

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

ТРУДЫ НИИСИ РАН

ТОМ 10 № 2

**МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ:**

ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ

МОСКВА
2020

Редакционный совет ФГУ ФНЦ НИИСИ РАН:

В.Б. Бетелин (председатель),
Е.П. Велихов, С.Е. Власов, 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.Г. Мадера

Тематика номера:

Информационные системы и сети, инструментальные средства для операционных систем реального времени, математическое моделирование тепловых процессов, математическое моделирование эволюционных процессов

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, опτικο-нейронные технологии, микро- и нанoeлектроника, математические исследования и вопросы численного анализа, история науки и техники.

The topic of the issue:

Information systems and networks, tools for real-time operating systems,
math modeling of thermal processes, math modeling of evolution processes

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>А.А. Рыбаков, С.С. Шумилин</i> Сравнение алгоритмов машинного обучения для предсказания времени работы пользовательских заданий в рамках оптимизации использования ресурсов суперкомпьютерного кластера МСЦ РАН	4
<i>А.Г. Мадера</i> Распределение температуры воздушной среды в электронной системе.....	14
<i>А.Г. Мадера, М.Ж. Акжолов</i> Исследование адиабатических и неадиабатических тепловых процессов в электронных системах около тепловыделяющего элемента	18
<i>М.Ж. Акжолов, П.И. Кандалов</i> Численное моделирование теплообмена тепловыделяющего элемента в окружающей среде в электронной системе при обдуве кулером	23
<i>М.Ж. Акжолов</i> Численное исследование зависимости конвективного охлаждения тепловыделяющего элемента электронной системы от скорости воздуха при обдуве кулером	28
<i>В.Г. Редько</i> Модель чувства причинности	34
<i>Н.Е. Каленов, С.А. Кириллов, А.Г. Рыбакова, И.Н. Соболевская, А.Н. Сотников</i> Технология формирования мультимедийного виртуального музейного пространства	39
<i>Т.К. Грингауз, А.Н. Онин</i> . Профилирование программ параллельной обработки сигналов для мультипроцессорных комплексов реального времени.....	44
<i>А.А. Рыбаков, Т.Т. Сохибов, С.С. Шумилин</i> Конструирование искусственных нейронных сетей с помощью меметических алгоритмов.....	55

Сравнение алгоритмов машинного обучения для предсказания времени работы пользовательских заданий в рамках оптимизации использования ресурсов суперкомпьютерного кластера МСЦ РАН

¹А.А. Рыбаков, ²С.С. Шумилин

МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия

E-mails: ¹rybakov@jscc.ru, ²shumilin@jscc.ru

Аннотация. Оценка ресурсов, необходимых для выполнения программы на суперкомпьютерном кластере, является трудновыполнимой задачей. Пользователю необходимо практически угадывать время работы программы, что приводит к тому, что некоторые задания принудительно завершаются планировщиком из-за превышения лимита по времени. Большое число подобных событий приводит к тому, что распределение ресурсов кластера теряет эффективность. Для того, чтобы оптимизировать распределение ресурсов, необходимо, чтобы планировщик получал более точные оценки параметров задания. Ранее для решения этой проблемы были предприняты попытки использовать алгоритмы машинного обучения. В данной статье рассмотрено применение таких алгоритмов к данным суперкомпьютерного кластера МСЦ РАН. Получены результаты по качеству регрессионных моделей на различных дополнительных признаках, среди которых проведен отбор наиболее информативных. Для отбора признаков использованы различные техники, широко используемые в машинном обучении. Данные признаки, которые характеризуют пользователя кластера, могут быть использованы для повышения точности уже полученных моделей.

Ключевые слова. НРС, машинное обучение, предсказательная аналитика, планировщик заданий, анализ поведения пользователя.

Введение

При формировании задания пользователь кластера МСЦ РАН должен указать время работы и количество процессорных ядер. Оценить время выполнения, основываясь на коде программы, очень сложно. Поэтому пользователи зачастую указывают приближенные оценки времени выполнения. Также следует отметить, что пользователи кластера МСЦ РАН в основном являются представителями сфер, не связанных с программированием, и поэтому было бы неправильно возлагать на них дополнительные задачи по оценке времени выполнения программы.

На основе указанных параметров в применяемой в МСЦ РАН СУППЗ [1] (Система управления прохождением параллельных заданий) формируется паспорт задания, вместе с которым задание попадает в очередь. Если задание завершается раньше указанного пользователем времени, то это вредит системе планирования, так как если

бы точное время выполнения задания было известно изначально, то ресурсы были бы распределены более эффективно. Поэтому ставится задача предсказания времени работы задания по указанным характеристикам.

То есть необходимо, чтобы на вход планировщика приходила не пользовательская оценка времени выполнения, а время, рассчитанное моделью (рис. 1).

Предсказание времени выполнения задания представляет собой задачу, основанную на представлении паспорта задания как входного набора признаков для алгоритмов машинного обучения (МО). Главная идея заключается в том, что каждому пользователю свойственны определенные шаблоны задач. То есть высока вероятность, что определенный пользователь постоянно запускает задачи, схожие по площади, или заказывает определенное время работы. То есть в целом его задачи имеют тенденцию обрабатываться за определенное время. Также высока вероятность, что пользователи из схожих

групп (проект, организация, сфера деятельности) имеют схожие статистические метрики. Данная задача называется задачей пользовательского моделирования (user modeling) и представляет собой использование поведенческих характеристик пользователя для предсказания его дальнейших действий [2, 3]. В нашем случае мы совмещаем признаки пользователя и задания и получаем модель, отражающую опыт пользователя.

Критическим остается вопрос выбора признаков для модели. В дополнение к исходным признакам в работе формируются новые: временные и персональные. Далее проводится отбор наиболее информативных признаков.

В данной работе приведены результаты применения алгоритмов машинного обучения для решения задачи регрессии для предсказания времени выполнения задания. Рассмотрены несколько моделей, которые относятся к различным классам алгоритмов. Основная метрика, используемая для оценки качества алгоритмов, это скорректированный

коэффициент детерминации R_{adj}^2 . На основе данной метрики выбраны наиболее качественные модели и произведено их сравнение с базовым алгоритмом, использующим скользящее среднее для оценки времени выполнения задания.

Обучение моделей проводилось на данных суперкомпьютера МВС-10П «Торнадо», являющегося одной из вычислительных систем МСЦ РАН. Данные представляют собой информацию об использовании ресурсов за 2013–2017 гг. В исходную таблицу входят 110 тыс. строк, каждая из которых представляет собой информацию об уникальном задании, в том числе данные указанные пользователем при регистрации задания.

Таким образом, данная работа привносит следующий вклад:

- Исследование данных МСЦ РАН.
- Выбор наиболее информативных признаков для обучения моделей.
- Сравнения с простым алгоритмом, использующим скользящее среднее.



Рис. 1 – Взаимодействие модели и планировщика заданий.

Машинное обучение для предсказания времени работы задания

Попытки использования алгоритмов машинного обучения для работы с данными суперкомпьютерного кластера для оптимизации работы планировщика проводились еще два десятилетия назад [4–6].

В работе [7] авторы приводят результаты построения моделей на данных суперкомпьютера. Чтобы определить количество используемых ресурсов (количество CPU и RAM) авторы используют алгоритмы регрессии. Для того чтобы определить, будет ли задание принудительно завершено планировщиком, авторы используют алгоритмы бинарной классификации.

В работе [8] авторы также используют регрессионные алгоритмы для предсказания количества заказанных процессоров. Для того, чтобы предсказать, будет ли задание завершено планировщиком принудительно, используются алгоритмы классификации. Авторы приходят к выводу, что при добавлении в обучающую выборку времени, запрошенного пользователем, качество регрессии значительно увеличивается (R^2), а качество классификации (F-мера) изменяется незначительно.

Авторы [9] описывают модель, которая предсказывает количество необходимой памяти для выполнения программы.

Стоит отметить, что полученные оценки моделей, указанные в приведенных работах, сложно сравнивать, потому что данные у каждой группы исследователей отличаются. Во-первых, отличаются размеры

обучающих выборок. Во-вторых, у каждого суперкомпьютерного центра может быть своя специфика пользователей. Предположим, у одного из центров большая часть пользователей – это люди, имеющие достаточный опыт для априорной оценки ресурсов задания. Следовательно, модель, обученная на таких данных, будет отличаться высокой точностью.

Построение новых признаков

Исходные признаки, указанные пользователем не отражают всей информации доступной для модели. Поэтому необходимо на основе имеющихся данных построить новые признаки, которые позволяют дифференцировать задания.

Для того, чтобы понять, какие признаки позволяют получить наиболее точную модель, были выделены следующие группы признаков:

- Исходные (И).
- Временные (В).
- Персональные (П).

Обозначения типов признаков: В – вещественные, К – категориальные, Б – бинарные.

Исходные признаки представляют собой данные, которые поступают на вход планировщика (табл. 1). К исходным признакам был добавлен признак *square*, представляющий собой площадь задания (ядер × час).

Табл. 1 – Исходные признаки.

Имя	Тип	Описание
userid	К	id пользователя
proc	В	заказанные ядра
ntime	В	заказанное время
quantum	В	квант
gid	К	id группы
domain	К	область исследований
orgid	К	id организации
square	В	площадь задания

Временные признаки представляют собой признаки, которые отражают момент, когда задание было зарегистрировано пользователем (табл. 2). Их применение основано на гипотезе о том, что

пользователям свойственны определенные шаблоны поведения, связанные с временем запуска заданий. Это может найти отражение в сезонных изменениях. К примеру, средняя длительность заданий к концу года или в период праздников может иметь определенную тенденцию. Так же, можно предположить, что время запуска определенного пользователя в течение рабочего дня может иметь выраженный характер и улавливаться моделью.

Табл. 2 – Временные признаки.

Имя	Тип	Описание
day	В	день
hour	В	час
month	В	месяц
dow	В	день недели

Персональные признаки формируются исходя из статистики пользователя (табл. 3). Использование данных признаков основано на гипотезе, что определенным пользователям свойственны определенные шаблоны поведения в использовании суперкомпьютерным кластером МСЦ РАН. На основе прошлых запусков пользователя формируется статистика, которая предположительно описывает данного пользователя и позволяет модели обучиться и выделять группы пользователей со схожей статистикой.

Табл. 3 – Персональные признаки.

Имя	Тип	Описание
duration_use	В	среднее время работы задания
proc_user	В	среднее заказанное число ядер
ntime_user	В	среднее заказанное время
resources_user	В	среднее число использованных ресурсов
exceedtime_user	В	среднее преувеличение заказанного времени
experience_user	В	количество дней между первым и последним запуском (опыт пользователя)

Таким образом, мы получаем 18 признаков.

Отбор признаков

Отбор признаков является распространенным этапом в подготовке данных. Его необходимость обуславливается рядом причин.

Очевидно, что не все признаки являются информативными и связаны с целевой переменной. Наличие таких признаков негативно влияет на качество модели и на ее обобщающую способность. Не всегда можно на этапе формирования данных отличить неинформативные признаки, поэтому изначально включаются все имеющиеся.

Для того чтобы отобрать важные признаки существуют несколько подходов. Самым простым подходом является метод при котором рассчитывается связь между каждым из признаков по отдельности и целевой переменной. На рисунке 2 приведены коэффициенты корреляции Пирсона для всех вещественных признаков. Отметим, что наибольшая корреляция прослеживается со средним временем выполнения задания пользователя. То есть каждому пользователю присущи задания определенной длины. Однако, проблема данного подхода заключается в том, что одномерный отбор не позволяет выявить комбинации признаков, которые имеют информативность выше, чем у каждого из входящих признаков в отдельности. Поэтому в МО существуют другие методы отбора признаков, позволяющие, учитывать их взаимодействие.

Линейные модели. Ответом линейной модели является взвешенная сумма признаков на данном объекте:

$$a(x) = \sum_{j=1}^d w_j x^j$$

где d - количество признаков, w - вес признака x .

Таким образом, признаки с большими весами вносят больший вклад в ответ модели, следовательно, являются более важными. Для этого модель обучается на признаках и веса, которые были получены в процессе обучения, интерпретируются как важность того или иного признака. Важным условием является масштаб признаков. То есть необходимо, чтобы признаки были выровнены относительно нуля. В данной работе к данным было применено масштабирование. Каждое значение признака заменяется на его z -оценку по формуле:

$$z = \frac{x - \bar{x}}{\sigma}$$

где x - значение признака, \bar{x} - среднее значение признака на всех объектах, σ - стандартное отклонение.

Для того, чтобы не допустить переобучения модели параметры масштабирования – среднее значение и стандартное отклонение вычисляются на обучающей выборке и потом применяются к тестовой.

Так как задача заключается в отборе признаков, то для того, чтобы занулить как можно больше неинформативных признаков, можно использовать L1-регуляризацию. Алгоритм линейной регрессии с L1-регуляризатором использует l_1 -норму вектора весов. Алгоритм описывается следующим образом:

$$\arg \min_{w \in \mathbb{R}^m} \left(\frac{1}{2m} \|Xw - y\|^2 + \lambda \|w\|_1 \right)$$

где X - матрица признаков, w - вектор весов, y - целевая переменная, m - количество объектов, λ - параметр регуляризации. С увеличением λ увеличивается количество зануленных неинформативных признаков.

Линейная модель с L2-регуляризатором использует l_2 -норму вектора весов.

Решающие деревья. Решающие деревья строятся жадно – от корня к вершине. Для того, чтобы произвести разбиение вершины нужно выбрать признак и порог, по которому определяется в какое из поддеревьев отправится данный объект.

Для этого в каждой вершине вычисляется взвешенная сумма критериев информативности левого и правого поддеревьев. Поэтому, как сильно признак уменьшил это значение можно судить о том, какой вклад он вносит в ответ модели.

Так же для этих целей можно использовать композиции деревьев в случайном лесе или градиентном бустинге. В данном случае уменьшение критерия информативности для данного признака вычисляется как сумма по всем деревьям композиции. Чем больше данная сумма, тем важнее признак для данной композиции.

Жадный метод отбора. Предположим мы имеем начальное множество признаков:

$$J_1 = \{i_1, \dots, i_k, \dots, i_n\}$$

На каждой итерации мы обучаем мо-

дель и получаем оценки важности признаков. Например, для линейных моделей оценками являются веса признаков. Далее признак с наименьшей важностью удаляется и модель обучается на новом наборе признаков:

$$J_2 = \{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_n\}$$



Рис. 2 - Корреляция между временем выполнения задания и вещественными признаками.

Описание выбранных алгоритмов

Для обучения моделей использовалась популярная библиотека алгоритмов машинного обучения *scikit-learn* [10]. Были выбраны несколько регрессионных алгоритмов, представляющие различные классы. Документация к ним может быть найдена в [11].

- *Decision Tree Regressor*: данный алгоритм использует дерево решений для выполнения регрессии целевой переменной.
- *Random Forest Regressor*: модель представляет собой композицию деревьев, составленных с использованием рандомизации. Так, каждое отдельное дерево составляется при помощи случайной подвыборки объектов из обучающей выборки. Также при составлении правила разбиения узла, выбираются либо все признаки, либо случайное подмножество.
- *Lasso Regression*: линейный алгоритм, который добавляет к функции потерь l_1 -регуляризатор, позволяющий снижать влияние несущественных признаков, обнуляя их.
- *Ridge Regression*: линейный алгоритм, который добавляет к функции потерь l_2 -регуляризатор, что позволяет уменьшать величину отдельных весов.
- *Gradient Boosting Regressor*: строит композицию деревьев с помощью жадной стратегии. Алгоритм обучает новое дерево, которое корректирует ошибку предыдущего. Таким образом, ответ модели является суммой ответов всех деревьев композиции.

Алгоритм продолжает работать пока количество признаков не соответствует минимальному указанному числу или пока ошибка не начнет увеличиваться.

- *K-Nearest Neighbors*: алгоритм, который предсказывает ответ для нового объекта при помощи интерполяции значений целевых переменных k соседей данного объекта.

Выбор метрики и обучение моделей

Для оценки регрессионных моделей обычно используется стандартная метрика R^2 – коэффициент детерминации, определяемый по формуле:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

где SS_{res} – сумма квадратов остатков регрессии, SS_{tot} – общая сумма квадратов.

Но данная метрика имеет существенный недостаток. По мере того как добавляются новые признаки к существующей модели R^2 может только увеличиваться. Предположим, что мы добавили к модели новые шумовые признаки. Существует вероятность, то они объясняют часть дисперсии зависимой переменной. Поэтому для того, чтобы сравнивать оценки регрессионных моделей с разным числом признаков, применяется скорректированный (adjusted) коэффициент детерминации:

$$R_{adj}^2 = 1 - (1 - R^2) \frac{(n-1)}{(n-k-1)} \leq R^2$$

где n – количество объектов в модели, k – количество признаков.

Несмотря на то, объектов значительно больше, чем признаков, а следовательно корректирующий коэффициент несущественно снижает исходный R^2 , для корректности в

качестве критерия качества модели применяется R_{adj}^2 .

Исходный набор данных был разделен на обучающую и тестовую выборки в пропорции 7:3, соответственно. Так как данные представляют временной ряд, то стандартная k-fold кросс-валидация здесь не подходит, так как в этом случае модель будет тестироваться на валидационной выборке, которая по времени находится раньше, чем обучающая выборка. Поэтому для кросс-валидации был выбран объект *TimeSeriesSplit* с пятью разбиениями.

Скользящее среднее

Сегодня в виду повсеместного применения машинного обучения часто возникает вопрос в целесообразности применения алгоритмов МО. Поэтому для проверки целесообразности применения приведенных алгоритмов в данной задаче был применен простой алгоритм,

основанный на скользящем среднем.

Принцип работы данного алгоритма заключается в том, что для вновь пришедшей задачи, время ее выполнения рассчитывается как среднее арифметическое времени работы последних n задач пользователя,

$$t_k = \frac{1}{n} \sum_{i=1}^n t_{k-i}$$

где t_k - оценка времени выполнения k -ого задания пользователя.

Данный алгоритм был применен для значений $n = 1 \dots 9$. Если пользователь на момент отправки нового задания не имеет ни одного задания, то в качестве оценки времени выполнения задания применяется указанное пользователем значение. Для каждого значения n рассчитано соответствующее значение R^2 . Результаты представлены на рисунке 3.

Наилучший показатель $R^2 = 0.425$ достигается при $n = 3$.

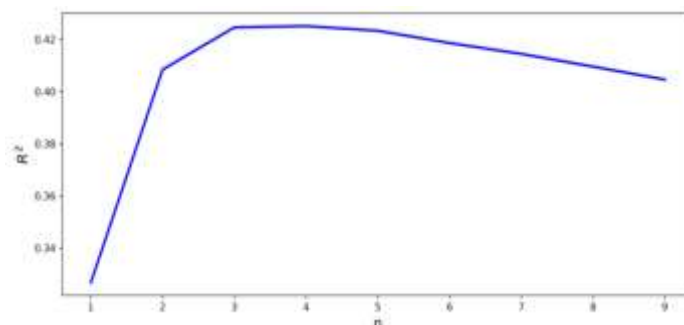


Рис. 3 – Зависимость R^2 от n (число последних задач пользователя) при использовании алгоритма скользящего среднего.

Результаты

Отбор признаков на основе моделей. Для отбора признаков была использована следующая схема: обучение на полном наборе признаков → отбор трех наиболее информативных признаков → обучение модели на выбранных признаках.

В каждом обучении для поиска параметров использовалась сетка, поэтому параметры моделей на разных наборах признаков не обязательно совпадают.

На рисунке 4а представлены результаты отбора признаков с использованием оценок различных алгоритмов. В таблице 4 представлены результаты обучения моделей на полном

наборе признаков и на трех самых информативных признаках.

Так как алгоритм К-ближайших соседей не оценивает важность признаков, то для сравнения был использован набор из трех признаков, отобранных градиентным бустингом.

Все алгоритмы на основе решающих деревьев выбирают основным признаком *n_time* (время, заказанное пользователем). Линейные алгоритмы ставят его на второе место, в то время как *duration_m_per_user* (среднее время длительности задания для пользователя) они оценивают выше всего.

Признак *time_exceeded_share_per_user* (средняя доля

неиспользованного заказанного времени для пользователя) является вторым по значимости признаком для всех алгоритмов, основанных на дереве решений. Данные алгоритмы показали лучший результат. Стоит отметить, что в смежных работах дерево решений так же часто оказывается лидером. Это может быть связано с тем, что значительная доля признаков является категориальными, это облегчает задачу разделения по вершинам дерева, но усложняет регрессию.

Необходимо отметить, что *userid*, *orgid*, *gid* являются плохими предикторами времени работы задания.

Из временных признаков *month* является самым информативным, что подтверждает предположение о сезонных тенденциях в запусах.

Главным выводом является то, что модель, обученная на меньшем числе признаков показывает такой же или больший результат. Следовательно, значительная часть выбранных признаков является малоинформативными.

Жадный отбор признаков. Отбор признаков производился с помощью объекта RFECV (Recursive Feature Elimination with Cross Validation) из библиотеки *sklearn*. Данный алгоритм производит отбор признаков, производя кросс-валидацию на каждой итерации, и исключает один из признаков. В качестве минимального количества признаков, которые необходимо оставить, выбрано $n = 3$. RFECV исключает признаки, пока не останется три или пока метрика не станет уменьшаться.

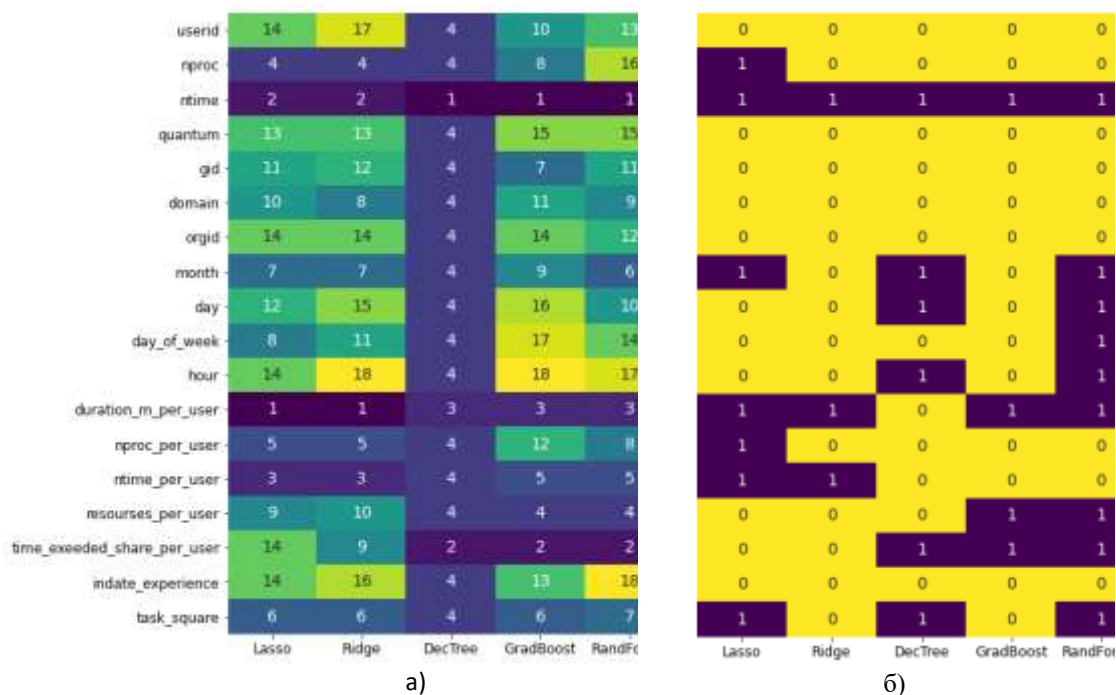


Рис. 4 – Результаты отбора признаков.

- а - Результат ранжирования признаков по степени информативности на основе моделей.
 б - Результаты жадного отбора признаков (1 – признак информативен, 0 – не информативен).

Таблица 4 – Результаты обучения моделей на полном наборе признаков и на отобранных на основе моделей.

	Lasso	Ridge	Dec. Tree	Grad. Boost.	Rand. Forest	KNN
R_{adj}^2 на всех признаках	0.35	0.35	0.66	0.68	0.68	0.28
R_{adj}^2 на 3-ех самых важных признаках	0.35	0.35	0.67	0.69	0.68	0.56

Таблица 5 – Результаты обучения моделей на признаках, отобранных жадным методом.

	Lasso	Ridge	Dec. Tree	Grad. Boost.	Rand. Forest
R_{adj}^2 на отобранных признаках	0.35	0.34	0.67	0.68	0.68

Использована следующая схема: жадный отбор признаков на обучающей выборке \rightarrow поиск лучших параметров модели на отобранных признаках \rightarrow окончательная оценка модели на тестовой выборке. Результаты приведены на рисунке 4б.

Отметим, что *duration_m* и *n_time* снова оказываются в лидерах. Также *month* и *time_exceeded_share_per_user* оцениваются как важные тремя алгоритмами из пяти.

В таблице 5 приведены результаты данных моделей. Они остаются почти идентичными к результатам в таблице 4.

Из результатов очевидно, что персональные признаки являются более информативными по сравнению с временными.

Интересно, что такие признаки как *user_id*, *orgid*, *gid* оказались не востребованными. Можно предположить, что персональные признаки помогают модели обобщать группы пользователей со схожей статистикой, в то время как *id* группы или пользователя является категориальным признаком и модель не может объединять их в группы. Также может играть роль небольшое число пользователей в данных $n = 264$ и организаций $n = 54$.

Отдельное внимание стоит обратить на скользящее среднее. Данный алгоритм демонстрирует приемлемый для своей простоты результат. Значение R^2 для него оказывается выше, чем для всех моделей, основанных на линейной регрессии. Это может объясняться спецификой пользователей кластера МСЦ РАН. Можно предположить, что пользователи часто запускают задания со схожим временем выполнения или в случае ошибки перезапускают задание снова и снова, поэтому среднее арифметическое времени работы трех последних заданий показывает такие хорошие результаты. Несомненным плюсом данного алгоритма является его интерпретируемость.

Дальнейшая работа

В дальнейшем планируется

использование полученных моделей для работы с потоком заданий, смоделированных с помощью симулятора, например, симулятор SLURM [12]. Это позволит дать конечную оценку для моделей, так как позволит рассчитать и сравнить показатели использования ресурсов.

Заключение

В работе приведены результаты применения алгоритмов МО для предсказания времени работы заданий, поступающих от пользователей. В качестве обучающей выборки были использованы данные суперкомпьютера МВС-10П «Торнадо» за 2013 – 2017 гг.

Для сравнения выбраны несколько алгоритмов, которые относятся к различным классами. Приведено сравнение алгоритмов на основе скорректированного коэффициента детерминации R_{adj}^2 . Лучшими алгоритмами оказались Decision Tree Regressor, Random Forest Regressor, Gradient Boosting.

В работе приведено описание методов отбора признаков. Рассмотрены методы отбора на основе моделей и жадный метод. Приведены результаты отбора.

Персональные признаки пользователя, на основе статистики пользования суперкомпьютерным кластером МСЦ РАН чаще чем временные признаки оказывались значимыми.

Также к данным был применен простой алгоритм скользящего среднего. Алгоритм показал результат, превышающий результаты моделей, основанных на линейной регрессии. Приемлемый результат данного алгоритма демонстрирует специфику пользователей МСЦ РАН. Пользователи склонны запускать задания со схожим временем выполнения, что позволяет предсказывать время выполнения эффективнее, чем с помощью линейной регрессии.

Работа выполнена при поддержке гранта РФФИ № 18-29-03236.

Comparison of machine learning algorithms for predicting the user job runtime in the framework of optimizing the use of resources of supercomputer cluster of JSCC RAS

A.A. Rybakov, S.S. Shumilin

Abstract. Assessing the resources needed to run a program on a supercomputer cluster is a difficult task. The user needs to practically guess the time of the program, which leads to the fact that some tasks are forcibly terminated by the scheduler due to exceeding the time limit. A large number of such events leads to the fact that the distribution of cluster resources loses effectiveness. In order to optimize the distribution of resources, it is necessary that the scheduler receive more accurate estimates of the task parameters. Earlier, attempts were made to use machine learning algorithms to solve this problem. This article discusses the application of these algorithms to the data of the supercomputer cluster of the JSCC RAS. The results are obtained on the quality of regression models on various additional features, among which the most informative ones were selected. To do this various techniques commonly used in machine learning were used. These features that characterize the user of the cluster can be used to improve the accuracy of already existing models.

Keywords. HPC, machine learning, predictive analytics, scheduler, user modeling.

Литература

1. Система управления прохождением параллельных заданий. Руководство программиста (пользователя), (2016). <http://www.jssc.ru/wp-content/uploads/2017/06/SUPPZ-user-guide-2016.pdf> (дата обращения: 14.02.2020).
2. Webb, G. I., Pazzani, M. J., and Billsus, D, "Machine learning for user modeling" User modeling and user-adapted interaction, vol. 11 pp. 19-29, 2001.
3. Yao, Y., Zhao, Y., Wang, J., and Han, S, "A model of machine learning based on user preference of attributes". International Conference on Rough Sets and Current Trends in Computing, pp. 587-596. Springer, Berlin, Heidelberg, 2006.
4. Warren Smith, Ian Foster, and Valerie Taylor. Predicting application run times using historical information. In Workshop Job Scheduling Strategies for Parallel Processing (JSSPP). pp. 122–142, 1998.
5. Warren Smith, Valerie Taylor, and Ian Foster. Using run-time predictions to estimate queue wait times and improve scheduler performance. In Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP). 202–219, 1999.
6. Dan Tsafirir, Yoav Etsion, and Dror G Feitelson. Backfilling using systemgenerated predictions rather than user runtime estimates. 2007 IEEE Transactions on Parallel and Distributed Systems 18, 6 (2007).
7. Dan Andresen, William Hsu, Huichen Yang, and AdedolapoOkanlawon. 2018. Machine Learning for Predictive Analytics of Compute Cluster Jobs. *CoRR* abs/1806.01116 (2018). <http://arxiv.org/abs/1806.01116> (дата обращения 20.03.20).
8. Rezaei, Mahdi &Salnikov, Alexey. (2018). Machine Learning Techniques to Perform Predictive Analytics of Task Queues Guided by Slurm. 1-6. 10.1109/GloSIC.2018.8570130.
9. Rodrigues, Eduardo & Cunha, Renato &Netto, Marco & Spriggs, Michael. (2016). Helping HPC Users Specify Job Memory Requirements via Machine Learning. 10.1109/HUST.2016.006.
10. 2019. Getting Started with Scikit-learn for Machine Learning. In *Python® MachineLearning*. John Wiley & Sons, Inc., 93–117. <https://doi.org/10.1002/9781119557500>.
11. Документация к sklearn. URL: <https://scikit-learn.org> (дата обращения 20.03.20).

12. Симулятор SLURM. URL: https://github.com/ubccr-slurm-simulator/slurm_simulator. (дата обращения 20.03.20).

Распределение температуры воздушной среды в электронной системе

А.Г. Мадера

ФГУ ФНЦ НИИСИ РАН alexmadera@mail.ru

Аннотация. Получено распределение температуры в потоке охлаждающей воздушной среды в зависимости от времени в аналитической форме

Ключевые слова. Аэродинамика, воздушная среда, электронная система, распределение температуры

Тепловые процессы в конструкции электронной системы (ЭС) определяются как мощностями энергопотребления активных элементов, так и теплообменом в потоке охлаждающей жидкостной или воздушной среды протекающей внутри конструкции ЭС. Математическое моделирование жидкостного охлаждения ЭС, осуществляемого потоком жидкости нагнетаемой по каналам (трубкам) проложенным в конструкции ЭС, разработано достаточно полно [1, 2, 4, 5, 6, 7], в то же время математическое моделирование воздушного охлаждения ЭС и ее элементов в потоке воздушной среды нагнетаемой через корпус ЭС во многом остается открытым. При проектировании радиоэлектронной аппаратуры и вычислительных комплексов, в основном принимается, что температура потока среды в ЭС является изотермической по всей конструкции ЭС и равной некоей усредненной температуре, определяемой как среднеарифметическое между температурами на входе и выходе ЭС. Точно также и в иерархическом методе моделирования априори задается линейный характер зависимости температуры воздушной среды от расстояния проходимого потоком от входа до выхода в ЭС [3]. При малых скоростях потока допущение о осредненной температуре среды или ее линейной зависимости от расстояния является приемлемым, однако, при больших скоростях среды (но с числом Маха меньшим единицы), распределение температуры в потоке среды может существенно отличаться от линейного. Сложность математического моделирования теплообмена ЭС при охлаждении ЭС воздушным потоком обуславливается различными причинами, а именно,

– сопряженным характером течения и теплообмена с элементами ЭС [4, 6],

– чрезвычайно сложной и хаотичной траекторией сквозь сформированные конструктивными и электронными элементами ЭС каналы самой разнообразной формой, переменным сечением и направлением движения,

– интервально-стохастическим характером течения потока воздуха и связанного с ним теплообмена, и др.

В отличие от иерархического метода моделирования [3], в котором распределение температуры в потоке охлаждающей жидкости не рассматривалось и заменялось усредненной изотермической температурой по всей ЭС в виде среднеарифметического входной и выходной температур потока, в данной работе определяется истинное распределение температуры в потоке движущейся жидкости, описываемое уравнением (3).

Теплообмен изотермических тел в оболочке ЭС с протекающей внутри нее средой определяется уравнением энергии в среде и уравнением, описывающим взаимодействие среды с телами системы и ее оболочкой, и взаимодействие между собой. Тем самым теплообмен между телами и средой, протекающей в ЭС и омывающей тела системы и оболочку, должно моделироваться как сопряженная задача.

Для решения указанной задачи, примем во внимание, что постоянные времени развития тепловых процессов в воздухе (τ_a) и твердотельных элементах (τ_s) ЭС (МС, ЭРЭ, элементы конструкции) связаны соотношением $\tau_a \ll \tau_s$. Это обуславливается тем, что отвечающий за динамику развития теп-

лового процесса комплекс, равный произведению плотности и теплоемкости (ρc) имеют существенно различные порядки величин, а именно: $\rho c_{\text{воздух}} \sim 1,2 \cdot 10^3$, $\rho c_{\text{вода}} \sim 4 \cdot 10^6$, $\rho c_{\text{текстолит}} \sim 2 \cdot 10^6$, $\rho c_{\text{медь}} \sim 3,4 \cdot 10^6$, Дж/м³К. Отсюда следует, что развитие тепловых процессов в твердотельных элементах происходит существенно медленнее (на три порядка), чем в воздушной среде, температура в воздухе устанавливается за время намного меньшее времени установления температуры в твердотельных элементах.

Учитывая все вышесказанное примем следующие допущения:

– температуры в твердотельных элементах описываются нестационарными уравнениями, а температура воздуха – стационарным,

– воздушная среда в стойке является несжимаемой,

– тепло, возникающее в результате вязкости воздуха пренебрежимо мало по сравнению с теплом, выделяемым активными элементами, установленными в ЭМ,

– конвективный тепловой поток существенно превосходит поток теплопроводности в воздухе по направлению его движения,

– теплообмен воздуха со стенками стойки пренебрежимо мал в сравнении с теплообменом между ЭМ и их совокупностей, составляющих панели,

– зависимостью физических характеристик воздушной среды от температуры пренебрегается,

– движение воздуха в стойке одномерно, конвекция вдоль потока по высоте стойки значительно превосходит конвекцию поперек потока.

Рассмотрим всю ЭС в целом и найдем в ней распределение температуры нагнетаемой воздушной среды по высоте, а также среднюю температуру твердотельной части ЭС. Если рассматривать ЭС целиком с усредненной температурой воздушной среды $T_a(t)$, на вход которой воздух поступает с температурой $T_{a,in}$ и выходит из ЭС с температурой $T_{a,out}(t)$, то уравнение моделирующее сопряженный теплообмен между нагнетаемой через ЭС средой и ее твердотельной частью по длине $l \in [0, c]$, отсчитываемой от начала ЭС $l = 0$, имеет вид

$$C_a U_a \frac{dT_a(l, t)}{dl} - \alpha S_a (T(t) - T_a(l, t)) = 0,$$

где S_a – суммарная площадь теплообмена ЭС; α – коэффициент теплоотдачи от суммарной теплоотдающей поверхности S_a ЭС; $U_a = G/F_a$ – средняя скорость движения воз-

духа через ЭС.

Средняя температура твердотельной части ЭС $T(t, \omega)$ определяется уравнением в пренебрежении теплообменом со стенкой ЭС:

$$C \frac{dT(t)}{dt} + J_a(T(t), T_a(t)) = P$$

где $J_a(T(t), T_a(t)) = \alpha S_a (T(t) - T_a(t))$ – конвективный тепловой поток от ЭС с температурой $T(t)$ в среду внутри ЭС с усредненной температурой $T_a(t)$; $C = \rho c V$, Вт, – полная теплоемкость конструкции ЭС объемом V и эффективными плотностью ρ и удельной теплоемкостью c .

Решая совместно оба приведенных уравнения получим:

– распределение температуры воздушной среды по высоте ЭС $l \in [0, c]$

$$T_a(l, t) = (T_{a,in} - T_a(t)) e^{-\frac{\alpha S_a l}{c_a U_a}} + T(t), \quad (1)$$

или иначе сгруппировав члены можно записать как, $l \in [0, c]$

$$T_a(l, t) = T_{a,in} e^{-\frac{\alpha S_a l}{c_a U_a}} + T(t, \omega) \left(1 - e^{-\frac{\alpha S_a l}{c_a U_a}}\right), \quad (2)$$

– температура воздушной среды на выходе из ЭС ($l = c$)

$$T_{a,out}(t) = (T_{a,in} - T(t)) A_a(c) + T(t), \quad (3)$$

или

$$T_{a,out}(t) = T_{a,in} A_a(c) + T(t)(1 - A_a(c)),$$

– разность температур среды на выходе и входе в стойку:

$$T_{a,out}(t) - T_{a,in} = T_{a,in}(A_a(c) - 1) - T(t)(A_a(c) - 1). \quad (5)$$

Уравнение для определения температуры $T(t)$, имеет вид:

$$C \frac{dT(t)}{dt} + C_a U_a \frac{1}{c} T(t)(1 - A_a(c)) = C_a U_a \frac{1}{c} T_{a,in}(1 - A_a(c)) + P, \quad (6)$$

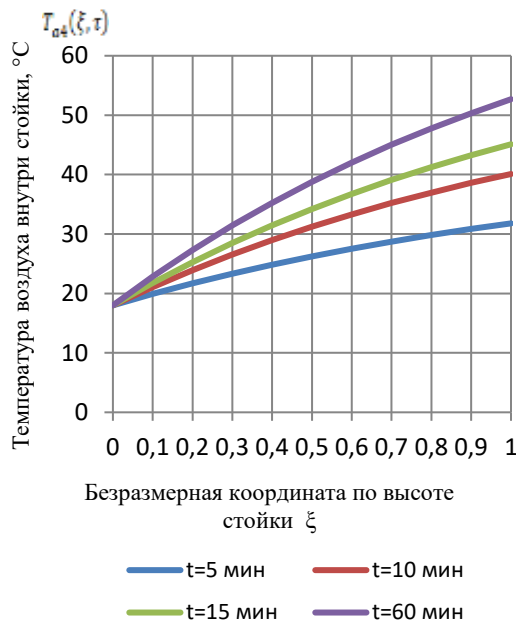
$$T_a(t = 0, \omega) = T_e(\omega),$$

где $A_a(c) = \exp(-\alpha S_a c / C_a U_a)$; c – высота ЭС.

И его решение равно:

$$T(t) = \left(T_{a,in} + \frac{c}{C_a U_a (1 - A_a(c))} \cdot P \right) \cdot \left(1 - \exp\left(-\frac{C_a U_a (1 - A_a(c)) t}{c \cdot C}\right) \right) + T_e \exp\left(-\frac{C_a U_a (1 - A_a(c)) t}{c \cdot C}\right) \quad (7)$$

Распределение температуры воздуха по высоте стойки в различные моменты времени



а)

Рис. 1. Распределение температуры воздушной среды по безразмерной координате ξ по высоте ЭС в зависимости от времени

или в иной записи после перегруппирования членов

$$T(t) = \left(T_e - T_{a,in} - \frac{c}{C_a U_a (1 - A_a)} \cdot P \right) \cdot \exp\left(-\frac{C_a U_a (1 - A_a(c)) t}{c \cdot C} \right) + T_{a,in} + \frac{c}{C_a U_a (1 - A_a(c))} \cdot P. \quad (8)$$

Подставляя выражение (7) или (8) для $T(t)$ в (1) или (2) находим распределение температуры воздушной среды $T_a(l, t)$ по высоте ЭС, а подставляя $T(t)$ в (4) или (5) находим температуру воздушной среды $T_{a,out}(t)$ на выходе ЭС.

Решение $T(t)$ в установившемся режиме при $(t \rightarrow \infty)$ имеет вид

$$T = T_{a,in} + \frac{c}{C_a U_a (1 - A_a(c))} \cdot P. \quad (9)$$

Результаты расчета распределения температуры воздушной среды по высоте ЭС в

зависимости от времени, приведены на рис. 1. Отсюда следует экспоненциальный рост температуры, в отличие от повсеместно принятого линейного. Таким образом, допущение о линейном характере зависимости температуры среды в потоке, нагнетаемом через ЭС, а тем более о усредненном, может приводить в значительным ошибкам при проектировании тепловых режимов ЭС.

Работа выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2019-0001 "Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях" (АААА-А19-119011790077-1).

Air temperature distribution in the electronic system

A.G. Madera

Abstract. The temperature distribution in the flow of cooling air is obtained as a function of time in the analytical form.

Keywords. Aerodynamics, air, electronic system, temperature distribution.

Литература

1. Л.Д. Ландау, Е.М. Лифшиц. Гидродинамика. М., Физматлит, 2017.
2. А.Г. Мадера Моделирование теплообмена в технических системах. М., НФ им.ак. В.А.Мельникова, 2005.
3. А.Г. Мадера. Иерархический метод математического моделирования стохастических тепловых процессов в сложных электронных системах. «Компьютерные исследования и моделирование», т. 11 (2019), №4, 613–630.
4. Т. Себиси, П. Брэдшоу. Конвективный теплообмен. Физические основы и вычислительные методы. М., Мир, 1987.
5. G.N. Ellison. Thermal computations for electronics. Conductive, radiative, and convective air cooling. N.Y., CRC Press, 2011.
6. H.G. Schlichting, K. Gersten. Boundary-Layer Theory. Berlin, Springer-Verlag, 2017.
7. D.B. Spalding, J. Taborek. Heat Exchanger Design Handbook. N.Y., Hemisphere Publishing Corporation, 1990.

Исследование адиабатических и неадиабатических тепловых процессов в электронных системах около тепловыделяющего элемента

¹А.Г. Мадера, ²М.Ж. Акжолов

ФГУ ФНЦ НИИСИ РАН, Москва, Россия

E-mail's: ¹omm@niisi.ras.ru, ²ak-1@mail.ru

Аннотация. Приведены результаты численного расчета конвективного теплообмена тепловыделяющих элементов с воздушной средой в электронных системах с различными граничными условиями

Ключевые слова. Электронные системы, тепловыделяющие элементы, конвективный теплообмен.

Рассматриваются процессы теплообмена между тепловыделяющими элементами (ТВЭ) и окружающей средой (воздух). Как показано в работах [1,2], при тепловом проектировании электронных систем (ЭС) необходимо учитывать такие факторы, как:

– зависимость мощностей тепловыделения ТВЭ от температуры собственного разогрева, а температуры ТВЭ, в свою очередь, от изменившейся мощности (тепловая обратная связь);

– интервально стохастический характер тепловых процессов, вызванных интервальным характером определяющих тепловые процессы параметров [2,3,4];

– неоднородность и сложность формы множества ТВЭ, входящих в конструкцию ЭС [3, 4, 5, 6] и др.;

– сопряженный характер теплообмена [7, 8, 9].

В работах [10-17] приведены результаты математического и компьютерного моделирования ламинарно-турбулентного конвективного теплообмена ТВЭ с воздушной средой внутри ЭС. При моделировании принималось, что газовая среда идеальная, вязкость отсутствует, процессы теплопроводности в газе пренебрежимо малы по сравнению с конвективными процессами.

В данной работе проводится сравнение и анализ расчетной области состоящей из области с границей являющейся неадиабатической и адиабатической (тепло изолированной). Математическая модель газодинамических процессов, протекающих при теплообмене ТВЭ со средой моделируется системой уравнений газовой динамики Эйлера

и решается численно методом крупных частиц [17].

Схематически расчетная область показана на рисунке (рис.1). Размеры расчетной

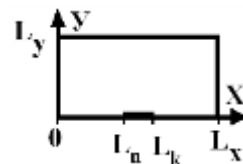


Рис. 1. Геометрия расчетной области

области составляли $L_x = 0.18$ м, $L_y = 0.09$ м, у основания области $y = 0$ в первой расчетной ячейке при $L_n \leq x \leq L_k$ находится ТВЭ. На нижней границе участка в интервале $[L_k, L_n]$ задан источник тепла (ТВЭ) длиной 40 мм. Начальная температура воздуха $T_0 = 300^\circ\text{K}$. Расчетная область разбивалась на 90×45 квадратных ячеек со сторонами $d_x = d_y = 0.002$ м, шаг по времени $dt = 10^{-8}$ сек. В мощность источника тепла (ТВЭ) равна $W=30$ Вт. Граничные условия, соответствующие нагреванию в источнике тепла задаются в виде [10]:

$$E(x_{ep}, y_{ep}, t) = \frac{Wt}{\rho(x_{ep}, y_{ep}, t)dxdy},$$

где E – полная энергия, ρ – плотность газа.

I. Неадиабатические условия

В качестве начальных условий используются параметры невозмущенного потока, которые задаются во всех расчетных точках области:

$$\rho_{ij}^0 = \rho_\infty^0 = 1.225 \text{ кг/м}^3,$$
$$u_{ij}^0 = 0.0,$$

$$\begin{aligned}v_{i,j}^0 &= 0.0, \\P_{i,j}^0 &= 1.01 \cdot 10^5 \text{ Па}, \\E_{i,j}^0 &= E_{\infty}^0 = P_{i,j}^0 / [(\gamma-1) \rho_{i,j}^0], \\T_{i,j}^0 &= 300 \text{ К},\end{aligned}$$

где u, v – скорости по оси ОХ и ОУ, P – давление, $i=1,2,3,\dots,N1$; $j=1,2,3,\dots,M1$ (рис.2).

В расчетном поле воздух покоится, значения плотности, давления, температура и полной энергий постоянны. На внешних границах расчетной области задавались следующие условия: векторы скорости на фиктивных ячейках (рис.2) были приравнены нулю, а скалярные величины плотности, давления, полной энергии и температуры сохранялись неизменными и соответствующими начальным условиям. На всех внешних границах применялись экстраполяции нулевого порядка.

Условия на границах расчетной области соответствуют (рис. 2):

– на левой границе АВ

$$\begin{aligned}\rho_{1,j}^n &= \rho_{\infty}^0, \\u_{1,j}^n &= 0.0, \\v_{1,j}^n &= 0.0, \\E_{1,j}^n &= E_{\infty}^0,\end{aligned}$$

– на нижней границе AD

$$\begin{aligned}\rho_{i,1}^n &= \rho_{\infty}^0, \\u_{i,1}^n &= 0.0, \\v_{i,1}^n &= 0.0, \\E_{i,1}^n &= E_{\infty}^0,\end{aligned}$$

– на верхней внешней границе ВС расчетного поля

$$\begin{aligned}\rho_{i,M1}^n &= \rho_{\infty}^0, \\u_{i,M1}^n &= 0.0, \\v_{i,M1}^n &= 0.0, \\E_{i,M1}^n &= E_{\infty}^0,\end{aligned}$$

– граничные условия на правой внешней границе CD расчетного поля:

$$\begin{aligned}\rho_{N1,j}^n &= \rho_{\infty}^0, \\u_{N1,j}^n &= 0.0, \\v_{N1,j}^n &= 0.0, \\E_{N1,j}^n &= E_{\infty}^0.\end{aligned}$$

Поток массы через правую границу CD определяется следующими соотношениями:

$$\Delta M_{(N-1)/2,j}^n = \rho_{N1-1,j}^n \frac{u_{N1-1,j}^n + u_{N1,j}^n}{2} \Delta t \Delta y,$$

если $u_{N1-1,j}^n + u_{N1,j}^n \geq 0$ и

$$\Delta M_{(N-1)/2,j}^n = \rho_{N1,j}^n \frac{u_{N1-1,j}^n + u_{N1,j}^n}{2} \Delta t \Delta y,$$

если $u_{N1-1,j}^n + u_{N1,j}^n < 0$.

Потоки массы на левой, нижней и верхних границах определялись аналогичными образами. Отметим, что потоки массы несут с собой потоки энергии, импульса и

тепла во внешнюю среду.

II. Адиабатическая условия

Адиабатические процессы обеспечивались следующими граничными условиями на внешних границах области (рис. 2):

– на левой границе АВ

$$\begin{aligned}\rho_{1,j}^n &= \rho_{2,j}^n, \\u_{1,j}^n &= -u_{2,j}^n, \\v_{1,j}^n &= -v_{2,j}^n, \\E_{1,j}^n &= E_{2,j}^n,\end{aligned}$$

– на нижней границе AD:

$$\begin{aligned}\rho_{i,1}^n &= \rho_{i,2}^n, \\u_{i,1}^n &= u_{i,2}^n, \\v_{i,1}^n &= -\gamma u_{i,2}^n, \\E_{i,1}^n &= E_{i,2}^n,\end{aligned}$$

– на верхней внешней границе ВС расчетного поля

$$\begin{aligned}\rho_{i,M1}^n &= \rho_{i,M1-1}^n, \\u_{i,M1}^n &= u_{i,M1-1}^n, \\v_{i,M1}^n &= v_{i,M1-1}^n, \\E_{i,M1}^n &= E_{i,M1-1}^n,\end{aligned}$$

– на правой внешней границе CD расчетного поля

$$\begin{aligned}\rho_{N1,j}^n &= \rho_{N1-1,j}^n, \\u_{N1,j}^n &= -u_{N1-1,j}^n, \\v_{N1,j}^n &= v_{N1-1,j}^n, \\E_{N1,j}^n &= E_{\infty}^0.\end{aligned}$$

Приведенные граничные условия обеспечивают отсутствие перетекания массы через внешние границы (твердая стенка), равно как энергии, импульса и тепла.

Результаты и анализ

На рис. 3а-7а приведены расчетные значения температурного поля воздуха в области с источником тепла при неадиабатическом процессе, а на рис. 3б-7б – при адиабатическом процессе в один и тот же момент времени $t=0.20$ сек, $t=0.50$ сек, $t=1.00$ сек, $t=1.50$ сек, $t=2.00$ сек соответственно. По вертикальной оси отложены значения температуры ($^{\circ}\text{C}$), по горизонтальной оси значения координат относительно расчетной области. Справа приведена черная белая палитра с оттенками серого идентифицирующая значения температуры в различных участках процесса и области.

Анализ полученных результатов показывает следующее. При нагреве области, занятой ТВЭ, над ним развивается конвективный процесс, происходящий под действием давления и сил гравитации (максимальная скорость $\leq \sqrt{2g_0H}$, где $H=0.09$ м – высота расчетной области). Полученное изображение развития конвективных процессов в воздухе над областью с ТВЭ достаточно хорошо

отражает моделируемое физическое явление. Тепловые потоки, возникшие за счет нагрева ТВЭ развиваются плавно (монотонно) начиная с приповерхностного слоя, а дальнейшее развитие процесса во времени показывает расширение области конвекции, охватывающей все больший объем воздуха. По результатам моделирования хорошо видно, что в один и тот же момент времени размер охватываемой тепловым возмущением области изменяется в зависимости от граничных условий (неадиабатические и адиабатические) на внешних границах расчетной области. За счет потоков на внешних границах фронт охвата шире для неизолированных граничных условий по сравнению с теплоизолированной системой, а увеличение температуры самого ТВЭ и около ТВЭ для неизолированной области слабее чем для изолированной, что достаточно убедительно демонстрируется на приведенных рисунках.

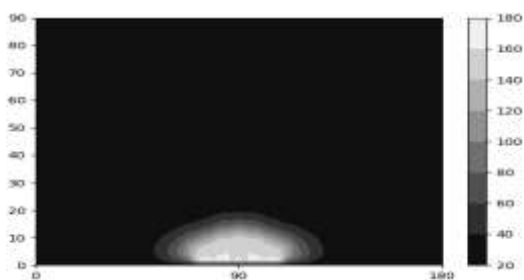


Рис.3а. Температурное поле воздуха в момент времени $t=0,20$ сек. при неадиабатическом условии.



Рис.3б. Температурное поле воздуха в момент времени $t=0,20$ сек. при адиабатическом условии.

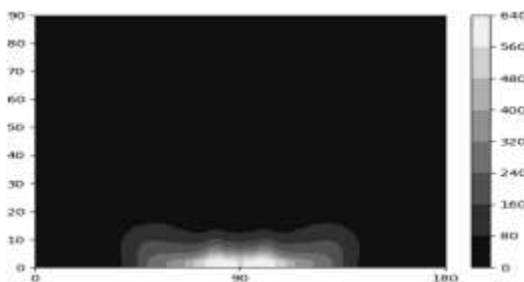


Рис.4а. Температурное поле воздуха в момент времени $t=0,50$ сек. при неадиабатическом условии.

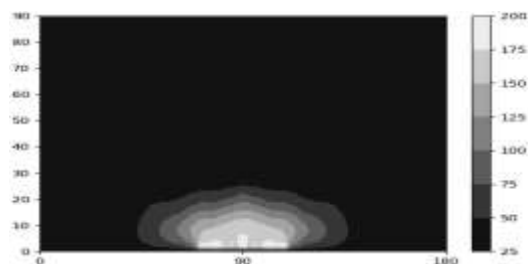


Рис. 4б. Температурное поле воздуха в момент времени $t=0,50$ сек. при адиабатическом условии.

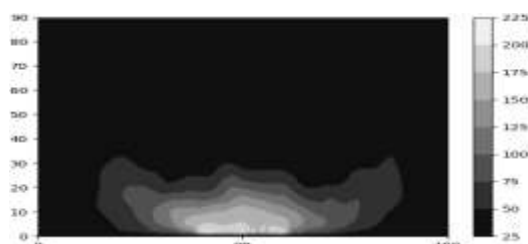


Рис.5а. Температурное поле воздуха в момент времени $t=1,00$ сек. при неадиабатическом условии.

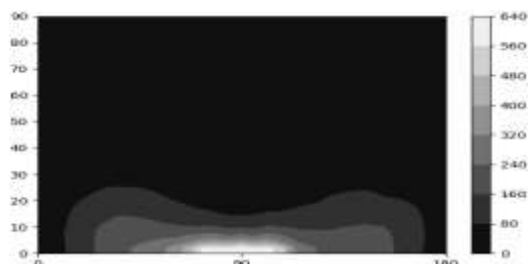


Рис.5б. Температурное поле воздуха в момент времени $t=1,00$ сек. при адиабатическом условии.

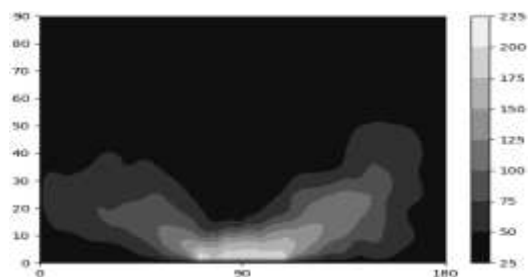


Рис.6а. Температурное поле воздуха в момент времени $t=1,50$ сек. при неадиабатическом условии.



Рис.6б. Температурное поле воздуха в момент времени $t=1,50$ сек. при неадиабатическом условии.

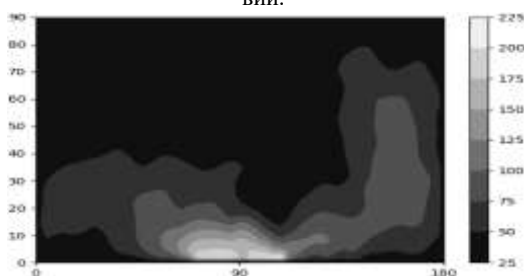


Рис.7а. Температурное поле воздуха в момент времени $t=2,00$ сек. при неадиабатическом условии.

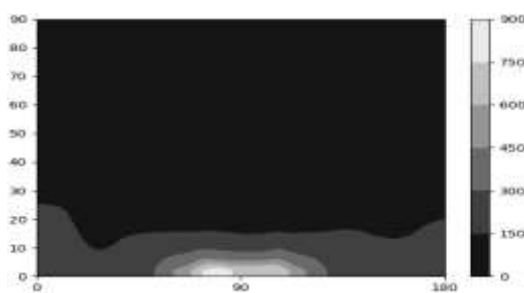


Рис.7б. Температурное поле воздуха в момент времени $t=2,00$ сек. при адиабатическом условии.

По результатам моделирования хорошо видно, что в один и тот же момент времени размеры и формы охватываемой тепловым возмущением области изменяется по разному в зависимости от внешних граничных условия т.е. адиабатические и неадиабатические процессы.

Полученное изображение развития конвективных процессов в воздухе над ТВЭ достаточно хорошо отражает моделируемое физическое явление. Тепловые потоки, возникшие за счет нагрева ТВЭ развиваются плавно (монотонно) начиная с приповерхностного слоя, а дальнейшее развитие процесса во времени показывает расширение области охвата конвекцией все большего объема воздуха.

На основе выше изложенной математической модели можно провести численные эксперименты для изучения газодинамические и тепловые процессы на ТВЭ и около него.

Работа выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2019-0001 "Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях" (AAAA-A19-119011790077-1).

Study of adiabatic and non-adiabatic thermal processes of electronic systems around heating elements

A.G. Madera, M.J. Akjolov

Abstract: The results of numerical calculation of convective heat transfer of heating elements with airflow and various boundary conditions are presented.

Keywords: electronic system, heating elements, convective heat transfer.

Литература

1. А.Г. Мадера, М.Ж. Акжолов, И.Г. Лебо Моделирование развития процессов конвекции плюс теплопроводность в воздухе вблизи процессора. // Труды НИИСИ РАН, Т. 3, №1, 2013. С. 90 – 93.
2. А.Г. Мадера Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 4. С.79 – 86

3. А.Г. Мадера, П.И. Кандалов. Анализ интервально стохастических температурных полей технических систем // Программные продукты и системы. 2014. №4. С. 41-45
4. А.Г. Мадера, П.И. Кандалов Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. Т.6. №5. С. 225-229
5. А.Г. Мадера Моделирование теплообмена в технических системах. –М.: НФ им. ак. В.А. Мельникова, 2005.
6. А.Г. Мадера, П.И. Кандалов Моделирование трехмерных температурных полей в электронных модулях // Программные продукты и системы. 2010. №2. С. 36
7. A.G. Madera Simulation of stochastic heat conduction processes // International Journal of Heat and Mass Transfer. 1994. Т. 37. № 16. С. 2571 – 2577.
8. А.Г. Мадера Математическое моделирование свободного конвективного теплообмена в электронных системах // Труды НИИСИ РАН, Том 1 №1, 2011. С. 31 – 37.
9. A.G. Madera Heat transfer from an extended surface at a stochastic heat-transfer coefficient and stochastic environmental temperature // International Journal of Engineering Science. 1996. Т. 34. № 9. С. 1093 – 1099.
10. М.Ж. Акжолов Численное моделирование конвективного теплообмена в электронной системе при различных мощностях источника тепла. // Труды НИИСИ РАН, Том 6 №1, 2016. С. 62 – 63.
11. М.Ж. Акжолов Математическое моделирование конвективного теплообмена тепловыделяющего элемента с воздушной средой в электронной системе. // Труды НИИСИ РАН, Том 5 №2, 2015. С. 103 – 105.
12. М.Ж. Акжолов, П.И. Кандалов, И.Г. Лебо, А.Г. Мадера Компьютерное моделирование конвективных процессов в воздушной среде вблизи электронных устройств. // Труды НИИСИ РАН, Т. 1, №2, 2011. С. 44 – 46.
13. М.Ж. Акжолов Влияние мощностей потребления и температуры окружающей среды на интегрально-стохастические тепловые процессы в электронном модуле. (статья) // Труды. Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН). том 7 №4 ISSN 2225-7349. Москва 2017. с.51-56.
14. М.Ж. Акжолов Анализ конвективного теплообмена с воздушной средой в электронных системах методом крупных частиц. Седьмой Российской национальной конференции по теплообмену (РНКТ-7): том 1. (22—26 октября 2018 г., Москва). С. 57-61.
15. М.Ж. Акжолов Воздействие температуры окружающей среды и мощностей потребления на тепловые процессы электронных систем в условиях стохастической неопределенности. Седьмой Российской национальной конференции по теплообмену (РНКТ-7): том 3. (22—26 октября 2018 г., Москва). С. 93-95.
16. М.Ж. Акжолов Численное исследование температурного поля воздуха при нагревании от теплового источника с учетом и без учета теплопроводности. // Труды. Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН). том 9 № 2 ISSN 2225-7349. Москва 2019. с. 46-49.
17. О.М. Белоцерковский, Ю.М. Давыдов Метод крупных частиц в газовой динамике. – М.: Наука, 1982.

Численное моделирование теплообмена тепловыделяющего элемента в окружающей среде в электронной системе при обдуве кулером

¹М.Ж. Акжолов, ²П.И. Кандалов

ФГУ ФНЦ НИИСИ РАН, Москва, Россия

E-mail's: ¹ak-1@mail.ru, ²petrki87@gmail.com

Аннотация: Приведены результаты численного моделирования конвективного теплообмена тепловыделяющих элементов и воздушной среды в электронных системах при обдуве кулером.

Ключевые слова: электронные системы, тепловыделяющие элементы, конвективный теплообмен.

Тенденция возрастания мощности потребления и соответствующего увеличения количества тепла, выделяемого рабочими элементами в современных компьютерах, приводит к необходимости исследования аэродинамики принудительного воздушного охлаждения узлов электронных систем (ЭС), как наиболее простого и до сих пор достаточно эффективного способа охлаждения. Причем при тепловом проектировании ЭС, наибольший интерес представляет охлаждение тепловыделяющих элементов (ТВЭ), как наиболее теплонагруженных элементов ЭС. В работе приводятся результаты численных исследований охлаждения ТВЭ при его обдуве кулером с заданными скоростями нагнетания.

Процессы теплообмена в ЭС между ТВЭ и окружающей воздушной средой рассматривались в [1-5], в которых принималось, что газовая среда идеальная, вязкость отсутствует, теплопроводность в газе пренебрежимо мала по сравнению с конвективным теплопереносом. Математическая модель газодинамических процессов, протекающих при теплообмене между ТВЭ и средой (воздух) моделируются системой уравнений газовой динамики Эйлера и решаются численно методом крупных частиц [6-13].

Проводились также исследования газодинамических процессов вблизи ТВЭ при неадиабатических условиях по периметру прямоугольной расчетной области с задан-

ной мощностью ТВЭ W для двух случаев: без обдува кулером (I) и с обдувом (II). Геометрия расчетной области приведена на рисунке (рис.1.). В обоих случаях размеры расчетной области и мощности источника тепла были одинаковые и составляли $L_x = 180$ мм, $L_y = 90$ мм, $W=100$ Вт (рис.1). На нижней границе участка в интервале $[L_n, L_k]$ задан

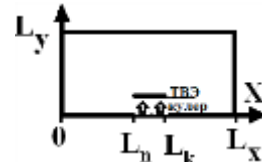


Рис. 1. Геометрия расчетной области

источник тепла (ТВЭ) длиной 40 мм. Расчетная область разбивалась на 90×45 квадратных ячеек со сторонами $d_x = d_y = 2$ мм, шаг по времени равен $dt = 10^{-8}$ сек. Начальные и граничные условия для случаев I и II. на внешних границах задавались одинаково:

– начальные условия (задавался невозмущенный воздух с нулевыми скоростями и плотностью воздуха при температуре 300°K):

$$U(x,y,0) = 0.0 \text{ м/с};$$

$$V(x,y,0) = 0.0 \text{ м/с};$$

$$\rho(x,y,0) = \rho_0;$$

$$\rho_0 = 1.225 \text{ кг/м}^3;$$

$$E(x,y,0) = E_0;$$

$$E_0 = T_0 * C_p;$$

$$T_0 = 300 \text{ K}^0;$$

$$C_p = 1004.800 \text{ Дж/(K*кг)},$$

где U, V – скорости воздуха по осям OX и по OY соответственно, ρ – плотность, E – полная энергия, T – абсолютная температура, C_p – теплоемкость воздуха при постоянном давлении.

– граничные условия (по всему периметру прямоугольной расчетной области задавались условия невозмущенного потока всего расчетного периода $t > 0$ (условия неадиабатические)):

$$U_{гр}(t) = 0.0 \text{ м/с};$$

$$V_{гр}(t) = 0.0 \text{ м/с};$$

$$\rho_{гр}(t) = \rho_0;$$

$$E_{гр}(t) = E_0.$$

При граничных слоях внешняя среда не возмущенная, т.е. значения физических параметров сохраняются во всем расчетном временном периоде. Внешняя среда рассматривалась как резервуар с бесконечной теплоемкостью.

Тепловыделяющий источник ТВЭ задавался на участке по оси OX [L_n, L_k] на пятой ячейке по оси OY (рис.1). На этом же участке на нижнем стенке установлен кулер, воздушный поток которого направлен снизу вверх вдоль оси OY .

Скорость воздушного потока от кулера составляла $V = 5 \text{ м/с}$. Мощность ТВЭ являлась линейно убывающей функцией зависящей от скорости как: $W = W_0(aV+b)$, где $a = -0.16$, $b = 1.0$.

Граничные условия, соответствующие нагреванию в источнике тепла задаются в виде [3]:

$$E(x_{гр}, y_{гр}, t) = \frac{W_0 t}{\rho(x_{гр}, y_{гр}, t) dx dy},$$

где $W_0 = 100 \text{ Вт}$.

Результаты численного моделирования приведены на рис. 2а – 6а, на которых показаны температурные поля для случая, когда ТВЭ не обдувается кулером т.е. скорость воздуха $V=0$. На рис.2б – 6б приведены температурные поля для случая, когда ТВЭ обдувается кулером со скоростью воздуха $V=5\text{м/с}$. На рис. 2в – 6в приведены поля векторов скоростей, когда скорость обдува ТВЭ $V=5\text{м/с}$. Рис. 2 а, б, в соответствуют моменту времени $t = 0.20 \text{ сек}$, рис. 3 а, б, в соответствуют моменту времени $t = 0.50 \text{ сек}$. Рис. 5а, б, в соответствуют моменту времени $t = 1.00 \text{ сек}$, рис. 5а, б, в соответствуют моменту времени $t = 1.50 \text{ сек}$, рис. 5а, б, в соответствуют моменту времени $t = 2.00 \text{ сек}$.

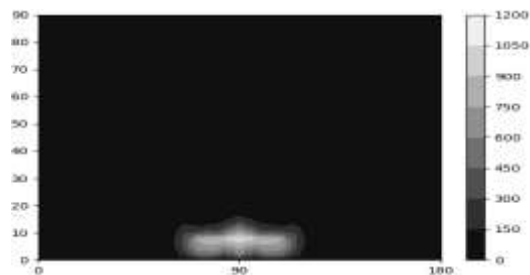


Рис.2а. Температурное поле воздуха в момент времени $t=0,20 \text{ сек}$. без охлаждения.



Рис. 2б. Температурное поле воздуха в момент времени $t=0,20 \text{ сек}$ с охлаждением.

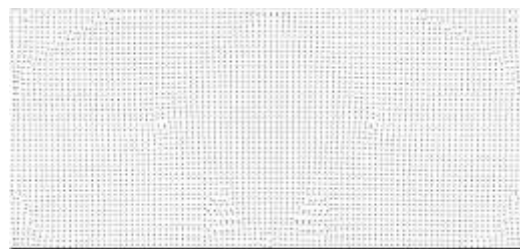


Рис.2в. Поле вектор скоростей воздуха в момент времени $t=0,20 \text{ сек}$ с охлаждением.

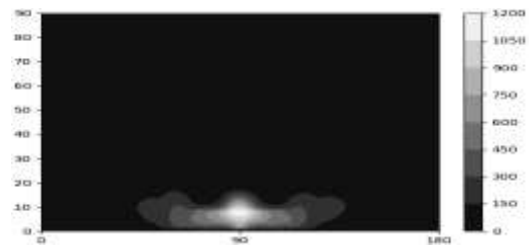


Рис.3а. Температурное поле воздуха в момент времени $t=0,50 \text{ сек}$ без охлаждения.



Рис. 3б. Температурное поле воздуха в момент времени $t=0,50 \text{ сек}$ с охлаждением.

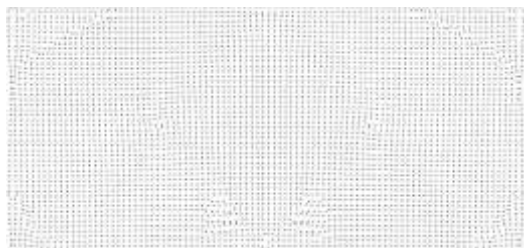


Рис.3в. Поле вектор скоростей воздуха в момент времени $t=0,50$ сек с охлаждением.

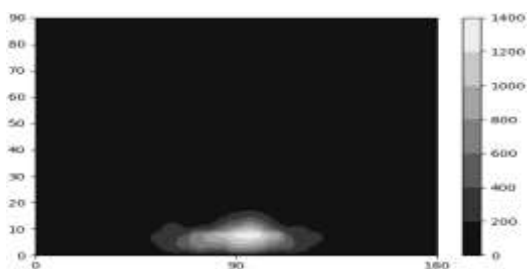


Рис.4а. Температурное поле воздуха в момент времени $t=1,00$ сек без охлаждения.



Рис.4б. Температурное поле воздуха в момент времени $t=1,00$ сек с охлаждением.

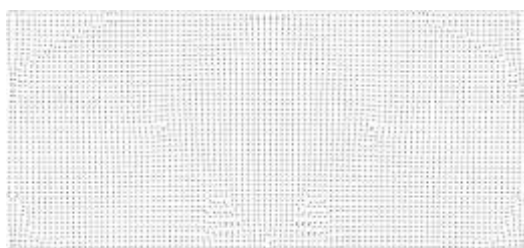


Рис.4в. Поле вектор скоростей воздуха в момент времени $t=1,00$ сек с охлаждением

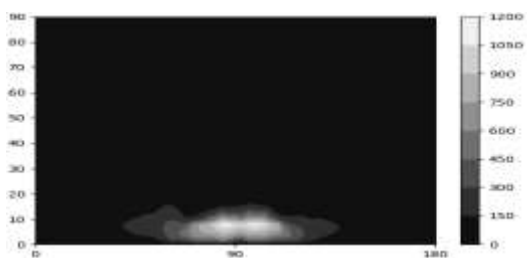


Рис.5а. Температурное поле воздуха в момент времени $t=1,50$ сек без охлаждения.

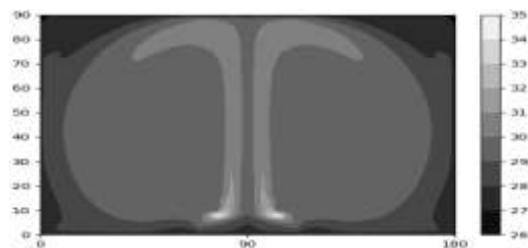


Рис.5б. Температурное поле воздуха в момент времени $t=1,50$ сек с охлаждением.

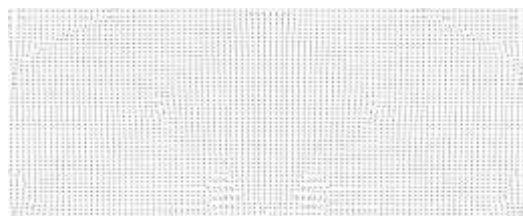


Рис.5в. Поле вектор скоростей воздуха в момент времени $t=1,50$ сек с охлаждением.

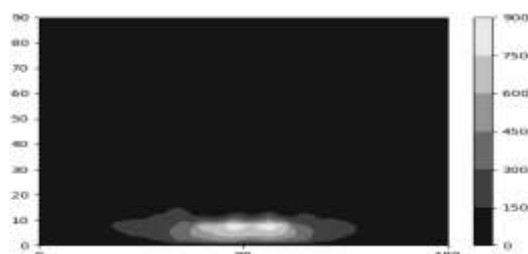


Рис.6а. Температурное поле воздуха в момент времени $t=2,00$ сек без охлаждения.



Рис.6б. Температурное поле воздуха в момент времени $t=2,00$ сек с охлаждением.

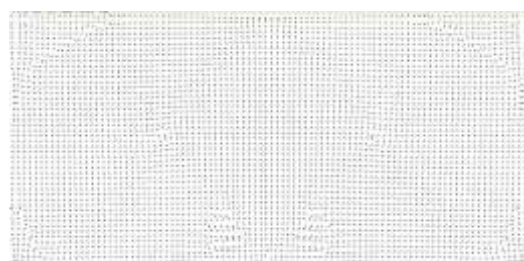


Рис.6в. Поле вектор скоростей воздуха в момент времени $t=2,00$ сек с охлаждением.

Результаты и анализ

Анализ полученных результатов (рис. 2а – 6а) показывает следующее. При нагревании области, занятой ТВЭ и при отсутствии обдува кулером, над ним развивается

конвективный процесс, который происходит за счет действия сил давления и гравитации (максимальная скорость $\leq \sqrt{2g_0H}$, где $H = 90$ мм – высота расчетной области). В этом случае тепло сконцентрировано в области возле ТВЭ и поэтому сильно нагревается, а область распространения тепла медленно расширяется со временем. Следует при этом отметить заметное влияние сил гравитационного притяжения.

На рис. 2б – 6б, видно, что при обдуве кулером в воздушных областях внутри расчетной области довольно быстро возникают циркуляционные зоны. Принудительная конвекция, создаваемая кулером, активно приводит к интенсивному охлаждению ТВЭ и быстрому расширению области охлаждаемой воздушной среды. В силу задания неадиабатических условий на внешних границах расчетной области интенсификация теплообмена с окружающей средой происходит намного быстрее.

На рис. 2в – 6в в области примыкающей к ТВЭ видны конвективные циркуляционные зоны динамических изменений полей векторов скорости имеющие форму "сердечек". Можно заметить, что на верхних и нижних, правых и левых уголках образуются застойные зоны с ослабленной динамикой.

Полученные результаты и визуализация развития конвективных процессов в воз-

духе над ТВЭ достаточно хорошо отражают моделируемое физическое явление при наличии обдува и в его отсутствии. Развитие тепловых потоков во времени, возникшее вследствие нагревания ТВЭ, зависит от наличия или отсутствия обдува кулером. Результаты моделирования свидетельствуют, что в один и тот же момент времени как размеры, так и форма области, охватываемой тепловым возмущением, существенно зависят от наличия (или отсутствия) принудительной конвекции осуществляемой кулером.

Разработанные модели и полученные результаты обеспечивают возможность проводить вычислительные эксперименты по решению задач охлаждения ЭС вычислительных систем при тепловом проектировании ЭС, размеров и размещения кулера и мощностей источников теплоты (ТВЭ).

Работа выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2019-0001 "Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях" (AAAA-A19-119011790077-1).

Numerical modeling of convective heat transfer in electronic system contained air cooler

M.J. Akjolov, P.I. Kandalov

Abstract. The results of numerical modeling of convective heat transfer for electronic system which consist of heating element blown by air cooler are presented.

Keywords. Electronic systems, heating elements, convective heat transfer.

Литература

1. А.Г. Мадера Моделирование теплообмена в технических системах. –М.: НФ им.ак. В.А.Мельникова, 2005.
2. А.Г. Мадера Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 4. С.79 – 86
3. А.Г. Мадера, П.И. Кандалов. Анализ интервально стохастических температурных полей технических систем // Программные продукты и системы. 2014. №4. С. 41-45
4. А.Г. Мадера, П.И. Кандалов Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. Т.6. №5. С. 225-229
5. A.G. Madera Simulation of stochastic heat conduction processes // International Journal of Heat and Mass Transfer. 1994. Т. 37. № 16. С. 2571 – 2577.

6. М.Ж. Акжолов Численное моделирование конвективного теплообмена в электронной системе при различных мощностях источника тепла. // Труды НИИСИ РАН, Том 6 №1, 2016. С. 62 – 63.

7. М.Ж. Акжолов Математическое моделирование конвективного теплообмена тепловыделяющего элемента с воздушной средой в электронной системе. // Труды НИИСИ РАН, Том 5 №2, 2015. С. 103 – 105.

8. М.Ж. Акжолов, П.И. Кандалов, И.Г. Лебо, А.Г. Мадера Компьютерное моделирование конвективных процессов в воздушной среде вблизи электронных устройств. // Труды НИИСИ РАН, Т. 1, №2, 2011. С. 44 – 46.

9. М.Ж. Акжолов Влияние мощностей потребления и температуры окружающей среды на интегрально-стохастические тепловые процессы в электронном модуле. (статья) // Труды. Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН). том 7 № 4 ISSN 2225-7349. Москва 2017. с.51-56.

10. М.Ж. Акжолов Анализ конвективного теплообмена с воздушной средой в электронных системах методом крупных частиц. Седьмой Российской национальной конференции по теплообмену (РНКТ-7): том 1. (22—26 октября 2018 г., Москва). С. 57-61.

11. М.Ж. Акжолов Воздействие температуры окружающей среды и мощностей потребления на тепловые процессы электронных систем в условиях стохастической неопределенности. Седьмой Российской национальной конференции по теплообмену (РНКТ-7): том 3. (22—26 октября 2018 г., Москва). С. 93-95.

12. М.Ж. Акжолов Численное исследование температурного поля воздуха при нагревании от теплового источника с учетом и без учета теплопроводности. // Труды. Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН). том 9 № 2 ISSN 2225-7349. Москва 2019. с. 46-49.

Численное исследование зависимости конвективного охлаждения тепловыделяющего элемента электронной системы от скорости воздуха при обдуве кулером

М.Ж. Акжолов

ФГУ ФНЦ НИИСИ РАН, Москва, Россия

E-mail's: ak-1@mail.ru

Аннотация. Приведены результаты численного моделирования конвективного теплообмена тепловыделяющих элементов и воздушной среды в электронных системах при обдуве кулером с различными скоростями воздуха и неизменном значении мощности источника .

Ключевые слова. Электронные системы, тепловыделяющие элементы, конвективный теплообмен, температурные и скоростные поля.

При тепловом проектировании электронных систем (ЭС), наибольший интерес представляет охлаждение тепловыделяющих элементов (ТВЭ), как наиболее теплонагруженных элементов ЭС. В работе приводятся результаты численных исследований охлаждения ТВЭ при его обдуве кулером с различными скоростями нагнетания воздуха при постоянном значении мощности ТВЭ.

Процессы теплообмена в ЭС между ТВЭ и окружающей воздушной средой достаточно подробно рассматривались в работах [1-6], в которых принималось, что газовая среда идеальная, вязкость отсутствует, теплопроводность в газе пренебрежимо мала по сравнению с конвективным теплопереносом.

Численные исследования газодинамических процессов, протекающих при теплообмене между ТВЭ и средой (воздух) с различными начальными и граничными условиями изложены в работах [7-13]

Геометрия расчетной области приведена на рисунке (рис.1.). Размеры расчетной области составляли $L_x = 180$ мм, $L_y = 90$ мм (рис.1). Расчетная область разбивалась на 90×45 квадратных ячеек со сторонами $d_x = d_y = 2$ мм, шаг по времени равен $dt = 10^{-8}$ сек. Начальные и граничные условия задавались следующим образом:

– начальные условия (задавался невозмущенный воздух с нулевыми скоростями и плотностью воздуха при температуре 300°K):

$$\begin{aligned}U(x,y,0) &= 0.0 \text{ м/с;} \\V(x,y,0) &= 0.0 \text{ м/с;} \\ \rho(x,y,0) &= \rho_0; \\ \rho_0 &= 1.225 \text{ кг/м}^3; \\ E(x,y,0) &= E_0; \\ E_0 &= T_0 * C_p; \\ T_0 &= 300 \text{ K}^0; \\ C_p &= 1004.800 \text{ Дж/(K*кг)},\end{aligned}$$

где U , V – скорости воздуха по осям OX и OY , соответственно, ρ – плотность, E – полная энергия, T – абсолютная температура, C_p – теплоемкость воздуха при постоянном давлении.

– граничные условия (по всему периметру прямоугольной расчетной области задавались условия невозмущенного потока всего расчетного периода $t > 0$ (условия неадиабатические)):

$$\begin{aligned}U_{\text{гр}}(t) &= 0.0 \text{ м/с;} \\V_{\text{гр}}(t) &= 0.0 \text{ м/с;} \\ \rho_{\text{гр}}(t) &= \rho_0;\end{aligned}$$

$$E_{гр}(t) = E_0.$$

В приграничных слоях внешняя среда не возмущенная, т.е. значения физических параметров сохраняются во всем расчетном временном периоде. Внешняя среда рассматривалась как резервуар с бесконечной теплоемкостью.

Тепловыделяющий источник ТВЭ задавался на участке по оси OX $[L_n, L_k]$ на пятой ячейке по оси OY (рис.1). На этом же участке на нижнем стенке установлен кулер, воздушный поток которого направлен снизу вверх вдоль оси OY.

Мощность ТВЭ являлась линейно убывающей функцией зависящей от скорости (V) воздуха, нагнетаемым кулером, и определялась как

$$W = W_0(aV + b),$$

где $a = -0.16$, $b = 1.0$, $W_0 = 100$ Вт.

Граничные условия, соответствующие нагреванию в источнике тепла задаются в виде [7,8]:

$$E(x_{гр}, y_{гр}, t) = \frac{Wt}{\rho(x_{гр}, y_{гр}, t)dxdy}.$$

Газодинамические процессы, протекающие при теплообмене между ТВЭ и средой (воздух) моделируются системой уравнений газовой динамики Эйлера и решаются численно методом крупных частиц [14].

Проводились подробные численные исследования газодинамических процессов, происходящих около ТВЭ при различных

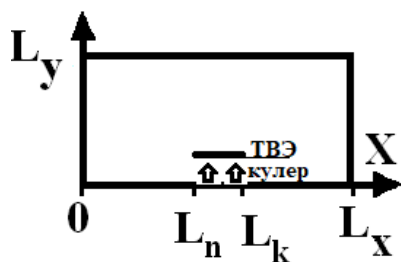


Рис. 1. Геометрия расчетной области

значениях скорости воздуха продуваемым кулером. Далее приведены результаты численных расчетов для трёх случаев: скорость воздуха продуваемые кулером $V = 1$ м/с, $V = 3$ м/с и $V = 5$ м/с при неизменном значении мощности ТВЭ $W = 100$ Вт.

Анализ полученных результатов

I. Для момента времени $t = 0.05$ сек. от начала расчета (рис. 2а-4а), показаны температурные поля, а на рис. 2б-4б поля векторов скоростей для случая, когда ТВЭ обдувается кулером при скорости воздуха $V = 1$ м/с, $V = 3$

м/с, $V = 5$ м/с, соответственно.

II. Для момента времени $t = 1.00$ сек. от начала расчета (рис. 5а-7а), показаны температурные поля, а на рис. 5а-7б поля векторов скоростей для случая, когда ТВЭ обдувается кулером при скорости воздуха $V = 1$ м/с, $V = 3$ м/с, $V = 5$ м/с, соответственно.

III. Для момент времени $t = 2.00$ сек. от начала расчета (рис. 8а-10а), показаны температурные поля, а на рис. 2б-4б поля векторов скоростей для случая, когда ТВЭ обдувается кулером при скорости воздуха $V = 1$ м/с, $V = 3$ м/с, $V = 5$ м/с, соответственно.

На рисунках, где показаны температурные поля по вертикальной оси, отложены значения температуры в °С, по горизонтальной оси значения координат относительно расчетной области. Справа приведена черно-белая палитра с оттенками серого, идентифицирующая значения температур в различных участках процесса и области.



Рис.2а. Температурное поле воздуха в момент времени $t = 0,05$ сек с охлаждением со скоростью обдува $V = 1$ м/с.

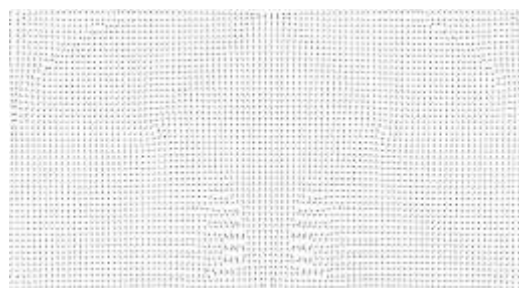


Рис. 2б. Поле вектор скоростей воздуха в момент времени $t = 0,05$ сек со скоростью обдува $V = 1$ м/с.

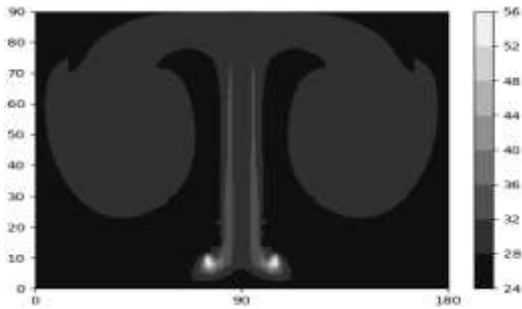


Рис.3а. Температурное поле воздуха в момент времени $t=0,05$ сек с охлаждением со скоростью обдува $V=3$ м/с.

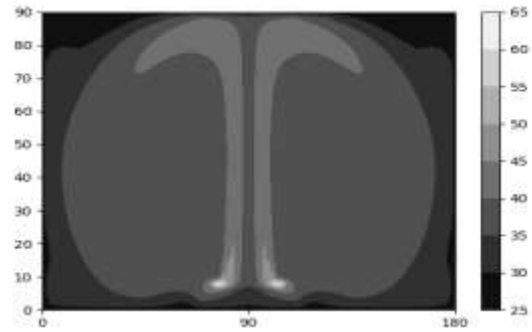


Рис.5а-5б. Температурное поле воздуха и поле вектор скоростей воздуха в момент времени $t=1,00$ сек с охлаждением со скоростью обдува $V=1$ м/с.

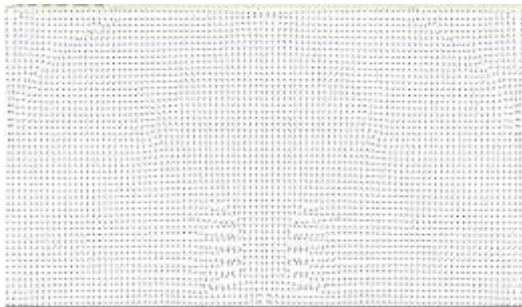


Рис.3б. Поле вектор скоростей воздуха в момент времени $t=0,05$ сек со скоростью обдува $V=3$ м/с.

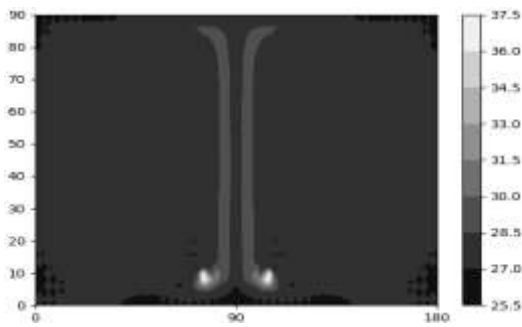
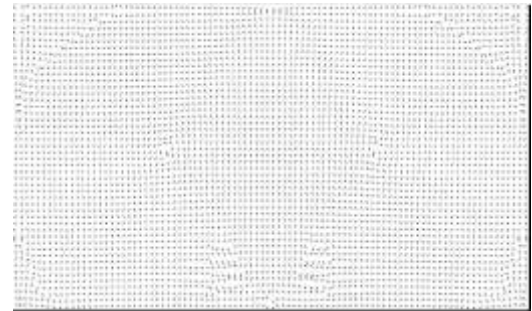


Рис.4а-4б. Температурное поле воздуха и поле вектор скоростей воздуха в момент времени $t=0,05$ сек с охлаждением со скоростью обдува $V=5$ м/с.



Рис. 6а. Температурное поле воздуха в момент времени $t=1,00$ сек с охлаждением со скоростью обдува $V=3$ м/с.

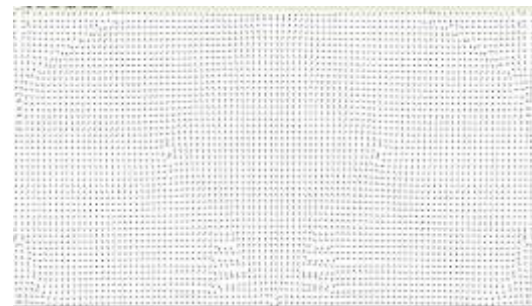
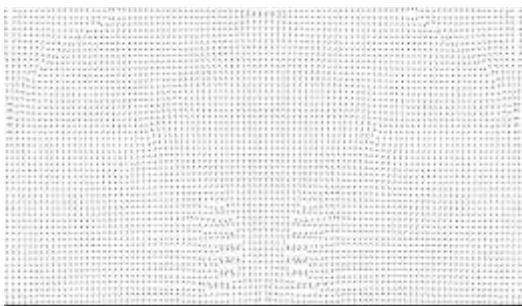


Рис.6б. Поле вектор скоростей воздуха в момент времени $t=1,00$ сек со скоростью обдува $V=3$ м/с.

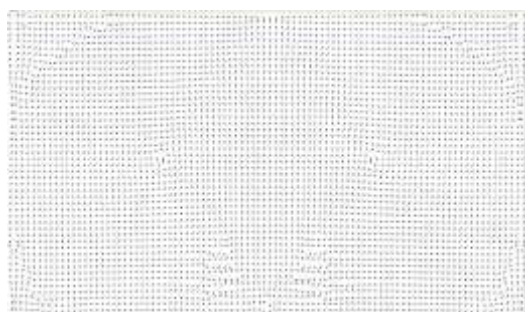
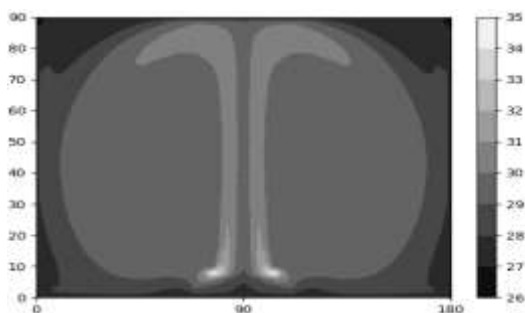


Рис.7а-7б. Температурное поле воздуха и поле вектор скоростей воздуха в момент времени $t=1,00$ сек с охлаждением со скоростью обдува $V=5$ м/с.

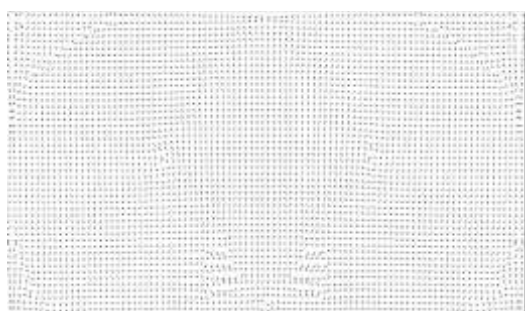


Рис.8а-8б. Температурное поле воздуха и поле вектор скоростей воздуха в момент времени $t=2,00$ сек со скоростью обдува $V=1$ м/с.

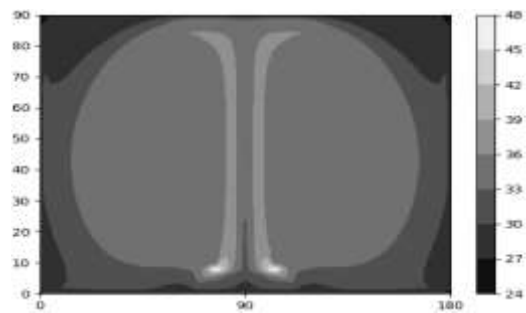


Рис.9а. Температурное поле воздуха в момент времени $t=2,00$ сек с охлаждением со скоростью обдува $V=3$ м/с.

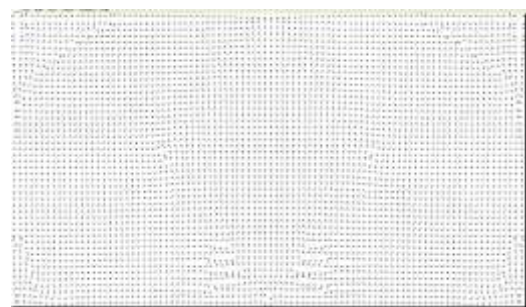


Рис.9б. Поле вектор скоростей воздуха в момент времени $t=2,00$ сек со скоростью обдува $V=3$ м/с.

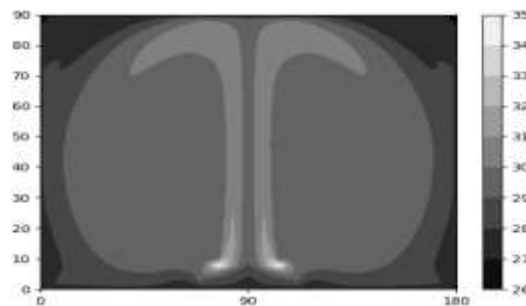


Рис.10а. Температурное поле воздуха в момент времени $t=2,00$ сек с охлаждением со скоростью обдува $V=5$ м/с.

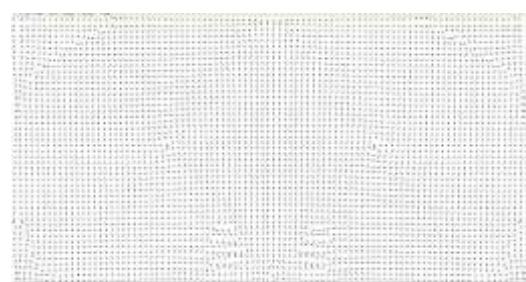


Рис.10б. Поле вектор скоростей воздуха в момент времени $t=2,00$ сек со скоростью обдува $V=5$ м/с.

На рисунках, где показаны поле вектора скорости видно, что при обдуве кулером в воздушных областях внутри расчетной области довольно быстро возникают циркуляционные зоны. Принудительная конвекция, создаваемая кулером, активно приводит к интенсивному охлаждению ТВЭ и быстрому

расширению области охлаждаемой воздушной среды. В силу задания неадиабатических условий на внешних границах расчетной области интенсификация теплообмена с окружающей средой происходит намного быстрее. В области, примыкающей к ТВЭ, видны конвективные циркуляционные зоны динамических изменений полей векторов скоростей имеющие форму "сердечек". Можно заметить, что на верхних и нижних, правых и левых уголках образуются застойные зоны с ослабленной динамикой.

На рисунках, где показаны температурные поля, можно провести количественные оценки. В таблице показаны температур в расчетных областях и на ТВЭ.

Таблица 1.

Время (сек)	Температура (°C)	Температура (°C)	Температура (°C)
	При V=1 м/с	При V=3 м/с	При V=5 м/с
0.05	24 – 72	24 – 56	25.5 – 37.5
1.00	25.5 – 65	25 – 47.5	26 – 35
2.00	25 – 65	24 – 48	26 – 35

Из таб.1 видно динамику охлаждения ТВЭ в зависимости от скорости воздуха обдуваемым кулером. Можно заметить, что после одной секунды температурные поля заметно стабилизируются

Полученные результаты и визуализация развития конвективных процессов в воз-

духе над ТВЭ достаточно хорошо отражают моделируемое физическое явление при наличии обдува с различными скоростями воздуха. Развитие тепловых потоков во времени, возникшее вследствие нагревания ТВЭ, зависит от скорости обдува кулером. Результаты моделирования свидетельствуют, что в один и тот же момент времени как размеры, так и форма области, охватываемой тепловым возмущением, существенно зависят от значений скоростей воздуха нагнетаемым кулером.

Разработанные модели и полученные результаты обеспечивают возможность проводить вычислительные эксперименты по решению задач охлаждения вычислительных систем (ВС) при их тепловом проектировании, размеров и размещения кулера и мощностей источников теплоты (ТВЭ).

Работа выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2019-0001 "Математическое обеспечение и инструментальные средства для моделирования, проектирования и разработки элементов сложных технических систем, программных комплексов и телекоммуникационных сетей в различных проблемно-ориентированных областях" (АААА-А19-119011790077-1).

Study of convective cooling of heating element in electronic system depending on cooler airspeed

M.J. Akjolov

Abstract: The results of numerical modeling of convective heat transfer for electronic system which consist of heating element blown by air cooler are presented. The electronic system was studied at various cooler airspeed and constant power dissipation of heating element.

Keywords: electronic systems, heating elements, convective heat transfer, temperature fields, fields of velocities.

Литература

1. А.Г. Мадера Моделирование теплообмена в технических системах. –М.: НФ им.ак. В.А.Мельникова, 2005.
2. А.Г. Мадера Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 4. С.79 – 86
3. А.Г. Мадера, П.И. Кандалов. Анализ интервально стохастических температурных полей технических систем // Программные продукты и системы. 2014. №4. С. 41-45

4. А.Г. Мадера, П.И. Кандалов Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. Т.6. №5. С. 225-229
5. A.G. Madera Simulation of stochastic heat conduction processes // International Journal of Heat and Mass Transfer. 1994. Т. 37. № 16. С. 2571 – 2577.
7. М.Ж. Акжолов Численное моделирование конвективного теплообмена в электронной системе при различных мощностях источника тепла. // Труды НИИСИ РАН, Том 6 №1, 2016. С. 62 – 63.
8. М.Ж. Акжолов Математическое моделирование конвективного теплообмена тепловыделяющего элемента с воздушной средой в электронной системе. // Труды НИИСИ РАН, Том 5 №2, 2015. С. 103 – 105.
9. М.Ж. Акжолов, П.И. Кандалов, И.Г. Лебо, А.Г. Мадера Компьютерное моделирование конвективных процессов в воздушной среде вблизи электронных устройств. // Труды НИИСИ РАН, Т. 1, №2, 2011. С. 44 – 46.
10. М.Ж. Акжолов Влияние мощностей потребления и температуры окружающей среды на интегрально-стохастические тепловые процессы в электронном модуле. (статья) // Труды. Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН). том 7 № 4 ISSN 2225-7349. Москва 2017. с.51-56.
11. М.Ж. Акжолов Анализ конвективного теплообмена с воздушной средой в электронных системах методом крупных частиц. Седьмой Российской национальной конференции по теплообмену (РНКТ-7): том 1. (22—26 октября 2018 г., Москва). С. 57-61.
12. М.Ж. Акжолов Воздействие температуры окружающей среды и мощностей потребления на тепловые процессы электронных систем в условиях стохастической неопределенности. Седьмой Российской национальной конференции по теплообмену (РНКТ-7): том 3. (22—26 октября 2018 г., Москва). С. 93-95.
13. М.Ж. Акжолов Численное исследование температурного поля воздуха при нагревании от теплового источника с учетом и без учета теплопроводности. // Труды. Научно-исследовательский институт системных исследований Российской академии наук (НИИСИ РАН). том 9 № 2 ISSN 2225-7349. Москва 2019. с. 46-49.
14. О.М. Белоцерковский, Ю.М. Давыдов Метод крупных частиц в газовой динамике. – М.: Наука, 1982.

Модель чувства причинности

В.Г. Редько

ФГУ ФНЦ НИИСИ РАН, Москва, Россия

E-mail: vgreddko@gmail.com

Аннотация. В статье построена и исследована модель чувства причинности автономных агентов. Рассмотрена модель эволюции популяции агентов. В популяцию входят агенты двух типов: 1) агенты с чувством причинности и 2) агенты без такого чувства. Каждый агент имеет свой собственный ресурс. Агенты с чувством причинности запоминают причинно-следственные связи между ситуациями во внешней среде. А именно, такие агенты запоминают многократно повторяющиеся пары ситуаций вида $\{A(t) \rightarrow B(t+1)\}$ (t – дискретное время), а затем, при появлении ситуации A , эти агенты готовятся к появлению ситуации B . За счет такой подготовки агенты с чувством причинности более экономно расходуют свой ресурс. Модель проанализирована путем компьютерного моделирования. Показано, что агенты с чувством причинности в процессе эволюции могут вытеснять агентов без чувства причинности из популяции.

Ключевые слова: автономные агенты, чувство причинности, эволюция, внутренний ресурс агента.

1. Введение

Познание причинно-следственных связей – одно из важных познавательных свойств человека. Но как это свойство могло бы возникнуть? В настоящей работе попытаемся подойти к исследованию этого вопроса. Будем отталкиваться от рассуждений Давида Юма. В «Исследовании о человеческом познании» (1748 г.) Юм подверг сомнению использование человеком понятия причинной связи [1]. А именно, он задался вопросом: почему, когда мы видим, что за одним явлением A постоянно следует другое B , то мы приходим к выводу, что A является причиной B ? Например, когда мы многократно наблюдаем, что Солнце освещает камень, и камень нагревается, то мы говорим, что солнечный свет есть причина нагревания камня. И почему мы можем делать вывод тогда, когда мы наблюдаем такую пару последовательных событий много раз, но не делаем вывод о причинно-следственной связи, когда мы только один раз наблюдаем такую пару?

Фактически Юм задался вопросом: что заставляет нас делать выводы о происходящих в природе явлениях? Что лежит в основе этих выводов? Юм попытался понять, откуда мы берем основание заключать, что A есть причина B . Он посмотрел на этот вопрос, как он пишет, со всех сторон и не нашел никакого другого основания, кроме некоторого внутреннего чувства привычки. То есть, имеется какое-то наше внутреннее свойство, которое заставляет нас утверждать, что если за A постоянно следует B , то A есть причина B . И это внутреннее чувство заставляет нас после того, когда мы сделали такое

умозаключение, и снова видим событие A , ожидать, что за A вновь последует и событие B .

В настоящей работе предлагается модель внутреннего чувства привычки у автономных агентов. Причем модель строится в максимально общем виде, чтобы характеризовать наиболее общие свойства понятия причинность. Модель исследуется путем компьютерного моделирования.

Отметим, что близкая модель была построена и исследована в работе [2], в которой агенты могли запоминать связи между ситуациями во внешнем мире $S(t)$, своими действиями $A(t)$ и следующими ситуациями $S(t+1)$:

$$\{S(t), A(t) \rightarrow S(t+1)\}, \quad (1)$$

где t – дискретное время, $t = 1, 2, \dots$. Рассматривались 1) агенты с определенным «чувством причинности», которые запоминали причинно-следственные связи вида (1) в своей базе знаний, и 2) агенты без такого чувства причинности. Число ситуаций во внешней среде и число возможных действий агентов предполагалось конечным. Одна из ситуаций была целевой. И рассматривался поиск целевой ситуации агентами указанных двух типов. Причем агенты с «чувством причинности» первоначально обучались, случайно находили целевую ситуацию и запоминали связи вида (1) в своей базе знаний, а агенты без чувства причинности искали целевую ситуацию просто случайным образом. Путем компьютерного моделирования было показано, что обученные агенты с рассмотренным «чувством причинности», используя свои базы знаний, находили целевую ситуа-

цию значительно быстрее, чем агенты без чувства причинности.

Однако модель работы [2] имеет и определенный недостаток. Предполагалось, что агенты не просто познавали закономерности внешнего мира, а познавали закономерности эффекта своего воздействия на внешнюю среду. И чтобы проанализировать свойства чувства причинности в более чистом виде, в настоящей работе строится и исследуется новая модель, в которой запоминаются причинно-следственные связи вида $A \rightarrow B$, независимые от действий агентов.

2. Описание новой модели

Предполагаем, что имеется эволюционирующая популяция агентов. Каждый агент имеет свой ресурс R . Рассмотрим мир (внешнюю среду агентов), в котором имеются различные ситуации $S_i(t)$. Время t дискретно. Предполагаем, что в основном ситуации случайные и чередуются во времени случайно. Но возможны и причинные связи между ситуациями. Считаем, что эпизодически появляются пары последовательно чередующихся ситуаций:

$$\{S_{1i}(t) \rightarrow S_{2j}(t+1)\}. \quad (2)$$

Эти пары представляют собой причинно-следственные связи мира. Некоторые из ситуаций-следствий S_{2j} (вторые элементы пар) существенны для агентов. В этих ситуациях агенты могут приобретать или терять существенную часть своего ресурса $R(t)$. Будем называть эти ситуации S_{2j} благоприятными, если агент приобретает в них ресурс, и неблагоприятными, если в такой ситуации агент теряет ресурс.

Рассмотрим эволюцию популяции агентов. Если в процессе жизни агента его ресурс стал меньше 0, то такой агент погибает. Если ресурс агента превысил определенный порог, то этот агент может «родить» потомка, при этом агент-родитель отдает половину своего ресурса потомку.

Считаем, что имеется два вида агентов: агенты первого вида и агенты второго вида. Агенты первого вида не имеют чувства причинности, агенты второго вида имеют чувство причинности.

Считаем, что при рождении новых агентов агент-потомок имеет тот же вид, что и агент-родитель, т.е. потомки агентов с чувством причинности также имеют чувство причинности. Потомки агентов без чувства причинности этого чувства не имеют.

Опишем свойства агентов первого вида (без чувства причинности). Эти агенты случайно попадают в существенные ситуации и, если ситуация удачная, то агент приобретает в этой ситуации ресурс Drp_1 ($R_i(t+1) = R_i(t) + Drp_1$), если ситуация неудачная, то агент теряет в ней ресурс Drm_1 ($R_i(t+1) = R_i(t) - Drm_1$).

Опишем свойства агентов второго вида (с чувством причинности). Считаем, что эти агенты могут необученными и обученными для определенных причинно-следственных связей $\{S_{1i}(t) \rightarrow S_{2j}(t+1)\}$, т.е. пар ситуаций вида (2). Изменения ресурса у необученных агентов второго вида происходит так же, как и у агентов первого вида. Считаем также, что обученный агент второго вида, запомнивший причинно-следственную связь для последовательной пары $\{S_{1i}(t) \rightarrow S_{2j}(t+1)\}$, обладает свойством подготовиться к существенной ситуации $S_{2j}(t+1)$.

Процесс обучения агентов второго вида происходит следующим образом. Когда агент первый раз наблюдает пару ситуаций $\{S_{1i}(t) \rightarrow S_{2j}(t+1)\}$, причем второй элемент пары S_{2j} существенен для агента, т.е. является благоприятной или неблагоприятной ситуацией, то агент заносит эту пару в свою базу знаний, в которой запоминает саму пару $\{S_{1i} \rightarrow S_{2j}\}$ и запоминает число наблюдений этой пары $N(S_{1i} \rightarrow S_{2j})$; естественно, что при первом наблюдении данной пары $N(S_{1i} \rightarrow S_{2j}) = 1$. При каждом повторном наблюдении этой пары, число $N(S_{1i} \rightarrow S_{2j})$ увеличивается на 1. Когда число наблюдений некоторой пары стало большим, больше некоторого порога: $N(S_{1i} \rightarrow S_{2j}) > Nth$, то считается, что агент запомнил данную причинную связь.

После такого обучения при появлении ситуации $S_{1i}(t)$ агент, запомнивший определенную причинно-следственную связь, делает предсказание $S_{1i}(t) \rightarrow S_{2j}(t+1)$ и готовится к появлению существенной ситуации $S_{2j}(t+1)$. Если ситуация S_{2j} удачная, то за счет подготовки агент приобретает ресурс Drp_2 , который больше, чем прирост ресурса Drp_1 у агента первого вида, не имеющего возможности предсказания данной ситуации и возможности подготовки к этой ситуации. В качестве аналога предсказания и подготовки к удачной ситуации можно вспомнить опыты И.П. Павлова, в которых после выработки условного рефлекса у собаки перед появлением пищи происходила подготовка к приему пищи (выделение слюны). Если ситуация S_{2j} неудачная, то у обученного агента второго вида также происходит подготовка к этой ситуации (например, агент может убежать от

опасности). За счет предсказания и подготовки уменьшение ресурса агента Drp_2 меньше, чем для агента первого вида, не запоминающего и не готовящегося заранее к неудачной ситуации.

Таким образом, агент с чувством причинности запоминает в своей базе знаний причинно-следственные связи между повторяющимися многократно парами ситуаций $S_{1i} \rightarrow S_{2j}$ и после обучения заранее готовится к появлению благоприятных и неблагоприятных ситуаций S_{2j} . За счет этого ресурс таких агентов становится больше, чем у агентов без чувства причинности.

Понятно, что в популяции агентов ресурс будут быстрее набирать агенты с чувством причинности. Агенты этого вида будут чаще рождать потомков и могут вытеснить из популяции агентов без чувства причинности.

Подчеркнем, что агенты с чувством причинности в настоящей модели прямо познают закономерности природы, закономерности, независимые от действий самих агентов. Познание этих агентов прямо соответствует представлениям Юма [1].

Агенты с чувством причинности явно используют свои познания закономерностей внешнего мира. Благодаря свойству предсказания, они (агенты второго вида) могут приспособиться к будущим событиям и за счет этого имеют преимущества по сравнению с агентами без чувства причинности. За счет этого ресурс агентов второго вида должен расти быстрее, чем у агентов первого вида. Хотя не все так просто. За счет рождения потомков (которым отдается половина ресурса) ресурс агентов второго вида может уменьшаться. Детали конкуренции агентов первого и второго вида в популяциях были проанализированы путем компьютерного моделирования.

3. Результаты компьютерного моделирования

Модель анализировалась путем компьютерного моделирования. Основные параметры моделирования определялись следующим образом.

Максимальная полная численность популяции агентов двух видов была равна 100. Общее число ситуаций внешнего мира равно 100. Число существенных пар ситуаций, т.е. число причинно-следственных связей вида (2) во внешнем мире равно 10. Изменения ресурса агентов таковы. Увеличение

ресурса первого вида и необученных агентов второго вида при попадании в благоприятную ситуацию равно $Drp_1 = 0.1$, уменьшение ресурса агентов первого вида и необученных агентов второго вида при попадании в неблагоприятную ситуацию равно $Drp_2 = 2.0$ либо $Drp_2 = 0.02$, соответственно. Считаем, что агент второго типа становится обученным определенной паре, т.е. запоминает определенную причинно-следственную связь, если он наблюдал эту связь более, чем $Nth = 10$ раз. После этого обученный агент может заранее подготовиться к появлению соответствующей благоприятной или неблагоприятной ситуации S_{2j} . Агент может родить потомка, если ресурс агента больше определенного порога Rth . Величина этого порога в компьютерных расчетах варьировалась. Начальные численности агентов первого и второго вида были примерно одинаковыми. Начальный ресурс любого агента был случайным, равномерно распределенным в интервале $[0, 10]$.

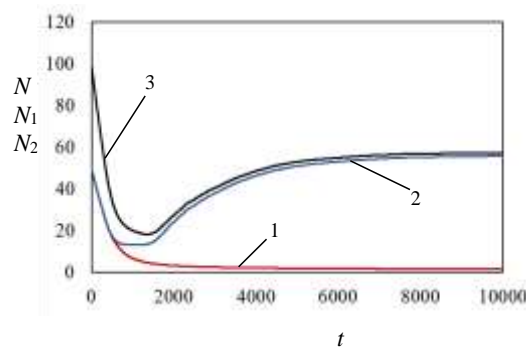


Рис. 1. Зависимости числа агентов первого вида N_1 (кривая 1) и второго вида N_2 (кривая 2), а также полной численности популяции N (кривая 3) от времени t . $Rth = 100$. Усреднено по 1000 независимым расчетам.

Типичные зависимости числа агентов первого и второго вида, а также полной численности популяции от времени представлены на рис. 1, 2.

Значение порогового ресурса, необходимого для размножения, составляло $Rth = 100$ (рис. 1) и $Rth = 10$ (рис. 2).

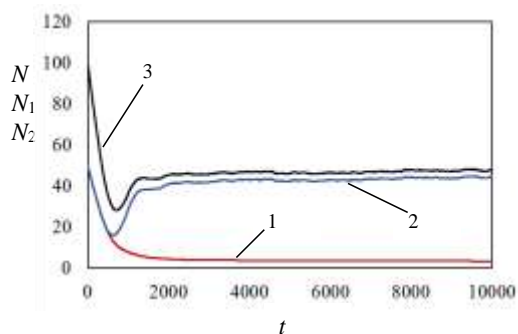


Рис. 2. Зависимости числа агентов первого вида N_1 (кривая 1) и второго вида N_2 (кривая 2), а также полной численности популяции N (кривая 3) от времени t . $R_{th} = 10$. Усреднено по 1000 независимым расчетам.

Пример динамики ресурсов агентов показан на рис. 3. Представленные зависимости усреднены по 1000 независимым расчетам, за счет усреднения ошибка расчетов не превышала 3%.

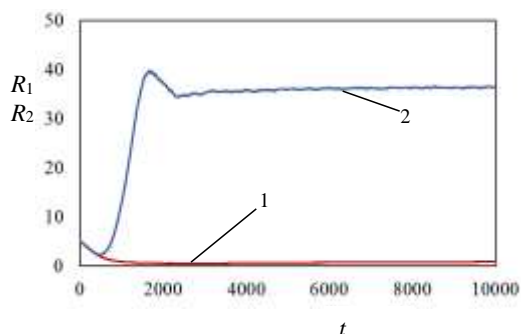


Рис. 3. Зависимости среднего по популяции ресурса агентов первого вида R_1 (кривая 1) и второго вида R_2 (кривая 2) от времени t . $R_{th} = 100$. Усреднено по 1000 независимым расчетам.

Приведенные рисунки показывают, что имеется начальный период развития популяции (порядка 1000 тактов времени), в котором как ресурсы, так и численности подпопуляций агентов первого и второго вида изменяются одинаково, независимо от вида агентов. Это означает, что сначала агенты второго вида еще не обучились, но затем, после обучения (при $t > 2000$) агенты второго вида имеют явное селективное преимущество по сравнению с агентами первого вида: их ресурс растет и они активно рожают потомков. Причем при $t \approx 2000$ (когда агенты второго вида только начинают рождать потомков, см. рис. 1) их ресурс максимален, но затем, за счет того, что эти агенты отдают половину своего ресурса потомкам, их ре-

сурс немного уменьшается.

В результате эволюции популяции агенты второго вида (с чувством причинности) в основном вытесняют агентов первого вида из популяции. Хотя небольшое число агентов первого вида в популяции остается.

Если пороговый ресурс для размножения R_{th} увеличивается, то преимущество агентов второго вида становится более выраженным, так как агенты первого вида не успевают набрать ресурс, требуемый для рождения потомков, и практически не размножаются (численность агентов первого вида значительно уменьшается, см. рис. 1). Если порог R_{th} уменьшается, то агенты начинают активно рождать потомков, отдавать им свой ресурс и их ресурс может значительно уменьшиться. Например, расчет при $R_{th} = 1$ показывает, что таком пороге все агенты быстро рожают потомков, отдают им половину своего ресурса; в результате ресурс агентов резко падает до нуля и популяция примерно за 1000 тактов времени вымирает. То есть, в настоящей модели значение порогового ресурса, необходимого для размножения, R_{th} играет важную роль.

4. Заключение

Таким образом, построена и проанализирована модель чувства причинности автономных агентов. Рассматривается эволюция популяции агентов двух видов: 1) агентов без чувства причинности и 2) агентов с чувством причинности. Агенты с чувством причинности запоминают причинно-следственные связи между существенными ситуациями во внешнем мире. Благодаря этому запоминанию, агенты с чувством причинности после обучения способны предвидеть благоприятные и неблагоприятные ситуации и заранее подготовиться к этим ситуациям. За счет такой подготовки агенты с чувством причинности увеличивают или сохраняют свой ресурс. Агенты с достаточно большим ресурсом рожают потомков.

Важно то, что агенты с чувством причинности явно используют это чувство. Они приспосабливаются к изменениям во внешней среде. Их ресурс растет быстрее, чем у агентов без чувства причинности. В результате эволюции популяции агенты с чувством причинности могут вытеснить из популяции агентов без чувства причинности.

Путем компьютерного моделирования

проанализированы условия конкуренции агентов указанных двух видов.

Подчеркнем, что использование свойства причинности, использование предсказаний – одно из ключевых свойств познания закономерностей внешнего мира. Изучение этого свойства важно при моделировании когнитивной эволюции, той эволюции, в результате которой произошло наше мышле-

ние, используемое в научном познании природы [3].

Автор благодарен Б.В. Крыжановскому за полезные замечания.

Настоящая работа выполнена при финансовой поддержке РФФИ, грант № 19-01-00331.

Model of a feeling of causality

V.G. Red'ko

Abstract. A model of a feeling of causality of autonomous agents has been created and investigated in the current article. The model of the evolution of a population of agents is considered. The population includes agents of two kinds: 1) agents with a feeling of causality and 2) agents without such a feeling. Each agent has its internal resource. Agents with a feeling of causality remember causal relationships between situations in the external environment. Namely, such agents memorize a repeated pair of situations of the form $\{A(t) \rightarrow B(t + 1)\}$ (t is the discrete time), and then, when the situation A appears, these agents prepare for the situation B . Due to this memorizing, agents with a feeling of causality spend their resources more economically. The model is analyzed by means of computer simulation. It is shown that agents with a feeling of causality in the process of evolution can displace agents without a feeling of causality from the population.

Keywords: autonomous agents, feeling of causality, evolution, agent's internal resource.

Литература

1. В.Г.Редько Моделирование чувства причинности. Первые результаты. «Труды НИИСИ РАН», т. 9 (2019), № 1, 66–68.
2. Д.Юм. Исследование о человеческом познании. Соч. в 2-х томах, т. 2. М.: Мысль, 1966, 5–169.
3. В.Г.Редько. Моделирование когнитивной эволюции: На пути к теории эволюционного происхождения мышления. Изд. 2, испр. и доп. М.: URSS, 2018.

Технология формирования мультимедийного виртуального музейного пространства

¹Н.Е. Каленов, ²С.А. Кириллов, ³А.Г. Рыбакова,
⁴И.Н. Соболевская, ⁵А.Н. Сотников

Межведомственный суперкомпьютерный центр Российской академии наук - филиал Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (МСЦ РАН - филиал ФГУ ФНЦ НИИСИ РАН)

E-mail's: ¹nekalenov@mail.ru, ²skirillov@jssc.ru, ³rybakova@jssc.ru, ⁴nikfirst@jssc.ru, ⁵ASotnikov@jssc.ru

Аннотация: в статье рассмотрены мультимедийные и интерактивные технологии, применяемые в музеех мира. Внедрение цифровых технологий в музейное пространство помогает в решении задач сохранения и популяризации культурного и научного наследия в целом. В работе приведен обзор некоторых музеев, внедривших новые мультимедийные технологии для формирования более подробного и яркого представления ряда своих фондов. Рассмотрены понятия «традиционный» музей и центр науки. Приведены сходства и различия этих объектов с точки зрения формирования и использования цифрового контента.

Ключевые слова: мультимедийный объект, электронная библиотека, Научное наследие России, 3D-модель, музейный объект, виртуальная выставка.

Одной из ключевых задач построения современного информационного общества является формирование цифрового пространства знаний. Для достижения качественно новых результатов в решении этой задачи необходимо объединение ресурсов и технического потенциала различных участников. Сетевые межинституциональные и межмузейные партнёрства позволяют поднять на качественно новый уровень просветительскую деятельность, повысить доступность для пользователей презентации научного, культурного и исторического наследия [1].

Сегодня многие музеи внедряют в свое пространство мультимедийные и интерактивные технологии, что помогает в решении задач сохранения и популяризации культурного и научного наследия в целом. Использование мультимедийных и интерактивных технологий позволяет:

1.Продемонстрировать музейные объекты, которые не включены в основную экспозицию музея (большая часть музейных предметов хранится в запасниках). Для демонстрации объектов, не входящих в экспозицию, применяется, в том числе, и технология видеомэппинга. Эта технология основана на создании аудиовизуального контента, включающего 3D-проекцию демонстрируемого объекта с учётом его геометрии и местоположения в пространстве. Видеомэппинг может создаваться под конкретный объект (например, архитектурный объект – собор, дворец и т.п.), или под «типовые»

объекты (например, гончарные изделия и т.п.). Кроме того, технология создания видеомэппинга допускает использование свойства интерактивности, т.е. возможность «взаимодействия» с человеком. Видеомэппинг позволяет создавать, например, голографические витрины и другие виды инсталляций;

2.Формировать виртуальный музейный контент, значительно расширяющий естественно-научное и культурно-просветительское пространство знаний;

3.Предоставить возможность получения более подробной и более яркой (в эмоциональном смысле) информации об объекте, явлении, историческом событии;

4.Превратить современное музейное пространство в образовательно-досуговый центр.

За последние два года круг музеев, использующих мультимедийные технологии, значительно расширился. В [2] мы описали ряд музеев, успешно применяющих различные возможности цифровых методов. Ниже приведен обзор некоторых музеев, внедривших новые мультимедийные технологии для формирования более подробного и яркого представления ряда своих фондов.

1.Третьяковская галерея [3]. В музее, в частности, используются интерактивные столы, позволяющие посетителям ознакомиться с цифровыми копиями архивных документов, к оригиналам которых невозможно предоставить открытый доступ. Также посетителям предлагается виртуальный кон-

структор, с помощью которого каждый может «создать» портрет, натюрморт или пейзаж на основе предлагаемых программой частей картин.

2. Государственный Исторический музей [4]. На одной из постоянных экспозиций применяются системы для точечной демонстрации музейных объектов. В музее также создана интерактивная инсталляция «Осколки империи», посвященная первой мировой войне.

3. Музей дизайна Купер-Хьюит (США) [5] предлагает посетителям ощутить себя дизайнером. А именно, в специальном «зале погружения» находится «черная ручка». Если поднести ее к образцу обоев на витрине, то на стене появится виртуальный интерьер выбранной эпохи. Если попробовать нарисовать этой ручкой какой-то орнамент (например, цветок), то на интерактивном столе можно услышать историю использования цветочных мотивов в дизайне помещений.

4. Национальный музей сланца (Уэльс, Великобритания) [6] на базе технологии iBeacon [7], внедрил сервис, позволяющий жителям выкладывать информацию о культурных достопримечательностях родного края. Технология iBeacon основана на рассылке автономными «маячками», работающими от батареек, по протоколу Bluetooth некоего уникального идентификатора (информации «о себе») [8].

5. Американский музей естественной истории (Нью-Йорк) [9], в частности, предлагает посетителям зала «Жизнь океанов» с помощью новейших цифровых технологий познакомиться с зарождением жизни в воде. А также, увидеть чучело голубого кита, расположенную под куполом зала, в масштабе один к одному. Кроме того, «Планетарий», представляющий собой полностью стеклянное здание с «плавающей» крышей в виде сферы, проводит сеансы мультимедийных инсталляций, воспроизводящих процесс формирования Вселенной в результате Большого взрыва.

6. Музей Ван Гога в Амстердаме [10] предлагает посетителям окунуться в атмосферу создания «Подсолнухов» посредством зеркального коридора, включающего интерактивную инсталляцию с изображением подсолнухов.

Мультимедийные и интерактивные решения для «классических» («традиционных»)

музеев (т.е. учреждений, занимающихся сбором, изучением, хранением и представлением памятников естественной истории, материальной и духовной культуры, а также просветительской и популяризаторской деятельностью являются вспомогательными инструментами или цифровым аналогом каких-либо временно недоступных объектов. При этом, в приведен обзор некоторых музеев, внедривших новые мультимедийные технологии для формирования более подробного и яркого представления ряда своих фондов. (особенно в художественных галереях) внимание посетителя полностью сосредоточено на предмете. В отсутствие экскурсовода, основное, что может понадобиться посетителю - это дополнительная информация об экспонате, представляющем научные, культурные или исторические ценности.

В то же время большой интерес вызывают современные музеи, представляющие собой, так называемые, центры науки. Здесь посетителю предлагается стать частью мира науки и самому выступить испытуемым или экспериментатором. Есть научные музеи, где, в том числе, можно посмотреть на мир глазами змеи или «почувствовать» запах лисы носом собаки, или понаблюдать за тем, как бьют молнии и образуются атмосферные разряды. Подобные интерактивные технологии предоставляют некоторые музеи, такие, как Политехнический музей, Государственный дарвиновский музей, Государственный биологический музей имени К. А. Тимирязева. В таких музеях современные мультимедийные технологии и есть экспонаты, представленные публике. Эти «зоны погружения», голографические витрины, проекторы, мультимедийные и аудио- инсталляции превращают музей не только в образовательный, но и развлекательный центр.

Таким образом, если мультимедийные технологии представляют собой хорошо продуманный инструмент, несущий информационно-познавательный характер, то внедрение этих технологий не только оправдано, но и, в некоторых случаях, необходимо. В то же время, не каждый «классический» естественно-научный музей может превратить свое пространство в развлекательно-образовательный центр.

Сегодня, на сайтах многих музеев есть раздел «Виртуальные выставки». Каждый музей сам определяет и интерпретирует понятие «виртуальная выставка». Некоторые музеи размещают в этом разделе сотни цифровых копий печатных и/или архивных документов, как это сделано, например, на сай-

те Камчатского краевого объединенного музея в разделе «виртуальные выставки» [11]. Или, например, Коми-Пермяцкий краеведческий музей им. П.И. Субботина-Пермяка в разделе «виртуальный музей» разместил 3D-панораму своих залов с кратким описанием экспонатов, находящихся в залах [12].

Другие музеи предлагают в качестве «виртуальной выставки» полный каталог, созданный по всем правилам мультимедийных технологий.

Например:

1. Музей Гуггенхайма [13] создал онлайн-коллекцию с возможностью поиска по именам и направлениям, объединив таким образом коллекции всех городов, где расположен музей, и других проектов фонда Гуггенхайма. На сайте Виртуального музея, также представлены видеоролики, связанные с тематикой музея.

2. На сайте Лувра [14] можно совершить виртуальную прогулку по нескольким залам и «посетить» виртуальную панораму Древнего Египта.

3. Музей Джорджа Вашингтона Маунт-Вернон [15] предлагает совершить «прогулку по месту, где работал и жил первый президент Америки. На сайте предложен подробный онлайн-тур, сопровождаемый аудиогидом на английском языке, а также видеоматериалами с участием актеров в костюмах конца XVIII века.

4. Американский Национальный музей естественной истории [16] предлагает виртуальную экспозицию окаменелостей древних существ, насекомых, птиц и египетских мумий

На ряду с разделами «Виртуальных выставок», представленными на сайте одного музея, может быть сформирован ресурс, объединяющий фонды нескольких музеев, связанных одной тематикой. Как например, виртуальный музей «Национальные культуры Сахалина и Курильских островов XIX–XX веков» [17]. Этот проект объединяет фонды Сахалинского областного краеведческого музея, Историко-литературного музея «А.П. Чехов и Сахалин», Сахалинского областного художественного музея, Охинского, Ногликского, Тымовского и Поронайского краеведческих музеев, а также виртуального филиала Государственного русского музея (г. Санкт-Петербург), созданного на базе Сахалинского областного художественного музея. Этот виртуальный музей позволяет пользователю сайта познакомиться с культурным наследием народов, населяющих Сахалин и Курильские острова и рассмотреть коллек-

ции предметов айнской культуры и японской культуры периода Карафутто.

Кроме краеведческих музеев, объединяющих в один виртуальный музей коллекции нескольких музеев, существуют проекты, объединяющие большие коллекции одного музея, имеющего несколько филиалов, физически удаленных друг от друга. Например, Государственный Русский музей создал портал «Виртуальный Русский музей» [18]. Этот ресурс открывает доступ к некоторым коллекциям Русского музея посредством виртуальной «прогулки» по некоторым залам музея, кроме этого, он позволяет познакомиться с некоторыми мультимедийными проектами Михайловского замка. «Посещение» каждого из представленных на сайте зала позволяет использовать «аудиогид» по данному залу.

Еще один пример современного виртуального музея – это Культурный институт Google [19]. Сегодня этот ресурс включает в себя разделы искусству, архитектуре, краеведению. Сайт позволяет познакомиться с некоторыми фондами галереи Тейт (Лондон), галереи Уффици (Флоренция), музея Метрополитен (Нью-Йорк), музея Д'Орсе (Париж) и др., а также с цифровыми копиями современного искусства, представляемого на различных площадках, в том числе с Венецианской выставки биеннале современного искусства, или с некоторыми произведениями Street Art.

Единой модели формирования виртуального выставочного пространства нет. Однако, как правило, «виртуальные выставки» различных музеев объединяют следующие цифровые объекты:

- материалы по завершённым реальным выставкам;
- материалы, подготовленные исключительно для этого раздела;
- публикации цифровых образов музейных объектов, тематически структурированные и снабженные комментариями;
- навигация по разделам выставки.

Виртуальный музей (или частный его случай – виртуальная выставка) представляет собой не самостоятельную структуру, а функциональную часть существующего музея, в которой реализуется некоторая часть музейных программ, в том числе, с целью популяризации самого музея [20].

Заключение

Создание единой методологии формирования виртуального выставочного музей-

ного пространства позволило бы эффективно интегрировать музейные объекты в открытое информационное пространство. В частности, погружать музейные коллекции в единое информационное пространство знаний [21].

Пространство знаний представляет один из важнейших элементов современного информационного общества. Предоставление доступа к объектам пространства знаний обеспечивается средствами сети интернет, где открываются широкие возможности объединения различных информационных источников, извлечения знаний и формирования на их основе виртуального информационного пространства. Электронная библиотека является эффективным средством интеграции информационных ресурсов и представляет собой комплекс технологических, технических и организационных решений, обеспечивающих формирование и предоставление широкому кругу пользователей информационных ресурсов по различным направлениям.

Одним из таких технологических ресурсов, позволяющих решать задачи по формированию и поддержке междисциплинарных виртуальных коллекций, является электронная библиотека «Научное наследие России» (ЭБ ННР). ЭБ ННР реализует открытый удаленный доступ к своим фондам для широкого круга пользователей, что существенно отличает этот сервис от многих музейных мультимедийных проектов.

В частности, на платформе ЭБ ННР

реализован совместный с Государственным биологическим музеем им. К.А. Тимирязева (ГБМТ) и Российским государственным архивом кинофотодокументов (РГАКФД) виртуальный выставочный проект «Портреты по скелетам», посвященный научному творчеству М.М. Герасимова [22]. Эта виртуальная выставка объединяет ресурсы библиотек, архивов и музея. В частности, этот проект предлагает посетителю познакомиться с историей антропологии, цифровыми печатными изданиями по антропологии, фото и видео документами, связанными с развитием антропологии и антропологических реконструкций, а также с 3D-моделями антропологических реконструкций М.М. Герасимова, хранящихся в фондах ГБМТ (рис. 1).



Рис. 1. – Стартовая страница виртуальной выставки «Портреты по скелетам»

Работа выполнена в рамках проекта РФФИ № 20-07-00773 и проекта РФФИ № 18-07-00893

Technologies of creating a multimedia virtual museums reviews

N.E. Kalenov, S.A. Kirillov, A.G. Rybakova, I.N. Sobolevskaya, A.N. Sotnikov

Annotation: the article presents multimedia and interactive technologies used in museums around the world. The digital technologies introduction into the museum space helps in solving the preserving and popularizing the cultural problems and scientific heritage in general. The work provides some museums overview that have introduced new multimedia technologies to form a more detailed and vivid presentation of a number of their collections. The concepts of a “traditional” museum and a scientific center are considered. The similarities and differences of these objects in terms of the formation and use of digital content are given.

Keywords: media item, digital library, Scientific heritage of Russia, 3D-model, museum object, virtual exhibition

Литература

1. А.Б. Антопольский, Н.Е. Каленов, В.А. Серебряков, А.Н. Сотников О едином цифровом пространстве научных знаний. Вестник Российской Академии наук, 2019, № 7, т.89, 728-735
2. С.А. Кириллов, И.Н. Соболевская О задаче формирования виртуального музейного пространства. Программные продукты, системы и алгоритмы, 2017, №4

3. <http://www.tretyakovgallery.ru/>
4. <https://shm.ru/>
5. <https://www.cooperhewitt.org/>
6. <https://museum.wales/slate/>
7. A.N. Sotnikov, I.N. Sobolevskaya, S.A. Kirillov, I.N. Cherednichenko 3D visualization technologies for web collections. CEUR Workshop Proceedings, 2018, № 2260, 438-447
8. Hammady Ramy, Minhua Ma, Powell Anna. User Experience of Markerless Augmented Reality Applications in Cultural Heritage Museums: 'MuseumEye' as a Case Study. 5th International Conference on Augmented Reality, Virtual Reality, and Computer Graphics (SALENTO AVR), Otranto, ITALY, Lecture Notes in Computer Science, 2018, v. 10851, 349-369
9. <https://www.amnh.org/>
10. <https://www.vangoghmuseum.nl>
11. <http://kamchatka-museum.ru/category/exp/virtual-exb/>
12. <http://www.subbotin-permyak.com/index.php?id=10>
13. <https://www.guggenheim.org/collection-online>
14. <https://www.louvre.fr/en/visites-en-ligne>
15. <http://www.mountvernon.org/site/virtual-tour/>
16. <http://www.mountvernon.org/site/virtual-tour/>
17. <http://sakh-virtualmuseum.ru/>
18. <https://rusmuseumvrm.ru/index.php>
19. <https://artsandculture.google.com/>
20. Cooper, P. John; Wetherelt, Andrew; Zazzaro, Chiara. From Boatyard to Museum: 3D laser scanning and digital modelling of the Qatar Museums watercraft collection. International journal of nautical archaeology, 2018, № 2, v. 47, 419-442
21. W. Lou, Pi RF., H. Wang, Y. Ju: Low-cost similarity calculation on ontology fusion in knowledge bases. Journal of information science (2019)
22. <http://acadlib.ru/>

Профилирование программ параллельной обработки сигналов для мультипроцессорных комплексов реального времени

Т.К. Грингауз¹, А.Н. Онин²

ФГУ «ФНЦ Научно-исследовательский институт системных исследований РАН»,
Отдел математического обеспечения, Москва, Россия,

E-mail's: ¹gring@niisi.ras.ru, ²alexii@niisi.ras.ru

1 – кандидат технических наук

Аннотация: В [1,2] представлена разработанная авторами технология сбора данных самоконтроля для программ параллельной обработки сигналов в мультипроцессорных комплексах реального времени с коммуникационной средой RapidIO [3]. Настоящая статья посвящена использованию данных самоконтроля для профилирования целевой программы. Разработана программа-профайлер, позволяющая: 1) графически отображать данные самоконтроля в реальном времени с выделением аномальных значений, 2) записывать поток данных самоконтроля в файл трассы, 3) просматривать трассу в замедленном темпе.

Ключевые слова: Самоконтроль системы, профилирование программ, библиотека параллельной обработки сигналов, вычислительная стадия, группа процессоров, поток данных, таймер, порт потока, стадия-аккумулятор, функция-обработчик, коммуникационная среда RapidIO.

1. Введение

Под профилем программы (далее – «целевая программа») будем понимать набор параметров, характеризующих ее поведение. Примеры таких параметров: время выполнения фрагмента кода, количество сбросов кэш-памяти в единицу времени, и т.п. Значение параметров профиля могут изменяться по ходу выполнения целевой программы. Мониторинг параметров профиля позволяет выявить нештатные ситуации. С этой целью параметры профиля измеряют периодически с заданной частотой. Результаты измерений параметров профиля будем называть данными профиля. Набор данных профиля, относящихся к одному периоду измерений, будем называть профилем там, где это не приведет к терминологической путанице. Для целевых программ с конвейерной обработкой данных естественный период измерений – итерация конвейерного цикла.

Профилирование программы - формирование ее профиля и его наглядного представления. Программу, выполняющую профилирование целевой программы, будем называть профайлером. Результат обработки

данных профиля, отображенный в цифровом или графическом виде, будем называть представлением профиля.

Термин «профилирование» будем употреблять также применительно к отдельному объекту или ресурсу программы. Под профилем объекта или ресурса будем понимать набор характеристик его состояния, поведения или эффективности использования.

В статье представлен профайлер программ параллельной обработки сигналов для мультипроцессорных комплексов реального времени. Объект профилирования – самоконтролируемые [1] мультипроцессорные программы с архитектурой БПОС [4]. Целевая программа функционирует под управлением операционной системы реального времени семейства ОС Багет 3.x [5] (далее – ОСРВ) на процессорах архитектуры КОМДИВ (1890ВМ6/7/8/9Я), объединенных коммуникационной средой RapidIO [3]. Процессоры, выполняющие целевую программу, далее будем называть «целевыми процессорами». Состав средств общего программного обеспечения и особенности их применения описаны в [6]-[9]. Профайлер функционирует на инструментальной ЭВМ под управле-

нием ОС Linux с поддержкой Java-машины (далее - «ИЭВМ»).

Профиль целевой программы формируется на основе данных самоконтроля [1,2]. Далее термины «данные профиля» и «данные самоконтроля» будем употреблять как синонимы. В соответствии с [1] данные самоконтроля собираются на целевых процессорах и пере-даются на аккумулирующий процессор в цикле вычислительного конвейера. Для передачи используется сеть RapidIO. Программа сбора данных самоконтроля [1,2] доработана. В ее новой версии появилась возможность ретрансляции аккумулированных данных по сети Ethernet. Поток данных самоконтроля целевой программы передается по сети Ethernet с аккумулирующего процессора на ИЭВМ. Там его обрабатывает профайлер.

Профайлер представляет данные самоконтроля в виде таблиц и диаграмм. Значения, свидетельствующие о нештатном поведении программы, при отображении выделяются контрастным цветом. Для визуализации профиля используется графический пакет Swing, входящий в состав средств разработки Java.

Профайлер позволяет: 1) наблюдать профиль целевой программы в реальном времени, 2) записывать поток данных самоконтроля в файл трассы, 3) формировать профиль на основе файла трассы и просматривать его в замедленном темпе.

В статье описаны состав данных профиля целевой программы, программная реализация и принцип функционирования профайлера, его графический интерфейс.

2. Архитектура целевой программы

Профайлер применим к целевым программам, разработанным на основе библиотеки БПОС [4] и программы сбора данных само-контроля СОСД [1]. Далее определенные в БПОС объекты или понятия будем употреблять с префиксом «БПОС» или с упоминанием БПОС, например: «БПОС-объект», «имя в нотации БПОС».

Архитектура программ на основе БПОС воплощает идеологию параллельного программирования в мультипроцессорной среде. Модель вычислений БПОС – конвейерная обработка данных, поступающих с фиксированной частотой. Основные объекты БПОС: процессор, группа процессоров, вычислительная стадия, поток данных. Конвейер вы-

полняет последовательность вычислительных стадий. Множество процессоров разбито на группы. Стадия может выполняться параллельно на группе процессоров. Группа процессоров выполняет одну и ту же последовательность стадий. Стадии обмениваются потоками данных. Поток данных – однонаправленная передача «отсчетов» от стадии-источника стадии-приемнику. Отсчет – трехмерный числовой массив фиксированной размерности. Отсчет может разбиваться на источнике или на приемнике по указанной пользователем размерности. В среде ОСРВ поток передает данные с процессора на процессор по каналу передачи сообщений mp (message passing) (далее – «mp-канал»). На процессоре mp-каналу соответствует порт: выходной порт на процессоре-источнике, входной порт на процессоре-приемнике. Порт - структура данных, идентифицируемая номером. В рамках процессора поддерживается сквозная нумерация портов. Передача сообщений по mp-каналу буферизуется. С портом связано кольцо буферов для отправляемых (принимаемых) сообщений. С потоком данных связаны выходные порты на процессоре-источнике, входные порты на процессоре-приемнике.

3. Состав данных профиля

В БПОС реализованы средства хронометража участков кода, а также функции опроса портов [1]. Результаты применения подобных средств по ходу выполнения целевой программы («системные» данные самоконтроля) составляют основу ее профиля.

Данные самоконтроля подразделяются на системные, пользовательские, и «данные об операциях с кэш-памятью». Системные данные определены в БПОС. Структура и семантика пользовательских данных определяются разработчиком целевой программы с использованием функций СОСД. Особенности работы с системными и пользовательскими данными описаны в [1]. Данные об операциях с кэш-памятью - новый тип данных, введенный в СОСД.

В текущей версии профайлера отображаются системные данные самоконтроля, данные об операциях с кэш-памятью и предопределенные в СОСД пользовательские данные.

К системным данным относятся:

- данные таймеров стадии БПОС;
- диагностическая информация о портах потоков;

- информация об операциях с кэш-памятью;

- значение системной переменной «errno» (код ошибки).

Пользовательские данные, предопределенные в СОСД:

- «текстовые строки»;

- «сообщения о критических ситуациях».

Таймеры - объекты БПОС, предназначенные для хронометража участков кода. Одна или несколько групп таймеров могут быть привязаны к стадии БПОС. Таймеры в группе идентифицируются номером и символическим именем. Таймер можно активировать и останавливать многократно. Таймер показывает: общее время активности (сумма всех периодов активности), минимальное и максимальное значения периодов активности, номера итераций, для которых были зарегистрированы минимальное и максимальное времена активности.

Примечание. Измерение времен в БПОС начинается по истечении некоторого числа начальных («разогревочных») стадий. Это число конфигурируется.

В БПОС предопределены таймер итерации стадии, таймер общего времени итерации, таймеры входных и выходных потоков.

Таймеры входного потока хронометрируют ожидание данных, сдачу обработанного буфера, копирование данных.

Таймеры выходного потока хронометрируют ожидание свободного буфера и сдачу обработанного буфера.

Диагностическая информация о портах входных и выходных потоков включает: число сбоев при передаче данных по заданному порту, номер последнего успешно переданного отсчета, число буферов с доступными для чтения данными (для входного порта), - число свободных буферов (для выходного порта).

Информация об операциях с кэш-памятью накапливается системными средствами на целевом процессоре по мере исполнения программы. Эффективность использования кэш-памяти характеризуется количествами операций сброса (flush), инвалидации (invalidate) и очистки (clear) за время наблюдения. Операции каждого типа классифицируются по следующим признакам: тип процесса, выполнявшего операцию (KERNEL или POSIX), объем (all, hit, miss).

Программа СОСД позволяет вести раздельный подсчет операций для всех комбинаций перечисленных типов и характеристик в разных стадиях БПОС, а также вне стадий. Для работы с данными об операциях

с кэш-памятью в СОСД реализована функция `statistic_add_cache`.

Значение переменной «errno» - код системной ошибки, формируемый операционной системой.

Пользовательские данные, предопределенные в СОСД – это два типа сообщений пользователя, различающиеся способом формирования, допустимым размером и способом обработки на аккумулирующем процессоре или в профайлере.

Максимальный размер текстовой строки – 2000 байт. Для приема текстовых строк в профайлере выделено по одному буферу на каждый процессор. Размер буфера - 2000 байт. Старые строки затираются новыми. Профайлер отображает только текущее содержимое буфера.

Максимальный размер сообщения о критической ситуации – 232 байта. Профайлер накапливает сообщения о критических ситуациях в структуре метаданных процессора (см. раздел 7.1 настоящей статьи). При выводе сообщения помечаются временем создания и номером итерации.

4. Сбор данных самоконтроля на целевых процессорах

На целевых процессорах программа работает следующим образом. На каждой итерации конвейерного цикла целевая программа с помощью функций СОСД собирает данные самоконтроля и передает их на аккумулирующий процессор.

Данные передаются пакетами NWRITE в среде RapidIO [3]. Применяются специальный протокол передачи и формат данных [1].

В качестве аккумулирующего процессора допустимо использовать процессоры с архитектурой КОМДИВ, реализованные в микросхемах 1890VM6/8/9Я. Текущая версия программы СОСД позволяет принимать на аккумулирующем процессоре потоки пакетов NWRITE от 256 целевых процессоров.

На аккумулирующем процессоре для приема пакетов NWRITE выделяется область памяти объемом 32 Мбайт. Область разделяется на 256 равных частей, по числу целевых процессоров. Целевой процессор передает поток пакетов NWRITE в предназначенную для него часть и заполняет ее по принципу кольцевого буфера (далее - «входной буфер»). Входной буфер вмещает 512 пакетов NWRITE. Запись входящих пакетов во входные буфера выполняется на аппа-

ратном уровне, без участия аккумулирующего процессора.

Выбор пакетов NWRITE из входных буферов и их обработку выполняет агент СОСД, функционирующий на аккумулирующем процессоре. Таким агентом может быть либо стадия-аккумулятор [1], либо клиентская часть Ethernet-ретранслятора (см. раздел 5 настоящей статьи). Выбор пакетов из входных буферов и их обработка производятся в асинхронном режиме. При переполнении кольцевых буферов старые данные затираются новыми.

5. Ретрансляция данных самоконтроля в сеть Ethernet

Если на аккумулирующем процессоре нагрузка на сеть RapidIO слишком велика, можно организовать доставку части данных самоконтроля по сети Ethernet. Для этого в СОСД реализована ретрансляция пакетов NWRITE из сети RapidIO в сеть Ethernet. Используется клиент-серверная технология передачи данных по протоколу TCP/IP.

С целевого процессора данные самоконтроля отправляются по сети RapidIO на процессор-ретранслятор. Ретранслятором может быть процессор с архитектурой 1890BM6/8/9Я. На аппаратном уровне прием потоков NWRITE на процессоре-ретрансляторе организован так же, как на процессоре-аккумуляторе (см. раздел 4 настоящей статьи).

На процессоре-ретрансляторе должен функционировать поток клиента-ретранслятора, на аккумулирующем процессоре - поток сервера-ретранслятора. Клиент-ретранслятор устанавливает соединение с сервером-ретранслятором по протоколу TCP/IP.

Клиент-ретранслятор в бесконечном цикле обходит входные буфера, в которые поступают пакеты NWRITE. При каждом проходе вновь поступившие пакеты выбираются из буферов и перенаправляются на аккумулирующий процессор по протоколу TCP/IP. Пакет NWRITE состоит из заголовка и тела. Адрес памяти, по которому пакет помещается во входной буфер, содержится в заголовке. Аппаратура копирует во входной буфер только тела входящих пакетов, заголовки утрачиваются. Клиент-ретранслятор перед отправкой пакета по TCP/IP реконструирует его NWRITE-заголовок и заполняет поле адреса.

На аккумулирующем процессоре выделены входные буфера для приема пакетов NWRITE по RapidIO (см. раздел 4 настоящей статьи). Сервер-ретранслятор получает поток пакетов формата NWRITE по сети Ethernet, отбрасывает заголовки и сохраняет тела пакетов во входных буферах в соответствии с полем адреса заголовка.

Схема передачи данных самоконтроля с использованием ретранслятора приведена на рисунке 1.

6. Передача данных самоконтроля на ИЭВМ

Для того, чтобы передать поток данных самоконтроля с аккумулирующего процессора в профайлер, необходимо на аккумулирующем процессоре запустить клиент-ретранслятор. Клиент-ретранслятор должен установить соединение TCP/IP по порту 1330 с ИЭВМ, на которой функционирует профайлер.

После подключения к профайлеру клиент-ретранслятор передает данные в соответствии с протоколом, определенным в СОСД. В первую очередь передаются статические данные о БПОС-объектах программы (группы процессоров, вычислительные стадии, потоки). Затем передается поток динамических данных, предназначенных для профилирования.

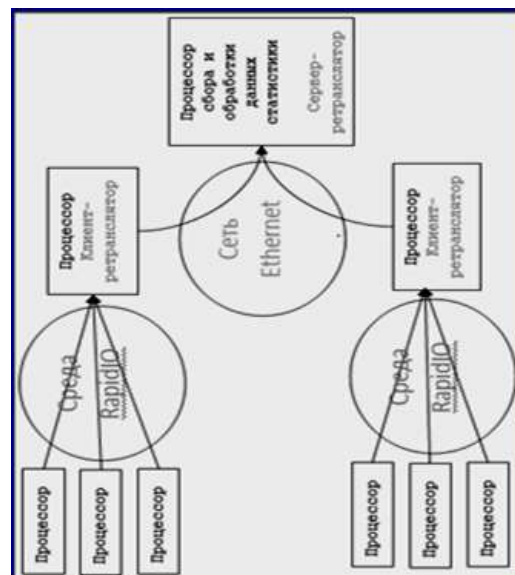


Рис.1 Передача данных самоконтроля с использованием ретранслятора

7. Архитектура и принцип работы профайлера

7.1 Потоки управления

Профайлер функционирует на ИЭВМ под управлением Java-машины версии 1.8 и выше. Получает данные самоконтроля целевой программы по сетевому протоколу TCP/IP от программы СОСД. Представляет собой многооконное графическое приложение с интерактивным интерфейсом. Отображает данные в виде таблиц, диаграмм и графических схем.

Профайлер реализован как многопоточное приложение. В его состав входят следующие потоки управления:

- поток TCP-сервера,
- потоки графических окон,
- поток трассы.

TCP-сервер выполняет следующие действия:

- прослушивает сокет Ethernet TCP с номером порта 1330 и устанавливает соединения с подключившимся клиентом. Возможные клиенты - клиент-ретранслятор СОСД или поток трассы профайлера;

- по команде пользователя устанавливает или сбрасывает режим записи трассы в указанный пользователем файл («файл трассы»),

- принимает данные по установленному соединению. Данные – это поток пакетов формата NWRITE. Обработывает каждый пакет следующим образом:

- если установлен режим записи трассы, добавляет текущий пакет NWRITE в файл трассы,

- извлекает данные самоконтроля из пакета NWRITE и интерпретирует их в соответствии с протоколом передачи и форматом данных СОСД [1],

- сохраняет данные профиля в структурах метаданных [1].

Профайлер создает структуру метаданных для каждого целевого процессора, передавшего данные на аккумулирующий процессор. Логически можно рассматривать структуру метаданных как таблицу. В строке таблицы записываются данные профиля, накопленные за одну итерацию БПОС на целевом процессоре: данные таймеров, портов, данные об операциях с кэш-памятью, сообщения о критических ситуациях, отображаемые пользовательские данные. Таблица рассчитана на хранение данных 1000 итераций. Таблица заполняется по принципу

кольцевого буфера. Количественные характеристики: целевых процессоров – до 256, таймеров – до 256 (на один процессор), портов потоков данных – до 256 (на один процессор), хранимых элементов информации об операциях с кэш-памятью – до 256 (на один процессор). Значения переменной егпно и текстовые сообщения не буферизуются. В структуре метаданных хранятся актуальное значение егпно и текущее текстовое сообщение. Значение егпно обновляется с каждым новым пакетом NWRITE.

Тракт данных профиля изображен на рисунке 2.

Потоки графических окон отображают и динамически обновляют представления профиля в окнах графического интерфейса профайлера. Поток графического окна извлекает данные профиля из структур метаданных в асинхронном режиме. Содержимое структур метаданных обновляется в темпе поступления данных в профайлер. Частота обращения потока окна к метаданным определяется производительностью и загруженностью процессора ИЭВМ. Настройка профайлера позволяет установить дополнительную задержку обновления информации в графических окнах в диапазоне от 0.1 до 3 секунд. Представление профиля отображает только те данные, которые находились в кольцевом буфере в моменты обращения потока.

Профайлер позволяет следить за профилем целевой программы в реальном времени. Такой режим позволяет своевременно обнаружить нештатную ситуацию. Если же ситуацию нужно исследовать, то может потребоваться повторный просмотр динамики изменений профиля. В этом случае используется файл трассы. Просматривая трассу в замедленном темпе, можно сопоставлять разные представления профиля или сравнивать профили, относящиеся к разным периодам наблюдения. Подачу данных трассы в профайлер выполняет поток трассы.

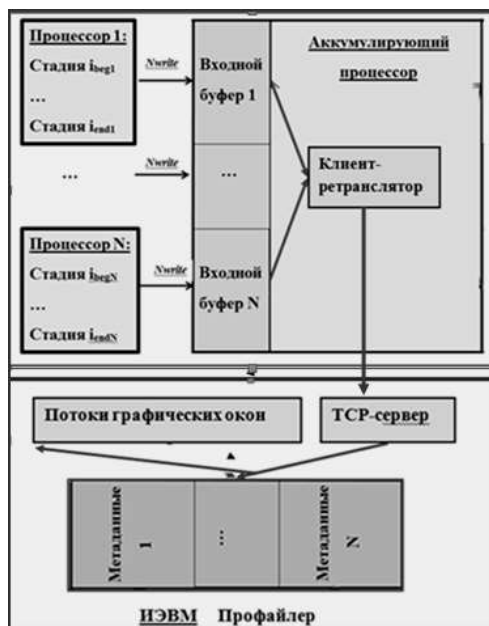


Рис.2 Тракт данных профайлера

Поток трассы устанавливает соединение с TCP-сервером профайлера по порту 1330 и последовательно передает в него данные из указанного пользователем файла трассы. Темп передачи данных определяет настройка профайлера. В файле трассы данные записаны в порядке поступления от клиента-ретранслятора СОСД, поэтому отображение трассы идентично отображению профиля целевой программы. Пользователь управляет потоком трассы двумя способами: 1) интерактивно, 2) за счет настройки профайлера. Команды пользователя позволяют начать, приостановить, возобновить и завершить воспроизведение трассы. Настройка профайлера позволяет автоматически останавливать воспроизведение трассы при проявлении предопределенных событий.

Профайлер контролирует приём данных от программы СОСД. При обнаружении потери или искажения данных профайлер выводит сообщение в окно «Сообщения о критических ситуациях» (см. раздел 7.4.3 настоящей статьи).

7.2 Отображение аномальных ситуаций

В профайлере предопределен набор условий («событий») для выявления аномального поведения целевой программы. Выполнение условия свидетельствует о наступлении или предупреждает об угрозе

нештатной ситуации. Профайлер визуализирует события путем изменения цвета элементов графических окон. Событие связывается с графическим элементом. Профайлер проверяет условие перед выводом элемента в графическое окно. При выполнении условия профайлер выделяет отображаемый элемент цветом.

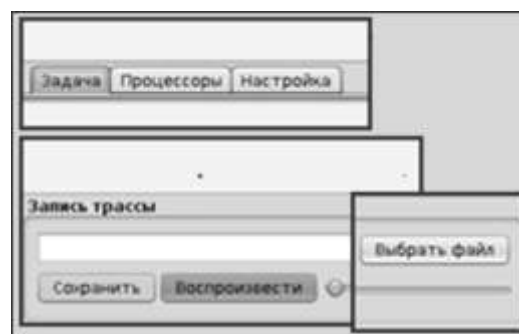


Рис.3 Фрагменты главного окна

8. Графический интерфейс профайлера

8.1 Главное окно

Профайлер запускается из командной строки ОС Linux. При запуске открывается главное окно (рисунок 3).

Графический интерфейс профайлера можно рассматривать как иерархию окон. Главное окно - корень иерархии.

Вкладки «Задача» и «Процессоры» открывают навигацию по двум ветвям иерархии. Окна в этих ветвях отображают представления профиля. В ветви «Задача» профилируются БПОС-объекты целевой программы, в ветви «Процессоры» - целевые процессоры.

Вкладка «Настройка» предназначена для установки конфигурационных параметров профайлера. Панель настройки отображается в закладке «Настройка» главного окна программы (см. рисунок 4).

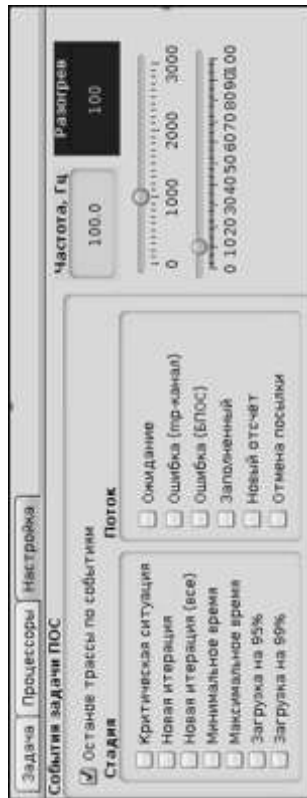


Рис.4 Панель настройки (поворот на 90°)

Настраиваемые параметры: частота итераций, количество «разогревочных» итераций, задержка обновления информации в графических окнах (в диапазоне от 0.1 до 3 секунд), количество итераций для усреднения показаний таймеров при отображении. Панель настройки содержит флажки для выбора событий. При обнаружении любого из выбранных событий профайлер автоматически останавливает воспроизведение трассы.

Графические элементы панели «Запись трассы» позволяют выбрать файл трассы, запустить запись или воспроизведение трассы, отрегулировать скорость воспроизведения, наблюдать за объемом переданных данных.

8.2 Вкладка «Задача». Графическая модель целевой программы

Графическая модель целевой программы выводится при выборе вкладки «Задача» главного окна. Модель отображает БПОС-объекты программы, связи между ними и текущие значения показателей производительности. Времена указываются в процен-

тах от периода наблюдения. Фрагменты представления модели целевой программы приведены на рисунке 5.

Обозначения БПОС-объектов:

- группа процессоров - прямоугольник с заголовком сиреневого цвета. В заголовке - порядковый номер группы, её название, количество процессоров в группе (в скобках) и относительное время простоя самого загруженного процессора группы. Под заголовком - прямоугольники вычислительных стадий;

- стадия - прямоугольник светло-бирюзового цвета. В центре прямоугольника - порядковый номер стадии, её название и относительное время вычислений (загрузка процессора). Слева от центра - входные потоки стадии, справа - выходные потоки стадии;

- поток данных - прямоугольник синего цвета. В центре - название потока. В прямоугольнике слева от центра - скорость передачи данных потоком. В прямоугольнике бледно-красного цвета справа от центра - количество ожиданий буфера выходного порта потока данных.

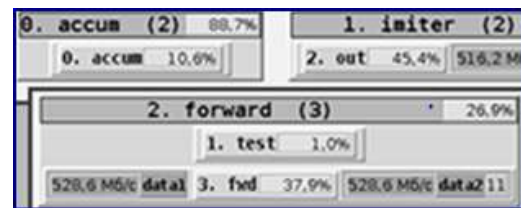


Рис.5 Фрагменты представления графической модели целевой программы

Поток данных изображается дважды: как выходной поток стадии-источника и как входной поток стадии-приемника. Такие изображения будем называть «связанными». Линии связи прямоугольников на графической модели отсутствуют. Связи визуализируются интерактивно: при выделении одного из прямоугольников потока «мышью» оба связанных прямоугольника подсвечиваются ярко-жёлтым цветом. Связанное изображение может находиться за пределами видимости окна. Для его перевода в предел видимости нужно дважды кликнуть видимый прямоугольник потока.

На графической модели отображаются следующие события: 1) превышение допустимого уровня загрузки в группе процессоров, 2) отсутствие свободных буферов на входе или на выходе потока данных.

Загрузка процессора за время выполнения итерации цикла вычисляется по данным

окна времени простоя в заголовке группы процессоров. Загрузка процессора за время выполнения отдельной стадии отображается в прямоугольнике стадии. Диапазон уровня загрузки разбит на четыре интервала: до 75%, от 75% до 95%, от 95% до 99%, более 99%. В первом интервале уровень считается нормальным, остальные три интервала интерпретируются как повышенный или недопустимый уровень.

События первого типа отображаются цветом окна данных в заголовке группы процессоров или в прямоугольнике стадии. Профайлер устанавливает взаимно однозначное соответствие между интервалами уровня загрузки процессора и цветами окна данных. При обновлении данных профайлер перекрашивает окно данных в соответствии с текущей загрузкой процессора.

События второго типа отображаются цветовой подсветкой прямоугольника потока данных.

Двойной клик «мышью» на заголовке группы процессоров открывает диалоговые окна для всех процессоров группы. Описание диалогового окна процессора приведено в разделе 8.4 настоящей статьи.

8.3 Сводная таблица процессоров

Сводная таблица процессоров выводится во вкладке «Процессоры» главного окна. Таблица агрегирует информацию о поведении программы на целевых процессорах. Фрагменты таблицы приведены на рисунке 6.

Задача	Процессоры										Настройка									
	Часы	Ю, ГА	Назван...	Итера...	Время, с	Ожида...	Средн...	Миним...	Итер...	Итер...	Итера...	Кэш	Ошибк...	Ожида...	Свобо...	Свобо...				
0	01:22:42	1	accum 0	197936	0.024521	91.9	0.024965	0.001061	1977			9059								
1	01:22:43	2	accum 1	49425	0.100189	99.9	0.100191	0.100179	4939			0	8518		1					
2	01:22:40	4	imiter 0	197968	0.024961	32.8	0.024972	0.015901	1977			0	14568		2					
3												0	0		2					
4												263	0		1					
5												263	0		31					
6												262	0		2					

Рис.6 Фрагменты таблицы процессоров (поворот на 90°)

Строка таблицы отображает данные профиля одного процессора. Номер строки соответствует логическому номеру процессора в нумерации БПОС [4].

Данные процессора в столбцах таблицы: порядковый номер, системное время, РИО-идентификатор и географический адрес, имя, номер текущей итерации, время выполнения текущей итерации, время простоя процессора на текущей итерации, среднее время итерации за период наблюдения, минимальное и максимальное времена итерации за период наблюдения, номера итерации с минимальным и максимальным временем выполнения, средняя частота итераций, текущий код ошибки «errno», текущее количество операций с кэш-памятью, количество ошибок передачи по тр-каналам потоков данных, количество ожиданий выходного буфера тр-каналов, минимальные числа свободных буферов (по всем входным и по всем выходным тр-каналам).

Подсветка поля таблицы предупреждает о возможной нештатной ситуации, связанной с отображаемым параметром. Детализировать информацию можно в диалоговом окне процессора.

Диалоговое окно процессора открывается двойным кликом «мышью» на строке таблицы. Описание диалогового окна процессора приведено в разделе 8.4 настоящей статьи.

8.4 Диалоговое окно процессора

Диалоговое окно процессора открывается двумя способами:

- двойным кликом «мышью» на заголовке группы процессоров (см. раздел 8.2),
- двойной клик «мышью» на строке таблицы процессоров (см. раздел 8.3).

В диалоговом окне выводится подробная информация о процессоре. Пример диалогового окна процессора приведён на рисунке 7.

В заголовке диалогового окна: РИО-идентификатор, имя в нотации БПОС, номер текущей итерации, средняя частота итераций, время выполнения текущей итерации, относительное время простоя процессора на текущей итерации, время работы других потоков управления на текущей итерации, текущий код ошибки «errno».

Вкладки диалогового окна: «Диаграмма», «Таймеры», «тр-каналы», «Сообщения», «Операции кэш-памяти».

8.4.1 Вкладка «Диаграмма»

Вкладка «Диаграмма» отображает динамику распределения процессорного времени. Метод визуализации – многоуровневая диаграмма с областями (см. рисунок 7). Изображение интерпретируется следующим образом. Горизонтальная сторона прямоугольника – ось итераций. Ось замкнута по принципу кольцевого буфера. Текущую итерацию отмечает вертикальный маркер. Данные динамически обновляются. При каждом обновлении маркер сдвигается на 1 пиксель вправо, по достижении конца отрезка – перемещается в начало. Вертикальная ось – процессорное время. Ось градуирована долями периода конвейерного цикла (верхняя отметка – длительность итерации). Верхняя сторона прямоугольника (линия синего цвета) – уровень длительности итерации. Линия черного цвета – средняя длительность итерации. Окрашенные области – вклад разных видов

деятельности процессора в длительность итерации, снизу вверх:

- вычислительные операции при выполнении стадий (бирюзовый цвет),
- вычислительные операции при выполнении всей итерации (зеленый цвет),
- простой процессора (голубой цвет),
- другие действия (красный цвет).

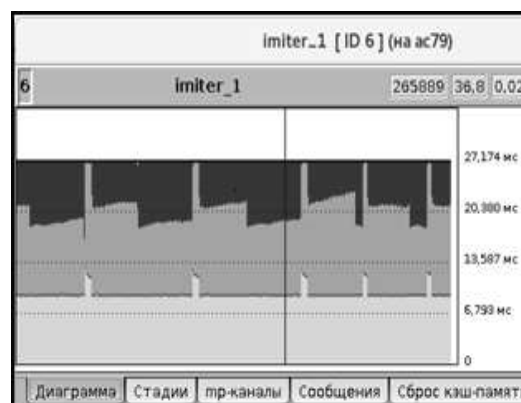


Рис. 7 Диалоговое окно процессора. Диаграмма распределения процессорного времени

8.4.2 Вкладка «Таймеры»

Вкладка «Таймеры» (рисунок 8) отображает БПОС-таймеры, привязанные к выполняемым на процессоре стадиям.

Для каждой стадии представлена таблица таймеров. Таблица разбита по строкам на секции: секция стадии, секции входных потоков, секции выходных потоков.

В заголовке секции стадии: название стадии, время выполнения стадии, время ожидания (простоя). В заголовке секции потока: название потока, размер отчета в байтах, размерности отчета по трем измерениям, способ разбиения потока, количество процессоров-источников, количество буферов на источнике, количество процессоров-приёмников, количество буферов на приёмнике, количество итераций прорезживания. В строке секции отображаются данные таймера стадии или потока. По столбцам: название таймера, времена (текущее, среднее, минимальное, максимальное), номера итераций с минимальным и максимальным временем.

Вход 0 data1			
Название	Время, %	Время, с	Средне..
-get	48,3	0,013616	0,011934
put	0,1	0,000016	0,000016
copy	0,0	0,000000	0,000000

Выход 0 data2			
Название	Время, %	Время, с	Средне..
-get	0,1	0,000036	0,000036
put	0,8	0,000213	0,000223

Минима...	Итерац...	Максим...	Итерац...
0,000025	312262	0,019664	312304
0,000014	313607	0,000158	313260
0,000000	312105	0,000000	312105

Рис.8 Диалоговое окно процессора. Фрагменты таблицы таймеров

8.4.3 Вкладка «тр-каналы»

Вкладка «тр-каналы» выводит таблицу тр-каналов потоков стадий, выполняемых на процессоре. Данные тр-канала отображаются в строке таблицы. По столбцам: имя стадии, имя потока данных, имена процессора-источника и процессора-приемника (в нотации БПОС), размер сообщения (в байтах), текущая скорость передачи данных (в байтах на итерацию), количество свободных буферов в порту тр-канала, количество захватов буфера (неудачных, успешных, общее число попыток), текущий номер отсчёта, количество ожиданий буфера, количество ошибок передачи данных, количество отказов передачи данных.

8.4.4 Вкладка «Сообщения»

Вкладка «Сообщения» отображает пользовательские данные: текстовые строки и сообщения о критических ситуациях. Перечисленные типы данных описаны в разделе 2 настоящей статьи. Сообщения в окне помечаются временем создания и номером выполняемой итерации. Общий вид вкладки приведен на рисунке 9.

8.4.5 Вкладка «Операции кэш-памяти»

Вкладка «Операции кэш-памяти» отображает выполненные процессором операции сброса, инвалидации и очистки кэш-памяти. Состав данных самоконтроля для операций кэш-памяти приведен в разделе 2 настоящей

статьи. Операции кэш-памяти отображаются в таблице. Каждый тип операции отображается в отдельной строке. По столбцам: имя стадии, название операции кэш-памяти, тип процесса, тип операции, объём, количество операций за время профилирования, количество операций за последнюю итерацию, номер итерации для последней операции с кэш-памятью, метка времени для последней операции с кэш-памятью.

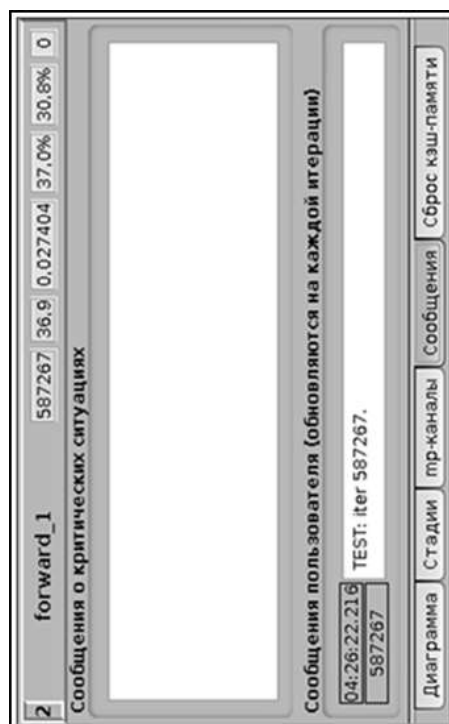


Рис.9 Диалоговое окно процессора. Вкладка «Сообщения» (поворот на 90°)

9 Заключение

Профайлер может применяться на разных этапах жизненного цикла многопроцессорной программы: при ее разработке, отладке, тестировании, эксплуатации. Профайлер позволяет:

- наблюдать за разными целевыми процессорами с единого экрана. Альтернатива – использовать эмуляторы терминала (например, minicom). Преимущество профайлера: в отличие от эмуляторов терминала он не требует подключения com-портов процессоров к ИЭВМ;

- наблюдать за ходом исполнения программы в реальном времени с возможностью переключаться «на лету» между различными представлениями профиля программы. Визуализация данных с выделением

аномальных значений помогает вовремя обратить внимание на нештатную ситуацию или ее угрозу;

- сохранять поток данных самоконтроля целевой программы в файле трассы и воспроизводить трассу в замедленном темпе. Такой режим полезен для поиска и исследования нештатных ситуаций постфактум, в лабораторных условиях.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (проведение фундаментальных научных исследований 47 ГП) по теме № 0065-2019-0002 «Исследование и реализация программной платформы для перспективных многоядерных процессоров» (рег. № АААА-А19-119012290074-2)

Parallel digital signal processing program profiling in multiprocessor real-time systems

T.K. Gringauz, A.N. Onin

Abstract: Parallel digital signal processing program self-diagnostic data collection technology in multiprocessor real-time systems with RapidIO communications was presented in [1,2]. This paper shows the self-diagnostic data usage for target program profiling. The profiler is developed with following features: 1) real time self-diagnostic data graphical representation with outliers highlighting, 2) self-diagnostic data flow recording into protocol file, 3) protocol file playback.

Keywords: system self-diagnostics, parallel digital signal processing library, program profiling, computing pipeline stage, processor group, data flow, timer, flow port, accumulator stage, handler function, RapidIO communication environment.

Литература

1. Т.К. Грингауз, А.Н. Онин. Технология сбора данных самоконтроля для программ параллельной обработки сигналов в мультипроцессорных комплексах реального времени. «Труды НИИСИ РАН», т.4 (2018), №2, 155-166.
2. Т.К. Грингауз, А.Н. Онин. Сканирование сети RapidIO в самоконтролируемых программах параллельной обработки сигналов для мультипроцессорных комплексов реального времени. Труды НИИСИ РАН», т.9 (2019), №3, 28-35.
3. RapidIO Interconnect Specification (Revision 1.3) Available from: <http://www.rapidio.org/specs/current>.
4. Райко Г.О. Библиотека параллельной обработки сигналов. «Труды НИИСИ РАН», т.5 (2015), №1, 64-69.
5. А.Н. Годунов, В.А. Солдатов. Операционные системы семейства Багет (сходства, отличия и перспективы). «Программирование», т.40 (2014), № 5, 68 – 76.
6. Т.К. Грингауз, А.Н. Онин. Программное обеспечение для приема и передачи данных по высокоскоростному каналу в мультипроцессорных комплексах реального времени. «Труды НИИСИ РАН», т.4(2014), №2, 22-32.
7. Т.К. Грингауз, А.Н. Онин. Инструментальное программное обеспечение для подготовки запуска задач в мультипроцессорных комплексах реального времени. «Труды НИИСИ РАН», т. 5 (2015), №2, 122-129.
8. Т.К. Грингауз, А.Н. Онин. Особенности использования конфигурационных файлов при интеграции технологий параллельной обработки сигналов, приема данных по высокоскоростному каналу, подготовки запуска задач в мультипроцессорных комплексах реального времени. «Труды НИИСИ РАН», т.7 (2017), №1, 58-69.
9. Т.К. Грингауз, А.Н. Онин. Технология разработки целевых программ для мультипроцессорных комплексов реального времени с приемом данных по высокоскоростному каналу. «Суперкомпьютерные технологии (СКТ-2018): V Всероссийская научно-техническая конференция. - Дивногорское, Геленджик 17-22 сентября 2018 г. (материалы конференции)», Ростов-на-Дону; Таганрог, Издательство Южного федерального университета, 2018, том 1, 76-79.

Конструирование искусственных нейронных сетей с помощью меметических алгоритмов

А.А. Рыбаков¹, Т.Т. Сохибов², С.С. Шумилин³

^{1,3}МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия;

²МГУ им. М.В. Ломоносова, Москва, Россия

e-mails: ¹rybakov@jssc.ru, ²dj.tolib.tj@mail.ru, ³shumilin@jssc.ru

Аннотация. В статье рассматривается семейство меметических алгоритмов конструирования искусственных нейронных сетей специального вида. В качестве целевых сетей рассматриваются сети, представленные ациклическими графами с фиксированным количеством входных и выходных нейронов. Меметический алгоритм представляет собой эволюционный алгоритм развития популяции нейронных сетей с итерационным частичным обучением, а также с выживанием, размножением и мутацией особей. В качестве завершения естественного отбора принимается момент, когда лидер популяции отвечает верно на все тестовые данные из целевого набора. Рассматриваемый алгоритм позволяет автоматически находить конфигурации нейронных сетей с меньшим количеством нейронов и связей, а также обладающие лучшими характеристиками по скорости обучения.

Ключевые слова. Искусственные нейронные сети, эволюционные алгоритмы, генетические алгоритмы, меметические алгоритмы, популяция, мутация.

Введение

В настоящее время искусственные нейронные сети находят широкое применение в различных областях жизни [1]. Распознавание и классификация изображений применяется в медицине (повышение осведомленности при диагностировании), безопасности (вопросы идентификации личности и определение потенциальных угроз), в геологоразведке, при анализе рукописных текстов. В экономике с помощью нейросетей производится предсказание состояния рынков, выявление коррупционных схем и оптимизация денежных потоков. Активно развивается рынок беспилотных объектов, начиная с летательных аппаратов и заканчивая автомобилями. Нейронные сети все глубже проникают с социальную сферу, они являются неотъемлемой частью маркетинга, адресной рекламы, автоматической рубрикации новостных лент.

Использование нейронных сетей позволяет существенно упростить процесс построения сложных нелинейных зависимостей, так как для этого требуется лишь задать структуру сети, а определение самих параметров (весов и смещений) выполняется автоматически в процессе обучения [2].

Однако само определение изначальной структуры сети может оказаться тонким местом построения сети. Так, при создании многослойной сети существует проблема как слишком простой структуры, так и слишком

сложной. В случае недостаточного количества слоев и нейронов в сети существует риск, что создаваемая сеть окажется неспособной обучиться для решения поставленной задачи. В случае же задания слишком большого количества слоев и нейронов существует опасность слишком медленного обучения или попадания сети в локальные экстремумы, из которых она будет неспособна выбраться в процессе дальнейшего обучения (неконтролируемое увеличение по модулю значений весов и смещений).

Для автоматического выбора подходящей начальной структуры нейронной сети предлагается использование эволюционного подхода, при котором в одной популяции одновременно существует несколько отличающихся друг от друга нейронных сетей, которые проходят параллельное частичное обучение и соревнуются друг с другом за право занимать место в популяции. При этом наиболее слабые особи вымирают, а более сильные получают право на размножение. Такие алгоритмы также называют меметическими, так как они являются объединением генетических алгоритмов и алгоритмов обучения нейросетей [3]. Модель эволюционной системы искусственных нейронных сетей рассматривается в данной статье.

Рассматриваемый тип нейронных сетей

В данной работе рассматриваются искус-

ственные нейронные сети, каждая из которых представлена ориентированным графом. В данном графе нейрон является вершиной, а связь между двумя нейронами – ребром. Ребра, входящие в вершину, называются входами соответствующего нейрона, по ним сигналы попадают в нейрон. Ребра, выходящие из вершины, аналогично, называются выходами соответствующего нейрона. В графе выделено фиксированное количество нейронов без входов, это начальные нейроны, или нейроны, относящиеся к сенсору сети (через них входной сигнал попадает в сеть). Также выделено фиксированное количество нейронов без выходов, это конечные нейроны, или нейроны, относящиеся к актуатору сети (сигналы на данных нейронах снимаются для получения ответа сети). Не рассматриваются сети, содержащие циклы. Таким образом сеть допускает нумерацию нейронов такую, что номер любого рассматриваемого нейрона больше номера любого его предшественника и меньше номера любого его последователя. При данной нумерации возможна последовательная обработка всех нейронов сети в порядке возрастания номеров.

Каждый нейрон, получив сигналы со всех своих входящих ребер, выполняет преобразование

$$z = b + \sum s_i w_i$$

$$a = \sigma(z)$$

и распространяет выходной сигнал по всем своим выходящим ребрам. Здесь s_i – входящий сигнал, w_i – вес входящего ребра, соответствующего данному сигналу, суммирование ведется по всем входящим ребрам, b – смещение, z – суммарный сигнал до применения активирующей функции, a – выходной сигнал после применения активирующей функции, в качестве активирующей функции берется сигмоидальная функция

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Будем рассматривать обучение нейронной сети с учителем, то есть когда предусмотрено наличие тестового набора, в котором каждому вектору входных значений x соответствует известный вектор $y(x)$. Если при этом нейронная сеть на поданный вектор x выдает ответ $a(x)$, то функцией ошибки (или функций стоимости) на этом элементе данных будем называть значение величины $0.5 \|y(x) - a(x)\|^2$. Значением функции полной стоимости при рассмотрении всех входных векторов будет являться следующее выражение (N обозначает конкретную конфигурацию

нейросети вместе с набором весов и смещений)

$$T(N) = \frac{1}{2|X|} \sum_{x \in X} \|y(x) - a(x)\|^2$$

где X – множество векторов входных данных из тестового набора [4].

Введем также другие обозначения. $R(N)$ – доля правильных ответов нейросети, при обучении с учителем идеальная верхняя граница данной величины равна единице. Порядком и размером сети будем называть порядок (количество вершин) и размер (количество ребер) соответствующего графа: $\nu(N)$, $\varepsilon(N)$.

Обучение рассматриваемых сетей будем проводить в помощь метода обратного распространения ошибки с фиксированной скоростью обучения. При этом количество итераций обучения будем запоминать в сети, данную величину обозначим $L(N)$ [5].

Описание семейства меметических алгоритмов

В основе генетических алгоритмов [6,7] лежит понятие популяции. Популяция – это множество особей, которые одновременно сосуществуют и соревнуются за право в этой популяции остаться. В процессе естественного отбора наиболее слабые особи популяции вымирают, а наиболее сильные дают потомство. В данной работе в качестве размножения особи будет пониматься просто ее клонирование. Таким образом восстановление популяции состоит всего лишь из клонирования ее лидеров и внесения с их структурой мутаций.

Одним из наиболее важных вопросов является критерий, по которому следует определять сильных и слабых особей. Конечно основным решающим показателем является параметр $R(N)$. Как только лидер популяции достигает значения 1.0 данного параметра, алгоритм может останавливаться. Однако, как показывают эксперименты, популяция способна довольно быстро сравняться по данному показателю и дальше должны учитываться другие параметры, такие как $T(N)$, $L(N)$, $\nu(N)$, $\varepsilon(N)$ и их комбинации. При этом следует отменить, что нейронная сеть не является статическим объектом во времени. При смене поколений каждая сеть частично обучается, поэтому только что созданные сети не следует оценивать наравне с более старыми. Поэтому в процессе тестирования алгоритмов была принята эвристика, моделирующая поведение защиты потомства:

никакая сеть не может быть уничтожена, пока она не достигла среднего возраста по-

пуляции (то есть каждая сеть получает шанс на начальное развитие).

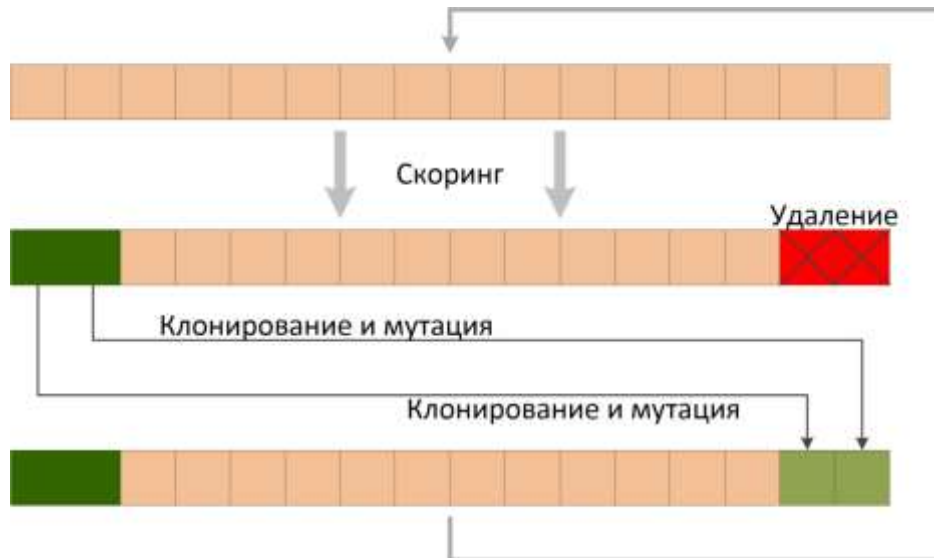


Рис. 1. Цикл простейшего генетического алгоритма: скоринг – удаление слабых особей – восстановление популяции – мутация.

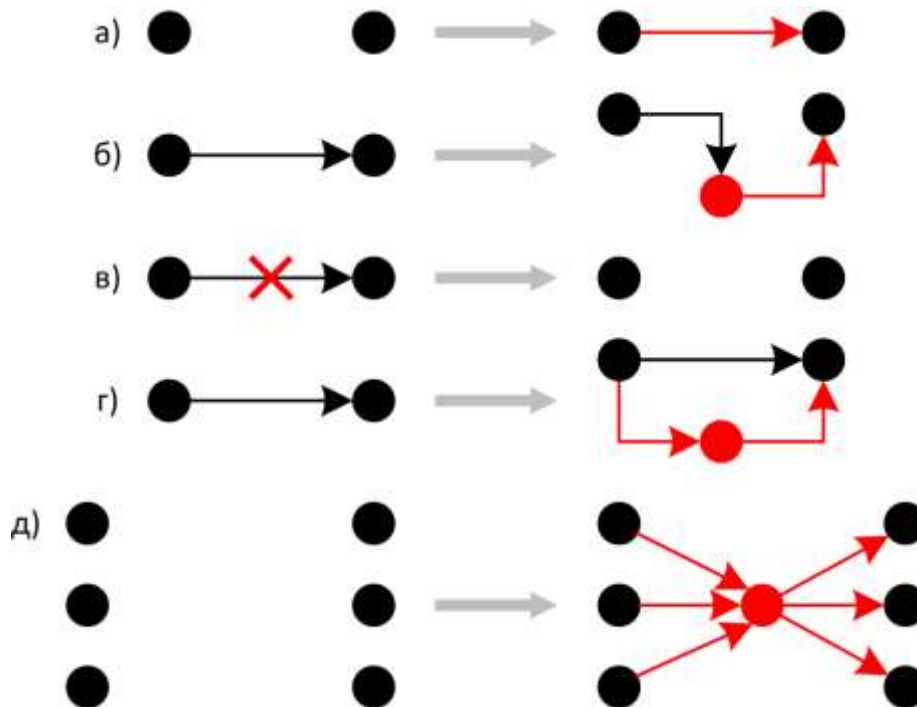


Рис. 2. Примеры используемых мутаций: а) добавление ребра между двумя нейронами; б) добавление нового нейрона на ребро (и еще одного ребра); в) удаление ребра (с возможным последующим удалением висячих нейронов); г) добавление дополнительной связи между двумя нейронами с помощью добавления нейрона и двух ребер; д) супермутация – связь двух подмножеств нейронов через новый нейрон.

С учетом замечания о защите недавно созданных сетей обработка поколения в генетическом алгоритме состоит из следующей

последовательности шагов: проведение скоринга (вычисление функций $R(N)$ и $T(N)$ для каждой сети); определение наиболее слабых

особей и их уничтожение; определение наиболее сильных особей и их клонирование для восстановления популяции (при этом в процессе клонирования производятся мутации с некоторой долей вероятности) (рис. 1).

Обычно в генетических алгоритмах структура и поведение особей кодируется в геноме, над которым и выполняются действия по клонированию, скрещиванию и мутациям. Такой подход можно наблюдать в системе DXNN [8]. Однако по сути это прямое кодирование нейросети, поэтому было принято решение в рамках данной работы вообще не использовать геном, а сразу выполнять мутации на готовом объекте, используя атомарные преобразования направленного графа по подобию преобразований из [9].

В качестве мутаций (рис. 2) были рассмотрены такие элементарные преобразования графа нейросети, как добавление и удаление ребра, добавление на ребро нового нейрона, добавление нового нейрона с одним входным и одним выходным ребром. Однако применение лишь элементарных преобразований не привело к достижению показателя $R(N) = 1.0$ даже на большом количестве поколений. Данный факт может свидетельствовать о том, что не всегда существует последовательность мелких атомарных преобразований, переводящих элементарный объект в процессе эволюционного развития в абсолютно оптимальный, который может достигнуть глобального максимума целевой функции.

В данном факте в принципе нет никакого противоречия, так как в процессе естественной эволюции в природе перед живым существом не ставится задача достижения глобального максимума, а стоит более приземленная задача – выживание в текущих локальных условиях в составе своей популяции [10]. В нашей же задаче конструирования оптимальной нейронной сети для данного тестового набора при обучении с учителем такой исход неприемлем.

Для преодоления данного препятствия была введена так называемая супермутация (рис. 2, д), в процессе которой добавляется один нейрон, связывающий два больших множества нейронов между собой. При этом для сохранения баланса была повышена ве-

роятность мутации, при которой происходит удаление ребра. Это помогло после возникновения супермутации быстро удалять избыточные ребра, оставляя только полезные связи, которые приводят к росту функции $R(N)$.

Применение к модельной задаче

В качестве модельной задачи была рассмотрена простая задача приближения логической функции с 5 входами и 2 выходами. Функция по сути принимает на вход число в двоичном виде (5 двоичных разрядов соответствуют числам от 0 до 31) и возвращает пару (1.0, 0.0) если число простое, и пару (0.0, 1.0) в противном случае. В процессе тестирования размер популяции брался равным 50, а количество вымирающих особей – не более двух (могло быть и менее, если в популяции слишком много молодых особей).

Также для борьбы с флуктуациями функции $T(N)$ лидера популяции было принято два дополнительных соглашения, которые в общем случае не гарантируют сходжение обучения. Во-первых, частичное обучение любой сети в рамках отработки текущего поколения не может завершиться, пока значение функции $T(N)$ для этой сети не уменьшится. Во-вторых, лидер популяции вообще не обучается. Если не придерживаться данного соглашения достаточно рано в популяции образуется компактная элита с одним показателем $R(N)$, и очень близкими значениями $T(N)$, и представители этой элиты начинают бесконечно поочередно становиться лидерами. После же принятия соглашения о монотонном убывании функции $T(N)$ данные проблемы исчезают, и неоптимальный лидер популяции рано или поздно оказывается смещенным своей копией, прошедшей через супермутацию.

Процесс эволюционного отбора на рассмотренной задаче сходится чуть более, чем за 1000 поколений. На рис. 3 представлена диаграмма продолжительности жизни всех особей популяции. На данной диаграмме видно, что практически в самом начале появились лидеры-долгожители, а остальные особи достаточно быстро вымирали (эпоха низких показателей количества верных ответов $R(N)$).

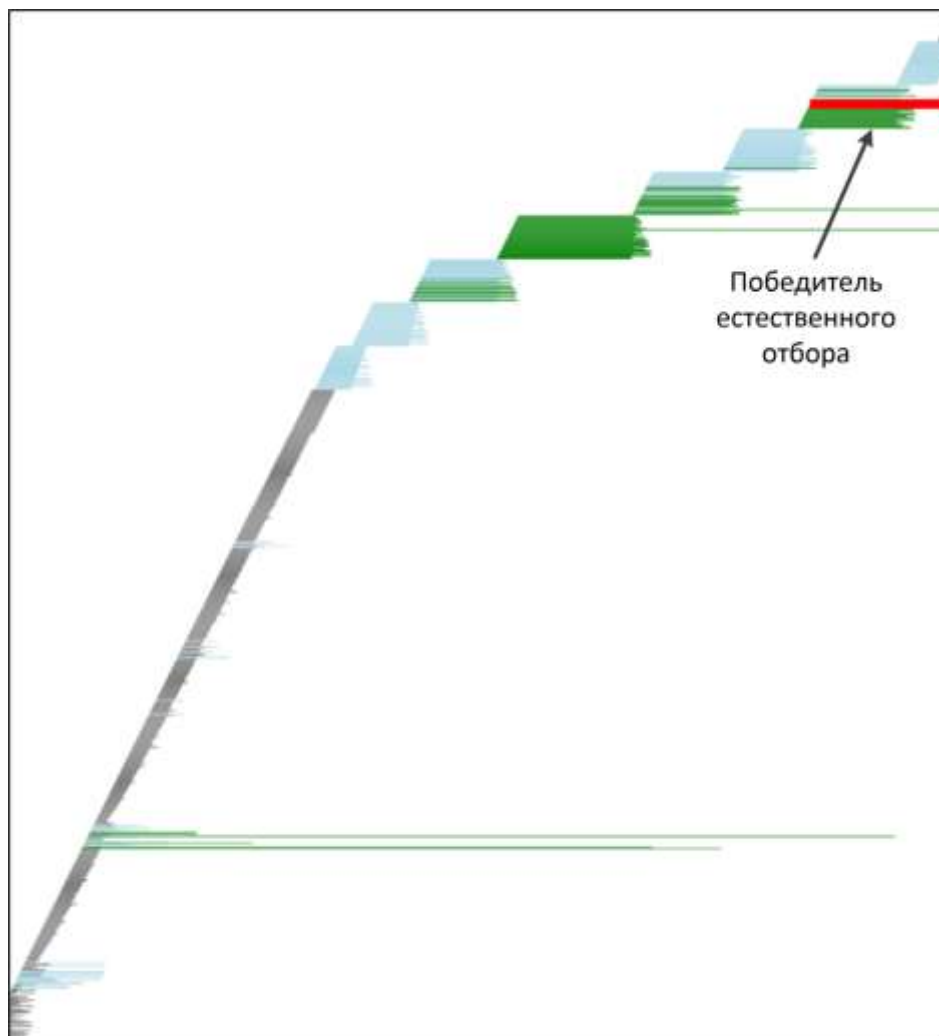


Рис. 3. Диаграмма, показывающая продолжительность жизни особей популяции. По вертикали показаны особи в порядке их рождения. Длина горизонтальной линии – время жизни особи (зеленые – время жизни более 100 поколений, голубые – от 30 до 100, серые – менее 30).

При возрастании показателя $R(N)$ лидера дальнейшее улучшение этой характеристики происходит только вследствие возникновения супермутаций, вероятность которых задана достаточно редкой. По сути каждая "ступенька" на диаграмме отражает реакцию на появление супермутации. В конце концов одна из особей, переживших супермутацию (отмечено на диаграмме красным), достигает показателя $R(N) = 1.0$, и алгоритм заканчивает работу. На рис. 4. представлены графики, на которых видно, как изменялись следующие показатели лидера популяции: возраст особи, а также $L(N)$, $R(N)$, $T(N)$, $v(N)$, $\varepsilon(N)$.

Для сравнения также было проведено обучение нескольких многослойных сетей. Результаты их обучения, а также результаты обучения победителя естественного отбора приведены в табл. 1. Видно, что первая многослойная нейросеть, достигшая показателя

$R(N) = 1.0$, больше по размеру и медленнее в плане обучения, чем лидер популяции.

Табл. 1. Сравнение показателей лидера популяции с обычными многослойными сетями.

Описание сети	$L(N)$	$R(N)$	$v(N)/\varepsilon(N)$
лидер	6700	1.0	11/25
[5, 2]	100 000	0.812 5	7/10
[5, 1, 2]	100 000	0.875	8/7
[5, 2, 2]	100 000	0.906 25	9/14
[5, 3, 2]	100 000	0.968 75	10/21
[5, 4, 2]	100 000	0.968 75	11/28
[5, 5,	100	0.968	12/35

2]	000	75	
[5, 6, 2]	74 866	1.0	13/42
[5, 7, 2]	6 728	1.0	14/49

[5, 8, 2]	6 043	1.0	15/56
--------------	-------	-----	-------

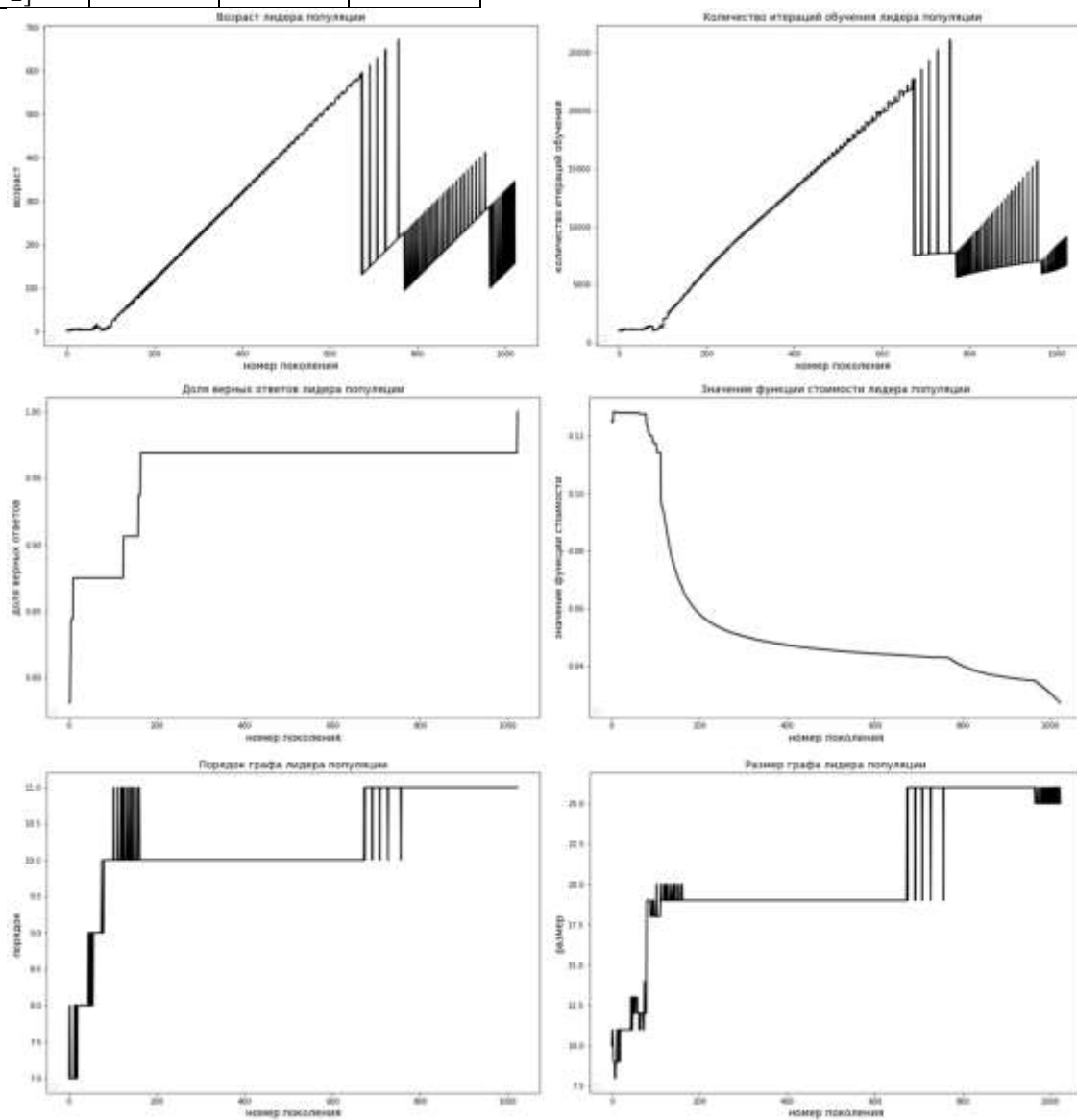


Рис. 4. Графики зависимостей лидера популяции от номера поколения (слева направо сверху вниз): возраст, количество итераций обучения, доля верных ответов, значение функции стоимости, количество нейронов, количество ребер.

Заключение

Было рассмотрено семейство меметических алгоритмов, с помощью которых возможно автоматическое конструирование искусственных нейронных сетей, обладающих высокой скоростью обучения и состоящих из меньшего количества нейронов и связей по сравнению со стандартными многослойными сетями (для сравнения из табл. 4 сеть "лидер" и сеть "[5, 6, 2]").

Механизмы эволюционного отбора явля-

ются довольно сложными объектами для моделирования, они обладают своей спецификой в зависимости от задачи и должны анализироваться отдельно. В частности ключевыми моментами для успеха эволюционного отбора оказались забота о молодых особях, разрушение стагнирующей элиты и наличие супермутаций, позволяющих преодолеть локальные экстремальные ямы.

Предложенный подход может быть использован для дальнейшего изучения меха-

низмов эволюционного отбора и для применения к практическим задачам.

Работа выполнена при поддержке гранта РФФИ № 18-29-03236.

Construction of artificial neural networks using memetic algorithms

A.A. Rybakov, T.T. Sokhibov, S.S. Shumilin

Abstract. The article considers a family of memetic algorithms for constructing artificial neural networks of a special kind. The networks represented by acyclic graphs with a fixed number of input and output neurons are considered as target networks. The memetic algorithm is an evolutionary algorithm for the development of neural networks population with iterative partial learning, as well as with the survival, reproduction and mutation of individuals. As the completion of natural selection, the moment is taken when the leader of the population answers correctly all test data from the target set. The considered algorithm allows to automatically find the configuration of neural networks with fewer neurons and connections, as well as having the best characteristics in terms of learning speed..

Keywords. Artificial neural networks, evolutionary algorithms, genetic algorithms, memetic algorithms, population, mutation..

Литература

1. С. Хайкин. Нейронные сети. Полный курс. // Диалектика/Вильямс, 2019.
2. Deep Learning. An MIT press book. // <https://www.deeplearningbook.org/>, дата обращения 10.02.2020.
3. G.I. Sher. Handbook of neuroevolution through Erlang. // Springer, 2013.
4. Neural networks and deep learning. // <http://neuralnetworksanddeeplearning.com/>, дата обращения 10.02.2020.
5. Т. Harju. Lecture notes on graph theory. // <http://users.utu.fi/harju/graphtheory/graphtheory.pdf>, дата обращения 10.02.2020.
6. Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. Генетические алгоритмы. // М.: Физматлит, 2006.
7. Ю.А. Скобцов. Основы эволюционных вычислений. // Донецк: ДонНТУ, 2008.
8. G.I. Sher. DXNN platform: The shedding of biological inefficiencies. // arXiv:1011.6022, 2011.
9. А.А. Рыбаков. Алгоритм создания случайных графов потока управления для анализа глобальных оптимизаций в компиляторе. // Parallel and distributed computing systems PDCS 2013 collection of scientific papers, Kharkiv, Ukraine, 2013, p. 269-275.
10. Р. Докинз. Слепой часовщик. Как эволюция доказывает отсутствие замысла во вселенной. // Династия, 2014.

Подписано в печать 18.5.2020 г.

Формат 60x90/8

Печать цифровая. Печатных листов 8

Тираж 100 экз. Заказ №

Отпечатано в ППП «Типография «Наука»
121099, Москва, Шубинский пер., 6