigated. The five-spot pattern was considered working during pilot production operation (all other wells are located far away from the pattern and don't interfere with pattern wells). Production of pattern wells was investigated by computer modelling. It was shown that: 1. If fault is located between injection and production well then water breakthrough to such well might take a lot of time due to the considerable water outflow along the fault; 2. Tectonic fault can be detected using flooding characteristics of pattern production wells in some favorable cases; 3. Tracer tests can provide meaningful additional information especially if are done several times during pilot operation by using different tracers.

Keywords: tectonic faults, tracer test, well flooding characteristic.

Литература

1. С.Г. Вольпин, И.В. Афанаскин, В.А. Юдин, Н.П. Ефимова. Возможность определения фильтрационных параметров дизъюнктивных нарушений на нефтяных месторождениях по данным гидродинамических исследований пластов и скважин. М., 2018.
2. А. Чодри. Гидродинамические исследования нефтяных скважин. М., ООО Премиум Инжиниринг, 2011.
3. Х. Азиз Х, Э. Сеттари. Математическое моделирование пластовых систем. М.- Ижевск, ИКИ, 2004.

4. Р.М. Кац, Е.Р. Волгин, И.В. Афанаскин. Численное моделирование двухфазной фильтрации нефти и воды. «Труды НИИСИ», Т. 4 (2014), № 2, 141-148.
5. П. Роуч. Вычислительная гидромеханика. М., Мир, 1980.
6. И.В. Афанаскин, А.В. Королев А.В., П.В. Ялов, Д.В. Солопов, Ф.Ю. Сорокин Ф.Ю. Трёхмерное моделирование заводнения нефтяных пластов с применением разностной схемы «классики», «Нефтепромысловое дело», 2017, № 11, 14–20.
7. А.Н. Лесной. Исследование влияния дизъюнктивных нарушений в юрских отложениях на строение залежей нефтяных месторождений Когалымского нефтегазоносного района. Дис. на соиск. уч. степ. к.г.-м.н. М., РГУ нефти и газа им. И.М. Губкина, 2015.
8. Э.В. Соколовский, С.Н. Чижов, Ю.И. Тренчиков и др. Методическое руководство по технологии проведения трассерных исследований и интерпретации их результатов для регулирования и контроля процесса заводнения нефтяных залежей. Грозный, СевКавНИПИнефть, 1989.
9. М.С. Хозяинов, М.В. Веселов, В.С. Иванов, Г.С. Кузьмина и др. Методические рекомендации по количественной интерпретации данных индикаторных исследований межскважинного пространства нефтяных месторождений. М., ВНИИГеоинформсистем, 1988.
10. Д.А. Чернокожев. Совершенствование технологии индикаторных исследований для оценки фильтрационной неоднородности межскважинного пространства нефтяных пластов. Дис. на соиск. уч. степ. к.т.н. Дубна, `Международный университет природы, общества и человека «Дубна», 2008.
11. W.E. Brigham, Maghsood Abbaszadeh-Dehghani. Tracer Testing for Reservoir Description, “Journal of Petroleum Technology”, V. 39 (1987), № 5, 519–527.

Распределенное моделирование глубины резкости при визуализации трехмерных сцен на GPU

А.В. Мальцев¹, М.А. Торгашев²

¹ФГУ ФНЦ НИИСИ РАН, Москва, Россия, avmaltcev@mail.ru;

²ФГУ ФНЦ НИИСИ РАН, Москва, Россия, mtorg@mail.ru

Аннотация: В работе предлагаются решения для моделирования глубины резкости на изображениях трехмерных виртуальных сцен, видимых посредством виртуальных камер. Разработанные распределенные методы и алгоритмы основаны на использовании современных многоядерных графических процессоров и обеспечивают рендеринг виртуальной среды в масштабе реального времени.

Ключевые слова: трехмерная сцена, визуализация, виртуальная камера, глубина резкости, фокусировка, графический процессор.

1. Введение

Одним из ключевых элементов моделируемой на компьютерах трехмерной виртуальной среды является виртуальная камера. От ее параметров во многом зависят изображения трехмерной сцены, получаемые в результате визуализации. Реализация виртуальной камеры предполагает наличие модели ее взаимодействия с окружающим пространством, а именно, каким образом попадающие в нее виртуальные световые лучи будут формировать видимое нами изображение.

Как правило, при рендеринге сцен используется модель «пинхол» [1, 2] – камера, входная апертура которой стремится к 0, а все объекты находятся в фокусе. Однако такой модели недостаточно для имитации реальных устройств фото- и видеонаблюдения, имеющих объективы с различным фокусным расстоянием и параметрами диафрагмы. В реальной жизни такие устройства обладают ограниченной глубиной резкости (ГРИП). Камера фокусируется на поверхности определенного объекта, который находится на некотором расстоянии от нее по главной оптической оси. Объекты, расположенные в пределах этого расстояния, отображаются в кадре с наибольшей резкостью, а остальные как бы размываются. Степень такого размывания зависит от относительного отверстия объектива и увеличивается с возрастанием разницы между расстоянием фокусировки и расстоянием от камеры до объекта.

Существуют два различных подхода к моделированию эффекта глубины резкости при рендеринге трехмерных виртуальных сцен. Первый подход основан на работе в пространстве объектов, то есть решении задачи непосредственно в процессе визуализации.

Например, в распределенной трассировке лучей (*distributed ray tracing* [3]) через каждый пиксел кадра пропускается ряд лучей, по физическим законам рассчитывается их преломление в линзе объектива, дальнейшее распространение в сцене и пересечение с объектами.

Второй подход состоит в постобработке изображений, полученных при рендеринге сцены с использованием стандартной виртуальной камеры типа пинхол. Эффект глубины резкости в таком случае моделируется путем размывания участков изображений с учетом буфера глубины. Постобработка, как правило, дает менее физически точный результат, чем методы, основанные на пространстве объектов, однако выполняется значительно быстрее. Это позволяет достичь оптимального баланса в отношении качества изображения к скорости его генерации. Последняя является одним из важных критериев применимости того или иного метода в области систем виртуального окружения и имитационно-тренажерных комплексов.

В данной работе предлагаются методы и алгоритмы для распределенной реализации эффекта глубины резкости на изображениях от виртуальных устройств наблюдения. Созданные решения основаны на постобработке изображений трехмерных сцен на многоядерном GPU с учетом модели виртуальной камеры, задаваемой несколькими основными параметрами объектива. Рассмотрим их подробнее.

2. Модель камеры

Для решения поставленной задачи в данной работе используется модель виртуальной камеры с тонкой линзой (рис. 1). Пусть для нее заданы фокусное расстояние f и расстояние u до объекта, находящегося в данный момент в фокусе, а также диафрагменное число k .

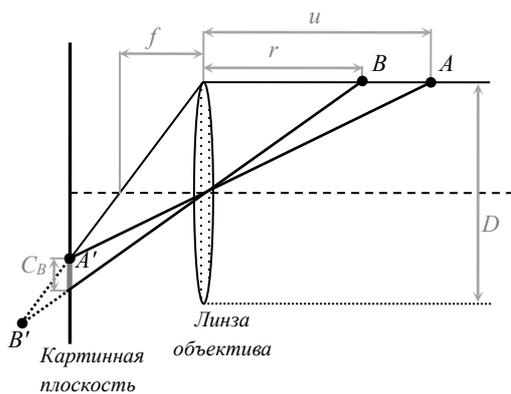


Рис. 1. Модель камеры с тонкой линзой

Все лучи, падающие на линзу из одной точки пространства, сходятся в одной точке за этой линзой. Для точек пространства, расположенных на удалении u от камеры, соответствующие им точки схождения лучей будут лежать в картинной плоскости. Например, на рисунке 1 точка A проецируется в точку A' . Лучи от остальных точек будут сходить за или перед картинной плоскостью и пересекать ее световым пучком, образуя на изображении так называемое пятно рассеяния (*Circle of Confusion*). Так точка B отображается на матрице камеры в виде пятна C_B .

Диаметр пятна рассеяния на картинной плоскости рассчитывается по формуле

$$C = \left| D \cdot f \frac{u - r}{r(u - f)} \right|,$$

где D – диаметр линзы, r – расстояние до объекта (см. [4]). Учитывая определение диафрагменного числа, значение диаметра линзы можно выразить через имеющиеся параметры, как $D = f/k$. Тогда

$$C = \left| \frac{f^2 (u - r)}{k \cdot r (u - f)} \right|. \quad (1)$$

3. Моделирование глубины резкости

Осуществим имитацию глубины резкости путем проведения постобработки изображения виртуальной среды, полученного с использованием стандартной камеры-пинхол, в ходе подготовки каждого кадра. Суть обработки заключается в размытии картинки на базе фильтра Гаусса с учетом величины пятен рассеяния для точек сцены, соответствующих пикселям изображения, а также расстояний между этими точками и камерой. Цвет каждого пикселя кадра заменяется на

средневзвешенную сумму цветов этого пикселя и его соседей, лежащих в пределах некоторого радиуса. Весовые коэффициенты вычисляются на основе закона нормального распределения. Значение радиуса выбирается пропорциональным максимальному размеру пятна рассеяния для точек, находящихся в пределах пирамиды видимости камеры.

Применяемый подход обладает свойством линейной сепарабельности, что позволяет разделить процесс размытия на два этапа. На первом изображение размывается вдоль горизонтальной оси кадра, на втором – вдоль вертикальной. Кроме того, весовой коэффициент пикселей, расположенных на расстоянии более 3σ от рассматриваемого (σ – среднеквадратичное отклонение в функции Гаусса), будет близок к 0, поэтому такие пиксели не будут оказывать значимого влияния на общую сумму цветов.

Перед выполнением обработки кадра необходимо предварительно скопировать текущие буферы цвета и глубины в текстуры. Обозначим их T_{img} и T_{depth} соответственно. Расчет цветов пикселей результирующего изображения T_{res} производится параллельно на GPU во фрагментном шейдере. Размеры T_{img} , T_{depth} и T_{res} одинаковы. Чтобы определить количество соседних пикселей, участвующих в фильтре размытия вычислим диаметры C_n и C_f пятен рассеяния для точек виртуального пространства, лежащих соответственно в ближней и дальней отсекающих плоскостях камеры. Для этого вначале рассчитаем расстояния n_p и f_p до этих плоскостей в системе координат (СК) VCS камеры, матрица M перспективного преобразования которой имеет следующий вид

$$M = \begin{pmatrix} \frac{2n_p}{r_p - l_p} & 0 & \frac{r_p + l_p}{r_p - l_p} & 0 \\ 0 & \frac{2n_p}{t_p - b_p} & \frac{t_p + b_p}{t_p - b_p} & 0 \\ 0 & 0 & \frac{f_p + n_p}{n_p - f_p} & \frac{-2f_p n_p}{f_p - n_p} \\ 0 & 0 & -1 & 0 \end{pmatrix},$$

где l_p , r_p , t_p и b_p – расстояния, задающие левую, правую, верхнюю и нижнюю границы пирамиды видимости. Применение данной матрицы к произвольной точке P в видовой системе VCS переводит P в пространство нормализованного объема видимости NDCS в виде куба со стороной 2 и центром в начале СК:

$$P_{ndcs} = M \cdot P_{ves},$$

$$P_{ves} = (x, y, z, 1), \quad P_{ndcs} = (x', y', z', w').$$

При этом

$$z' = m_{22}z + m_{23}, \quad w' = -z, \quad (2)$$

где

$$m_{22} = \frac{f_p + n_p}{n_p - f_p}, \quad m_{23} = \frac{-2f_p n_p}{f_p - n_p}.$$

Чтобы получить координаты P в экранном пространстве необходимо выполнить перспективное деление координат (x', y', z', w') на w' , а также произвести сдвиг начала СК NDCS в вершину $(-1, -1, -1)$ куба видимости и растяжение ее осей в 2 раза:

$$P_{scr} = (x'', y'', z'', 1) = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \frac{P_{ndcs}}{w'} \quad (3)$$

Из формул (2) и (3) получаем, что

$$\frac{z'}{w'} = -m_{22} - \frac{m_{23}}{z} = 2z'' - 1, \\ z = \frac{-m_{23}}{m_{22} + 2z'' - 1}.$$

Поскольку взгляд виртуальной камеры направлен вдоль оси $-Z$ этой СК, то для любой видимой ей точки P значение $z < 0$. Поэтому расстояние r от камеры до точки P в видовой системе будет равно $-z$:

$$r = \frac{m_{23}}{m_{22} + 2z'' - 1} \quad (4)$$

Полученная формула (4) позволяет по известному значению глубины z'' точки в экранном пространстве, хранящемуся в буфере или текстуре глубины, вычислить расстояние от этой точки до камеры в системе VCS. Передняя и задняя плоскости отсечения содержат точки, глубины которых в экранных координатах равны 0 и 1 соответственно. Следовательно, расстояния n_p и f_p определяются, как

$$n_p = \frac{m_{23}}{m_{22} - 1}, \quad f_p = \frac{m_{23}}{m_{22} + 1}.$$

Искомые диаметры пятен рассеяния для этих плоскостей вычислим с использованием формулы (1):

$$C_n = \min \left(\left| \frac{f^2 (u - n_p)}{k \cdot n_p (u - f)} \right|, 10 \right), \\ C_f = \min \left(\left| \frac{f^2 (u - f_p)}{k \cdot f_p (u - f)} \right|, 10 \right).$$

Теперь рассмотрим процесс размытия изображения T_{img} с применением фрагментного шейдера. Сначала произведем размытие вдоль горизонтальной оси кадра. При этом результирующий цвет каждого пиксела вычис-

лим на основе средневзвешенной суммы его исходного цвета, а также цветов n пикселей слева и справа от него, где

$$n = \lceil 3 \cdot \max(C_n, C_f) \rceil,$$

скобки обозначают взятие целой части числа. Обозначим индексы рассматриваемого пиксела, для которого вычисляется цвет, через x_c и y_c (отсчет ведется от левого нижнего угла кадра). Тогда его весовой коэффициент

$$w_c = \min \left(1, \frac{1}{\sigma(x_c, y_c) \sqrt{2\pi}} \right), \quad \sigma(x, y) = C(d), \quad (5)$$

где $d = T_{depth}(x, y)$ – значение глубины пиксела (x, y) , записанное в текстуре T_{depth} , $C(d)$ – диаметр пятна рассеяния для соответствующей ему точки виртуального пространства.

Чтобы во фрагментном шейдере вычислить $C(d)$ по значению глубины d , представленному в экранной СК, используем формулу (1) с подстановкой в нее выражения для r из (4) при $z'' = d$:

$$C = \left| \frac{f^2 (u - r)}{k \cdot r (u - f)} \right| = \left| \frac{f^2}{k (u - f)} \cdot \left(\frac{u}{r} - 1 \right) \right| = \\ = \left| \frac{f^2}{k (u - f)} \cdot \left(\frac{u (m_{22} + 2d - 1)}{m_{23}} - 1 \right) \right| = \\ = \left| \frac{f^2}{k (u - f)} \cdot \left(\frac{2u}{m_{23}} \cdot d + \frac{u (m_{22} - 1)}{m_{23}} - 1 \right) \right|.$$

Вводя параметры

$$a_0 = \frac{f^2}{k (u - f)}, \quad a_1 = \frac{2u}{m_{23}} \cdot a_0, \\ a_2 = \left(\frac{u (m_{22} - 1)}{m_{23}} - 1 \right) \cdot a_0,$$

получим

$$C(d) = |a_1 \cdot d + a_2|.$$

Значения a_0 , a_1 и a_2 являются постоянными, как минимум, в рамках визуализации одного кадра изображения виртуальной сцены, поэтому они могут быть вычислены один раз до запуска шейдера и переданы в него в качестве входных переменных.

Весовой коэффициент для каждого из пикселей, расположенных по горизонтали слева и справа от (x_c, y_c) на расстоянии $i \in [1, n]$, $i \in \mathbb{Z}$ равен

$$w(x, y) = q \cdot \min \left(1, \frac{1}{\sigma(x, y) \sqrt{2\pi}} \cdot e^{-i^2/2\sigma^2(x, y)} \right), \quad (6)$$

$$q = \begin{cases} 1, & d < d_c \\ \min \left(1, \frac{1}{|\sigma(x, y) - \sigma(x_c, y_c)|} \right), & d \geq d_c \end{cases},$$

где d и d_c – глубины пикселей (x, y) и (x_c, y_c) , записанные в текстуре T_{depth} . Учитывая (5) и (6), искомый цвет пикселя размытого изображения вычислим по формулам

$$T(x_c, y_c) = \frac{1}{W} \cdot \left(w_c T_{\text{img}}(x_c, y_c) + \sum_{i=1}^n \left(w(x_c - i, y_c) T_{\text{img}}(x_c - i, y_c) + w(x_c + i, y_c) T_{\text{img}}(x_c + i, y_c) \right) \right),$$

$$W = w_c + \sum_{i=1}^n \left(w(x_c - i, y_c) + w(x_c + i, y_c) \right)$$

и запишем его в промежуточную текстуру T . Результатом работы шейдера является размытое вдоль горизонтальной оси изображение T_{img} , сохраненное в T . Далее к нему необходимо применить размытие по вертикальной оси, которое осуществляется аналогичным образом с помощью фрагментного шейдера. Отличие состоит в том, что соседние пиксели выбираются уже не по горизонтали, а по вертикали. Тогда результирующий цвет пикселя изображения с имитацией эффекта глубины резкости, записываемый в T_{res} , вычисляется по формулам

$$T_{\text{res}}(x_c, y_c) = \frac{1}{W} \cdot \left(w_c T(x_c, y_c) + \sum_{i=1}^n \left(w(x_c, y_c - i) T(x_c, y_c - i) + w(x_c, y_c + i) T(x_c, y_c + i) \right) \right),$$

$$W = w_c + \sum_{i=1}^n \left(w(x_c, y_c - i) + w(x_c, y_c + i) \right).$$

4. Заключение

В данном исследовании была рассмотрена задача имитации эффекта глубины резкости, наблюдаемого на изображениях от реальных фото- и видеокамер, при рендеринге трехмерных виртуальных сцен. Предложенные методы и подходы обеспечивают распределенную реализацию данного эффекта с использованием современных многоядерных GPU в масштабе реального времени. На основе представленных решений были разработаны программные модули, которые были успешно апробированы в составе системы визуализации трехмерных виртуальных сцен, созданной в ФГУ ФНЦ НИИСИ РАН.

На рисунке 2 представлен пример изображения виртуальной модели Международной космической станции, полученного с помощью созданных программных компонентов. Фокусировка производилась на модуле, расположенном в центре кадра.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00393.

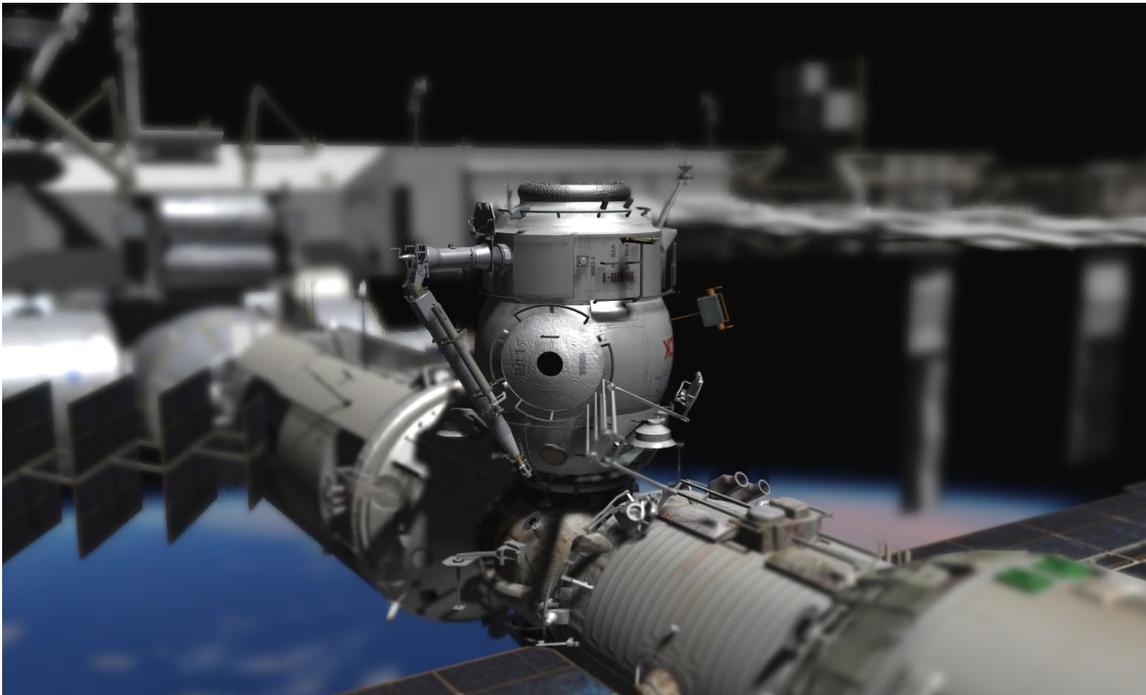


Рис. 2. Имитация глубины резкости на изображении от виртуальной камеры

Distributed simulation of DOF when rendering virtual scenes on GPU

A.V. Maltsev, M.A. Torgashev

Abstract: The paper proposes solutions for DOF (depth of field) simulation on images of three-dimensional virtual scenes, which are visible by virtual cameras. Developed methods and algorithms are based on using distributed computing by means of modern multicore graphics processors and provide real-time visualization of virtual environment.

Keywords: three-dimensional scene, visualization, virtual camera, depth of field, focusing, graphics processor.

Литература

1. Pinhole camera. URL: https://en.wikipedia.org/wiki/Pinhole_camera (дата обращения: 24.09.2019).
2. R. Hartley, A. Zisserman. Multiple View Geometry in computer vision. Cambridge University Press, 2003.
3. R.L. Cook, T. Porter, L. Carpenter. Distributed ray tracing. "ACM SIGGRAPH Computer Graphics". – V. 18 (1984), № 3, 137-147.
4. Circle of confusion. URL: https://en.wikipedia.org/wiki/Circle_of_confusion (дата обращения: 24.09.2019).

Визуализация векторных полей цифровой модели керна

М.В. Михайлюк¹, И.Н. Чередниченко²

¹ ФГУ ФНЦ НИИСИ РАН, Москва, Россия, mix@niisi.ras.ru;

² ФГУ ФНЦ НИИСИ РАН, Москва, Россия, inch@jscs.ru

Аннотация: В статье описываются методы и алгоритмы визуализации векторных полей цифровой модели керна в формате NetCDF, основанные на обработке данных как в центральном процессоре (CPU), так и на процессорах графической карты (GPU). Для исключения условных операторов, замедляющих обработку данных на графических процессорах, при вычислении цветов отображаемых векторов предлагается использовать индикаторные функции. Интерфейс разработанного программного модуля позволяет пользователю выбирать режим отображения векторных данных и гибко настраивать параметры визуализации.

Ключевые слова: визуализация, векторные поля, OpenGL, индикаторные функции, шейдеры.

1. Введение

Цифровое моделирование керна в настоящее время широко востребовано для решения многих задач, возникающих при разведке и добыче трудно извлекаемых запасов нефти. Цифровая модель керна строится на основе рентгеновской томографии и представляет собой трехмерный массив коэффициентов поглощения рентгеновского излучения нефтеносной породой в ячейках объема моделирования. Использование цифровой модели керна позволяет избежать разрушений керна во время исследования, повышая, тем самым, его эффективность. Результатами моделирования процессов в цифровой модели керна являются скалярные и векторные поля (температуры, давления, скорости, напряжения и др.). Изучение этих полей без визуального представления весьма затруднительно. Поэтому, важной задачей является создание систем, ориентированных на визуальное представление как скалярных, так и векторных результатов моделирования процессов в модели керна, позволяющее изучать их интерактивно [1].

Существующие системы визуализации векторных полей являются либо закрытыми коммерческими разработками (например, Amira-Avizo [2]), либо открытыми универсальными пакетами, которые не оптимизированы под визуализацию больших векторных полей и не справляются с их интерактивным отображением. Например, широко известная программа ParaView [3] для визуализации научных данных, резко замедляется при попытке визуализировать поле из более, чем 1 млн. векторов. В данной работе предлагаются метод и алгоритмы

визуализации векторного поля на примере поля скоростей, полученного в результате цифрового моделирования протекания жидкости через керн. Визуализация проводится с использованием возможностей современных видеокарт, таких, как дисплейные списки [4] и шейдеры [5]. Такой подход позволил разработать и использовать методы и алгоритмы, соответствующие требованиям интерактивной обработки и визуализации графической информации данных цифровой модели керна.

2. Визуализация векторного поля

2.1. Предобработка исходных данных

Мы рассматриваем трехмерные (3D) векторные поля, записанные в формате NetCDF. Этот формат был разработан Notre Dame of Maryland University [6] и широко применяется для хранения геоданных, данных по наблюдению за погодой, рентгенографии и во многих других областях, требующих хранения больших объемов структурированной информации. Достоинством этого формата является его самодокументированность и наличие библиотеки функций чтения и записи хранящихся данных.

Исходные данные для объема моделирования из $L = N_x \times N_y \times N_z$ ячеек записаны в файле в виде трех отдельных массивов, содержащих x, y, z компоненты векторного поля. После считывания эти данные переводятся в одномерный массив V векторов. Ячейке с индексами (i, j, k) будет соответствовать элемент массива с номером

$n = i + jN + kN^2$. Отсюда видно, что i, j и k вычисляются по n следующим образом:

$$\begin{aligned} i &= n \pmod{N}, \\ j &= (n \div N) \pmod{N}, \\ k &= ((n \div N) \div N) \pmod{N}, \end{aligned} \quad (1)$$

где операция \div означает целочисленное деление.

Для корректного начального отображения, выполним масштабирование всех входных векторных данных, приведя значения их длин в полуинтервал $[0,1)$:

$$M = \max_{1 \leq n \leq L} (|\vec{v}_n|) + \varepsilon, \quad \vec{v}'_n = \vec{v}_n / M.$$

Малая величина ε добавлена для того, чтобы векторы длины M попали внутрь рассматриваемого полуинтервала.

При визуализации векторов \vec{v}'_n с таким начальным масштабом, ориентированные графические примитивы, отображающие векторы, не будут выходить за границы своих ячеек, и мешать отображению соседних значений.

Коэффициент масштабирования и максимальная длина векторов отображаются в интерфейсе программы.

2.2. Визуализация векторного поля

При визуализации векторного поля мы будем считать, что длины сторон всех ячеек равны единице, и для отображения использовать стрелки (хотя возможно использование и других ориентированных примитивов). Если отображать все векторы поля, то, из-за большого их количества, может быть затруднено рассмотрение характерных особенностей поля. Поэтому реализована возможность отображения векторов с некоторым шагом, задаваемым пользователем. Первоначальные значения шага и коэффициента масштабирования могут изменяться пользователем в процессе работы.

В проекте реализовано три варианта визуализации векторного поля. В первом случае все отображаемые векторы имеют одинаковую длину, а значение реальной длины вектора отображается цветом. Во втором варианте цвет всех векторов одинаков, а длина вектора задается масштабированной длиной стрелки. Третий вариант является объединением двух предыдущих (т.е. длины и цвета векторов зависят от их исходных размеров). В программе пользователь также может задать минимальный порог для длин отображаемых векторов. Это введено для того, чтобы пользователь мог убрать из рассмотрения области с малыми значениями

векторов скорости и сосредоточиться на изучении структуры основных потоков.

2.3. Цветовая палитра

Для отображения длин векторов с помощью цвета необходимо выбрать цветовую палитру. В качестве палитры в данной работе предлагается использовать термальную цветовую шкалу.

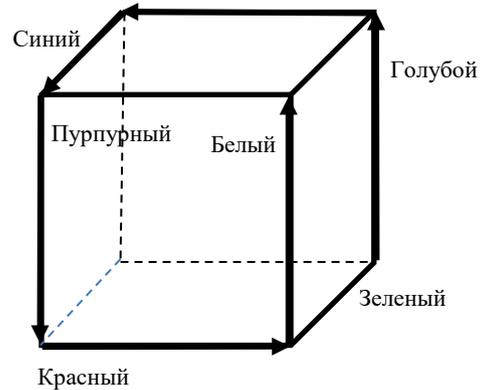


Рис. 1. Обход RGB-куба по термальной шкале.

Термальная палитра может быть представлена как траектория по шести ребрам цветового RGB куба, начиная с зеленого цвета и заканчивая белым (см. рис. 1). Графики изменения значений цветовых компонент RGB показаны на рис. 2, из которого видно, что эти компоненты являются кусочно-линейными функциями.

Разделим полуинтервал $[0, M)$, которому принадлежат длины векторов, на шесть полуинтервалов одинаковой длины $L = M / 6$ и обозначим j -й полуинтервал $[j \cdot L, (j + 1) \cdot L)$ через I_j , $j = 0, \dots, 5$.

Для пояснения алгоритма вычисления кусочно-линейных функций, рассмотрим для примера красную компоненту $R(x)$ (верхний график на рис. 2). Обозначим через $f_{R,j}(x)$ функцию, совпадающую с $R(x)$ на j -м полуинтервале. Из графика на рис. 2 видно, что

$$\begin{aligned} f_{R,0}(x) = f_{R,1}(x) &= 0, \quad f_{R,2}(x) = x / L - 2, \\ f_{R,3}(x) = f_{R,4}(x) &= f_{R,5}(x) = 1. \end{aligned}$$

Для вычисления значений такой функции можно было бы использовать несколько условных операторов *if*, однако в шейдерных программах нежелательно использовать операторы ветвлений и условных переходов, поскольку они снижают общую производительность системы. Вместо использования условных операторов, применим аппарат индикаторных функций

[7, 9] и построим функцию $R(x)$ для вычисления значений красной компоненты. Для j -го полуинтервала определим его индикаторную функцию

$$g_j(x) = \begin{cases} 1, & \text{если } x \in I_j \\ 0 & \text{в прот. сл.} \end{cases}$$

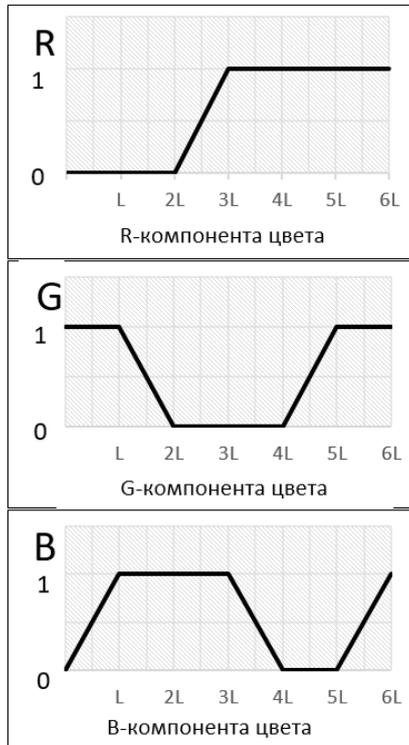


Рис. 2. Изменение значений RGB компонент

Функция $g_j(x)$ может быть вычислена по формуле

$$g_j(x) = \text{float}([x/L] == j).$$

Действительно, если $x \in I_j$, то $j \cdot L \leq x < (j+1) \cdot L$, откуда получаем $j \leq x/L < j+1$ или $[x/L] = j$. Явное указание на преобразование типов нужно, чтобы подчеркнуть, что в функции происходит преобразование значений логического выражения *true-false* в значение переменной в плавающем формате – *0.0* и *1.0*.

Тогда функцию $R(x)$ можно представить в виде суммы произведений функций $f_{R,j}(x)$ и соответствующих полуинтервалам индикаторных функций $g_j(x)$:

$$R(x) = \sum_{j=0}^5 g_j(x) f_{R,j}(x).$$

Аналогичным образом вычисляются значения кусочно-линейных функций зеленой $G(x)$ и синей $B(x)$ компонент.

Таким образом, вместо использования конструкций *if-else*, вычисляется значение выражения. Так как выбор цвета окраски происходит для каждого отображаемого вектора внутри шейдерной программы, выполняющейся на GPU (Graphic Processor Unit), то такой подход оказывается оправдан, так как не приводит к снижению производительности из-за использования условных операторов в цикле обработки данных.

2.4. Построение стрелок

В данной работе визуализация векторов выполнялась в виде стрелок с использованием GPU (Graphic Processor Unit). Построение геометрии каждой стрелки выполнялось при помощи геометрического шейдера. Шейдер написан с использованием языка программирования GLSL [5]. Так как геометрический шейдер имеет особенность, что на его выходе возможны графические примитивы одного типа, то стрелка представляется в виде двух тетраэдров: тела стрелки и ее наконечника (см. рис. 3). Оси этих тетраэдров совпадают с отображаемым вектором.

Рассмотрим произвольный вектор \vec{v} из векторного поля, записанный в n -м элементе массива L . Координаты (i, j, k) его ячейки вычисляется по формулам (1), а координаты начала P_0 этого вектора равны $(i+s, j+s, k+s)$, где s – половина размера ячейки.

Вычислим теперь координаты вершин наконечника стрелки; на рис. 3 они обозначены цифрами. Точка P_1 , задающая острие стрелки равна $P_1 = P_0 + \vec{v}$. Остальные



Рис. 3. Наконечник стрелки из ленты треугольников

точки наконечника лежат в плоскости, перпендикулярной оси вектора. Предполагая, что длина наконечника равна 0.2 длины всей стрелки, получаем, что плоскость пересекает вектор в точке $P'_0 = P_0 + 0.8 \cdot \vec{v}$. Построим два вектора, лежащие в этой плоскости. В качестве одного из векторов можно взять векторное произведение $\vec{v}_1 = [\vec{v}, \vec{z}]$, если только вектор \vec{v}

не коллинеарен оси \vec{z} . Если же имеет место коллинеарность, то тогда вектор \vec{v} не коллинеарен оси \vec{x} и можно взять векторное произведение $\vec{v}_2 = [\vec{v}, \vec{x}]$. Эти случаи можно объединить при вычислении в одной формуле при помощи описанной ранее индикаторной функции:

$$\vec{u}_1 = g \cdot \vec{v}_1 + (1-g) \cdot \vec{v}_2,$$

$$\text{где } g = \begin{cases} 1, & \text{если } |\vec{v}_1| > \varepsilon \\ 0 & \text{в прот. сл.} \end{cases}.$$

Второй вектор $\vec{u}_2 = [\vec{v}_1, \vec{u}_1]$ будет ненулевым, перпендикулярным векторам \vec{v}_1 , \vec{u}_1 и принадлежит рассматриваемой плоскости. Точки P_2 , P_3 и P_4 будут расположены в плоскости, как показано на рис. 4:

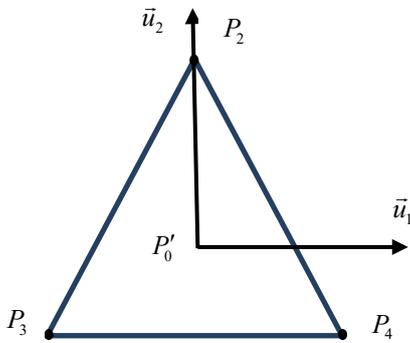


рис. 4. Основание наконечника стрелки

Обозначая длину стороны треугольника через a и, учитывая, что высота равностороннего треугольника равна $\sqrt{3}a/2$ и делится центром треугольника в отношении 1:2, получаем:

$$P_2 = P_0 + \frac{a \cdot \vec{u}_2}{\sqrt{3}},$$

$$P_3 = P_0 - \frac{a}{2} \cdot (\vec{u}_1 + \frac{\vec{u}_2}{\sqrt{3}}),$$

$$P_4 = P_0 + \frac{a}{2} \cdot (\vec{u}_1 - \frac{\vec{u}_2}{\sqrt{3}}).$$

Имея координаты всех вершин, можно построить тетраэдр наконечника стрелки. Одним из эффективных способов является его задание в виде треугольного стрипа (Triangle Strip, [8]). В нем только первый треугольник задается всеми тремя вершинами, а каждый последующий – лишь одной вершиной, которая добавляется к двум последним использованным. Таким образом, можно уменьшить избыточность используемых геометрических данных. Для тетраэдра, например, можно использовать стрип с

порядком точек 2,3,1,4,2 и с вершиной тетраэдра с индексом 1 в центре последовательности. Аналогичным образом строится и тело стрелки.

2.5. Интерфейс программы

После открытия файла (формата NetCDF) с векторным полем, его имя отображается в верхней части окна. Для управления отображением векторного поля в интерфейсе системы (см. рис. 5) можно задать следующие параметры: верхний и нижний порог длины отображаемых векторов, шаг вывода по сетке данных, масштаб длин выводимых векторов, тип отображения и цвет фона.

Как отмечалась выше, задание порогов позволяет визуализировать только те векторы, длина которых попадает в заданный диапазон. Шаг вывода дает возможность выводить только каждый r -й вектор в строке, столбце и слое. На рисунке 6 слева показан результат визуализации всех векторов, а справа – визуализации с шагом 8. Очевидно, что второй вариант позволяет более детально изучить направление векторов в интересующих областях.

Масштабирование длин позволяет пользователю задать такой коэффициент, при котором вектора будут отображаться наиболее наглядно. Результат использования функции масштабирования приведен на рисунке 7.

Если пользователь выбирает однотонный режим отображения, то все векторы рисуются одним цветом, но разной длины (с учетом коэффициента масштабирования). Если выбирается цветовой режим отображения, то все векторы рисуются одинаковой длины, а их реальная длина кодируется цветом, при этом в левой части окна появляется палитра со шкалой, соответствующей данным открытого файла. На палитре показываются границы отображаемого диапазона длин. Поскольку в некоторых режимах цвета отображаемых векторов может быть плохо различимы на текущем фоне, реализована функция изменения его цвета в оттенках серого.

Для подробного изучения векторного поля доступны стандартные интерфейсные возможности управления отображением, такие как приближение / удаление модели, а также поворот и сдвиг ее в двух плоскостях при помощи манипулятора мышь.

С технической точки зрения, программа оформлена как DLL (Dynamic Link Library), подключаемая как один из модулей к программному комплексу “Цифровая модель керна” [10].

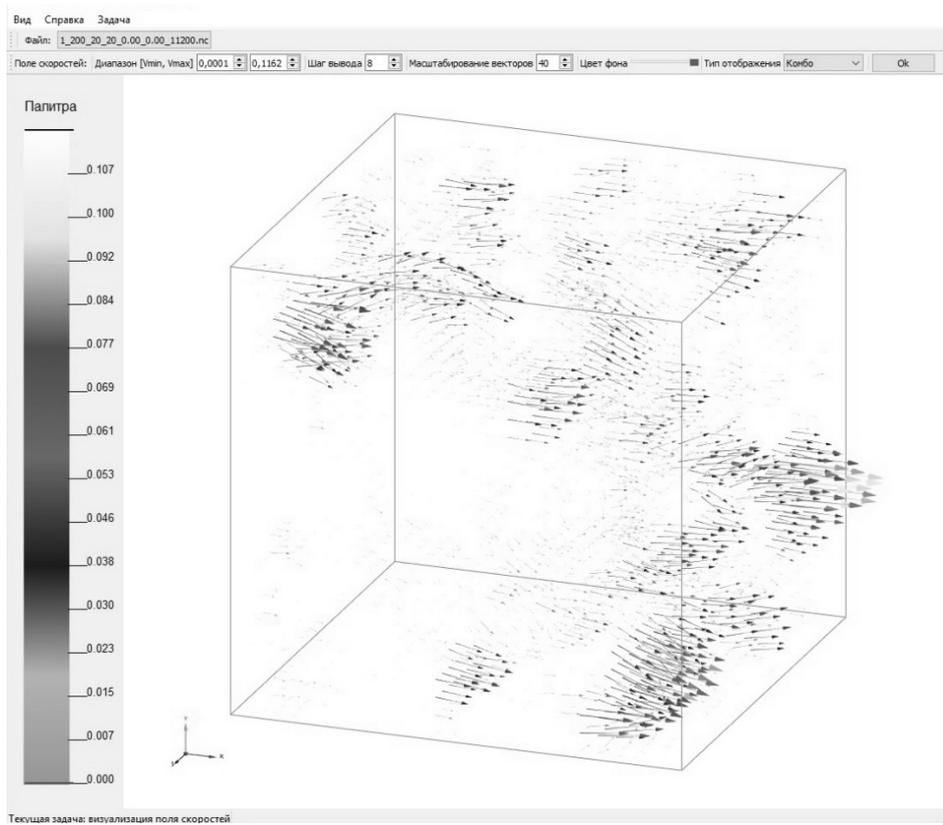


Рис. 5. Интерфейс программы

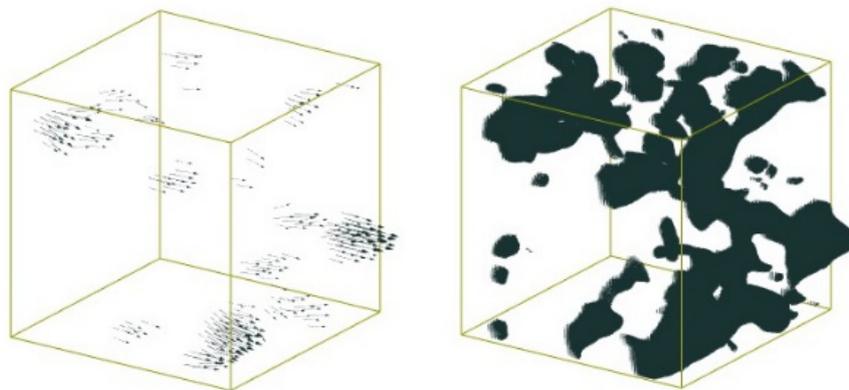
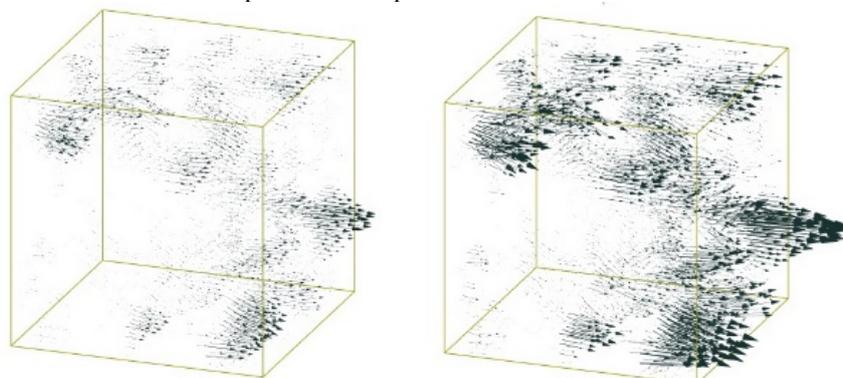
Рис. 6. Отображение векторных данных с шагом $r=8$ и $r=1$ 

Рис. 7. Отображение данных с масштабированием векторов в 20 и 40 раз

3. Заключение

В данной работе описан метод, алгоритмы и результат программной реализации системы визуализации векторных данных для цифровой модели ядра.

Программа активно использует параллельную обработку данных на GPU и технологию программирования шейдеров.

Во время работы над реализацией эффективных параллельных алгоритмов для GPU был разработан и апробирован метод вычисления значений кусочно-линейных

функций при помощи аппарата индикаторных функций. Этот подход к построению алгоритмов параллельной обработки данных позволил отказаться от использования операторов ветвления и условных переходов при создании программ для GPU.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований (ГП 14) по теме (проекту) «34.9. Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации» (0065-2019-0012).

Vector field visualization in the study of core structures

M.V. Mikhaylyuk, I.N. Cherednichenko

Abstract: The paper describes the methods and algorithms for visualization of the digital core model vector fields, given in NetCDF format. They are based on the data processing both on the CPU and GPU. To exclude condition statements which slow down data processing on GPU, we propose to use indicator functions. Interface of the developed software module allows the user to select the display mode of vector data and to flexibly configure visualization options.

Keywords: visualization, vector fields, OpenGL, indicator functions, shaders.

Литература

1. Э. Эйнджел. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. Москва: Издательский дом "Вильямс", 2001.
2. Amira-Avizo Software [Электронный ресурс] // Thermo Fisher Scientific.
URL: <https://www.fei.com/software/avizo/>
3. ParaView [Электронный ресурс], 2019, URL: <https://www.paraview.org>
4. S.H. Ahn. OpenGL Display List. URL: http://www.songho.ca/opengl/gl_displaylist.html
5. John Kessenich, Dave Baldwin, Randy Rost. The OpenGL Shading Language. 2013.
URL: <https://www.khronos.org/registry/OpenGL/specs/gl/GLSLangSpec.4.30.pdf>
6. Network Common Data Form (NetCDF). 2019.
URL: <https://www.unidata.ucar.edu/software/netcdf>
7. Folland G.B. Real Analysis: Modern Techniques and Their Applications. 2nd ed., John Wiley & Sons, Inc., 1999.
8. Graham Sellers, Richard S. Write, Nicholas Haemel. Open GL Superbible: Comprehensive Tutorial and Reference. Addison-Wesley Professional, 2013.
9. А.Н. Колмогоров, С.В. Фомин. Элементы теории функций и функционального анализа. М.: Наука, 2004.
10. П.Ю. Тимохин, М.В. Михайлюк. Технология создания мультизадачной графической оболочки системы визуализации цифровой модели ядра. «Вестник кибернетики», 2018, №. 3 (31), 247-254.

Определение коллизий аппроксимирующих сфер с рельефом местности

Е.В. Страшнов¹, Л.А. Финагин²

¹ФГУ ФНЦ НИИСИ РАН, Москва, Россия, strashnov_evg@mail.ru;

²ФГУ ФНЦ НИИСИ РАН, Москва, Россия, fnagin_la@mail.ru

Аннотация: В работе рассматривается задача определения пересечения аппроксимирующих сфер с рельефом местности, заданным в виде полигональной сетки. Для решения этой задачи предлагается подход, основанный на построении регулярной прямоугольной сетки высот. При таком подходе проверка на пересечение выполняется между аппроксимирующей сферой и ячейками сетки высот, которые в зависимости от типа рельефа являются прямоугольниками или четырехугольниками. Предлагаемые в статье алгоритмы определения коллизий основаны на вычислении проекции центра сферы на ячейку сетки высот и сортировке контактных точек по глубинам проникновения. Апробация этих алгоритмов проводилась в комплексе виртуального окружения на примере моделирования взаимодействия сфер с различным типом рельефа.

Ключевые слова: определение коллизий, аппроксимирующая сфера, рельеф местности, сетка высот, области Вороного, барицентрические координаты, система виртуального окружения.

1. Введение

При моделировании в тренажерных системах и комплексах виртуального окружения объекты могут сталкиваться (пересекаться) друг с другом. Это приводит к необходимости решения задачи определения коллизий (обнаружения столкновений) между этими объектами. Наиболее распространенный подход для решения этой задачи заключается в том, чтобы окружать объекты набором аппроксимирующих контейнеров [1] стандартной формы (сферы, параллелепипеды, цилиндры и т.д.). Тогда определение коллизий двух объектов сводится к определению коллизий аппроксимирующих контейнеров, окружающих эти объекты. Однако ландшафт местности для многих виртуальных сцен имеет сложную форму, что затрудняет применение для него такого подхода. Вместо этого используется представление рельефа местности в виде полигональной сетки, состоящей из множества связанных треугольников. Разработка методов и алгоритмов определения коллизий полигональной сетки с объектами виртуального окружения является важной и актуальной задачей.

В данной работе рассматривается задача определения коллизий рельефа местности с аппроксимирующими сферами, которая состоит в том, чтобы определить сам факт пересечения сферы с рельефом, а также сформировать контактное многообразие о пересечении, которое задается набором контактных точек, нормальными и глубинами проникновения. Для решения этой задачи один

из распространенных подходов заключается в вычислении ближайших точек между сферой и треугольниками полигональной сетки [2], [3], [4]. Тогда сфера пересекается с треугольником, если расстояние между ближайшей точкой и центром сферы меньше ее радиуса. Одна из проблем такого подхода заключается в том, что количество треугольников полигональной сетки может оказаться достаточно большим. Поэтому необходимо каким-то образом отбрасывать из рассмотрения треугольники, с которыми сфера заведомо не пересекается. Другая проблема состоит в том, что получаемый набор контактных точек может оказаться избыточным, что в дальнейшем может привести к нереалистичному движению объектов. Для решения этой проблемы в работе [5] был предложен алгоритм, в котором точки контакта формируются сначала для граней треугольников, затем для их ребер и, наконец, для вершин. Идея алгоритма заключается в том, что он позволяет отбросить избыточные точки контакта, которые образуются при обработке двух соседних треугольников. Однако в общем случае такое решение не является универсальным и получается сложным в реализации.

В данной статье для эффективной реализации алгоритмов определения коллизий аппроксимирующих сфер с рельефом предлагается решение, которое основано на построении для рельефа регулярной прямоугольной сетки высот. В этом случае задача сводится к проверке пересечения аппроксимирующей сферы с ячейками сетки,

которые в зависимости от сложности рельефа представляют собой прямоугольники или четырехугольники. Для определения пересечения сферы с сеткой высот сначала путем проецирования сферы на горизонтальную плоскость определяется множество ячеек, с которыми сфера может потенциально пересекаться. Затем для каждой ячейки из этого множества осуществляется проверка на пересечение, которая заключается в вычислении расстояния между центром сферы и ближайшей к ней точки ячейки. Для определения ближайшей точки треугольника применяется концепция, основанная на разбиении пространства на области Вороного [6] и вычислении точки с помощью барицентрических координат [2]. В случае наличия пересечения сферы с ячейкой, для построения контактного многообразия предлагается подход, в рамках которого осуществляется сортировка контактных точек по глубинам проникновения с применением различных критериев. Разработанные в статье алгоритмы были реализованы в программном комплексе виртуального окружения и апробированы на примере моделирования взаимодействия сферы с различным типом поверхности рельефа, включая горки, холмы и лестницы.

2. Регулярная сетка высот

В системах виртуального окружения рельеф местности создается в системе компьютерного моделирования (например, в системе 3ds Max). Как правило, результатом получается полигональная модель, в которой рельеф задается в виде множества связанных треугольников различной величины и формы.

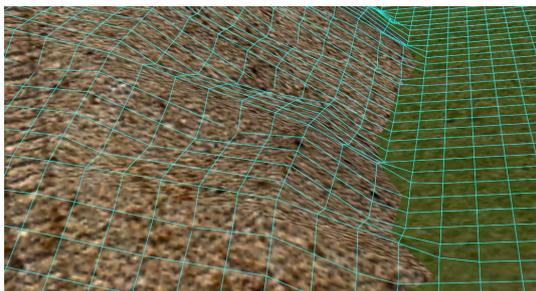


Рис. 1. Регулярная сетка высот

Для эффективной реализации алгоритмов определения коллизий в данной работе предлагается модель, основанная на построении регулярной прямоугольной сетки высот (см. рис. 1). В этом случае плоскость XU разбивается на M горизонтальных и N вертикальных линий, образуя множество ячеек

размером d_x на d_y . Вертикальные и горизонтальные линии сетки определяются с помощью двух индексов i и j , что позволяет вычислять точки сетки как

$$P = (x_0 + jd_x, y_0 + id_y), \quad (1)$$

где точка (x_0, y_0) – левая нижняя точка прямоугольной сетки.

Для каждого узла прямоугольной сетки вычисляется значение высоты z_{ji} . Если участок рельефа местности является ровным, то соответствующая ячейка сетки будет прямоугольником, в противном случае – четырехугольником.

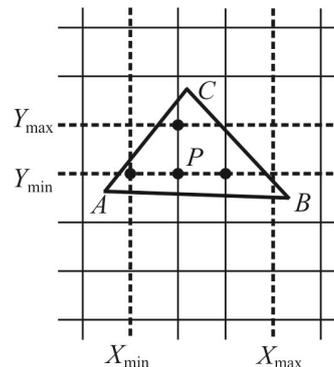


Рис. 2. Проекция треугольника на плоскость

Опишем реализацию построения сетки высот на основе полигональной модели. Идея заключается в том, чтобы определить точки сетки, которые попадают внутрь треугольника, и вычислить в этих точках высоту. Рассмотрим проекцию треугольника ABC , вершины которого упорядочены против часовой стрелки, на плоскость XU (см. рис. 2). Тогда получим прямоугольную область с границами $[X_{\min}, X_{\max}]$ и $[Y_{\min}, Y_{\max}]$, для узлов сетки которой необходимо определить принадлежность треугольнику. Вычисление этих границ осуществляется путем преобразования координат точек треугольника в индексы прямоугольной сетки. Для этого сначала координаты точек треугольника сортируются по возрастанию, что дает значения $[x_{\min}, x_{\max}]$ и $[y_{\min}, y_{\max}]$. Тогда индексы прямоугольной сетки вычисляются следующим образом

$$i_{\min} = \left\lceil \frac{y_{\min} - y_0}{d_y} M \right\rceil, \quad i_{\max} = \left\lfloor \frac{y_{\max} - y_0}{d_y} M \right\rfloor;$$

$$j_{\min} = \left\lceil \frac{x_{\min} - x_0}{d_x} N \right\rceil, \quad j_{\max} = \left\lfloor \frac{x_{\max} - x_0}{d_x} N \right\rfloor,$$

где для левых границ используется округление сверху, а для правых – снизу.

По полученным индексам для границ по формуле (1) определяется множество точек прямоугольной области. Если для проекции точки P на плоскость XY (точка $P_0 = (x, y, 0)$) выполнены условия

$$(\mathbf{P}_0 \mathbf{A} \times \mathbf{P}_0 \mathbf{B})_z \geq 0;$$

$$(\mathbf{P}_0 \mathbf{B} \times \mathbf{P}_0 \mathbf{C})_z \geq 0;$$

$$(\mathbf{P}_0 \mathbf{C} \times \mathbf{P}_0 \mathbf{A})_z \geq 0,$$

то луч, проходящий через точку P попадает внутрь треугольника.

Далее задача заключается в том, чтобы для точки P определить координату z (высоту z_{ji} прямоугольной сетки). Для этого воспользуемся алгоритмом Моллера-Трумбора [7], который предназначен для реализации трассировки лучей с использованием полигональных сеток. Этот алгоритм позволяет вычислить расстояние от начала луча до точки пересечения с треугольником. В данном случае луч, задаваемый началом P_0 и вектором $\mathbf{v} = (0, 0, 1)^T$, пересекает треугольник в точке P (см. рис. 3).

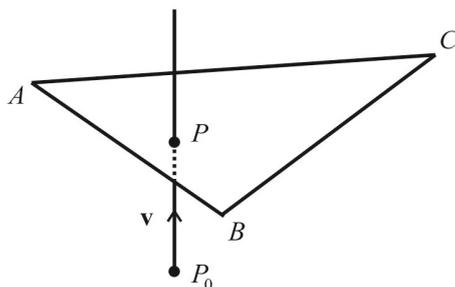


Рис. 3. Пересечение луча и треугольника

С помощью алгоритма Моллера-Трумбора координата z точки P вычисляется следующим образом

$$z = \frac{\mathbf{AC} \cdot (\mathbf{AP}_0 \times \mathbf{AB})}{\mathbf{AB} \cdot (\mathbf{v} \times \mathbf{AC})}.$$

Данное соотношение выводится путем решения системы линейных уравнений относительно параметра луча и барицентрических координат треугольника для точки P .

3. Определение коллизий сферы и сетки высот

Рассматриваемая задача заключается в том, чтобы проверить, пересекается ли сфера с сеткой высот, и в случае пересечения определить необходимую контактную информацию, включая точки контакта, глубины и нормали. Проверка на пересечение осуществляется между сферой и ячейками

сетки высот. На первом этапе определим множество ячеек Ω , с которыми сфера может потенциально пересекаться. Для этого спроецируем сферу с центром в точке P_s и радиусом r на плоскость XY (см. рис. 4). Тогда получим, что множество Ω ограничено прямоугольником, индексы линий границ которого вычисляются как

$$i_{left} = \left\lfloor \frac{P_{sy} - r - y_0}{d_y} M \right\rfloor, \quad i_{right} = \left\lceil \frac{P_{sy} + r - y_0}{d_y} M \right\rceil;$$

$$j_{left} = \left\lfloor \frac{P_{sx} - r - x_0}{d_x} N \right\rfloor, \quad j_{right} = \left\lceil \frac{P_{sx} + r - x_0}{d_x} N \right\rceil,$$

где (P_{sx}, P_{sy}) – координаты центра сферы P_s в плоскости XY . Здесь для вычисления левых границ индексов прямоугольника применяется округление снизу, а для правых – сверху.

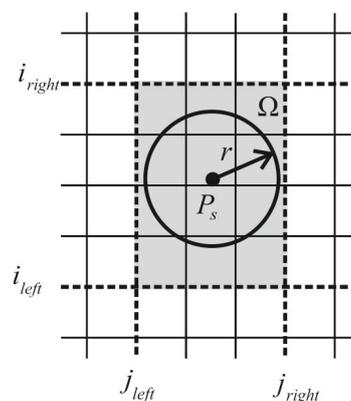


Рис. 4. Проекция сферы на плоскость

Для полученного множества Ω ячеек сетки высот необходимо определить их пересечение со сферой. Предлагаемое решение состоит в том, чтобы сначала выполнить быстрые тесты на пересечение. Для этого для каждого узла полученного множества сетки вычисляются минимальное z_{min} и максимальное z_{max} значение высоты. Если координата z нижней точки сферы больше, чем z_{max} , т.е. $P_{sz} - r > z_{max}$, то сфера и сетка высот не пересекаются.

В противном случае мы сначала проверяем, является ли участок поверхности горизонтальной плоскостью. Это соответствует условию $|z_{max} - z_{min}| \leq \varepsilon_z$, где ε_z – задаваемая точность. Если это условие выполнено, то нижняя точка сферы Q является единственной точкой контакта и вычисляется как

$$Q = P_s - r\mathbf{n},$$

где $\mathbf{n} = (0, 0, 1)^T$ – нормаль к поверхности.

Глубина проникновения сферы внутрь плоскости определяется как разность высоты с координатой z нижней точки сферы:

$$h = z_{\max} - P_{sz} + r.$$

Если участок поверхности рельефа не является горизонтальной плоскостью, то требуется проверить пересечение сферы с ячейками сетки высот, которые могут быть, как прямоугольниками, так и четырехугольниками, в свою очередь разбиваемые на два треугольника. В этом случае задача определения коллизий сферы с рельефом местности сводится к поиску ближайшей точки прямоугольника или треугольника. Тогда если расстояние между ближайшей точкой и центром сферы меньше радиуса, то сфера и ячейка сетки высот пересекаются. Далее опишем, как вычислять ближайшую точку в зависимости от типа ячейки, а также приведем алгоритмы сортировки контактных точек и определения коллизий сферы с сеткой высот.

3.1. Вычисление ближайшей точки прямоугольника

Пусть ячейка $ABCD$ сетки высот является прямоугольником. Тогда задача заключается в том, чтобы определить точку Q прямоугольника, которая является ближайшей к центру сферы P_s . Каждая точка прямоугольника задается в параметрическом виде

$$Q = A + u \cdot \mathbf{AB} + v \cdot \mathbf{AC}, \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1. \quad (2)$$

Параметры u и v для точки P_s вычисляются путем ее проецирования на отрезки AB и AC :

$$u = \begin{cases} 0, & \mathbf{AP} \cdot \mathbf{AB} \leq 0 \\ \frac{\mathbf{AP} \cdot \mathbf{AB}}{\|\mathbf{AB}\|^2}, & 0 < \mathbf{AP} \cdot \mathbf{AB} < \|\mathbf{AB}\|^2; \\ 1, & \mathbf{AP} \cdot \mathbf{AB} \geq \|\mathbf{AB}\|^2 \end{cases}$$

$$v = \begin{cases} 0, & \mathbf{AP} \cdot \mathbf{AC} \leq 0 \\ \frac{\mathbf{AP} \cdot \mathbf{AC}}{\|\mathbf{AC}\|^2}, & 0 < \mathbf{AP} \cdot \mathbf{AC} < \|\mathbf{AC}\|^2; \\ 1, & \mathbf{AP} \cdot \mathbf{AC} \geq \|\mathbf{AC}\|^2 \end{cases}$$

3.2. Вычисление ближайшей точки треугольника

Рассмотрим треугольник ABC , для которого требуется определить точку Q , ближайшую к центру сферы P_s . Для этого воспользуемся концепцией разбиения

треугольником пространства на области Вороного [6]. Идея заключается в том, что в зависимости от того в какую область попадает точка P_s ближайшей точкой будет либо вершина треугольника, либо находится на стороне треугольника, либо внутри треугольника. На рис. 5 показано, что точка P_s может располагаться в одной из семи областей. Опишем вычисление точки Q для каждой из этих областей более подробно.

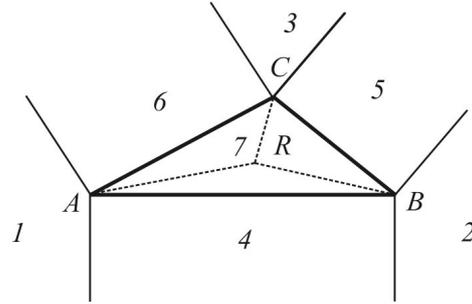


Рис. 5. Области Вороного для треугольника

Область 1 определяется следующими условиями

$$\mathbf{AP}_s \cdot \mathbf{AB} \leq 0, \quad \mathbf{AP}_s \cdot \mathbf{AC} \leq 0.$$

Если эти условия выполнены, то ближайшей точкой треугольника является точка A , то есть $Q = A$.

Аналогичным образом определяются области 2 и 3, для которых ближайшими точками являются B и C , соответственно.

Точка P_s лежит внутри области 4, если для нее выполнены следующие условия

$$S_{RAB} \leq 0, \quad \mathbf{AP}_s \cdot \mathbf{AB} \geq 0, \quad \mathbf{BP}_s \cdot \mathbf{BA} \geq 0. \quad (3)$$

где R – проекция точки P_s на плоскость треугольника, S_{RAB} – площадь треугольника RAB .

Площадь треугольника RAB вычисляется следующим образом

$$S_{RAB} = \frac{1}{2} (\mathbf{RA} \times \mathbf{RB}) \cdot \mathbf{n},$$

где $\mathbf{n} = (\mathbf{AB} \times \mathbf{AC}) / \|\mathbf{AB} \times \mathbf{AC}\|$ – нормаль к плоскости треугольника.

Так как R является проекцией точки P_s на плоскость треугольника, то можно показать [2], что при вычислении площади S_{RAB} точку R можно заменить на точку P_s . Тогда первое условие в (3) эквивалентно неравенству

$$S'_{RAB} = (\mathbf{P}_s \mathbf{A} \times \mathbf{P}_s \mathbf{B}) \cdot (\mathbf{AB} \times \mathbf{AC}) \leq 0, \quad (4)$$

где $S'_{RAB} = 4S_{RAB}S_{ABC}$.

Если условия (3) выполнены, то ближайшая точка Q вычисляется путем проецирования точки P_s на отрезок AB :

$$Q = A + t \cdot \mathbf{AB}, \quad t = \frac{\mathbf{AP}_s \cdot \mathbf{AB}}{\|\mathbf{AB}\|^2}. \quad (5)$$

Аналогично (3) задаются условия для проверки расположения точки P_s в областях 5 и 6. В случае выполнения этих условий точка Q будет лежать на отрезках BC и AC , соответственно, и вычисляться аналогично соотношению (5).

В противном случае точка P_s будет находиться внутри треугольника в области 7. Ближайшую точку Q будем вычислять с помощью барицентрических координат [2] в виде

$$Q = uA + vB + wC, \quad u + v + w = 1.$$

После преобразований получим, что точка Q определяется как

$$Q = A + v \cdot \mathbf{AB} + w \cdot \mathbf{AC}.$$

Барицентрические координаты v и w вычисляются как отношения площадей:

$$v = S'_{RBC} / S'_{ABC}, \quad w = S'_{RCA} / S'_{ABC},$$

где $S'_{RBC} = 4S_{RBC}S_{ABC} = (\mathbf{P}_s \mathbf{B} \times \mathbf{P}_s \mathbf{C}) \cdot (\mathbf{AB} \times \mathbf{AC})$,
 $S'_{RCA} = 4S_{RCA}S_{ABC} = (\mathbf{P}_s \mathbf{C} \times \mathbf{P}_s \mathbf{A}) \cdot (\mathbf{AB} \times \mathbf{AC})$,
 $S'_{ABC} = S'_{RAB} + S'_{RBC} + S'_{RCA}$, а S'_{RAB} вычисляется по формуле (4).

При выводе соотношения для координаты v было использовано выражение

$$\begin{aligned} \frac{S'_{RBC}}{S'_{ABC}} &= \frac{S'_{RBC}}{S'_{RAB} + S'_{RBC} + S'_{RCA}} = \\ &= \frac{4S_{RBC}S_{ABC}}{4(S_{RAB} + S_{RBC} + S_{RCA})S_{ABC}} = \\ &= \frac{S_{RBC}}{S_{RAB} + S_{RBC} + S_{RCA}} = \frac{S_{RBC}}{S_{ABC}}. \end{aligned}$$

Аналогичное равенство справедливо и для соотношения, с помощью которого вычисляется координата w .

3.3. Сортировка контактных точек

Если сфера и ячейка сетки высот пересекаются, то есть выполнено условие $\|\mathbf{P}_s \mathbf{Q}\| \leq r$, тогда точка Q является потенциальной точкой контакта сферы с рельефом. Однако, эта точка может оказаться избыточной, приводя к тому, что моделирование движения тел становится нереалистичным (например, из-за лишней точки контакта катящаяся сфера сильно теряет свою скорость или останавливается, когда должна продолжать движение). Для решения этой проблемы здесь рассматривается задача сортировки и отбрасывания лишних точек, согласно заданному набору критериев.

Напомним, что контактная информация (контактное многообразие) о пересечении двух объектов формируется из точек контакта, нормалей к поверхностям и глубин проникновения. В данном случае точкой контакта является ближайшая точка ячейки Q . Нормаль к поверхности вычисляется, как нормаль к ячейке, и для прямоугольника $ABCD$ и треугольника ABC определяется следующим образом:

$$\mathbf{n} = \frac{\mathbf{AB} \times \mathbf{AC}}{\|\mathbf{AB} \times \mathbf{AC}\|}. \quad (6)$$

В свою очередь глубина проникновения h для ячейки вычисляется как

$$h = r - \|\mathbf{P}_s \mathbf{Q}\|. \quad (7)$$

Предлагаемое решение для построения контактного многообразия сферы с рельефом местности заключается в том, чтобы выбирать точки с максимальными глубинами проникновения. При этом для сортировки используются три критерия для удаления точки из контактного многообразия: точки имеют общую нормаль, расположены на небольшом расстоянии друг от друга и их количество превышает заданное число K .

Пусть \mathbf{n}_1 и \mathbf{n}_2 – нормали двух точек контакта Q_1 и Q_2 . Тогда близость этих нормалей означает выполнение условия

$$\mathbf{n}_1 \cdot \mathbf{n}_2 \geq \varepsilon_\alpha,$$

где $\varepsilon_\alpha = \cos(\Delta\alpha)$, $\Delta\alpha$ – угол между нормальями, задающий их близость.

Критерий малости расстояния между двумя точками Q_1 и Q_2 заключается в выполнении условия

$$\|\mathbf{Q}_1 \mathbf{Q}_2\| \leq \varepsilon_L,$$

где ε_L – задаваемая точность для расстояния.

Если точки имеют разные нормали и удалены друг от друга, то среди них выбираются точки с максимальными глубинами проникновения, число которых не превышает заданного числа K .

Алгоритм добавления контактной точки Q с нормалью \mathbf{n} и глубиной h в контактное многообразие будет следующим:

1. $isNew = true$ – инициализация флага добавления новой точки;
2. Цикл $s = 1, \dots, S$ по всем точкам Q_s с нормальями \mathbf{n}_s и глубинами h_s из контактного многообразия:

Если $\mathbf{n} \cdot \mathbf{n}_s \geq \varepsilon_\alpha$ или $\|\mathbf{Q} \mathbf{Q}_s\| \leq \varepsilon_L$, то

$isNew = false$;

Если $h_s < h$, то заменяем точку:

$$Q_s = Q, \quad \mathbf{n}_s = \mathbf{n} \text{ и } h_s = h.$$

3. Если $isNew == true$, то
 если $S == K$ (количество точек максимальное), то
 $h_{min} = h_i, i = 1$;
 Цикл $s = 2, \dots, S$:
 если $h_s < h_{min}$, то $h_{min} = h_s$ и $i = s$;
 Если $h_{min} < h$, то заменяем точку:
 $Q_i = Q, \mathbf{n}_i = \mathbf{n}$ и $h_i = h$.
- в противном случае
 добавляем точку Q в контактное многообразие;
 $S = S + 1$.

3.4. Алгоритм определения коллизий сферы с сеткой высот

Описанные этапы определения коллизий аппроксимирующей сферы с сеткой высот объединяются в следующий алгоритм:

1. Находим множество ячеек Ω , с которыми сфера может потенциально пересекаться. Для этого множества вычисляем границы индексов $[i_{left}, i_{right}]$ и $[j_{left}, j_{right}]$.
2. Выполняем цикл по всем узлам сетки множества ячеек Ω . Вычисляем z_{min} и z_{max} .

3. Если выполнено условие $P_{sz} - r > z_{max}$, то сфера и сетка высот не пересекаются.

4. В противном случае если выполнено условие $|z_{max} - z_{min}| \leq \varepsilon_z$, то сфера пересекается с рельефом местности в точке $Q = P_s - r\mathbf{n}$ с нормалью $\mathbf{n} = (0, 0, 1)^T$ и глубиной проникновения $h = z_{max} - P_{sz} + r$.

5. В противном случае осуществляем цикл по всем ячейкам множества Ω :

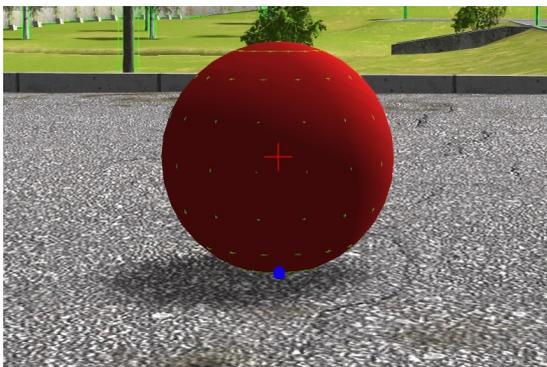
Получаем ячейку $ABCD$;

Если $\mathbf{AB} \cdot \mathbf{AC} \leq \varepsilon$ и $\mathbf{DC} \cdot \mathbf{DB} \leq \varepsilon$, то ячейка является прямоугольником и выполняем следующее:

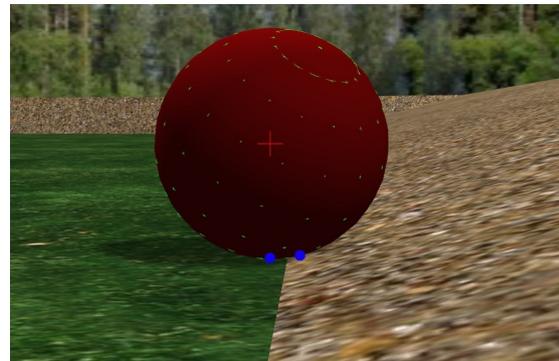
- находим ближайшую точку Q прямоугольника $ABCD$ по формуле (2);

- если $\|\mathbf{P}_s \mathbf{Q}\| \leq r$, то применяем алгоритм добавления контактной точки Q с нормалью и глубиной, которые вычисляются по формулам (6) и (7).

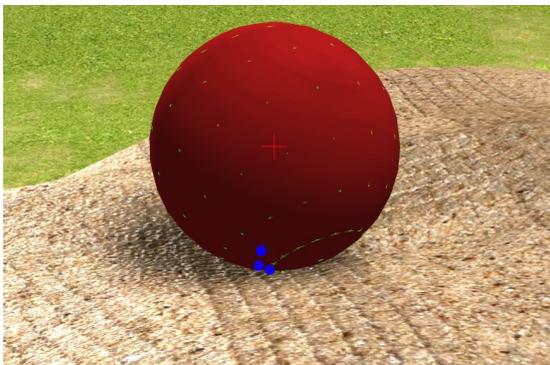
в противном случае ячейка является четырехугольником, которую разбиваем на треугольники ABC и DCB :



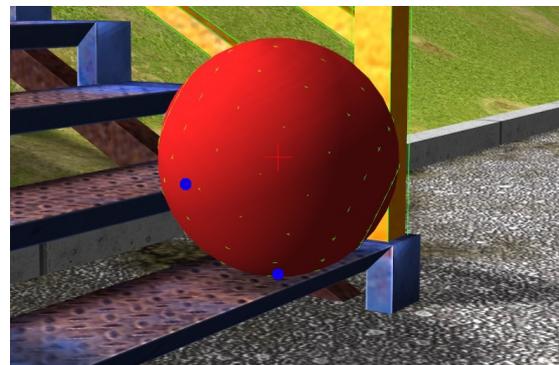
а)



б)



в)



г)

Рис. 6. Моделирование определения коллизий сферы с рельефом различного типа

- находим ближайшую точку Q треугольника ABC ;
- если $\|P_s Q\| \leq r$, то применяем алгоритм добавления контактной точки Q с нормалью и глубиной, которые вычисляются по формулам (6) и (7);
- аналогичные действия выполняем для треугольника DCB (вычисляем точку Q и в случае пересечения этого треугольника со сферой добавляем ее в контактное многообразие).

4. Результаты моделирования

Предлагаемые в статье алгоритмы определения коллизий аппроксимирующей сферы с сеткой высот были реализованы в виде программных модулей в комплексе виртуального окружения, разработанного в ФГУ ФНЦ НИИСИ РАН. В этом программном комплексе осуществляется моделирование динамики объектов в виртуальных сценах, размер местности которых не превышает 500 метров. Поэтому сетка высот рельефа местности в этих сценах создается из ячеек размером $d_x = d_y = 4$ см. Для реализации алгоритмов определения коллизий сферы с рельефом были выбраны следующие параметры моделирования: $K = 4$, $\varepsilon = 10^{-6}$, $\varepsilon_z = 0.01$ см, $\varepsilon_\alpha = \cos(2.5^\circ) \approx 0.999$, $\varepsilon_L = 2$ см. Моделирование проводилось на примере

контакта сферы с различными участками поверхности рельефа. Получаемая контактная информация о пересечении виртуальных объектов была задействована для разрешения коллизий, которое реализуется с помощью ранее разработанного метода последовательных импульсов [8]. На рис. 6 показано четыре случая контакта сферы с поверхностями, заданными в виде сетки высот: а) – на ровном асфальте, б) – на горке, в) – на холме, г) – на лестнице. Результаты показывают, что разработанные алгоритмы позволяют успешно определять пересечение сферы с рельефом и вычислять минимально необходимое количество точек контакта сферы с поверхностью (на рис. 6 эти точки изображены синим цветом).

Также для апробации предлагаемых в статье решений было проведено моделирование взаимодействия сферы и колеса виртуальной модели мобильного робота КРТ-100МП. При этом для определения коллизий колеса со сферой был применен ранее разработанный алгоритм определения коллизий аппроксимирующего цилиндра со сферой [9]. На рис. 7 показано положение мобильного робота КРТ-100МП, когда при заезде на горку его колесо контактирует со сферой. Из рисунка видно, что предложенные в статье алгоритмы позволяют успешно определять пересечение сферы с виртуальными объектами в сложной динамической ситуации.



Рис. 7. Моделирование взаимодействия колесного робота КРТ-100 МП со сферой

5. Заключение

В работе были разработаны эффективные алгоритмы для определения коллизий аппроксимирующей сферы с рельефом местности. Предложенный подход, основанный на модели построения сетки высот для рельефа, имеет преимущество по сравнению с полигональной моделью. С одной стороны, он позволяет существенно сократить количество ячеек, для которых осуществляется проверка на пересечение, а с другой – ускорить вычисления для ровных участков поверхностей, ячейки которых представляют

собой прямоугольники. В свою очередь разработанный алгоритм сортировки контактных точек по глубинам проникновения показал свою адекватность для решения задачи вычисления оптимального набора контактных точек сферы для различных участков рельефа поверхности. В дальнейшем планируется разработать алгоритмы для определения коллизий аппроксимирующих параллелепипедов и цилиндров с рельефом местности.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00387.

Collision detection of bounding spheres with terrain

E.V. Strashnov, L.A. Finagin

Abstract: This paper considers the task of intersection detection of bounding spheres with a terrain specified in the form of a polygonal mesh. To solve this task, an approach based on the construction of a regular rectangular height grid is proposed. With this approach, an intersection verification is performed between the approximating sphere and the height grid cells, which, depending on the type of relief, are rectangles or quadrangles. The collision detection algorithms proposed in the article are based on calculating the projection of the sphere center onto the cell of the height grid and sorting the contact points by penetration depth. The approbation of these algorithms was carried out in the virtual environments complex using the example of interaction simulation for spheres with a different type of relief.

Keywords: collision detection, bounding sphere, terrain, height grid, Voronoi diagram, barycentric coordinate system, virtual environment system.

Литература

1. Bounding volume. Wikipedia. The Free Encyclopedia.
URL: https://en.wikipedia.org/wiki/Bounding_volume (дата обращения: 14.10.2019).
2. C. Ericson. Real-time collision detection. Morgan Kaufmann, 2004.
3. G. van den Bergen. Collision detection in interactive 3D computer animation. PhD dissertation, Eindhoven University of Technology, 1999.
4. P.J. Schneider and D.H. Eberly. Geometric tools for computer graphics. Morgan Kaufmann, 2003.
5. P. Terdiman. Contact generation for meshes. URL: www.codercorner.com/MeshContacts.pdf (дата обращения: 14.10.2019).
6. F. Aurenhammer. Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure, «ACM Computing Surveys», V. 23 (1991), 345-405.
7. T. Möller, and B. Trumbore. Fast, minimum storage ray-triangle intersection, «In Journal of graphics tools», V. 2 (1997), No. 1, 21–28.
8. А.М. Трушин. Обработка коллизий виртуальных объектов с помощью метода последовательных импульсов. «Труды НИИСИ РАН», Т. 4 (2014), № 2, 95-105.
9. Е.В. Страшнов, Д.В. Омельченко. Определение коллизий аппроксимирующих цилиндров со сферами и плоскостью. «Труды НИИСИ РАН», Т. 8 (2018), № 2, 92-97.

Настройка и оптимизация системы ввода-вывода в среде виртуализации GNU Linux/QEMU/KVM/Libvirt

А.Б. Бетелин¹, И.Б. Егорычев², А.А. Прилипко³, Г.А. Прилипко⁴,
С.Г. Романюк⁵, Д.В. Самборский⁶

ФГУ ФНЦ НИИСИ РАН, Москва, Россия

E-mails: ¹ab@niisi.msk.ru, ²egorychev@niisi.msk.ru, ³aaprilipko@niisi.msk.ru, ⁴prilipko@niisi.msk.ru,
⁵sgrom@niisi.ras.ru, ⁶samborsky_d@fastmail.com

Аннотация: В данной статье выполнен обзор и тестирование различных режимов работы систем хранения данных в среде виртуализации с открытым исходным кодом QEMU/KVM/Libvirt на базе операционных систем (ОС) GNU Linux. Измерение и анализ производительности ввода-вывода в разных программных окружениях (виртуальные ОС Linux и Windows, реальная ОС Linux) и на разных носителях (HDD, SSD, RAM-диски) позволили выработать практические рекомендации по использованию и настройке виртуальных ОС для оптимальной эксплуатации различных прикладных программных систем.

Ключевые слова: виртуализация, ввод-вывод данных, Libvirt, QEMU, KVM, Linux, virtio, SPDK, HDD, SSD, NVMe, RAM

Введение

Надежное долговременное хранение данных, устойчивое к отказам оборудования и перебоям электропитания, является важнейшим требованием, предъявляемым к программно-аппаратным комплексам. Среда виртуализации ОС имеет дополнительную свободу выбора системы хранения данных и режимов ее работы. Часто виртуальные ОС используются для продления эксплуатации устаревающих прикладных программ, которые не могут быть непосредственно адаптированы к новому оборудованию и общесистемному программному обеспечению (ПО). В этом случае такие программы продолжают работать в среде виртуализации, причем с более высокой скоростью и надежностью (по сравнению со старым аппаратно-программным окружением).

Характеристики производительности систем хранения данных

Производительность систем хранения данных характеризуется: максимальной пропускной способностью (bandwidth), общим количеством единичных операций в секунду (IOPS) и временем подтверждения единичной операции (latency). Эти показатели не всегда стабильны, поэтому если есть большой разброс измеренных

значений, то нужно обращать внимание на их распределение (например, в случае latency – на среднее и медианное значения, 99-й перцентиль). Кроме того, оказывается, что эти три величины имеют широкий диапазон возможных значений даже при измерении в одном и том же аппаратном окружении, если применяются разные режимы ввода-вывода и системные настройки. Важность того или иного показателя (bandwidth / IOPS / latency) также зависит от режима работы прикладных программ. Можно описать как минимум четыре типичных сценария:

- *Мультимедийный или архивный сервер.* Последовательное чтение/запись больших объемов данных с подтверждением записи в конце операции либо после передачи больших блоков. Это обычно операции копирования больших файлов или операции потоковой передачи мультимедийных или иных данных. Производительность в этом случае определяется пропускной способностью системы хранения данных (bandwidth), а скорость единичных операций (IOPS) и задержки (latency) имеют гораздо меньшее значение.
- *Файл-сервер общего назначения.* Асинхронное чтение и запись малых порций данных в произвольном порядке с периодическим опросом или ожиданием завершения запрошенных операций. Такие сценарии характерны для сетевого файл-

сервера. В этом режиме оказывается важно общее количество выполняемых операций, IOPS (без требования их немедленного подтверждения), хотя к файл-серверу могут также предъявляться требования минимальной bandwidth и максимальной latency.

- *Сервер баз данных.* Асинхронное чтение и запись малых порций данных в произвольном порядке с частыми синхронизациями для фиксирования согласованного состояния моделей данных. Такую нагрузку порождают серверы управления баз данных (СУБД), которые для завершения транзакций должны получать подтверждение выданных ранее запросов на запись. Параллельное исполнение нескольких транзакций, когда между ними нет зависимостей или они могут быть решены на стадии финализации, позволяет создать

высокую нагрузку с помощью асинхронных запросов. Поэтому для СУБД важны два показателя: IOPS и latency.

- *Однопоточковые прикладные программы.* Операции чтения и записи в произвольном порядке с ожиданием их завершения. Вызовы синхронизации записи могут быть либо периодическими, либо сопровождать каждую операцию записи. Такую нагрузку обычно создают старые однопоточковые приложения, не использующие библиотеки асинхронного ввода-вывода. При неблагоприятном стечении обстоятельств скорость работы таких приложений ограничивается исключительно задержкой выполнения единичной операции (latency).

С другой стороны, разные типы устройств имеют разные уровни производительности, см. табл. 1.

Таблица 1. Производительность основных типов накопителей данных (для операций записи)

Тип носителя	Bandwidth, Mb/s	IOPS	Latency
HDD	40 .. 130	55 .. 200	4 .. 15 ms
SSD SATA/SAS	150 .. 500	2000 .. 90000	120 .. 500 μ s
SSD NVMe	150 .. 2000	2000 .. 500000	10 .. 500 μ s

В датацентрах широко применяется объединение набора накопителей одного или разных типов в массивы для повышения надежности хранения данных и скорости доступа к ним. Тем не менее этот подход не гарантирует увеличения сразу всех основных характеристик (труднее всего сохранить latency, то есть не потерять скорость выполнения единичной операции чтения или записи по сравнению с прямым подключением).

Архитектура подсистем ввода-вывода

Среда виртуализации операционных систем QEMU/KVM/Libvirt [1,2,3] поддерживает разные способы подключения устройств хранения данных. Множество допустимых конфигураций велико и может быть разделено по нескольким критериям. Например, с точки зрения виртуальной ОС дисковый накопитель может быть:

- виртуальным устройством, для которого есть стандартные драйверы (устройство на шине данных IDE, SATA, SCSI, NVMe);
- реальным PCI-устройством (накопитель NVMe или контроллер SCSI), доступ к которому гипервизор делегировал виртуальной ОС;
- Virtio-устройством со специальным драйвером, поставляемым вместе с QEMU.

С точки зрения гипервизора QEMU передаваемое виртуальной ОС устройство может быть:

- файлом виртуального диска в одном из поддерживаемых форматов (RAW, QCOW2, VMDK, VHD);
- блочным устройством хоста виртуализации (устройство дискового накопителя или один из его разделов, LVM-том, сетевое устройство iSCSI, NBD, или RBD);

- PCI-устройством, делегируемым виртуальной ОС.

По способу использования физического устройства накопителя данных:

- устройство данных может использоваться монополюсно (целиком передано виртуальной ОС) или совместно (например, поделено на разделы);
- работа с устройством может выполняться из ядра системы гипервизора либо быть передана на уровень процессов, в userspace. Вторым вариантом реализуется передача управления устройством непосредственно виртуальной ОС или отдельному процессу ввода-вывода по экспериментальным технологиям Vhost и Intel SPDK (см. ниже).

Не все сочетания перечисленных вариантов допустимы. Кроме того, выбор конфигурации часто является результатом компромисса между требованиями к производительности и гибкости настройки. Если не рассматривать случаи требования поддержки сильно устаревших версий ОС, когда невозможно использование новых драйверов и шин данных, то остаются две стратегии выбора конфигурации подсистемы ввода-вывода в среде QEMU/KVM/Libvirt:

- Достижение максимальной производительности с сохранением гибкости настройки и простоты эксплуатации. Например, драйверы семейства Virtio обеспечивают наиболее высокую скорость доступа к виртуальным дискам при условии сохранения всех функций блочных устройств QEMU (включающих создание моментальных снимков состояния виртуальной ОС, с последующим архивированием дисковых образов [4], возможность безостановочной миграции ОС между хостами виртуализации и др.).
- Применение специальных программных решений для получения максимальной скорости доступа к накопителям данных. Это может быть делегирование прямого доступа к устройству или использование технологий Vhost и Intel SPDK.

Описание методики тестирования

В статье рассматривается тестирование различных конфигураций устройств хранения данных и вариантов их настройки в ряде операционных систем как с использованием технологий виртуализации,

так и без них. При этом выбирались такие сочетания, в которых скорость обмена данными предполагалась максимальной.

Для тестирования скорости доступа к накопителям данных была использована утилита "Flexible Input/Output tester" (FIO) [10], которая работает в Windows, Linux, и Unix-подобных ОС и способна создавать разные нагрузки на систему ввода-вывода и измерять все необходимые показатели (bandwidth, IOPS и распределение latency).

Программа FIO запускалась в виртуальных ОС двух типов – Linux Centos 7 и Windows 2012R2, а также в системе хоста виртуализации, работающим под управлением Linux Centos 7.

Аппаратные конфигурации тестовых стендов

Стенд 1, сервер: 2x Intel Xeon CPU E5-2643v2 @3.50GHz, 16x16Gb DDR3 PC3-12800, 2x480Gb SSD Intel DC серии S3500, 2x4Tb HDD HGST Ultrastar 7K6000.

Стенд 2, персональная рабочая станция: Intel Core i5-3330 CPU @3.00GHz, 4x8Gb DDR3 PC3-10600, Intel Optane 900P 270Gb.

Конфигурация используемых дисков

На стенде 1 файлы тестируемых виртуальных дисков располагались на устройствах HDD и SSD. Была использована следующая цепочка технологий: файл виртуального диска для QEMU в RAW-формате / файловая система ext4 / менеджер логических томов LVM / программный массив RAID1, состоящий из двух идентичных устройств.

Кроме того, для сравнения производительности тестировались виртуальные накопители данных, расположенные в оперативной памяти (так называемые RAM-диски). Эти диски были созданы в среде системы гипервизора с помощью встроенной файловой системы tmpfs и устройств /dev/ram*, а также внутри виртуальной ОС Windows несколькими сторонними программами менеджеров RAM-дисков.

На стенде 2 тестировался доступ ко всему устройству NVMe SSD, то есть без использования уровня файловой системы и менеджера логических томов.

Настройки кэширования виртуальных дисков QEMU

Драйвер виртуальных дисков QEMU имеет параметр `cache`, определяющий режим, в котором работает QEMU с файлом или устройством виртуального диска. Четыре возможных значения этого параметра соответствуют четырем комбинациям флагов `O_DIRECT` и `O_DSYNC` системного вызова открытия файла: `cache = none` – `O_DIRECT`; `cache = writethrough` – `O_DSYNC`; `cache = directsync` – `O_DIRECT` и `O_DSYNC`; `cache = writeback` – без флагов. Как следует из стандарта POSIX, опция `O_DIRECT` запрещает кэширование данных в общесистемном файловом кэше, а опция `O_DSYNC` указывает на необходимость ожидания подтверждения каждой операции записи данных. Еще один режим включается значением `cache = unsafe` – он работает так же, как `cache = writeback`, но дополнительно игнорирует команды синхронизации данных со стороны виртуальной ОС. Тестирование выполнялось только для режимов `none`, `writethrough` и `directsync`, поскольку остальные режимы, `writeback` и `unsafe`, допускают значительную потерю данных в случае аварийного выключения аппаратуры хоста виртуализации.

Варианты состояний кэша виртуальных дисков

Если виртуальный диск использует файловый кэш (режим `cache=writethrough`), то с помощью утилиты `vmtouch` [9] можно наблюдать за статистикой кэширования данных виртуального диска, а также загружать данные в кэш или, наоборот, удалять страницы данных из кэша.

Для оценки влияния состояния дискового кэша на выполнение операции чтения тестирование производилось в следующих режимах:

- «холодный» кэш. В этом режиме перед началом теста выполнялась команда сброса кэша: `vmtouch -e`;
- «горячий» кэш. В этом режиме перед началом теста выполнялась команда полного зачитывания используемого файла в кэш и поддержания в нем: `vmtouch -d -l`.

Режимы тестирования FIO

Тестовая программа FIO также может использовать прямой или буферизованный ввод-вывод с синхронизацией или без (опции `O_DIRECT`, `O_DSYNC`). Для этого служат параметры теста `direct=[0|1]`, `fdatasync=[0|1]`.

Программа FIO может использовать различные механизмы ввода-вывода, в настоящей работе применялись:

- **psync**. Механизм, использующий синхронные вызовы `pread` и `pwrite`.
- **libaio**. Данный режим задействует библиотеку асинхронного ввода-вывода `libaio` в системе Linux. Это самый быстрый механизм асинхронного ввода-вывода в системе Linux (быстрее, чем `posixaio`).
- **windowsaio**. Представляет собой вариант асинхронного ввода-вывода, который включен по умолчанию в операционных системах семейства Windows.
- **mmap**. Этот вариант подразумевает отображение файла в памяти при помощи системного вызова `mmap`.

Вызов `mmap` создает новое отображение в виртуальном адресном пространстве вызывающего процесса, после которого каждое копирование данных в/из этой области памяти означает операции отложенной записи и чтения данных файла, соответственно. Для синхронизации записанных данных используется системный вызов `msync`. Понятие прямого ввода-вывода (`direct=1`) плохо применимо к механизму `mmap`, хотя программа FIO в среде Linux использует вызов `posix_madvise(POSIX_MADV_DONTNEED)`, чтобы указать, что удержание данных в памяти нежелательно, и тем самым форсировать сброс данных в файл. Возможно, по причине этих особенностей реализации механизм ввода-вывода `mmap` редко применяется для измерения скорости работы накопителей данных, но он бывает полезен для оценки выигрыша от использования отображения данных в память по сравнению с обычным доступом к файлам.

Результаты

Выполненные тесты помогли ответить на вопросы о способах оптимальной настройки драйверов ввода-вывода и ограничениях, накладываемых на скорость ввода-вывода со стороны среды виртуализации.

Таблица 2. Скорость выполнения операций произвольного чтения блоков данных 4Кб, IOPS

Тест	Накопитель HDD		Накопитель SSD	
	Состояние файлового кэша			
	«холодное»	«горячее»	«холодное»	«горячее»
Linux хост гипервизора QEMU				
psync	169	693500	8200	757640
Виртуальная ОС Linux				
psync	174	20170	5734	19170
Виртуальная ОС Windows				
windowsaio	159	17160	5290	16810

В табл. 2 приведены результаты тестов чтения данных на дисках HDD и SSD, выполненных в разных режимах файлового кэша гипервизора: "холодного", в котором данные были предварительно сброшены утилитой vmtouch, и "горячего", в котором данные виртуального диска были зачитаны в кэш этой утилитой и далее удерживались в кэше в течении теста. В данном случае FIO использовала механизмы psync (в Linux) и windowsaio (в Windows), причем тестирование проводилось в режиме direct, чтобы исключить влияние файлового кэша внутри виртуальных ОС. Для дисков HDD, как и ожидалось, использование кэша ускоряет на два порядка чтение произвольных блоков данных малого размера. Для дисков SSD относительный выигрыш также есть, но он менее значительный. Это объясняется тем, что скорость работы SSD дисков значительно выше скорости HDD, а при извлечении данных из файлового кэша скорость операций ограничивается накладными расходами гипервизора. Например, отдельное измерение показало, что при работе с электронным диском в виртуальных ОС количество синхронных операций чтения и записи не превышает ~20000 IOPS, что позволяет оценить задержку одной операции в ~50µs. Данные измерения были выполнены в конфигурациях виртуальных ОС, использующих специально назначенный поток управления ввода-вывода QEMU (iothread) для каждого тестируемого устройства. Такая настройка обеспечивает максимально

скоростной режим ввода-вывода из всех режимов, доступных с применением стандартных драйверов QEMU.

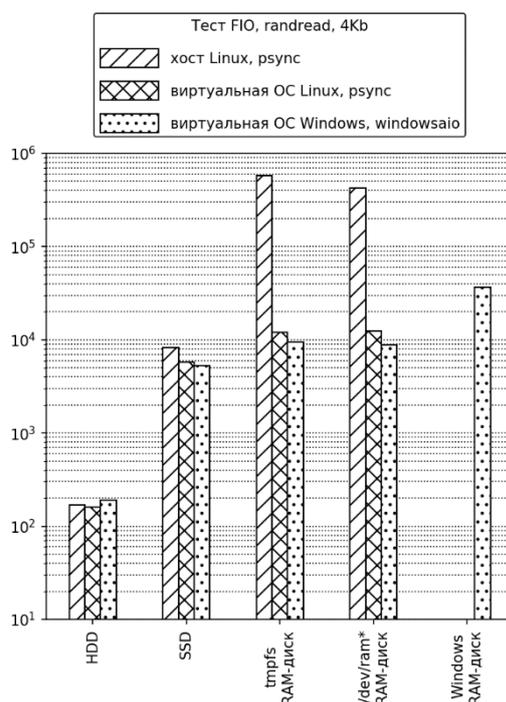


Рис. 1. Скорости чтения блоков данных размером 4Кб в произвольном порядке с различных носителей (состояние файлового кэша – "холодное"), IOPS.

Сравнение скоростей чтения с разных носителей из виртуальных и реальных операционных систем (см. рис. 1), показывает, что виртуализация не снижает существенно

производительность ввода-вывода на HDD и SSD носителях в случае, когда файловый кэш виртуальных дисков еще "холодный" и не ускоряет чтение.

Тем не менее вышеупомянутое предельное значение ~20000 IOPS может стать ограничивающим фактором при использовании более быстрых SSD накопителей, либо когда чтение данных

ускоряет файловый кэш. Результаты, приведенные на рис. 1, были измерены без использования отдельных потоков для обслуживания ввода-вывода (iothread), поэтому максимальные значения IOPS, достигаемые в среде виртуальных ОС при работе с RAM-дисками, здесь не ~20000, как в табл. 2, а только ~12500.

Таблица 3. Производительность и медианная задержка операций записи на SSD-диск блоков данных размером 4Kb в произвольном порядке в разных режимах кэширования

Система	Тест	Синхронизация (fdatasync)	Режим кэширования виртуального диска					
			none		writethrough		directsync	
			IOPS	Latency, μ s	IOPS	Latency, μ s	IOPS	Latency, μ s
Linux	psync	нет	9850	92	4300	220	4490	205
	psync	да	1650	100	1505	220	1440	215
	libaio, 5 потоков	нет	19150	980	24160	780	9710	1975
Windows	windowsaio	нет	9855	80	3920	222	4400	198
	windowsaio	да	1730	82	1740	222	1790	206
	windowsaio, 5 потоков	нет	18775	1030	22030	815	9720	1960

Руководства администратора, например Proxmox [5], обычно рекомендуют режим кэширования `cache=writethrough` (опция `O_DSYNC`) для случаев, когда важнее минимизировать задержку чтения, и режим `cache=none` (опция `O_DIRECT`), когда более важна скорость записи. Действительно, тесты подтверждают, что операции записи одного потока без синхронизации несколько быстрее выполняются в режиме `cache=none` (см. табл. 3, файловый кэш виртуальной ОС в этих тестах не использовался). Однако при необходимости синхронизации уже не наблюдается выигрыша от `cache=none` по сравнению с `cache=writethrough`, а тест многопоточковой записи, наоборот, показывает преимущество режима `writethrough`. В этом смысле рекомендации использовать `cache=none` не подтверждаются. Кроме того, когда утверждается, что для прикладной программы наиболее критична скорость записи на диск, операции чтения, как

правило, предваряют операции записи, и их ускорение тоже важно.

Ограничение на количество IOPS, наблюдаемое в среде виртуальных ОС, заставляет искать иные способы оптимизации. Одним из решений может быть использование программ для имитации накопителей в области оперативной памяти (RAM-диск). Такие RAM-диски могут периодически синхронизировать данные с внешним накопителем, что допускает их применение в прикладных системах, где потеря недавно записанных данных может быть исправлена (то есть, не в серверах управления баз данных). Существует ряд бесплатных и условно-бесплатных программ для создания RAM-дисков в среде Windows. В табл. 4 приведены результаты тестов FIO с RAM-дисками: одним, созданным в системе гипервизора, и четырьмя, созданными различными менеджерами RAM-дисков для

ОС Windows (из которых только ImDisk является бесплатной программой с открытым исходным кодом, остальные распространяются по модели условно-бесплатного программного обеспечения). Из

этих результатов следует, что на операциях с небольшими блоками данных получить существенный прирост IOPS можно только в режиме mmap без синхронизации.

Таблица 4. Производительность RAM-дисков в среде Windows при операциях с блоками данных 4Кб

Тест	Запись, IOPS		Чтение, IOPS
	асинхронная, fdatasync=0	синхронная, fdatasync=1	
RAM-диск на гипервизоре (tmpfs)			
mmap	200190	470	200340
windowsaio, direct=0	35990	6040	25120
windowsaio, direct=1	17210	7390	19080
ImDisk			
mmap	182030	2317	179560
windowsaio, direct=0	30400	1467	10340
windowsaio, direct=1	15300	7186	12220
UltraDisk			
mmap	182050	2810	178570
windowsaio, direct=0	36250	1727	29980
windowsaio, direct=1	33850	16600	35170
SmartTool RAM disk			
mmap	184090	2600	180790
windowsaio, direct=0	23780	1734	10340
windowsaio, direct=1	23500	14990	30320
Primo Ramdisk			
mmap	180050	2780	175940
windowsaio, direct=0	31670	1520	23480
windowsaio, direct=1	27580	9550	28470

Для изучения скоростей работы SSD накопителей нового поколения, NVMe 3D XPoint, в среде гипервизора QEMU/KVM была выбрана плата накопителя Intel Optane 900P в составе персонального компьютера (стенд 2). Технология 3D XPoint демонстрирует существенно большую скорость (запись – до 2Gb/sec, чтение – до 3Gb/sec) и меньшую задержку (мин. 10µs) при произвольном доступе к данным. На фоне таких скоростей становятся большими накладные расходы по передаче запросов и переключению контекста исполнения из режима процесса ОС в режим гипервизора, далее в режим ядра и обратно. Это замедляет работу с данными и создает дополнительную нагрузку на процессоры. Для ускорения доступа к NVMe SSD из виртуальных ОС и других прикладных программ компанией Intel была разработана библиотека Storage Performance Developer Kit (SPDK) [6,11], создающая специальные процессы драйверов ввода-вывода, которые работают вне контекста ядра. В этом случае работа с устройством NVMe осуществляется без обработки прерываний (т.е. в режиме цикла опроса, polling), а взаимодействие с потребителями и передача данных выполняются через страницы разделяемой памяти. Так как драйвер NVMe и процессы потребителей данных одновременно выполняются на разных ядрах процессора, то отпадает необходимость переключения контекста.

Чтобы подключить NVMe-устройство с помощью драйвера SPDK к виртуальной ОС гипервизора QEMU, требуется:

1. Установить и инициализировать библиотеку SPDK и создать с ее помощью целевое устройство vhost, соответствующее выбранному устройству. После создания устройства vhost в системе появится отдельный процесс для опроса состояния и работы с NVMe-устройством, работающий в цикле опроса и использующий 100% одного из ядер процессора.
2. Настроить виртуальную систему QEMU со специальным объектом разделяемой памяти memory-backend-file и драйвером vhost-user-blk. Версия программы QEMU должна иметь номер не ниже 2.12.

Поскольку эта технология еще находится в состоянии экспериментальной, и пользовательская документация QEMU еще не содержит всех необходимых сведений, настройка стенда 2 и запуск тестов ввода-вывода с драйверами SPDK были выполнены в соответствии с инструкциями, изложенными на web-страницах [7] и [8]. Накопитель NVMe в этих тестах использовался непосредственно как блочное устройство, то есть без разбиения на разделы и без создания файловой системы. Все тесты выполнялись в синхронном однопоточном режиме (ioengine=libaio, iodepth=1, numjobs=1, direct=1, fdatasync=1), как представляющем наибольший интерес. Результаты измерений приведены в табл. 5.

Таблица 5. Скоростные показатели записи на устройство Intel Optane 900P с разными драйверами ввода-вывода

Размер блока, Kb	Bandwidth, Mb/s	IOPS	Latency (медианная), µs
Linux хост (без виртуализации)			
4	234	58440	10.5
32	881	27530	23.4
2048	1893	925	897.0
QEMU virtio-scsi*			
4	63	15850	49.4
32	383	11980	65.3
2048	1802	880	987.0

Размер блока, Kb	Bandwidth, Mb/s	IOPS	Latency (медианная), μ s
QEMU virtio-scsi + iothread**			
4	87	21840	34.6
32	506	15800	47.9
2048	1805	880	987.0
QEMU vhost-user-blk + SPDK***			
4	112	27910	15.0
32	536	16750	28.5
2048	1614	790	1044.0

(*) – NVMe-устройство передается виртуальной ОС как SCSI устройство с помощью драйвера virtio-scsi (с прямым доступом и без кэширования – cache=directsync, io=native).

(**) – то же, что и (*), но для ввода-вывода устройства используется специальный поток управления, прикрепленный к выделенному ядру. Это самый быстрый из режимов, полностью совместимых со штатными версиями QEMU/KVM/Libvirt в среде RedHat/CentOS Linux 7.

(***) – SPDK передает устройство виртуальной ОС как Virtio BLK устройство, то есть используется стандартный Virtio QEMU драйвер, доступный для всех популярных ОС. Также есть возможность представить устройство как Virtio SCSI (vhost-user-scsi), хотя этот режим работает несколько медленнее.

Из таблицы видно, что в худшем случае, когда размер блока равен 4Kb, потеря IOPS от виртуализации с драйвером SPDK составляет около 50%, а latency выше на 50%. При использовании штатного драйвера QEMU virtio-scsi со специально назначенным потоком управления ввода-вывода для виртуального устройства (iothread) скорость меньше уже в 3 раза. Если используется virtio-scsi без iothread, то скорость ниже еще на 30%.

Согласно спецификации Intel максимальное количество операций записи в однопоточном режиме должно быть приблизительно 100000 IOPS, поскольку минимальная задержка заявлена как равная 10 μ s. Тем не менее тест FIO показывает 58440 IOPS (первая строка табл. 5), что существенно меньше – это косвенно указывает на накладные расходы программы FIO и системной библиотеки по формированию запросов. Действительно, самое высокое значение синхронных IOPS в однопоточном режиме, равное 81000 IOPS, удастся получить при тестировании в режиме ioengine=pvsync2 вместе с опцией hipri, которая включает режим опроса

устройства в цикле с помощью атрибута RWF_HIPRI системных вызовов preadv2 и rwitev2. Такой тест нельзя выполнить внутри виртуальной ОС, поэтому он не приведен в табл. 5, такое сравнение не имело бы смысла.

При нагрузке устройства большим количеством асинхронных запросов, то есть при увеличении параметров iodpth и numjobs, можно наблюдать почти линейный рост IOPS вплоть до насыщения на уровне 484000 IOPS без виртуализации и 303000 IOPS в виртуальной ОС с драйвером SPDK. Это несколько меньше максимального значения 500000 IOPS, указанного в спецификации Intel.

Обсуждение

Среда виртуализации QEMU/KVM/Libvirt может использовать разнообразные блочные локальные и сетевые устройства для эмуляции виртуальных накопителей данных. Если используются медленные устройства хранения (HDD, SSD первых поколений) или нужны все функции блочных устройств QEMU (доступные только при использовании стандартных драйверов), то наиболее скоростной режим работы виртуальных

дисков достигается при использовании драйверов VirtIO в сочетании с опцией `cache=writethrough`. Максимальные показатели IOPS можно получить, назначив каждому виртуальному накопителю специальный поток управления ввода-вывода (`iothread`) и привязав его к определенному ядру процессора.

Если возникает подозрение, что медленная работа прикладных программ в виртуальной ОС ограничена скоростью доступа к виртуальным дискам, то рекомендуется изучить подробнее конфигурацию всей программной системы, а именно ответить на следующие вопросы:

1. Может ли файловый кэш внутри виртуальной ОС ускорить работу прикладной программы? Простое увеличение размера оперативной памяти виртуальной ОС может обеспечить кэширование часто используемых данных. Этот подход не даст результата, если прикладная программа использует прямой ввод-вывод (`O_DIRECT`) или выполняет частые операции записи с синхронизацией.
2. Каков состав операций ввода-вывода прикладной программы? Какие скорости ввода-вывода ожидаемы для такой нагрузки? Определить это можно специальными системными программами (`blktrace` в Linux, `ProcMon` в Windows), протоколирующими системные вызовы. Если воспроизвести похожую нагрузку тестовой программой `FIO`, то результаты запуска в виртуальной ОС и в системе гипервизора помогут понять, в чем причина. При тестировании без виртуализации нужно использовать те же накопители данных и использовать файловый кэш и синхронизацию (опции `FIO direct=0, fdatasync=1`), чтобы симитировать режим кэширования `cache=writethrough` диска QEMU.
3. Является ли "узким местом" именно ограничения среды виртуализации? Если после предыдущего шага выяснилось, что необходимое количество IOPS выше

20000 и такая скорость может быть обеспечена накопителями данных, то ответ положительный.

Существенно более высокую производительность ввода-вывода в QEMU можно получить либо полностью передавая управление накопителем данных виртуальной ОС (проброс PCI-устройства), либо используя экспериментальные драйверы, работающие вне ядра Linux (`Vhost` и `SPDK`). Оба варианта имеют свои преимущества и недостатки: предоставление прямого доступа к устройству не позволяет использовать накопитель совместно, а технология `SPDK` еще находится в стадии разработки. В будущем `SPDK` может стать наиболее удобным решением задачи построения скоростных систем хранения данных. Так, `SPDK` предусматривает уровень разбиения NVMe SSD на тома, аналогичный системе `LVM`, и доступ к устройствам через `RDMA`-сеть по технологии `NVMe over Fabric (NVMeoF)`, аналогичный `iSCSI`. Разработчики библиотеки `SPDK` регулярно публикуют результаты тестирования производительности подсистем ввода-вывода, созданных на базе этой библиотеки [6].

Заключение

Среда виртуализации QEMU/KVM/Libvirt предоставляет широкий набор функций для работы с разными системами хранения данных. К сожалению, пользовательская документация этих продуктов разрознена и отстает от текущих версий ПО. В данной работе было выполнено тестирование нескольких типичных программно-аппаратных конфигураций и показано, как можно достичь максимальной производительности ввода-вывода прикладных программ в среде виртуальных операционных систем.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2019-0007 "36.20 Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления." (№АААА-А19-119011590093-3).

I/O configuration and performance tuning in GNU Linux/QEMU/KVM/Libvirt environment

A.B. Betelin, I.B. Egorychev, A.A. Prilipko, G.A. Prilipko, S.G. Romanyuk,
D.V. Samborskiy

Abstract: In this paper, we describe and test various configurations of storage subsystems in QEMU/KVM/Libvirt virtual OS instances. We performed I/O tuning and benchmarking in Linux and Windows virtual instances with different storage medias (HDD, SSD, RAM-disks). The experimental results allowed us to provide basic guidelines for improving application performance in virtualized environment.

Keywords: virtualization, I/O, Libvirt, QEMU, KVM, Linux, virtio, SPDK, HDD, SSD, NVMe, RAM

Литература

1. Сайт "QEMU, the FAST! processor emulator", <https://www.qemu.org>. Дата обращения 11.11.2019.
2. Сайт проекта KVM, <https://www.linux-kvm.org/page>. Дата обращения 11.11.2019.
3. Сайт "Libvirt: The virtualization API", <https://libvirt.org>. Дата обращения 11.11.2019.
4. А.Б. Бетелин, А.А. Прилипко, Г.А. Прилипко, С.Г. Романюк, Д.В. Самборский, "Оптимизация архивирования виртуальных операционных систем в среде GNU Linux/QEMU/KVM/Libvirt" // Труды НИИСИ РАН. 2018. Т. 8 №5, С. 32-41
5. Сайт "Proxmox User Guide: Performance Tweaks", https://pve.proxmox.com/wiki/Performance_Tweaks. Дата обращения 11.11.2019.
6. Сайт проекта SPDK "Storage Performance Development Kit", <https://spdk.io>. Дата обращения 11.11.2019.
7. Сайт "Vhost vs local NVMe-over-fabrics targets", <https://www.circuitblvd.com/post/vhost-vs-local-nvme-over-fabrics-targets>. Дата обращения 11.11.2019.
8. Раздел документации SPDK "vhost Target", <https://spdk.io/doc/vhost.html>. Дата обращения 11.11.2019.
9. Сайт "vmtouch – the Virtual Memory Toucher", <https://hoitech.com/vmtouch>. Дата обращения 11.11.2019.
10. Сайт "FIO – Flexible I/O tester", <https://fio.readthedocs.io>. Дата обращения 11.11.2019.
11. Yang Z, Harris JR, Walker B, Verkamp D, Liu C, Chang C, Cao G, Stern J, Verma V, Paul LE. Spdk: A development kit to build high performance storage applications. – In 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) 2017 Dec 11 (pp. 154-161). IEEE.

Подготовка статей для журнала «Труды НИИСИ РАН»

А.А. Асонов¹, А.Н. Годунов²

¹ФГУ ФНЦ НИИСИ РАН, Москва, Россия, asonow@niisi.ras.ru;

²ФГУ ФНЦ НИИСИ РАН, Москва, Россия, nkag@niisi.ras.ru

Аннотация. Статья содержит правила и рекомендации по оформлению научных публикаций для журнала «Труды НИИСИ РАН». В ней также даются рекомендации по использованию текстового редактора Microsoft Word.

Ключевые слова: подготовка статей, Труды НИИСИ РАН, форматирование, MS Word

1. Введение

Журнал «Труды НИИСИ РАН» публикует оригинальные статьи, ранее не опубликованные в других изданиях (все статьи проверяются на наличие заимствований, в том числе и из собственных статей авторов). Объем статьи, как правило, не должен превышать 20 страниц. Статья направляется научному редактору номера по электронной почте в виде файла формата .doc или .docx. В сопроводительном письме необходимо указать номера мобильных телефонов всех авторов. Номера мобильных не публикуются и нужны только для оперативной связи редакции с авторами.

Для подготовки статьи следует использовать текстовый редактор Microsoft Word 2010 или более новой версии. Данный документ можно использовать в качестве шаблона. Весь текст статьи набирается шрифтом Times New Roman. Сноски по всему тексту (кроме списка авторов статьи) не допускаются.

Размер всех полей (верхнее, нижнее, внутри, снаружи) в документе равен 3 см. Размер верхнего и нижнего колонтитулов – 1,25 см.

Для задания размеров полей необходимо на вкладке «Макет» нажать кнопку «Параметры страницы» и в появившемся окне указать необходимые размеры полей.

Для добавления колонтитулов необходимо на вкладке «Вставка» нажать кнопку «Верхний (Нижний) колонтитул», во всплывающем меню выбрать пункт «Изменить верхний (нижний) колонтитул» и на появившейся вкладке «Конструктор» указать необходимые размеры колонтитулов.

Статья должна содержать заголовок, основную часть, краткую информацию о статье на английском языке и список литературы.

2. Заголовок

Заголовок статьи включает в себя название статьи, список авторов, краткую информацию об авторах, аннотацию и ключевые слова.

В *названии статьи* используется форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 20 кегль.

В *списке авторов* статьи к фамилии каждого автора добавляется нумерованная ссылка на сноску с краткой информацией об авторе. Авторы статьи перечисляются через запятую. В конце списка авторов точка не ставится. Для списка авторов статьи используется форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 12 кегль. Для создания сноски необходимо выделить ее цифры и на вкладке «Главная» нажать кнопку «Надстрочный знак».

Сноска с краткой информацией об авторе содержит номер ссылки на нее, название организации, где работает автор, город и страна нахождения данной организации и адрес электронной почты автора, разделенные запятыми (для сотрудников НИИСИ указывается краткое наименование – ФГУ ФНЦ НИИСИ РАН, Москва, Россия, электронная почта автора). Сноски начинаются с новой строки и отделяются друг от друга точками с запятой. В конце последней сноски точка не ставится. Для сноски с краткой информацией об авторе используется форматирование абзаца по левому краю с отступом 3 см, междустрочный интервал одинарный, интервал перед абзацем –

0 пт, интервал после абзаца – 6 пт, шрифт обычный (не полужирный, не курсив, без подчеркивания), 9 кегль.

Аннотация должна содержать краткое описание основных положений статьи и не должна совпадать с разделом «Введение». Размер аннотации ограничен 10–15 строками текста. Ссылки на литературу в аннотации не допускаются, также желательно не использовать формулы, за исключением простых. В аннотации используется форматирование абзаца по ширине с отступом первой строки 1 см, междустрочный интервал одинарный, интервал перед абзацем – 12 пт, интервал после абзаца – 12 пт, шрифт обычный, 9 кегль, заглавие выделяется полужирным шрифтом.

Ключевые слова перечисляются через запятую. В конце списка ключевых слов точка не ставится. Для ключевых слов используется форматирование абзаца по ширине, отступ первой строки 1 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный, 10 кегль, заглавие выделяется полужирным шрифтом.

3. Основная часть статьи

Основная часть статьи состоит из нумерованных разделов, которые в свою очередь могут содержать подразделы, рисунки, таблицы и формулы.

Для смыслового выделения текста используйте курсив, по возможности не используйте подчеркивание и полужирный шрифт.

Ссылки на источники из списка литературы следует указывать в квадратных скобках (например, [3]). В случае ссылок на несколько источников, ссылки указываются через запятую (например, [5], [6], [9] или [5], [6] и [9]). На каждый источник из списка литературы должна быть ссылка в тексте статьи.

Для *основного текста* статьи используется форматирование абзаца по ширине с отступом первой строки 0,5 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный, 10 кегль.

3.1. Разделы и подразделы

Названия всех разделов и подразделов, включая «Введение» и «Заключение», нумеруются. Нумерация разделов и подразделов представляет собой

последовательность арабских цифр и точек. В конце номера раздела и подраздела точка также ставится.

В *названии разделов* используется форматирование абзаца по левому краю без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт полужирный, 14 кегль.

В *названии подразделов* используется форматирование абзаца по левому краю без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт полужирный, 12 кегль.

3.2. Рисунки, формулы и таблицы

Таблицы и рисунки вставляются непосредственно в текст выровненными по центру (см. рис. 1 и таблицы 1, 2). Все рисунки и таблицы нумеруются в порядке их следования.

Подпись к рисунку состоит из номера и названия рисунка в виде «Рис. 1. Название рисунка» и помещается снизу от самого рисунка. Для подписи к рисунку используется форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт обычный, 9 кегль.

Для добавления рисунка в редакторе Microsoft Word необходимо на вкладке «Вставка» нажать кнопку «Рисунки» и в появившемся окне выбрать файл с рисунком. Кликнуть правой кнопкой мыши по вставленному рисунку и во всплывающем меню выбрать пункт «Обтекание текстом» | «Сверху и снизу». Также допускается использовать полотно.

Заголовок таблицы состоит из номера и названия таблицы в виде «Таблица 1. Название таблицы». Заголовок таблицы помещается сверху от самой таблицы, для него используется форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт обычный, 9 кегль.

Таблица может быть создана средствами редактора Microsoft Word (см. таблицу 1) или вставлена в виде рисунка хорошего качества (см. таблицу 2).

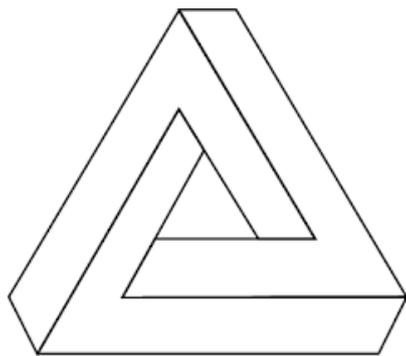


Рис. 1. Пример добавления рисунка

Таблица 1. Пример таблицы, созданной средствами текстового редактора

Название столбца 1	Название столбца 2	Название столбца 2
Название строки 1	25	Слово
Название строки 2	@#\$\$%	

Таблица 2. Пример таблицы в виде рисунка

Числовой формат	Пример
Числовой	1234,56
Процентный	57%
Дробный	1/8
Дата	12.09.2019
Время	12:34:56

Для создания таблицы средствами редактора необходимо на вкладке «Вставка» нажать кнопку «Таблица» и во всплывающем меню выделить необходимое количество строк и столбцов.

$$\lim_{x \rightarrow \infty} \frac{1}{x^2 + 1} = 0 \quad (1)$$

$$NPV = \sum_{t=0}^n \frac{CF_t}{(1+R)^t} \quad (2)$$

Математические формулы набираются средствами встроенного редактора (1) или вставляются в виде рисунка хорошего качества (2) и выравниваются по центру строки. Также допускается добавление формул, набранных с помощью редактора MathType.

Для создания математической формулы средствами редактора необходимо на вкладке «Вставка» нажать кнопку «Уравнение» и

подобрать подходящий шаблон для формулы.

На все рисунки и таблицы должны быть ссылки в тексте статьи. Ссылки на формулы представляют собой номер формулы, заключенный в скобки. Нумеровать нужно только те формулы, на которые есть ссылки в тексте статьи. Не рекомендуется уменьшать размер используемого в формуле шрифта или масштабировать рисунок с формулой с целью уместить длинную формулу в одну строку.

4. Краткая информация о статье на английском языке

После окончания основного текста статьи приводится краткая информация о ней на английском языке. Данный раздел набирается в одну колонку. Раздел содержит (см. пример оформления ниже):

- название статьи (форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 20 кегль);

- имена и фамилии авторов (форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт полужирный, 12 кегль);

- Abstract (форматирование абзаца по ширине с отступом первой строки 1 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт обычный, 9 кегль, заглавие выделяется полужирным шрифтом);

- Keywords (форматирование абзаца по ширине с отступом первой строки 1 см., междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный, 10 кегль, заглавие выделяется полужирным шрифтом).

5. Список литературы

В самом конце статьи (после краткой информации о статье на английском языке) приводится список литературы. Данный раздел оформляется в одну колонку (см. пример ниже).

Для названия раздела «Литература» используется форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт полужирный, 14 кегль.

Цитируемая литература оформляется в виде нумерованного списка форматированием абзаца по ширине с отступом первой строки 0,5 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0

пт, шрифт обычный, 10 кегль. Нумерация списка состоит из арабских цифр с точкой на конце (например, «3.»).

При оформлении списка литературы следует руководствоваться следующими правилами.

Для книг указываются инициалы и фамилии авторов через запятую, название книги без кавычек, город, издательство, год. Например,

В.Б. Бетелин. Системы автоматизации труда программиста. М., Наука, 1990.

Для статей в периодических изданиях указываются инициалы и фамилии авторов через запятую, название статьи без кавычек, название периодического издания в кавычках, номер тома, год в скобках, номер журнала, номера страниц. Например,

А.А. Асонов, А.Н. Годунов, В.А. Солдатов. Методы снижения ресурсоемкости операционной системы жесткого реального времени. «Труды НИИСИ», Т. 8 (2018), № 1, 4–13.

Для материалов конференции указываются инициалы и фамилии авторов через запятую, название доклада без кавычек, название конференции в кавычках, место проведения, даты проведения, где изданы тезисы. Например,

А.Г. Петров, И.Р. Шилова. Имитационное моделирование в сложных технических системах. «Моделирование сложных систем: IV Международная научная конференция. Москва, 21–23 апреля 2005 г. (тезисы)», М., Изд-во МИЭМ, 2006, 53–54.

Для диссертаций указываются инициалы и фамилии авторов через запятую, название диссертационной работы без кавычек, тип диссертации (кандидатская, докторская),

организация, в которой проходила защита, город, год, URL-ссылка на полный текст диссертации или автореферат. Например,

Н.Д. Байков. Численные методы решения некоторых задач механики жидкости со свободной границей. Кандидатская диссертация. МГУ имени М.В. Ломоносова, Москва, 2019, https://istina.msu.ru/download/200148295/1iMZAy:2D2bOWmXo_5GIbqecdQJCj2sQg4/.

6. Заключение

В конце основного текста статьи можно выразить благодарности людям и организациям, оказавшим помощь в ее написании. Также указывается в рамках какой работы выполнена публикация, например,

«Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме (проекту) «Исследование принципов построения компактной операционной системы для отечественных радиационно-стойких процессоров» (№ 0065-2016-0003).»

В таблице 3 приведены правила оформления всех частей статьи и названия соответствующих стилей.

Актуальную версию статьи, относящуюся к конкретному номеру, следует запрашивать у научного редактора номера.

Авторы выражают надежду, что использование данной работы в качестве руководства по подготовке статьи поможет сократить время подготовки и повысит ее качество.

Таблица 3. Правила оформления статьи

Тип объекта	Форматирование	Стиль
Заголовок статьи		
Название статьи	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 20 кегль	_НАЗВАНИЕ СТАТЬИ_
Список авторов	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 12 кегль	_СПИСОК АВТОРОВ_
Сноска с краткой информацией об авторе	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 6 пт, шрифт обычный, 9 кегль	_СНОСКА_
Аннотация	Форматирование абзаца по ширине с отступом первой строки 1 см,	_АННОТАЦИЯ_

Тип объекта	Форматирование	Стиль
	междустрочный интервал одинарный, интервал перед абзацем – 12 пт, интервал после абзаца – 12 пт, шрифт обычный, 9 кегль, заглавие выделяется полужирным шрифтом	
Ключевые слова	Форматирование абзаца по ширине с отступом первой строки 1 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный, 10 кегль, заглавие выделяется полужирным шрифтом	_КЛЮЧЕВЫЕ СЛОВА_
Основная часть статьи (в две колонки)		
Основной текст	Форматирование абзаца по ширине с отступом первой строки 0,5 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный или курсив, 10 кегль	_ОСНОВНОЙ ТЕКСТ_
Название раздела	Форматирование абзаца по левому краю без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт полужирный, 14 кегль	_РАЗДЕЛ_
Название подраздела	Форматирование абзаца по левому краю без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт полужирный, 12 кегль	_ПОДРАЗДЕЛ_
Подпись к рисунку	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт обычный, 9 кегль. Номер и название рисунка добавляются снизу от самого рисунка	_РИСУНОК ТАБЛИЦА_ И
Заголовок таблицы	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт обычный, 9 кегль. Номер и название таблицы помещаются сверху от самой таблицы	_РИСУНОК ТАБЛИЦА_ И
Краткая информация о статье на английском языке		
Название статьи	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 20 кегль	_НАЗВАНИЕ СТАТЬИ_
Имена и фамилии авторов	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт полужирный, 12 кегль	_СПИСОК АВТОРОВ_

Тип объекта	Форматирование	Стиль
Abstract	Форматирование абзаца по ширине с отступом первой строки 1 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 12 пт, шрифт обычный, 9 кегль, заглавие выделяется полужирным	_АННОТАЦИЯ_
Keywords	Форматирование абзаца по ширине с отступом первой строки 1 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный, 10 кегль, заглавие выделяется полужирным	_КЛЮЧЕВЫЕ СЛОВА_
Список литературы		
Заголовок «Литература»	Форматирование абзаца по центру без отступа первой строки, междустрочный интервал одинарный, интервал перед абзацем – 6 пт, интервал после абзаца – 6 пт, шрифт полужирный, 14 кегль	_СПИСОК ЛИТЕРАТУРЫ_
Список цитируемой в статье литературы	Нумерованный список. Форматирование абзаца по ширине с отступом первой строки 0,5 см, междустрочный интервал одинарный, интервал перед абзацем – 0 пт, интервал после абзаца – 0 пт, шрифт обычный, 10 кегль	_ОСНОВНОЙ ТЕКСТ_
Примечание: Для всего текста статьи следует использовать шрифт Times New Roman. Размер полей (верхнее, нижнее, внутри, снаружи) – 3 см. Размер верхнего и нижнего колонтитулов – 1,25 см.		

Preparation of Papers for Proceedings of SRISA RAS

Alexander Asonov, Alexander Godunov

Abstract. The article contains rules and recommendations on the design of scientific publications for the journal "Proceedings of NIISI RAS". It also provides recommendations for using the Microsoft Word text editor.

Keywords: preparation of papers, Proceedings of NIISI RAS, formatting, MS Word

Литература

1. В.Б. Бетелин. Системы автоматизации труда программиста. М., Наука, 1990.
2. А.А. Асонов, А.Н. Годунов, В.А. Солдатов. Методы снижения ресурсоемкости операционной системы жесткого реального времени. «Труды НИИСИ», Т. 8 (2018), № 1, 4–13.
3. Н.Д. Байков, А.Г. Петров. О формировании кумулятивных струй в плоско-параллельном потоке идеальной жидкости. «Всероссийская конференция с международным участием «Современные проблемы механики сплошных сред и физики взрыва», Россия, Новосибирск, 4–8 сентября 2017.
4. I. Danilov, M. Gorbunov, A. Antonov. SET tolerance of 65 nm CMOS majority voters: A comparative study. "14th European Conference on Radiation and Its Effects on Components and Systems (RADECS) 2013", Great Britain, Oxford, 2013, 1–6.
5. I. Danilov, M. Gorbunov, A. Shnaider, A. Balbekov, Y. Rogatkin, S. Bobkov. On board electronic devices safety provided by DICE-based Muller C-elements. "Acta Astronautica", V. 150 (2018), 28–32.

6. F. de Dinechin, D. Defour, C. Lauter. Fast correct rounding of elementary functions in double precision using double extended arithmetic. INRIA, Research Report RR-5137, 2004, <https://hal.inria.fr/inria-00071446/document>.

7. Н.Д. Байков. Численные методы решения некоторых задач механики жидкости со свободной границей. Автореферат кандидатской диссертации. МГУ имени М.В. Ломоносова, Москва, 2019, <https://istina.msu.ru/download/200148293/1iMZAy:-kdxksZwfcH7TLfnuPPfkDhoeLk/>.