

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 22
2016
№ 5

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Саак А. Э. Диспетчеризация массивов заявок кругового и гиперболического типа в Grid-системах 323
- Мохов А. С., Толчеев В. О. Способы учета структуры научных документов в задачах обработки и анализа текстовой информации 332

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Мухутдинов А. Р., Вахидова З. Р., Ефимов М. Г. Компьютерное моделирование бризантного действия взрыва 340
- Инютин С. А. Метод вычисления количественной характеристики модулярной величины 343

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Богатырев В. А., Богатырев А. В. Модель резервированного обслуживания запросов реального времени в компьютерном кластере 348
- Микова С. Ю., Оладько В. С. Сравнение алгоритмов выявления сетевых аномалий с помощью меры Ван Ризбергена 356

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Авдошин С. М., Лазаренко А. В. Методы деанонимизации пользователей Tor 362

БАЗЫ ДАННЫХ

- Коровин А. С., Скирневский И. П. Система динамической визуализации больших массивов данных сложных физических экспериментов 373

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИМЕДИЦИНСКИХ СИСТЕМАХ

- Агеева У. О., Агеева В. Г., Барский А. Б. Бионическое интеллектуальное протезирование конечностей и логические нейронные сети 379
- Прасолова А. Е. Коллективный нейросетевой алгоритм диагностики инфаркта миокарда 386

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Гриняк В. М., Иваненко Ю. С., Девятисильный А. С. Погрешность измерения координат компьютеризированными РЛС, обусловленная скоростью передачи данных в распределенных информационных системах 391
- Скоробогатов Р. Ю. Расширение интерактивности компьютерной модели в телевизионной среде 396

Главный редактор:
СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:
ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:
БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
чл.-корр. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
чл.-корр. РАН, д. т. н., проф.
СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
чл.-корр. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:
АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ГАЛУШКИН А. И., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. М., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕЩУК О. М., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:
БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Vol. 22
2016
No. 5

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Saak A. E.** Scheduling of Sets of Hyperbolic-Type and Circular-Type Tasks in Grid-Systems 323
- Mokhov A. S., Tolcheev V. O.** Approaches to Considering Patterns of Scientific Documents in Processing and Analysis of Text Information 332

MODELING AND OPTIMIZATION

- Mukhutdinov A. R., Vahidova Z. R., Efimov M. G.** Computer Modelling of Brisant Action of Explosion 340
- Inyutin S. A.** Method Calculation Quantitative Characteristic Computer Modular Value 343

COMPUTING SYSTEMS AND NETWORKS

- Bogatyrev V. A., Bogatyrev A. V.** The Model of Redundant Service Requests Real-Time in a Computer Cluster 348
- Mikova S. Yu., Oladko V. S.** Comparison of Algorithms to Identify Network Anomalies Using Measures Van Rizbergen 356

CRYPTOSAFETY INFORMATION

- Avdoshin S. M., Lazarenko A. V.** Tor Users Deanonimization Methods 362

DATABASE

- Korovin A. S., Skirnevskij I. P.** Web-Application for Real-Time Big Data Visualization of Complex Physical Experiments 373

INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

- Ageeva U. O., Ageeva V. G., Barskiy A. B.** Bionic Intelligence Limbs Prosthetic and Logical Neural Networks 379
- Prasolova A. E.** Collective Neural Network Algorithm for the Diagnosis of Myocardial Infarction 386

APPLIED INFORMATION TECHNOLOGIES

- Grinyak V. M., Ivanenko Yu. S., Devyatisilny A. S.** Data Rate in Distributed Information System and its Effect on Digital Radar Measurement Error 391
- Skorobogatov R. Yu.** Introduction Virtual Characters in the Space of TV Studios 396

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Galushkin A. I., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. M., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Grigorin-Ryabova E. V.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

Диспетчеризация массивов заявок кругового и гиперболического типа в Grid-системах

В среде ресурсных прямоугольников предлагаются варианты уровневых алгоритмов, позволяющие диспетчирование массивами ресурсных прямоугольников, не обладающих свойством монотонности горизонтальных или вертикальных измерений. Для указанных массивов, состоящих из заявок кругового и гиперболического типа, предлагаются варианты уровневых алгоритмов по высоте и протяженности (с недостатком, превышением, минимальным отклонением). Сравнительный анализ позволяет рекомендовать ряд таких алгоритмов к использованию в Grid-системах с централизованной структурой.

Ключевые слова: Grid-система, централизованная структура системы диспетчирования, мультисайтный режим обслуживания заявки, обслуживание заявки без прерывания, заявка кругового типа, заявка гиперболического типа, не-эвклидова эвристическая мера, полиномиальная трудоемкость алгоритма, уровневый алгоритм по высоте, уровневый алгоритм по протяженности

Введение и постановка задачи

Возрастающая потребность пользователей в вычислительной мощности, наряду с развитием технологии, ростом числа процессоров, увеличением быстродействия телекоммуникационных сетей, стимулировала в конце 90-х годов переход от многопроцессорных систем, метакомпьютинга к Grid-компьютингу, который развивается и в настоящее время [1, 2], наряду с такими парадигмами распределенных вычислений [3], как облачные вычисления [4] и вычислительные джунгли [5].

В Grid-диспетчировании выделяют четыре основных этапа: выявление ресурсов (*resource discovery*), выбор ресурсов (*resource selection*), генерация расписания (*schedule generation*), выполнение задачи (*job execution*) [6]. Наше основное внимание сосредоточено на этапе генерации расписания алгоритмом диспетчирования (*scheduling algorithm*). При этом алгоритм диспетчирования определяет способ, которым задачи назначаются на ресурсы, а под расписанием понимается назначение задач на ресурсы в определенные периоды времени [7].

В настоящей работе рассматриваются вычислительные заявки [8] с заранее известным временем решения [9], в которых число требуемых процессоров определяет пользователь при подаче в систему [10]. Предполагается, что политика диспетчирования поддерживает обслуживание заявки без прерывания (*non-preemptive, run-to-completion*), т. е. выделенные заявке процессоры удерживаются в течение всего времени ее выполнения от начала до

окончания [11, 12]. В этих условиях при выделении для обработки заявки процессоров с последовательными номерами задача генерации расписания алгоритмом диспетчирования эквивалентна задаче распределения ресурсных прямоугольников. При представлении заявки пользователя для обслуживания диспетчером Grid-системы ресурсным прямоугольником горизонтальное и вертикальное измерения принимаются равными соответственно числу единиц ресурса времени и процессоров, требуемому для обработки заявки [13]. Символом $[a(j), b(j)]$ обозначается j -я заявка, требующая $a(j)$ единиц времени и $b(j)$ единиц процессоров.

Grid-системы с централизованной структурой системы диспетчирования [8, 14], состоящие из сайтов, содержащих параллельные системы, функционирующие в режиме мультисайтного диспетчирования [14, 15], моделируются ресурсным квадратом [16, 17].

Для управления распределением вычислительных-временных ресурсов и заявками пользователей Grid-систем в качестве теоретической основы алгоритмического обеспечения диспетчирования с полиномиальной трудоемкостью в [16–20] определена среда ресурсных прямоугольников. В [16] введены операции динамического интегрирования ресурсных прямоугольников по вертикали и по горизонтали, состоящие в наилучшем приближении заданного значения с недостатком. Состав операций расширен динамическим интегрированием ресурсных прямоугольников по вертикали [21] и по

горизонтали [22] с превышением и с минимальным отклонением. На основе этих операций разработаны полиномиальные алгоритмы [21, 22], адаптированные под соответствующий квадратичный тип массива заявок. В [16] круговой, гиперболический и параболический типы определены для массивов из не менее двух заявок. Квадратичный тип одной заявки введен в [23], где исследована адаптируемость полиномиальных алгоритмов под массивы, состоящие из заявок кругового типа.

В настоящей статье ставится вопрос об адаптированности полиномиальных алгоритмов для массивов, состоящих из заявок кругового и гиперболического квадратичного типа.

Расширение класса полиномиальных уровней алгоритмов обслуживания в Grid-системах

В работе [20] предложен уровень алгоритм по высоте (с недостатком), в [16] — уровень алгоритм по протяженности (с недостатком). Уровень алгоритм по высоте с превышением и уровень волновой алгоритм (с минимальным отклонением) введены в работе [21]. Рассмотрим варианты этих алгоритмов для массивов ресурсных прямоугольников, не обладающих свойством монотонности горизонтальных или вертикальных измерений, присутствующих массивам кругового и гиперболического типов: V -уровневых алгоритмов по высоте (с недостатком, с превышением, с минимальным отклонением) и H -уровневых алгоритмов по протяженности (с недостатком, с превышением, с минимальным отклонением).

Функционирование V -уровневых алгоритмов по высоте аналогично приведенным в [21], с тем отличием, что на каждом шаге правая сторона ресурсной оболочки определяется суммой значения правой стороны достигнутой ресурсной оболочки и максимального горизонтального измерения элементов в слое, а не горизонтальным измерением начального элемента слоя. Так, на s -м шаге (рис. 1) значение правой стороны ресурсной оболочки Y_{s+1} равно $Y_{s+1} = Y_s + \max_{j_{s-1} \leq j \leq j_s} a(j)$, где j_{s-1}, j_s — начальный и конечный номера ресурсных прямоугольников в вертикальном слое.

В рассматриваемых V -уровневых алгоритмах по высоте значение уровня определяется величиной

$$\sqrt[k]{\sum_{j=0}^{k-1} a(j)b(j)}, \text{ где } k \text{ — число}$$

заявок в массиве.

Функционирование H -уровневых алгоритмов по протяженности аналогично приведенному в работе [16]. Для H -уровневого алгоритма по протяженности с превышением отличие состоит в том, что на каждом шаге используется введенная в [22] операция динамического интегрирования ресурсных прямоугольников по горизонтали с превыше-

нием, состоящая в наилучшем приближении заданной протяженности с избытком. Для H -уровневого алгоритма по протяженности с минимальным отклонением отличие состоит в том, что на каждом шаге используется введенная в работе [22] операция динамического интегрирования ресурсных прямоугольников по горизонтали с минимальным отклонением. При этом верхняя сторона ресурсной оболочки определяется суммой значения верхней стороны достигнутой ресурсной оболочки и максимального вертикального измерения элементов в слое, а не вертикальным измерением начального элемента слоя. Так, на s -м шаге (рис. 2) значение

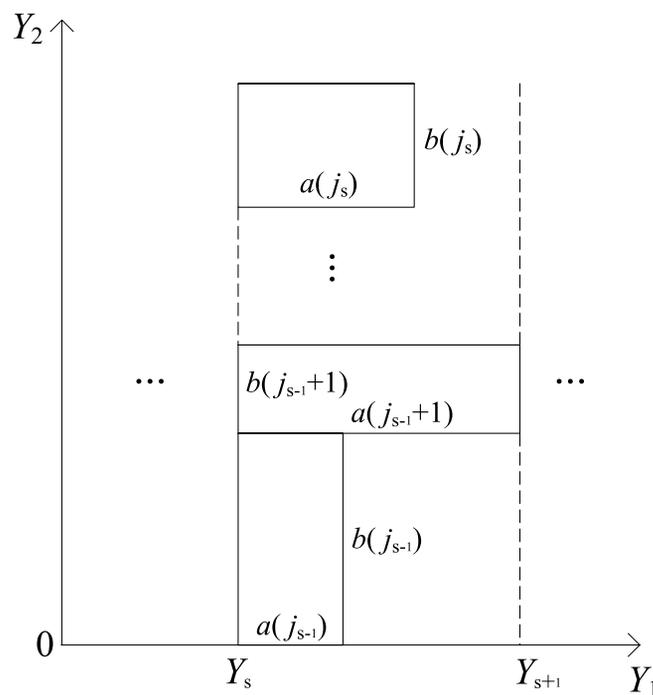


Рис. 1. Вертикальный s -й слой V -уровневых алгоритмов

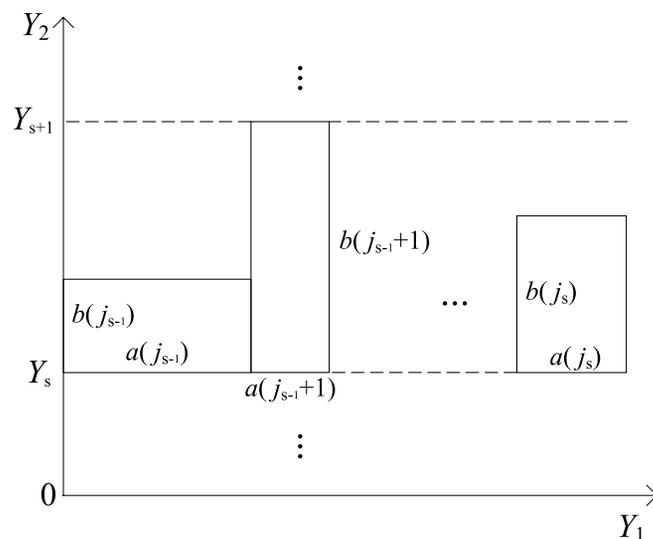


Рис. 2. Горизонтальный s -й слой H -уровневых алгоритмов

верхней стороны ресурсной оболочки Y_{s+1} равно $Y_{s+1} = Y_s + \max_{j_{s-1} \leq j < j_s} b(j)$, где j_{s-1}, j_s — начальный и конечный номера ресурсных прямоугольников в горизонтальном слое.

В рассматриваемых H -уровневых алгоритмах по протяженности значение требуемой протяженности определяется величиной

$$\sqrt{\sum_{j=0}^{k-1} a(j)b(j)}.$$

Далее исследуется адаптированность полиномиальных V -уровневых и H -уровневых алгоритмов для массивов, состоящих из заявок кругового и гиперболического квадратичного типа.

Диспетчеризация уровней алгоритмами массива, состоящего из заявок кругового и гиперболического типов

Для оценки качества диспетчеризации эвристических алгоритмов в работе [16] была предложена

неевклидова эвристическая мера $\frac{1}{2} \left(\frac{AB + (A - B)^2}{\sqrt{\sum_{j=0}^{k-1} a(j)b(j)}} \right)^2$,

учитывающая как площадь AB , так и форму $(A - B)^2$ занятой ресурсной области, где A — протяженность, B — уровень по вертикали ресурсной прямоугольной оболочки. Возможный минимум эвристической меры, равный $1/2$, достигается при беспустотной укладке в квадрат.

Массивы прямоугольников, индуцированные элементами диссекции квадрата на различные прямоугольники с отношением сторон $1:2$ ($\frac{a(j)}{b(j)} = \frac{1}{2}$

или $\frac{a(j)}{b(j)} = 2$), используются нами в качестве тестового примера [24]. Такой массив, в соответствии

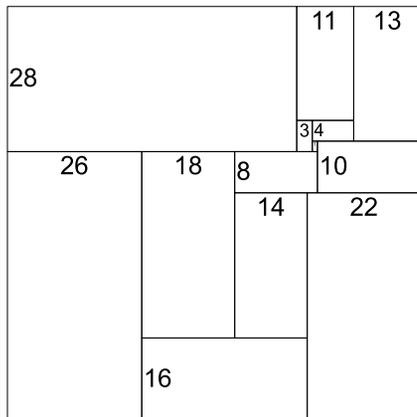


Рис. 3. Диссекция квадрата на 13 прямоугольников с отношением сторон $1:2$

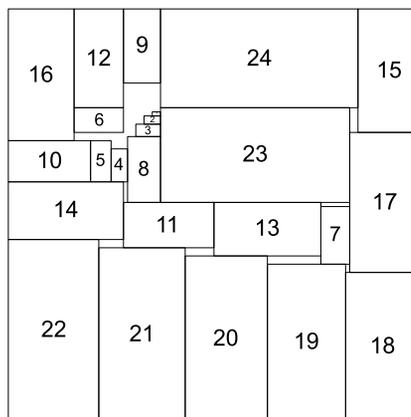


Рис. 4. Укладка 24 последовательных прямоугольников с отношением сторон $1:2$ в квадрат с минимальной стороной [25]

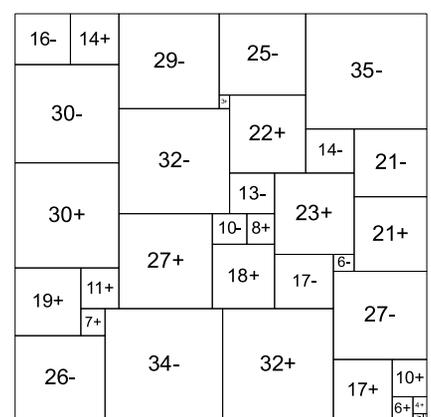


Рис. 5. Облицовка почти квадрата, содержащая 34 последовательных почти квадратов [28]

с определениями [23], содержит прямоугольники кругового типа при горизонтальной ориентации $a(j) \geq b(j)$ и гиперболического типа — при вертикальной $a(j) < b(j)$. В [24] приведены диссекции квадрата на 11, 12 и 13 (рис. 3) прямоугольников. На прямоугольнике указано значение меньшей стороны.

Указанные массивы ресурсных прямоугольников, упорядоченные по убыванию высот, обозначим следующим образом: для диссекции порядка 11 — массив I, для диссекции порядка 12 — массив II, для диссекции порядка 13 — массив III.

Следующие тестовые примеры порождены укладками последовательных прямоугольников $[j, 2j]$, $j = 1, 2, \dots, k$, в квадрат с минимальной стороной, полученные Э. Фридманом (Erich Friedman) в 2014 г. для $k = 21, 22, 23, 24$ (рис. 4) [25]. На прямоугольнике указано значение меньшей стороны.

Массивы прямоугольников, индуцированные такими укладками, содержат прямоугольники с отношением сторон $1:2$ горизонтальной и вертикальной ориентации. Соответствующие массивы ресурсных прямоугольников, упорядоченные по убыванию высот, обозначим следующим образом: для $k = 21$ — массив IV, для $k = 22$ — массив V, для $k = 23$ — массив VI, $k = 24$ — массив VII.

Заключительные тестовые примеры индуцированы укладкой массива $[j, (j + 1)]$, $j = 1, 2, \dots, k$, последовательных прямоугольников (измерения сторон отличаются на единицу, называемых почти квадратами [26—28], в соответствующий почти квадрат без пустот. В [28] приведены такие укладки, называемые облицовкой (*tiling*), для $k = 20, 34$ (рис. 5). Следуя [28], на прямоугольнике указано значение горизонтального измерения, а знаки "+" или "-" означают больше или меньше на 1 значение вертикального измерения.

Так как условие отнесения прямоугольника к круговому или гиперболическому типу $(b(j) - a(j))^2 \leq a(j)b(j)$, сформулированное в работе [23], для

почти квадратов приводит к очевидному неравенству $1 \leq j(j + 1)$, то почти квадраты относятся к круговому или гиперболическому типу в зависимости от горизонтальной или вертикальной ориентации [23]. Массивы ресурсных прямоугольников, индуцированные облицовкой, упорядоченные по убыванию высот, обозначим следующим образом: для $k = 20$ — массив VIII, $k = 34$ — массив IX.

Вычислим эвристические меры ресурсных оболочек, получаемых при диспетчировании этих массивов V -уровневыми алгоритмами по высоте (с недостатком, с превышением, с минимальным отклонением) и H -уровневыми алгоритмами по протяженности (с недостатком, с превышением, с минимальным отклонением).

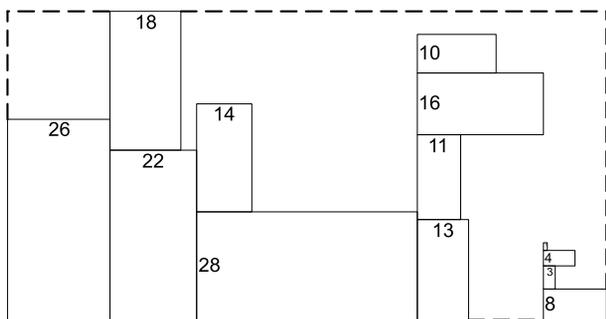


Рис. 6. Укладка массива III V -уровневым алгоритмом по высоте с недостатком

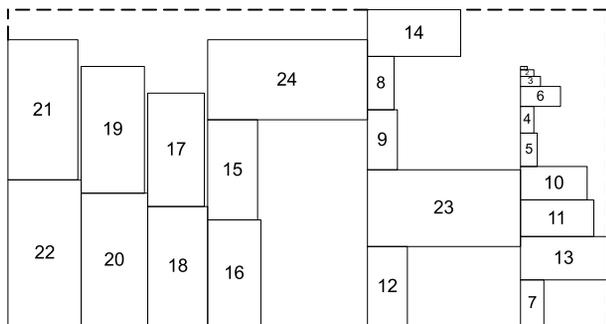


Рис. 7. Укладка массива VII V -уровневым алгоритмом по высоте с недостатком

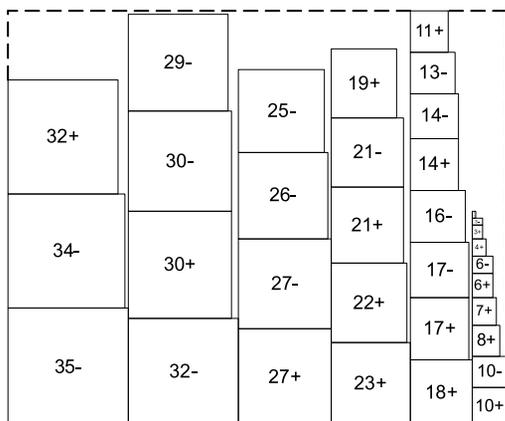


Рис. 8. Укладка массива IX V -уровневым алгоритмом по высоте с недостатком

Построения V -уровневым алгоритмом по высоте с недостатком для массивов III, VII, IX приведены на рис. 6—8.

Эвристические меры ресурсных оболочек V -уровневого алгоритма по высоте с недостатком для массива из заявок кругового и гиперболического квадратичного типов приведены в табл. 1.

Видим, что эвристические меры ресурсных оболочек V -уровневого алгоритма по высоте с недостатком не превосходят значения $0,5 + 0,86$.

Построения V -уровневым алгоритмом по высоте с превышением для массивов III, VII, IX приведены на рис. 9—11.

Эвристические меры ресурсных оболочек V -уровневого алгоритма по высоте с превышением для

Таблица 1
Эвристические меры ресурсных оболочек V -уровневого алгоритма по высоте с недостатком

Номер массива	Эвристическая мера	Номер массива	Эвристическая мера	Номер массива	Эвристическая мера
I	1,03	IV	0,81	VII	1,24
II	1,3	V	0,76	VIII	0,71
III	1,36	VI	0,96	IX	0,63

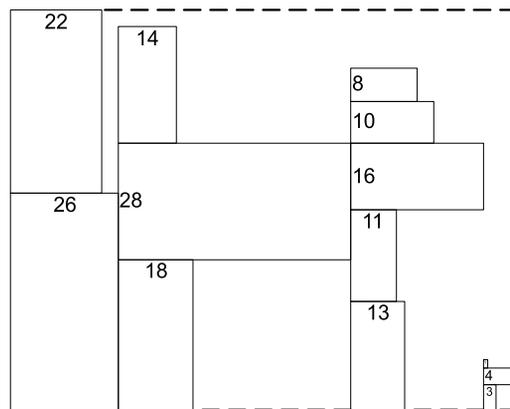


Рис. 9. Укладка массива III V -уровневым алгоритмом по высоте с превышением

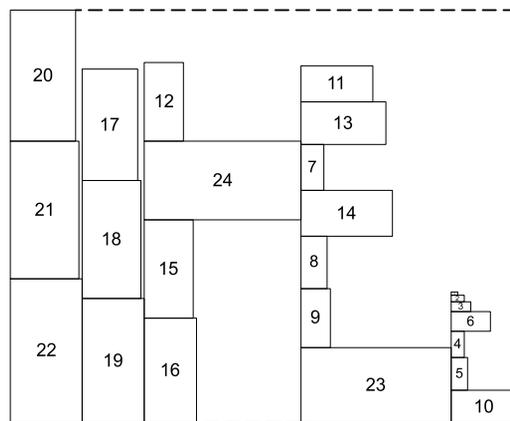


Рис. 10. Укладка массива VII V -уровневым алгоритмом по высоте с превышением

массива из заявок кругового и гиперболического квадратичного типов приведены в табл. 2.

Видим, что эвристические меры ресурсных оболочек *V*-уровневого алгоритма по высоте с превышением не превосходят значения $0,5 + 0,54$.

Построения *V*-уровневым алгоритмом по высоте с минимальным отклонением для массивов III, VII, IX приведены на рис. 12—14.

Эвристические меры ресурсных оболочек *V*-уровневого алгоритма по высоте с минимальным отклонением для массива из заявок кругового и гиперболического квадратичного типа приведены в табл. 3.

Видим, что эвристические меры ресурсных оболочек *V*-уровневого алгоритма по высоте с минимальным отклонением не превосходят значения $0,5 + 0,47$.

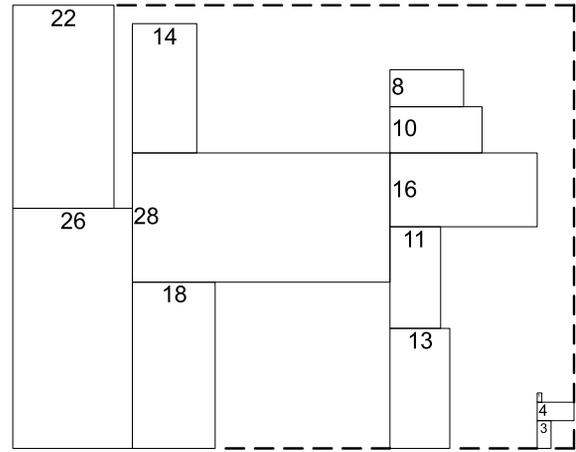


Рис. 12. Укладка массива III *V*-уровневым алгоритмом по высоте с минимальным отклонением

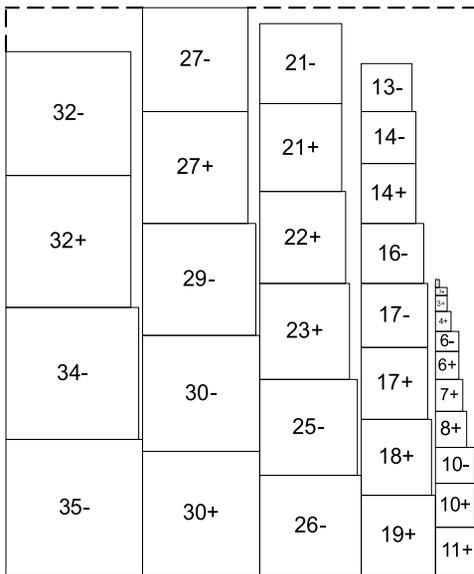


Рис. 11. Укладка массива IX *V*-уровневым алгоритмом по высоте с превышением

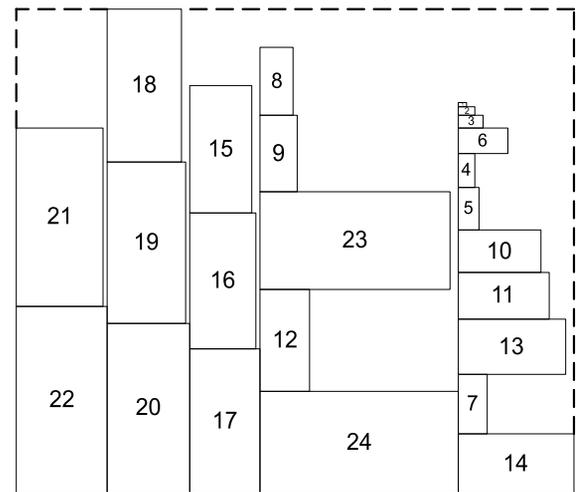


Рис. 13. Укладка массива VII *V*-уровневым алгоритмом по высоте с минимальным отклонением

Таблица 2

Эвристические меры ресурсных оболочек *V*-уровневого алгоритма по высоте с превышением

Номер массива	Эвристическая мера	Номер массива	Эвристическая мера	Номер массива	Эвристическая мера
I	0,86	IV	0,74	VII	1,04
II	0,81	V	0,88	VIII	0,64
III	0,97	VI	0,92	IX	0,62

Таблица 3

Эвристические меры ресурсных оболочек *V*-уровневого алгоритма по высоте с минимальным отклонением

Номер массива	Эвристическая мера	Номер массива	Эвристическая мера	Номер массива	Эвристическая мера
I	0,86	IV	0,72	VII	0,81
II	0,78	V	0,93	VIII	0,62
III	0,97	VI	0,77	IX	0,61

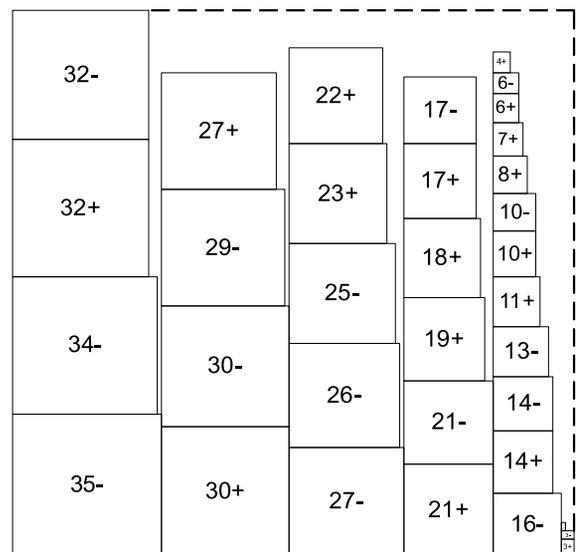


Рис. 14. Укладка массива IX *V*-уровневым алгоритмом по высоте с минимальным отклонением

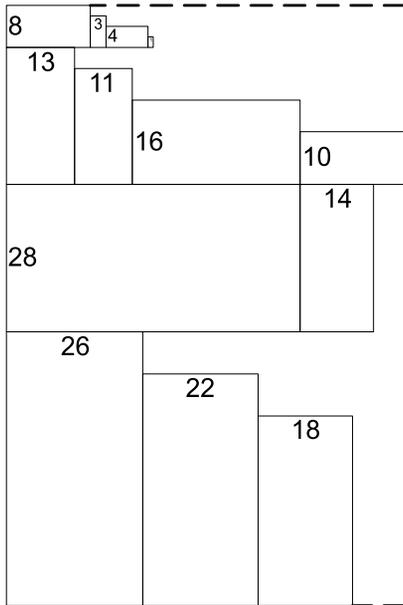


Рис. 15. Укладка массива III *H*-уровневым алгоритмом по протяженности с недостатком

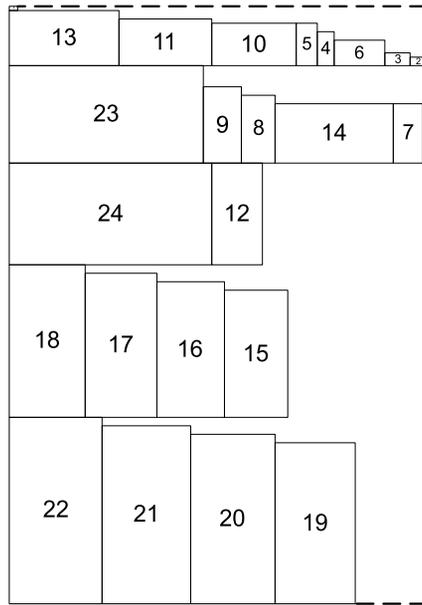


Рис. 16. Укладка массива VII *H*-уровневым алгоритмом по протяженности с недостатком

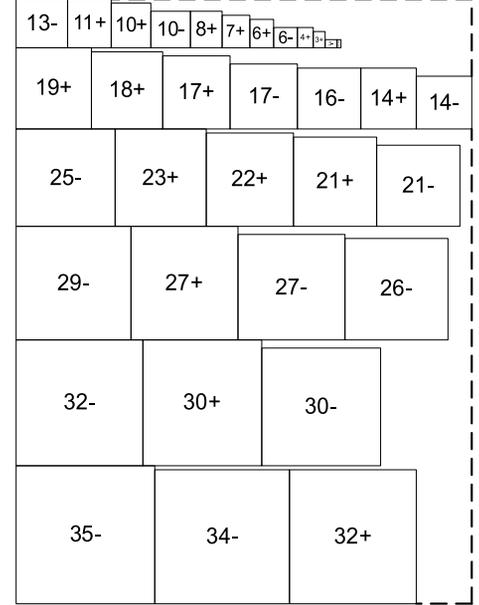


Рис. 17. Укладка массива IX *H*-уровневым алгоритмом по протяженности с недостатком

Таблица 4

Эвристические меры ресурсных оболочек *H*-уровневого алгоритма по протяженности с недостатком

Номер массива	Эвристическая мера	Номер массива	Эвристическая мера	Номер массива	Эвристическая мера
I	0,71	IV	0,69	VII	0,80
II	0,73	V	0,70	VIII	0,67
III	0,79	VI	0,72	IX	0,64

Построения *H*-уровневым алгоритмом по протяженности с недостатком для массивов III, VII, IX приведены на рис. 15–17.

Эвристические меры ресурсных оболочек *H*-уровневого алгоритма по протяженности с недостатком для массива из заявок кругового и гиперболического квадратичного типов приведены в табл. 4.

Видим, что эвристические меры ресурсных оболочек *H*-уровневого алгоритма по протяженности с недостатком не превосходят значения $0,5 + 0,30$.

Построения *H*-уровневым алгоритмом по протяженности с превышением для массивов III, VII, IX приведены на рис. 18–20.

Эвристические меры ресурсных оболочек *H*-уровневого алгоритма по протяженности с превышением для массива из заявок кругового и гиперболического квадратичного типов приведены в табл. 5.

Видим, что эвристические меры ресурсных оболочек *H*-уровневого алгоритма по протяженности с превышением не превосходят значения $0,5 + 0,43$.

Построения *H*-уровневым алгоритмом по протяженности с минимальным отклонением для массивов III, VII, IX приведены на рис. 21–23.

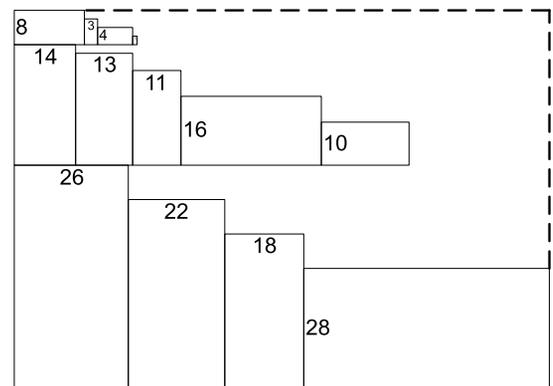


Рис. 18. Укладка массива III *H*-уровневым алгоритмом по протяженности с превышением

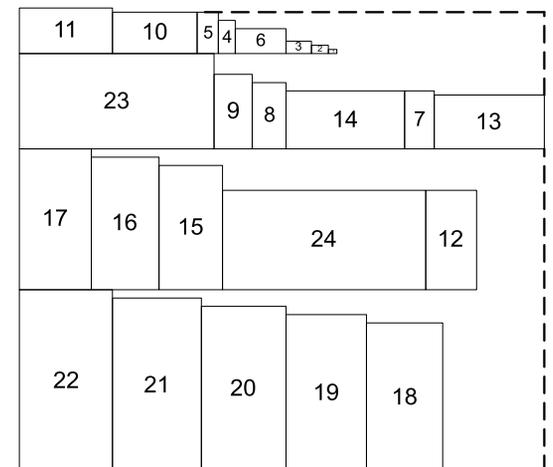


Рис. 19. Укладка массива VII *H*-уровневым алгоритмом по протяженности с превышением

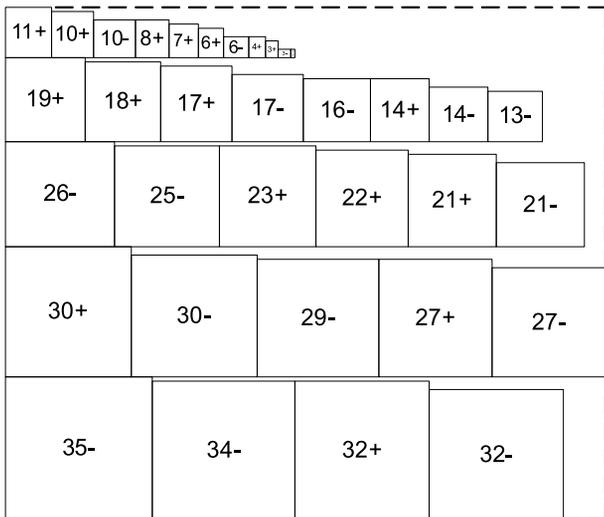


Рис. 20. Укладка массива IX H -уровнем алгоритмом по протяженности с превышением

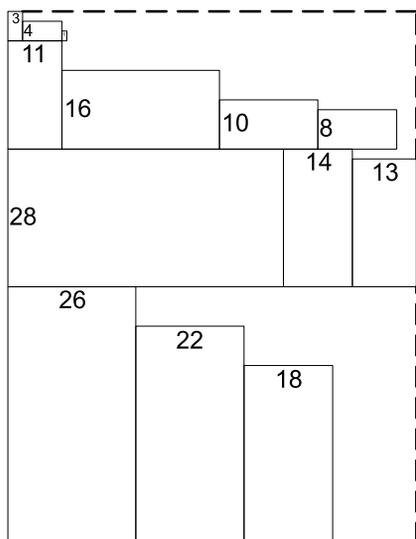


Рис. 21. Укладка массива III H -уровнем алгоритмом по протяженности с минимальным отклонением

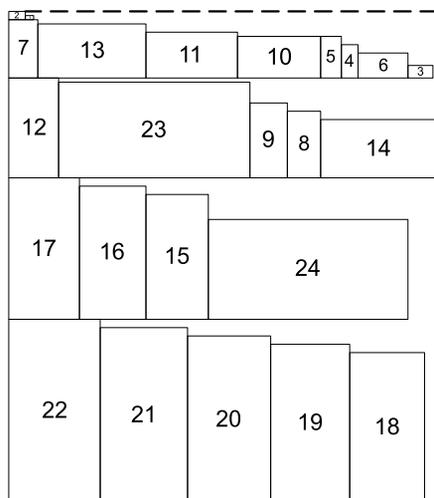


Рис. 22. Укладка массива VII H -уровнем алгоритмом по протяженности с минимальным отклонением

Таблица 5

Эвристические меры ресурсных оболочек H -уровневого алгоритма по протяженности с превышением

Номер массива	Эвристическая мера	Номер массива	Эвристическая мера	Номер массива	Эвристическая мера
I	0,67	IV	0,62	VII	0,72
II	0,81	V	0,68	VIII	0,62
III	0,93	VI	0,64	IX	0,63

Таблица 6

Эвристические меры ресурсных оболочек H -уровневого алгоритма по протяженности с минимальным отклонением

Номер массива	Эвристическая мера	Номер массива	Эвристическая мера	Номер массива	Эвристическая мера
I	0,71	IV	0,70	VII	0,63
II	0,67	V	0,66	VIII	0,64
III	0,75	VI	0,71	IX	0,61

Эвристические меры ресурсных оболочек H -уровневого алгоритма по протяженности с минимальным отклонением для массива из заявок кругового и гиперболического квадратичного типов приведены в табл. 6.

Видим, что эвристические меры ресурсных оболочек H -уровневого алгоритма по протяженности с минимальным отклонением не превосходят значения $0,5 + 0,25$.

Графики эвристической меры ресурсных оболочек V -уровневыми алгоритмами по высоте (с недостатком, с превышением, с минимальным отклонением) и H -уровневыми алгоритмами по протя-

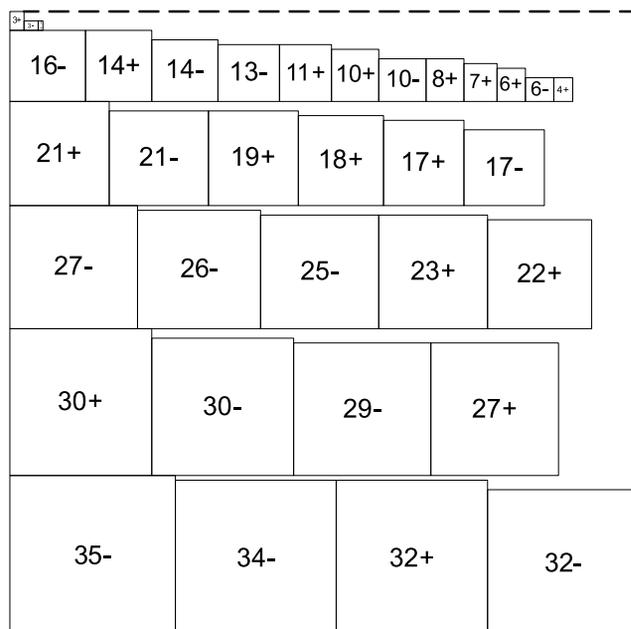


Рис. 23. Укладка массива IX H -уровнем алгоритмом по протяженности с минимальным отклонением

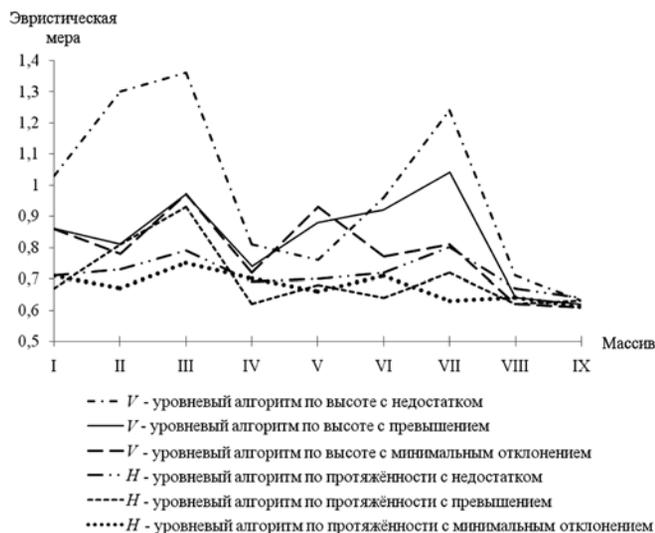


Рис. 24. Эвристические меры ресурсных оболочек V-уровневых и H-уровневых алгоритмов

женности (с недостатком, с превышением, с минимальным отклонением) при диспетчеризации массивами I—IX показаны на рис. 24.

Видим, что H-уровневые алгоритмы по протяженности имеют меньшую эвристическую меру ресурсных оболочек. При этом H-уровневый алгоритм по протяженности с минимальным отклонением имеет наименьшее значение максимума эвристической меры $0,5 + 0,25$ на рассматриваемых тестовых массивах ресурсов прямоугольников. Проведенный анализ позволяет рекомендовать предложенные полиномиальные алгоритмы к использованию в Grid-системах с централизованной структурой и мультисайтным режимом диспетчеризации при обслуживании массивов, состоящих из заявок кругового и гиперболического квадратичного типов.

Заключение

В среде ресурсных прямоугольников предлагаются варианты уровневых алгоритмов по высоте и по протяженности, позволяющие диспетчеризацию массивами ресурсных прямоугольников, не обладающих свойством монотонности горизонтальных или вертикальных измерений. Для девяти тестовых массивов заявок кругового и гиперболического квадратичного типов вычисляются эвристические меры ресурсных оболочек шести вариантов уровневых алгоритмов. Исследование показало адаптированность ряда предложенных полиномиальных алгоритмов к рассматриваемому классу массивов заявок пользователей Grid-систем.

Список литературы

- Schwiegelshohn U., Badia R., Bubak M., Danelutto M., Dustdar S., Gagliardi F., Geiger A., Hluchy L., Kranzlmüller D., Laure E., Priol T., Reinefeld A., Resch M., Router A., Rienhoff O., Rütter T., Sloot P., Talia D., Ullmann K., Yahyapour R., Voigt G.

Perspectives on grid computing // Future Generation Computer Systems. 2010. Vol. 26, Iss. 8. P. 1104—1115.

- Bencivenni M., Michelotto D., Alfieri R., Brunetti R., Ciccanti A., Cesini D., Costantini A., Fattibene E., Gaido L., Misurelli G., Ronchieri E., Salomoni D., Veronesi P., Venturi V., Vistoli M. Accessing Grid and Cloud services through a scientific web portal // J. Grid Computing. 2015. Vol. 13, Iss. 2. P. 159—175.

- Kahanwal B., Singh T. The distributed computing paradigms: p2p, grid, cluster, cloud, and jungle // International Journal of Latest Research in Science and Technology. 2012. Vol. 1, Iss. 2. P. 183—187.

- Buyya R., Broberg J., Goscinski A. (eds.). Cloud computing: principles and paradigms. Hoboken: John Wiley & Sons Inc., 2011, 644 p.

- Seinstra F., Maassen J., Nieuwpoort R., Drost N., Kessel T., Werkhoven B., Urbani J., Jacobs C., Kielmann T., Bal H. Jungle computing: Distributed supercomputing beyond clusters, grids, and clouds / M. Cafaro, G. Aloisio (eds.) // Grids, Clouds and Virtualization. London: Springer-Verlag, 2011, 235 p., pp. 167—197.

- Li M., Baker M. The grid: core technologies. Chichester: John Wiley & Sons Ltd, 2005, 452 p.

- Pugliese A., Talia D., Yahyapour R. Modeling and Supporting Grid Scheduling // J. Grid Computing. 2008. Vol. 6, Iss. 2. P. 195—213.

- Rahman M., Ranjan R., Buyya R., Benattallah B. A taxonomy and survey on autonomic management of applications in grid computing environments // Concurrency Computat.: Pract. Exper. 2011. Vol. 23, Iss. 16. P. 1990—2019.

- Ye D., Zhang G. On-Line Scheduling of Parallel Jobs // LNCS. 2004. Vol. 3104. P. 279—290.

- Keitelson D., Rudolph L. Toward convergence in job schedulers for parallel supercomputers // LNCS. 1996. Vol. 1162. P. 1—26.

- Feitelson D., Rudolph L., Schwiegelshohn U., Sevcik K., Wong P. Theory and practice in parallel job scheduling // LNCS. 1997. Vol. 1291. P. 1—34.

- Majumdar S., Parsons E. Parallel job scheduling: a performance perspective // LNCS. 2000. Vol. 1769. P. 233—252.

- Caramia M., Giordani S., Iovanella A. Grid scheduling by on-line rectangle packing // Networks. 2004. Vol. 44, N. 2. P. 106—119.

- Hamscher V., Schwiegelshohn U., Streit A., Yahyapour R. Evaluation of job-scheduling strategies for grid computing // LNCS. 2000. Vol. 1971. P. 191—202.

- Sonmez O., Mohamed H., Epema D. On the benefit of processor coallocation in multicluster grid systems // IEEE Trans. on Paral. and Distrib. Syst. 2010. Vol. 21, N. 6. P. 778—789.

- Саак А. Э. Полиномиальные алгоритмы распределения ресурсов в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2013. № 7. Приложение. 32 с.

- Саак А. Э. Управление ресурсами и заявками пользователей в Grid-системах с централизованной архитектурой // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. Москва, 16—19 июня 2014 г. М.: Институт проблем управления им. В. А. Трапезникова РАН, 2014. С. 7489—7498.

- Саак А. Э. Локально-оптимальные ресурсные распределения // Информационные технологии. 2011. № 2. С. 28—34.

- Саак А. Э. Алгоритмы диспетчеризации в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2011. № 11. С. 9—13.

- Саак А. Э. Сравнительный анализ полиномиальных алгоритмов диспетчеризации в Grid-системах // Информационные технологии. 2012. № 9. С. 28—32.

- Саак А. Э. Уровневые алгоритмы диспетчеризации массивами заявок кругового типа в Grid-системах // Известия ЮФУ. Технические науки. 2015. № 6 (167). С. 223—231.

- Саак А. Э. Кольцевые алгоритмы диспетчеризации массивами заявок в Grid-системах // Информационные технологии. 2016. № 3. С. 163—169.

- Саак А. Э. Диспетчеризация заявок кругового типа в Grid-системах // Информационные технологии. 2016. № 1. С. 37—41.

- Jepsen C. Dissections into 1:2 rectangles // Discrete Mathematics. 1996. Vol. 148. P. 107—117.

- Friedman E. Dominoes in squares. 2014. URL: <http://www2.stetson.edu/~efriedma/domino/>

- Korf R., Moffitt M., Pollack M. Optimal rectangle packing // Annals of Operations Research. 2010. Vol. 179. Iss. 1. P. 261—295.

- Huang E., Korf R. Optimal rectangle packing: an absolute placement approach // Journal of Artificial Intelligence Research. 2013. Vol. 46. P. 47—87.

- Friedman E. Tiling Almost Squares. 2013. URL: <http://www2.stetson.edu/~efriedma/almost/>

Scheduling of Sets of Hyperbolic-Type and Circular-Type Tasks in Grid-Systems

The annotation: Users' growing demand for computational capability, along with the progress in technologies, stimulate the transition from metacomputing to Grid-computing at the end of past century, which is still stepping up today jointly with such distributed computing paradigms as cloud computing and jungle computing. This paper considers computer tasks with a priori defined time of execution, in which the number of required processors is specified by a user at the moment of their delivery to the system. In these conditions when assigning sequentially numbered processors for task handling, the task of schedule generation by a scheduling algorithm is equivalent to the task of resource rectangles distribution. User's task, which needs service from Grid system's scheduler, is presented as a resource rectangle with horizontal and vertical dimensions, correspondingly, are set equal to the number of processors and time units required to complete a task. Grid-systems with centralized structure of their scheduling system which consist of sites which contain parallel systems which function in the mode of multisite scheduling are simulated by the resource quadrant. The author defined the resource rectangles environment, as a theoretical base of algorithmic support of polynomial complexity scheduling for the purpose of distribution management of computer and time resources and users' tasks in Grid-systems. In the environment of resource rectangles versions of level algorithms by height and by length are suggested. They allow scheduling by the sets of resource rectangles which don't have the feature of monotony of horizontal and vertical dimensions. This paper poses a question of suggested polynomial algorithms adaptiveness for the sets which consists of tasks of hyperbolic and quadratic, and circular and quadratic types. The comparative analysis allows recommending some of the algorithms to use in Grid systems with centralized structure.

Keywords: Grid system, centralized structure of scheduling system, multisite mode of task service, non-preemptive task service, circular-type task, hyperbolic-type task, Non-Euclidean heuristic measure, polynomial complexity of an algorithm, level algorithm by height, level algorithm by length

References

1. Schwiegelshohn U., Badia R., Bubak M., Danelutto M., Dustdar S., Gagliardi F., Geiger A., Hluchy L., Kranzlmüller D., Laure E., Priol T., Reinefeld A., Resch M., Reuter A., Rienhoff O., Rüttger T., Shot P., Talia D., Ullmann K., Yahyapour R., Voigt G. Perspectives on grid computing, *Future Generation Computer Systems*, 2010, vol. 26, Iss. 8, pp. 1104–1115.
2. Bencivenni M., Michelotto D., Alfieri R., Brunetti R., Ciccanti A., Cesini D., Costantini A., Fattibene E., Gaido L., Misurelli G., Ronchieri E., Salomoni D., Veronesi P., Venturi V., Vistoli M. Accessing Grid and Cloud services through a scientific web portal, *J. Grid Computing*, 2015, vol. 13, Iss. 2, pp. 1159–175.
3. Kahanwal B., Singh T. The distributed computing paradigms: p2p, grid, cluster, cloud, and jungle, *International Journal of Latest Research in Science and Technology*, 2012, vol. 1, Iss. 2, p. 183–187.
4. Buyya R., Broberg J., Goscinski A. (eds.). *Cloud computing: principles and paradigms*. Hoboken: John Wiley & Sons Inc., 2011, 644 p.
5. Seinstra F., Maassen J., Nieuwpoort R., Drost N., Kessel T., Werkhoven B., Urbani J., Jacobs C., Kielmann T., Bal H. Jungle computing: Distributed supercomputing beyond clusters, grids, and clouds. M. Cafaro, G., Aloisio (eds.). *Grids, Clouds and Virtualization*. London: Springer-Verlag, 2011, 235 p., pp. 167–197.
6. Li M., Baker M. *The grid: core technologies*. Chichester: John Wiley & Sons Ltd, 2005, 452 p.
7. Pugliese A., Talia D., Yahyapour R. Modeling and Supporting Grid Scheduling, *J. Grid Computing*, 2008, vol. 6, Iss. 2, pp. 195–213.
8. Rahman M., Ranjan R., Buyya R., Benatallah B. A taxonomy and survey on autonomic management of applications in grid computing environments, *Concurrency Computat: Pract. Exper*, 2011, vol. 23, Iss. 16, pp. 1990–2019.
9. Ye D., Zhang G. On-Line Scheduling of Parallel Jobs, *LNCS*, 2004, vol. 3104, pp. 279–290.
10. Feitelson D., Rudolph L. Toward convergence in job schedulers for parallel supercomputers, *LNCS*, 1996, vol. 1162, pp. 1–26.
11. Feitelson D., Rudolph L., Schwiegelshohn U., Sevcik K., Wong P. Theory and practice in parallel job scheduling, *LNCS*, 1997, vol. 1291, pp. 1–34.
12. Majumdar S., Parsons E. Parallel job scheduling: a performance perspective, *LNCS*, 2000, vol. 1769, pp. 233–252.
13. Caramia M., Giordani S., Iovanella A. Grid scheduling by on-line rectangle packing, *Networks*, 2004, vol. 44, no. 2, pp. 106–119.
14. Hamscher V., Schwiegelshohn U., Streit A., Yahyapour R. Evaluation of job-scheduling strategies for grid computing, *LNCS*, 2000, vol. 1971, pp. 191–202.
15. Sonmez O., Mohamed H., Epema D. On the benefit of processor co location in multicluster grid systems, *IEEE Trans. on Paral. and Distrib. Syst.*, 2010, vol. 21, no. 6, pp. 778–789.
16. Saak A. Eh. Polinomialnye algoritmy raspredeleniya resursov v Grid-sistemakh na osnove kvadrachnoy tipizatsii massivov zayavok [Polynomial algorithms for resource allocation in Grid-based systems for quadratic typing, arrays applications], *Informacionnye tekhnologii* [Information Technologies], 2013, no. 7. Prilozhenie, 32 p.
17. Saak A. Eh. Upravleniye resursami i zayavkami polzovatelye v Grid-sistemakh s centralizovannoy arkhitektyroy [Resource and multi-processor task management in Grid system of centralized architecture], *Trudy XII Vserossiyskogo soveshanya po problem upravleniya VSPU-2014*. Moskva, 16 ijunya — 19 ijunya 2014 g. [Proceedings of XII all-Russian conference "Control problems" RCCP'2014. Moscow, 16 June — 19 June 2014]. Moscow: Institut problem upravleniya im. V. A. Trapeznikova RAN, 2014, pp. 7489–7498.
18. Saak A. Eh. Lokalno-optimalnye resursnye raspredeleniya [Locally optimal resource allocation], *Informatsionniye tekhnologii* [Information Technologies], 2011, no. 2, pp. 28–34.
19. Saak A. Eh. Algoritmy dispatcherizatsii v Grid-sistemakh na osnove kvadrachnoy tipizatsii massivov zayavok [Algorithms scheduling in Grid-based systems for quadratic typing, arrays applications], *Infoniatsionnye tekhnologii* [Information Technologies], 2011, no. 11, pp. 9–13.
20. Saak A. Eh. Sravnitelniy analiz polinomialnykh algoritmov dispatcherizatsii v GRID-sistemakh [Comparative analysis of poly-

nomial algorithms for scheduling in GRID-systems], *Informatsionnye tekhnologii* (Information Technologies), 2012, no. 9, pp. 28–32.

21. **Saak A. Eh.** Urovnevye algoritmy dispatcherizatsii massivamy zayavok krygovogo tipa v Grid-sistemakh [Level algorithms of scheduling by circle type task sets in grid systems], *Izvestiya YuFU. Tekhnichskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, no. 6 (167), pp. 223–231.

22. **Saak A. Eh.** Kolcivie algoritmy dispatcherizatsii massivamy zayavok v Grid-sistemakh [Ring algorithms for scheduling in grid systems by sets of tasks], *Informatsionnye tekhnologii* [Information Technologies], 2016, no. 3, pp. 163–169.

23. **Saak A. Eh.** Dispatcherizatsiya zayavok krygovogo tipa v Grid-sistemakh [Circular-typed multiprocessor tasks scheduling in

grid systems], *Informatsionnye tekhnologii* (Information Technologies), 2016, no. 1, pp. 37–41.

24. **Jepsen C.** Dissections into 1:2 rectangles, *Discrete Mathematics*, 1996, vol. 148, pp. 107–117.

25. **Friedman E.** *Dominoes in squares*. 2014. URL: <http://www2.stetson.edu/~efriedma/domino/>

26. **Korf R., Moffitt M., Pollack M.** Optimal rectangle packing, *Annals of Operations Research*, 2010, vol. 179, Iss. 1, pp. 261–295.

27. **Huang E., Korf R.** Optimal rectangle packing: an absolute placement approach, *Journal of Artificial Intelligence Research*, 2013, vol. 46, pp. 47–87.

28. **Friedman E.** *Tiling Almost Squares*. 2013. URL: <http://www2.stetson.edu/~efriedma/almost/>

УДК 004.912; 519.254

А. С. Мохов¹, ассистент, e-mail: asmokhov@mail.ru,
В. О. Толчеев², д-р техн. наук, проф., e-mail: tolcheevvo@mail.ru
НИУ "МЭИ"

Способы учета структуры научных документов в задачах обработки и анализа текстовой информации

Рассмотрены процедуры обработки и анализа текстовой информации на основе учета структуры документа. Приведены основные модели представления текстов в задачах машинного обучения. Показана эффективность использования частично структурированных моделей для информационного поиска, автоматического аннотирования, выявления нечетких дубликатов и классификации. Наряду с известными подходами в работе излагаются предложенные авторами процедуры, учитывающие особенности двуязычных библиографических документов и позволяющие проводить высокоточную классификацию.

Ключевые слова: интеллектуальный анализ текстовых данных, модель текстового документа, информационный поиск, автоматическое аннотирование, выявление нечетких дубликатов, классификация двуязычных библиографических документов

Введение

В рамках работ по интеллектуальному анализу текстовых данных (Text Mining) решаются задачи поиска информации, кластеризации, фильтрации и классификации документов, автоматического аннотирования и извлечения ключевых понятий из текстов, выявления плагиата и нечетких дубликатов в документальных массивах, создания онтологий и построения моделей предметных областей. Общей особенностью вышеуказанных задач является их слабая формализованность, сложность математического описания текстов.

Под структурой документа далее понимается его логическое построение в виде взаимосвязанных фрагментов. Структура текстов во многом обуславливается функциональным стилем, которому соответствует публикация. Различают публицистический, официально-деловой, научно-технический, разговорный, художественный стили. В данной работе рассматривается научно-технический стиль, который используется в монографиях, статьях, учебни-

ках, диссертациях, отчетах по НИР и т. д. Для научных статей в качестве фрагментов целесообразно рассматривать название, аннотацию, ключевые слова, введение, основную часть, заключение. Для учебного пособия или монографии это могут быть главы, подглавы, параграфы и т. п.

Характерной особенностью научно-технических документов является наличие наряду с полным текстом короткого библиографического описания (БО), включающего имена и фамилии авторов, название, аннотацию, ключевые слова, место издания и другую вспомогательную информацию. Чаще всего БО представлены на двух языках: русском и английском. В библиографических описаниях, как и в других коротких текстах (сообщения электронной почты, новостная информация, описания товаров, инструкции по применению, анкеты, справочники), наиболее четко прослеживается общая структура. В простейшем случае библиографический научный документ может быть описан кортежем $\langle T, A, K \rangle$, в который включены название (title) — T , аннотация (abstract) — A , ключевые слова (key

words) — *K*. Именно научные публикации, заданные своими двуязычными (русско-английскими) БО, рассматриваются в данной статье, и применительно к ним показана целесообразность использования заранее известной структуры для более эффективного решения прикладных задач Text Mining.

Основные модели представления текстовых документов

Результаты поиска, обработки и анализа текстовой информации существенным образом зависят от выбранного способа представления и математического описания документа. Чаще всего используется широко известная в литературе модель "мешок слов" ("bag of words"), которая представляет публикацию в виде набора не связанных между собой слов [1]:

$$X_j = [x_j^{(1)}, \dots, x_j^{(i)}, \dots, x_j^{(M)}]^T, \quad (1)$$

где $x_j^{(i)}$ — вес термина i в документе j ($j = 1, \dots, N$; N — число документов в выборке; $i = 1, \dots, M$; M — число информативных терминов после удаления служебных слов). Значение $x_j^{(i)}$ будет существенно зависеть от способа взвешивания терминов (в задачах Text Mining обычно используются *tf*-, *tf-idf*-, *tfc*-, *lfc*-взвешивания) [2, 3]. К информативным словам относятся:

- общенаучные (высокочастотные) термины, единые для различных предметных областей;
- междисциплинарные и узкоспециализированные понятия (среднечастотные слова), характеризующие тематику статьи;
- редкие (низкочастотные) термины, соответствующие конкретному научному направлению или отражающие авторский стиль.

Выборка текстовых документов может быть представлена в виде матрицы "документ—термин":

$$X = \begin{bmatrix} x_1^{(1)} & \dots & x_1^{(i)} & \dots & x_1^{(M)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j^{(1)} & \dots & x_j^{(i)} & \dots & x_j^{(M)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^{(1)} & \dots & x_N^{(i)} & \dots & x_N^{(M)} \end{bmatrix}. \quad (2)$$

Более полное описание документов благодаря использованию всей доступной информации достигается в *частично структурированных* и *полностью структурированных* моделях. В частично структурированных моделях при определении весов терминов учитываются дополнительные характеристики: порядок следования слов (символов), их расположение в документе (заголовки, аннотация, ключевые слова, первый абзац и т. д.), наличие специального оформления термина стилистически или с

помощью шрифта. Кроме того, в ряде случаев проводится выделение гиперонимов — слов с более широким значением (нож — (это) оружие, бульдог — собака — животное) и словосочетаний — устойчивых групп терминов, которые образуют общие понятия для данной предметной области [4–6]. Частично структурированные модели весьма эффективны при обработке и анализе библиографических документов, новостной и справочной информации, анкет, т. е. коротких текстов, которые обладают заранее известной структурой.

В полностью структурированных моделях применяются заранее сформированные базы знаний, содержащие ключевые слова, их сочетания, а также иерархические связи, свойственные предметной области. Традиционно для этого разрабатываются онтологии и тезаурусы, на верхних уровнях которых находятся общие термины, уточняемые на более низких уровнях [7].

В специализированной литературе активно обсуждается вопрос, насколько высокая сложность и существенные трудозатраты при разработке частично структурированных и полностью структурированных моделей гарантируют значимое улучшение целевых показателей (например, точности-быстродействия классификации документов или полноты-точности поиска информации). Необходимо признать, что в профессиональном сообществе пока не выработано единой позиции по этой проблеме [4, 7, 8]. Справедливо отмечается, что выбор модели, прежде всего, зависит от области применения (сложности решаемой задачи Text Mining), требований целевого критерия, имеющейся априорной информации [2].

Решение проблемы обработки и анализа больших массивов информации (так называемой проблемы больших данных — Big Data), требующее распределения колоссального числа документов по многочисленным классам, целесообразно организовывать на базе полностью структурированных моделей — онтологий. В настоящее время для описания в онтологиях семантики лексических единиц (слов, словосочетаний, предложений и дискурсов) разработаны эффективные подходы, в частности, теория *K*-представлений (концептуальных представлений) [18–20].

Для широкого круга задач классификации, предусматривающих разнесение небольших выборок по сравнительно малому числу классов, достаточно использовать модель "мешок слов". При этом более высокая точность может быть достигнута путем проведения комплексного обучения решающего правила на большом числе выборок, тщательного отбора информативных терминов с помощью специализированных процедур, формирования коллективов решающих правил, использования дополнительной априорной информации (например, структуры БО) [2, 9].

Частично структурированные модели в задачах поиска, обработки и анализа информации

В центре внимания данной статьи находятся такие задачи Text Mining, для решения которых, на наш взгляд, целесообразно применять частично структурированные модели. Рассмотрим эти задачи более подробно.

Информационный поиск. На раннем этапе разработка частично структурированных моделей проводилась, прежде всего, в рамках работ по теории информационного поиска в целях повышения релевантности выдаваемых документов. В этих исследованиях проверялась эффективность применения цитирования (перекрестных ссылок), учета местоположения и частоты появления терминов, анализа стиля и шрифта написания слов для улучшения показателя полнота-точность. В результате были разработаны достаточно универсальные подходы, в частности PageRank, широко известная процедура Google, предназначенная для оценки качества Web-страниц на основании информации о количестве имеющихся на нее ссылок [5]. Более комплексный подход, применимый как к Web-документам, так и журнальным публикациям, предложили В. Фресно и А. Рибейро (V. Fresno, A. Ribero) [10].

В данной модели рассчитываются следующие характеристики текстов:

- частота i -го термина в документе: $f_f(i) = n_f(i)/N_{tot}$;
- частота i -го термина в заголовках документа: $f_t(i) = n_t(i)/N_{tit}$;
- частота i -го термина среди выделенных слов (жирное начертание, курсив): $f_e(i) = n_e(i)/N_{emph}$;
- функция положения i -го термина в документе:

$$f_p(i) = \frac{\frac{3}{4}n_{1,4}(i) + \frac{1}{4}n_{2,3}(i)}{\sum_{l=1}^k \left(\frac{3}{4}n_{1,4}(l) + \frac{1}{4}n_{2,3}(l) \right)}$$

Дадим расшифровку обозначений: $n_{1,4}(i)$ — число вхождений слова в первую и четвертую четверти документа; $n_{2,3}(i)$ — число вхождений слова во вторую и третью четверти документа; k — число различных терминов в соответствующих частях документа; $n_f(i)$, $n_t(i)$, $n_e(i)$ — число раз, которые i -й термин встречался в документе, заголовке, среди выделенных слов; $N_{tot}(i)$, $N_{tit}(i)$, $N_{emph}(i)$ — число терминов в документе, заголовке, общее число выделенных слов.

Итоговый вес вычисляется как линейная комбинация вышеуказанных показателей:

$$r_i = C_f f_f(i) + C_t f_t(i) + C_e f_e(i) + C_p f_p(i). \quad (3)$$

Здесь C_f , C_p , C_e , C_p — настраиваемые коэффициенты, определяющие степень влияния различных характеристик термина на значение результирующего веса.

Эмпирически авторы получили следующие значения коэффициентов:

$$r_i = 0,3f_f(i) + 0,15f_t(i) + 0,25f_e(i) + 0,3f_p(i). \quad (4)$$

В модели В. Фресно и А. Рибейро настройка параметров проводится на основе экспериментальных исследований. Поэтому в новых условиях (другая предметная область, измененный размер и структура Web-страниц) придется заново настраивать весовые коэффициенты и искать их наилучшее сочетание. Возможно, этим объясняется весьма ограниченное практическое применение формулы (4) в задачах Text Mining.

Автоматическое аннотирование. Автоматическое аннотирование (реферирование) заключается в выявлении наиболее информативных фрагментов полнотекстового документа и их объединении в короткую аннотацию-реферат (отметим, что такие рефераты во многом являются аналогом библиографического описания) [5]. Кроме анализа частоты появления термина, его стиля и шрифта, контекста (окружение термина, очередность следования слов вокруг него), при автоматическом аннотировании анализируется местоположение термина или фрагмента в документе. Это позволяет корректно оценить их значимость и полезность для включения в короткое описание. В литературе предложены различные способы определения ценности фрагмента, например можно использовать формулу [5]

$$Weight(r) = Location(r) + KeyPhrase(r) + StatTerm(r) + AddTerm(r), \quad (5)$$

где r — фрагмент исходного документа ($r = 1, \dots, R$, R — общее число фрагментов в тексте); $Location$ — весовой коэффициент, значение которого зависит от места появления фрагмента (начало, середина или конец параграфа, введение, заключение); $KeyPhrase$ — весовой коэффициент, увеличивающий значимость фрагмента, который входит в ключевые конструкции-клише (например, для научных публикаций: статья посвящена..., рассматривается актуальная задача..., результаты эксперимента... и т. п.); $StatTerm$ — весовой коэффициент, учитывающий частоту встречаемости фрагмента в тексте; $AddTerm$ — весовой коэффициент, повышающий вес фрагмента в случае наличия в нем важных терминов (определенных пользователем или экспертом).

После расчета весов отбираются фрагменты, обладающие наибольшими значениями $Weight(r)$. Слабое место данного подхода заключается в неформализованности расчета весовых коэффициентов, которые в ряде случаев могут назначаться самими разработчиками исходя лишь из собственного опыта и предпочтений.

Выявление нечетких дубликатов. К дубликатам (неуникальным публикациям) принято относить документы с идентичным содержанием. Нечеткими (неполными) дубликатами или почти дубликатами

считаются документы, в содержательную часть которых внесены незначительные изменения. Проблема выявления нечетких дубликатов возникает при поиске информации (устранение одинаковых документов, выдаваемых информационно-поисковой системой), формировании цифровых библиотек и банков данных (исключение повторяющихся сведений, полученных из разнородных источников), проверке на оригинальность представленных для издания научных статей и докладов (обнаружение работ, которые были уже ранее опубликованы авторами в других журналах) [11, 12].

В литературе по Text Mining предложены и исследованы различные способы выявления нечетких дубликатов (специализированные расстояния, меры близости и коэффициенты ассоциативности, метод шинглов и его модификации, процедуры на основе расчета весов терминов и т. п.) [11]. В известных подходах практически не используется априорная информация о строении документа. Вместе с тем идентификация дубликатов часто проводится именно среди коротких документов, хорошо поддающихся структурированию (библиографические описания научных статей, новостные сообщения, анкеты и т. п.).

Рассматриваемый в данной работе метод — *Обобщенный Коэффициент Ассоциативности (ОКА)* — предназначен для обнаружения нечетких дубликатов по библиографическим описаниям научных статей. Его специфика заключается в выявлении не только совпадающих терминов двух сравниваемых текстов, но и в учете того, в каком месте БО (название или аннотация) это совпадение произошло. Расчет ОКА основан на вычислении различных коэффициентов ассоциативности, определяемых отдельно для названий и аннотаций [13].

Введем следующие обозначения, используемые при анализе таблиц сопряженности размера 2×2 : $A_{\text{назван}}$ и $A_{\text{аннот}}$ — число совпавших терминов соответственно в названиях и аннотациях двух документов X_j и X_i ; $B_{\text{назван}}$ и $B_{\text{аннот}}$ — число терминов (соответственно из названий и аннотаций), имеющих в X_j и отсутствующих в X_i ; $C_{\text{назван}}$ и $C_{\text{аннот}}$ — число терминов (соответственно из названий и аннотаций), имеющих в X_i и отсутствующих в X_j .

ОКА определяется по формуле [13] :

$$\begin{aligned} \text{ОКА} &= \frac{1}{2} (K_1 + K_2) = \\ &= \frac{1}{2} \left\{ \frac{A_{\text{назван}}}{\max(A_{\text{назван}}, B_{\text{назван}}, C_{\text{назван}})} + \right. \\ &\left. + \min\left(\frac{A_{\text{аннот}}}{A_{\text{аннот}} + B_{\text{аннот}}}; \frac{A_{\text{аннот}}}{A_{\text{аннот}} + C_{\text{аннот}}}\right) \right\}. \quad (6) \end{aligned}$$

Здесь K_1 — коэффициент, который вычисляется только по терминам, встречающимся в названиях

статей ($K_1 \in [0; 1]$). Такой расчет K_1 позволяет более качественно выявлять статьи с терминологически близкими названиями и публикации с "непохожими" заголовками; K_2 — коэффициент, который рассчитывается только по терминам из аннотаций ($K_2 \in [0; 1]$), равен наименьшему из выражений, стоящих в скобках. Это позволяет учесть наличие в двух аннотациях несовпадающих терминов. Уменьшение значений K_1 и (или) K_2 снижает ОКА, возрастание значений K_1 и (или) K_2 увеличивает ОКА. Усреднение суммы коэффициентов K_1 и K_2 приводит значения ОКА в диапазон $[0; 1]$ (равенство ОКА единице означает обнаружение полного дубликата).

Результаты исследования ОКА (расчет показателя полнота-точность на тестовых массивах) и сопоставление ОКА с известными процедурами (метод шинглов, коэффициент Жаккара, расстояние Джаро и Джаро-Винклера) приведены в работе [13]. Сравнительный анализ показал, что учет структуры документа в ОКА позволяет увеличить точность распознавания нечетких дубликатов без уменьшения показателя полноты.

Классификация библиографических текстовых документов. Несмотря на широкое использование модели "мешок слов" в задачах классификации, усиливается интерес к применению частично структурированных моделей для снижения ошибки группировки документов [2, 6, 14]. В данной работе рассматриваются подходы, которые позволяют повысить точность за счет анализа информации о структуре текстового документа, в частности, сведений о местоположении терминов. Так, в работе [9] предлагается следующая формула линейного взвешивания:

$$x_j^{(1)} = \alpha t_j^{(i)} + \beta a_j^{(i)} + \gamma k_j^{(i)}, \quad (7)$$

где $x_j^{(i)}$ — результирующий вес термина i в документе j ; $t_j^{(i)}$ — вес термина i в названии документа j ; $a_j^{(i)}$ — вес термина i в аннотации документа j ; $k_j^{(i)}$ — вес термина i в ключевых словах документа j ; α , β , γ — настраиваемые весовые коэффициенты.

Как и в других рассмотренных выше подходах, наиболее неформализованной проблемой при использовании формулы (7) является расчет весовых коэффициентов. В работе [9] для определения величин α , β , γ были использованы две стратегии, основанные на экспериментальных исследованиях. Согласно первой стратегии коэффициенты рассчитывались как отношение ошибок, получаемых при классификации текстовых документов при раздельном использовании терминов из названий, аннотаций и ключевых слов (для классификации использовались метод k -ближайших соседей и косинусоидальная мера близости). При использовании второй стратегии коэффициенты определялись как отношение числа терминов, встречавшихся в аннотациях, названиях и ключевых словах

обучающей выборки. Настройка коэффициентов α , β , γ по второй стратегии позволила увеличить точность классификации на 5 % [9].

В отличие от задач информационного поиска и автоматического аннотирования, для которых определение местоположения термина является стандартным приемом, при классификации документов этот подход не стал общепринятым, возможно, из-за большей трудоемкости и негарантированно-го улучшения точности. В данной работе исследуется целесообразность учета структуры текстов при классификации двуязычных (русско-английских) библиографических документов, к которым относятся, например, научные публикации, размещаемые на сайтах журналов и в цифровых библиотеках.

Классификация двуязычных библиографических текстовых документов

Дадим краткое описание используемых методов классификации на основе вычисления профилей [15]. Профиль класса состоит из информативных (классоразделяющих) терминов, способных характеризовать (представлять при классификации) все остальные элементы класса. Одним из наиболее известных в литературе профилей является центроид [1].

Для выявления информативных терминов и вычисления их весов на этапе обучения применяются специальные статистические, теоретико-информационные и эвристические критерии, рассчитываемые, как и в случае метода ОКА, на основе таблиц сопряженности размера 2×2 . Введем следующие обозначения: A — число раз, когда термин $x^{(i)}$ и класс Q_g встречаются вместе; B — число раз, когда $x^{(i)}$ встречается без Q_g ; C — число раз, когда Q_g встречается без $x^{(i)}$; D — число раз, когда ни Q_g , ни $x^{(i)}$ не встречаются; N — общее число наблюдений в выборке, χ^2 — величина Хи-квадрат критерия [6].

Приведем формулы для вычисления профилей разными методами с учетом введенных обозначений [15]:

РО-профиль (ρ):

$$\rho = \sqrt{\frac{\chi^2}{N}} = \frac{(AD - CB)}{\sqrt{(A + B)(C + D)(A + C)(B + D)}}. \quad (8)$$

Нормированный МИ-профиль (НМИ):

$$NMI = \frac{A \log \frac{AN}{(A + B)(A + C)}}{(A + B) \log \frac{N}{A + B}}. \quad (9)$$

Профиль Жаккара (J-профиль):

$$J = \frac{A}{A + B + C}. \quad (10)$$

В работе [15] также исследуются специально разработанные комбинированные профили UNI3 и UNI6, созданные на основе РО-, НМИ- и J-профилей. В профиль UNI3 прежде всего отбираются наиболее информативные русские термины, общие для РО- и J-профилей. Затем профиль дополняется самыми важными общими английскими терминами. При этом вес выбирается наибольший из РО- и J-профилей [16].

UNI3-профиль:

$$\omega_{UNI3} = \max(\omega_{PO}, \omega_J). \quad (11)$$

Для составления UNI6 используется аналогичная процедура, но на основе трех профилей (РО-, НМИ- и J-) веса исходных профилей суммируются:

UNI6-профиль:

$$\omega_{UNI6} = \omega_{PO} + \omega_{НМИ} + \omega_J. \quad (12)$$

В формулах (11) и (12) использованы следующие обозначения: ω_{UNI3} , ω_{UNI6} — веса слов в профилях UNI3 и UNI6, ω_{PO} , $\omega_{НМИ}$, ω_J — веса слов, рассчитанных по формулам (8)–(10).

На этапе классификации вычисляются значения весов классов для каждого нового документа (документа, который не использовался для построения профиля на стадии обучения):

$$W_g = \sum_{i=1}^{M_g} tf_i \text{Prof}(x^{(i)}, Q_g). \quad (13)$$

Здесь tf_i — частота встречаемости i -го признака в новом документе; M_g — число информативных терминов, включенных в профиль g -го класса на этапе обучения (в наших исследованиях все классы имели профиль одинакового размера $L = M_g$); $\text{Prof}(x^{(i)}, Q_g)$ — вес i -го термина в профиле, вычисленном по одной из формул (8)–(12).

Согласно решающему правилу (см. соотношение (13)) новый документ относится к тому классу, которому соответствует наибольший вес: $W_g = \max(W_g)$, $g = 1, \dots, G$.

Рассмотрим возможность использования априорной информации о структуре двуязычных библиографических описаний для проведения классификации. Логично предположить, что термины из различных фрагментов (название, аннотация, ключевые слова) имеют неодинаковую ценность и учет этого факта в формуле взвешивания улучшает результирующую точность. Далее проверяется предположение, что значимость (ценность, информативность) термина зависит не только от величин, вычисляемых по формулам (8)–(12), но и от весовых коэффициентов, характеризующих место появления слова. Предлагается общий вес i -го термина определять следующим образом:

$$\omega_{part}^{(i)} = k_{part} \text{Prof}_{part}(x^{(i)}, Q_g), \quad (14)$$

где "part" обозначает раздел БО: T — название, A — аннотация, K — ключевые слова и слова, выделенные стилем или шрифтом в тексте БО; k_{part} — весовой коэффициент, повышающий или уменьшающий значимость соответствующих разделов БО; $Prof_{part}(x^{(i)}, Q_g)$ — профиль класса, который используется для данного раздела БО.

Решающее правило (13) модифицируется следующим образом:

$$W_g = \sum_{i=1}^{Mg} tf_T^{(i)} \omega_T^{(i)} + tf_A^{(i)} \omega_A^{(i)} + tf_K^{(i)} \omega_K^{(i)} =$$

$$= \sum_{i=1}^{Mg} \alpha tf_T^{(i)} Prof_T(x^{(i)}, Q_g) + \beta tf_A^{(i)} Prof_A(x^{(i)}, Q_g) +$$

$$+ \gamma tf_K^{(i)} Prof_K(x^{(i)}, Q_g), \quad (15)$$

где α, β, γ — коэффициенты, повышающие или уменьшающие значимость термина в зависимости от его появления в одном из фрагментов документа; $\omega_T^{(i)}, \omega_A^{(i)}, \omega_K^{(i)}, tf_T^{(i)}, tf_A^{(i)}, tf_K^{(i)}$ — соответственно вес и частота i -го термина в названии, аннотации, ключевых словах классифицируемого документа;

$$\omega_T^{(i)} = Prof_T(x^{(i)}, Q_g),$$

$$\omega_A^{(i)} = Prof_A(x^{(i)}, Q_g), \quad \omega_K^{(i)} = Prof_K(x^{(i)}, Q_g).$$

Отметим также, что нами рассматриваются двуязычные БО, поэтому в профили, описывающие разные фрагменты, входят как русские, так и английские информативные слова. Далее проводятся экспериментальные исследования профильных методов; рассчитанных по формулам (8)—(12) и (14), на специально сформированных выборках.

Экспериментальное исследование разработанных методов

Для проведения исследований было составлено по 20 обучающих и экзаменационных выборок, состоящих из двуязычных БО научных статей. Выборки были сформированы из электронных библиотек eLibrary, Киберленинка, а также электронных журналов по 36 инженерно-техническим, естественнонаучным и гуманитарным тематикам. Каждая выборка состояла из семи классов, содержащих по 65 и 15 документов для обучения и экзамена. Ошибка рассчитывалась как число неправильных решений метода на экзаменационной выборке.

В ходе экспериментальных исследований ставилась задача подтвердить (или опровергнуть) предположения:

а) в формуле (15) целесообразно использовать различные профили для взвешивания терминов из разных фрагментов БО;

б) введение (и настройка) дополнительных коэффициентов α, β, γ для расчета общего веса термина в формуле (15) увеличивают точность классификации.

В табл. 1 приведены ошибки классификации, полученные с помощью профильных (РО, НМИ, J) и комбинированных (UNI3, UNI6) методов. Для них использовалось решающее правило, рассчитываемое по формуле (13). Кроме того, в табл. 1 содержатся ошибки составных профилей, в которых применяются различные профильные методы для взвешивания терминов из названий, аннотаций и ключевых слов. Например, UNI3-J-UNI6 означает, что для расчета веса слова в названии использовался метод UNI3, в аннотации — J, ключевых слов — UNI6. При проведении исследований значимость трех фрагментов при определении класса документа считалась одинаковой $\alpha = \beta = \gamma = 1$ и использовалось решающее правило, заданное формулой (15).

Как следует из табл. 1, использование различных способов взвешивания фрагментов (с одинаковыми коэффициентами $\alpha = \beta = \gamma = 1$) не дает значимого прироста в точности по сравнению с профилями, рассчитанными по формулам (8)—(12). В связи с этим главный акцент в экспериментальных исследованиях был сделан на снижении ошибки классификации за счет настройки весовых коэффициентов α, β, γ в соотношении (15). В качестве начальных приближений были взяты следующие значения коэффициентов по методу Фишберна $\langle \alpha, \beta, \gamma \rangle$: $\langle 1/6, 2/6, 3/6 \rangle$; $\langle 2/6, 1/6, 3/6 \rangle$; $\langle 1/6, 3/6, 2/6 \rangle$; $\langle 2/6, 3/6, 1/6 \rangle$; $\langle 3/6, 2/6, 1/6 \rangle$; $\langle 3/6, 1/6, 2/6 \rangle$, при этом $(\alpha + \beta + \gamma = 1)$ [17].

Наилучшие результаты для всех методов показал вариант, при котором названиям статей присваивался коэффициент $3/6$, аннотациям $1/6$ и ключевым словам $2/6$, т. е. $\alpha = 3/6; \beta = 1/6; \gamma = 2/6$. Результаты классификации для данных коэффициентов и выбранных методов приведены в табл. 2.

Целесообразность применения НМИ-профиля, обладающего достаточно большой ошибкой (см. табл. 1), для ключевых слов объясняется тем, что этот профиль устанавливает высокие веса для редких,

Таблица 1

Минимальные, максимальные и средние ошибки классификации

Метод	Ошибки классификации, %		
	Минимальная	Максимальная	Средняя
РО	5,71	20,95	12,04
НМИ	8,57	18,09	12,75
J	7,61	16,19	11,52
UNI3	8,57	16,19	12,13
UNI6	3,81	18,09	10,52
UNI3-J-UNI6	5,71	16,19	10,52
UNI3-J-НМИ	5,71	14,28	10,23
J-UNI6-НМИ	5,71	16,19	10,37

Минимальные, максимальные и средние ошибки при классификации с использованием коэффициентов Фишберна

Метод	Ошибки классификации, %		
	Минимальная	Максимальная	Средняя
UNI3-J-UNI6-Fishburn	5,71	15,23	10,33
UNI3-J-НМИ-Fishburn	6,67	14,28	9,71
J-UNI6-НМИ-Fishburn	3,81	15,23	9,56

специализированных терминов (и это представляется важным в контексте проводимого исследования). Остальные профили (прежде всего J и UNI6) были отобраны для исследования, так как они относятся к наиболее точным профильным методам. Кроме того, UNI3 и UNI6 являются комбинированными процедурами, учитывающими преимущества статистического, теоретико-информационного и эвристического взвешивания (см. формулы (11)—(12)).

Наши экспериментальные исследования показали, что способ расчета профилей для различных фрагментов БО практически не влияет на результирующую точность. Вместе с тем, введение дополнительных коэффициентов, присваивающих различный вес терминам из названий, аннотаций, ключевых и выделенных слов, позволяет улучшить точность на 0,96 % по сравнению с наиболее точным профильным методом UNI6 (на 1,96, 2,48 и 3,19 % соответственно для "классических" J-, РО- и НМИ-профилей).

Заключение

На наш взгляд, широко распространенная практика использования априорных знаний о структуре документов при решении задач информационного поиска и автоматического аннотирования может быть успешно применена для выявления нечетких дубликатов и увеличения точности классификации текстовых данных. Резюмируя основные результаты, приведенные в статье, отметим следующие:

- исследованы профильные методы, на базе которых предложены новые модификации;
- сформированы обучающие и экзаменационные выборки, состоящие из двуязычных (русско-английских) библиографических описаний;
- экспериментально установлено снижение ошибки классификации для комплексного решающего правила (см. формулу (15)), в котором учитывается структура текста (местоположение термина в БО).

При проведении дальнейших исследований планируется особое внимание уделить уточнению значений коэффициентов α , β , γ в процессе машинного обучения (Machine Learning) на сформированных выборках и проверке полученных результатов на новых коллекциях двуязычных библиографических документов.

1. Сэлтон Г. Автоматическая обработка, хранение и поиск информации. М.: Советское радио, 1973. 560 с.
2. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. М.: ВИЛЬЯМС, 2014. 528 с.
3. Aas K., Eikvil L. Text Categorization: A Survey. Technical Report Raport NR 941. Norwegian Computing Center. Oslo. 1999. P. 1—37.
4. Scott S., Matwin S. Text Classification using WordNet hypernyms // Proceedings of the COLING/ACL WorkShop on Usage of WordNet in Natural Language Processing Systems. Montreal. 1998. P. 45—51.
5. Барсегян А. А., Куприянов М. С., Холод И. И., Тесс М. Д., Елизаров С. И. Анализ данных и процессов. СПб: БХВ-Петербург, 2009. 512 с.
6. Толчеев В. О. Модели и методы классификации текстовой информации // Информационные технологии. 2004. № 5. С. 6—14.
7. Iwazume M., Takeda H., Nishida T. Ontology-based Information Gathering and Text Categorization from the Internet // Proceedings of the 9th ACM International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. 1996. P. 305—314.
8. Peng X., Choi B. Document classifications based on word semantic hierarchies. IASTED International Conference on Artificial Intelligence and Applications 2005. February 14—16. Innsbruck, Austria. 2005. P. 362—367.
9. Некрасов И. В., Толчеев В. О. Построение модели представления библиографического документа // Информационные технологии. 2005. № 11. С. 57—63.
10. Fresno V., Ribero A. An Analytical Approaches to Concept Extraction in HTML Environments // Journal of Intelligent Information Systems. 2004. N. 22. P. 213—236.
11. Зеленков Ю. Г., Сегалович И. В. Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9-й Всероссийской научной конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции", Переславль-Залесский: Изд. ИПС РАН. 2007. С. 166—174.
12. Толчеев В. О. Анализ проблемы и разработка процедуры выявления нечетких дубликатов научных статей по библиографическим описаниям // Информационные технологии. 2011. № 2. С. 17—21.
13. Дербенев Н. В., Толчеев В. О. Разработка метода выявления нечетких дубликатов по библиографическим описаниям // Тр. Междунар. конф. "Интеллектуализация обработки информации", Черногория, Будва: Изд-во Торус. 2012. С. 613—616.
14. Губин М. В. Модели и методы представления текстового документа в системах информационного поиска. Автореферат дисс. канд. физ.-мат. наук. СПб., 2005. 15 с.
15. Мохов А. С., Толчеев В. О. Разработка методов высокоточной классификации двуязычных текстовых библиографических документов // Информационные технологии. 2014. № 5. С. 8—13.
16. Мохов А. С., Толчеев В. О., Юров Р. С. Разработка процедуры взвешивания терминов в зависимости от структуры двуязычного библиографического документа // Тр. XX Байкальской Всероссийской конф. "Информационные и математические технологии в науке и управлении". Ч. 3. Иркутск: ИСЭМ СО РАН. 2015. С. 43—50.
17. Фишберн П. Теория полезности для принятия решений: Пер. с англ. Под ред. Н. Н. Воробьева. М.: Наука, 1978. 352 с.
18. Фомичев В. А. Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть I // Информационные технологии. 2002. № 10. С. 16—25.
19. Фомичев В. А. Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть II // Информационные технологии. 2002. № 11. С. 34—45.
20. Fomichov V. A. Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. New York, Dordrecht, Heidelberg, London: Springer, 2010. 354 p.

Approaches to Considering Patterns of Scientific Documents in Processing and Analysis of Text Information

The procedures of processing and analysis of text information, based on document structure are considered in the article. The main models of text document representation are reviewed. The effectiveness of partly structured models is shown for such tasks as information retrieval, automatic annotation and fuzzy duplicates identification. Along with the known approaches the new authors' methods are presented. These methods successfully consider distinctive features of bilingual bibliographic documents and decrease classification error.

Keywords: Text mining, text document representation, informational retrieval, automatic annotation, fuzzy duplicates identification, classification of bilingual bibliographic documents

References

1. **Salton G.** *Avtomaticheskaya obrabotka, hraneniye i poisk informatsii*. Moscow: Sovetskoe radio, 1973. 560 p.
2. **Manning K., Raghavan P., Shutce H.** *Vvedenie v informatzionnyy poisk*. Moscow: Williams, 2014. 528 p.
3. **Aas K., Eikvil L.** *Text Categorization: A Survey*. Norwegian Computing Center. Oslo, 1999, pp. 1–37.
4. **Scott S., Matwin S.** Text Classification using WordNet hypernyms, *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*. Montreal, 1998, pp. 45–51.
5. **Barsegian A. A., Kupriyanov M. S., Holod I. I., Tess M. D., Elizarov S. I.** *Analiz danih i processov*. SPb: BHV-Peterburg, 2009. 512 s.
6. **Tolcheev V. O.** Modeli i metody klassifikatsii tekstovoi informatsii. *Informatsionnye tekhnologii*, 2004, no. 5, pp. 6–14.
7. **Iwazume M., Takeda H., Nishida T.** Ontology-based Information Gathering and Text Categorization from the Internet. *Proceedings of the 9th ACM International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 1996, pp. 305–314.
8. **Peng X., Choi B.** Document classifications based on word semantic hierarchies *IASTED International Conference on Artificial Intelligence and Applications*, 2005, pp. 362–367.
9. **Nekrasov I. V., Tolcheev V. O.** Postroeniye modeli predstavleniya bibliograficheskogo dokumenta, *Informatsionnye tekhnologii*, 2005, no. 11, pp. 57–63.
10. **Fresno V., Ribero A.** An Analytical Approaches to Concept Extraction in HTML Environments. *Journal of Intelligent Information Systems*, 2004, no. 22, pp. 213–236.
11. **Zelenkov Y. G., Segalovich I. V.** Sravnitelnyy analiz metodov opredeleniya nechetkikh dublikatov dlya Web-dokumentov. *Trudy 9-y Vserossiyskoy nauchnoy konferentsii "Elektronnye biblioteki: perspektivnye metody tekhnologii, elektronnye kollektcii"*, Pereslavl-Zalesskiy, Izd-vo IPS RAN, 2007, pp. 166–174.
12. **Tolcheev V. O.** Analiz problemi i razrabotka procedury viyavleniya nechetkikh dublikatov po bibliograficheskim opisaniyam, *Informatsionnye tekhnologii*, 2011, no. 2, pp. 17–21.
13. **Derbenev N. V., Tolcheev V. O.** Razrabotka metoda viyavleniya nechetkikh dublikatov po bibliograficheskim opisaniyam. *Trudy mezhdunarodnoy konferentsii "Intellektualizatsiya obrabotki informatsii"*, Chernogoriya, Budva: Izd-vo Tous, 2012, pp. 613–616.
14. **Gubin M. V.** *Modeli i metody predstavleniya tekstovogo dokumenta v sistemah informatsionnogo poiska*. Avtoreferat diss. kand. fizmat nauk. SPb., 2005 g. 15 p.
15. **Molhov A. S., Tolcheev V. O.** Razrabotka metodov visokotochnoy klassifikatsii dvuyazichnikh tekstovikh bibliograficheskikh dokumentov, *Informatsionnye tekhnologii*. 2014, no. 5, pp. 8–13.
16. **Mokhov A. S., Tolcheev V. O., Yurov R.S.** Razrabotka procedury vzyeshivaniya terminov v zavisimosti ot struktury dvuyazychnogo bibliograficheskogo dokumenta, *Trudy XX Baykalstoy vserossiyskoy konferentsii "Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii"*. Ch. 3. Irkutsk: ISEM SO RAN, 2015, pp. 43–50.
17. **Fishburn P.** *Teoriya poleznosti dlya prinyatiya resheniy*. Per. s angl. pod red. N. N. Vorobieva. Moscow: Nauka, 1978. 352 p.
18. **Fomichev V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatzionnykh tekhnologiy. Chast' I, *Informatsionnye tekhnologii*. 2002, no. 10, p. 16–25.
19. **Fomichev V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatzionnykh tekhnologiy. Chast' II, *Informatsionnye tekhnologii*. 2002, no. 11, p. 34–45.
20. **Fomichev V. A.** *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms*. New York. Dordrecht. Heidelberg. London: Springer, 2010, 354 p.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 543.4:5.44.2

А. Р. Мухутдинов¹, докт. техн. наук, профессор, e-mail: muhutdinov@rambler.ru,

З. Р. Вахидова², канд. техн. наук, доц., e-mail: MRZulphiya@rambler.ru,

М. Г. Ефимов¹, магистрант, jero07@bk.ru

¹ Казанский национальный исследовательский технологический университет

² Казанский национальный исследовательский технический университет им. А. Н. Туполева — КАИ

Компьютерное моделирование бризантного действия взрыва

Представлена разработанная и отработанная методика создания компьютерной модели бризантного действия взрыва в ANSYS AUTODYN. Показано, что модель позволяет прогнозировать бризантное действие взрыва для различных взрывчатых веществ с ошибкой до 7 %. Проведено сравнительное изучение результатов моделирования и эксперимента для зависимости бризантности тротила от плотности его заряда.

Ключевые слова: компьютерная модель, бризантность, эксплуатационные параметры, взрывчатые вещества

Введение

Известно, что бризантность является одной из важнейших характеристик взрывчатого вещества (ВВ), на основании которой проводят сравнительную оценку и выбор ВВ для разных целей (боеприпасы, подрывные средства и т. д.) [1]. Количественному определению бризантности посвящено много работ [2–8]. Следует отметить, что экспериментальное определение бризантности ВВ является опасным и дорогим мероприятием [подготовленный персонал и специально оборудованная площадка с необходимым инвентарем (свинцовый столбик, стальной диск, навеска ВВ и электродетонатор)]. Поэтому использование информационных технологий для компьютерного моделирования бризантного действия взрыва является более удобным, наглядным, дешевым и безопасным по сравнению с реальным экспериментом. На сегодняшний день разработан широкий спектр прикладных программных средств для моделирования быстропротекающих процессов [9–11]. Так, для моделирования явлений, которые характеризуются малым временем протекания, большими деформациями и перемещениями, частичным или полным разрушением материала, наиболее приемлемым является прикладное программное обеспечение (ППО) ANSYS AUTODYN.

Анализ литературы показал, что компьютерным моделированием бризантного действия взрыва занимается большое число исследователей, однако подробных методик в этих работах не представлено. В связи с этим разработка методики компьютерного моделирования бризантного действия взрыва яв-

ляется актуальной задачей, имеющей научный и практический интерес.

Методика и объект исследования

Цель данной работы — разработка и отработка методики создания компьютерной модели бризантного действия взрыва для определения бризантности ВВ (проба Гесса) с помощью ППО ANSYS AUTODYN. Объектом исследования является процесс бризантного действия взрыва тротила (вторичного взрывчатого вещества нормальной мощности (ВВВ НМ 1) [8, 9, 14]). Предлагаемая методика включает следующие этапы:

- 1-й этап — запуск ППО ANSYS AUTODYN;
- 2-й этап — создание новой модели;
- 3-й этап — выбор материалов, которые будут использоваться в построении модели;
- 4-й этап — определение плотности тротила;
- 5-й этап — создание начальных граничных условий для воздуха;
- 6-й этап — создание конечных граничных условий;
- 7-й этап — создание Лагранжевой части для свинцового цилиндра;
- 8-й этап — создание Лагранжевой части для стальной пластины;
- 9-й этап — создание Лагранжевой части для стальной плиты;
- 10-й этап — создание Лагранжевой части для взрыва;
- 11-й этап — определение условий оттока Эйлеровой части для взрыва за границы модели;
- 12-й этап — установление Лагранж—Лагранж взаимодействия;

13-й этап — установление Эйлер—Лагранж взаимодействия;

14-й этап — установление точки детонации;

15-й этап — определение контроля решений;

16-й этап — определение выходных параметров;

17-й этап — настраивание отображений модели;

18-й этап — проведение расчета бризантности;

19-й этап — создание GIF-анимации (*Graphics Interchange Format* — формат для обмена изображениями) решенной задачи.

Построение модели осуществляли в двумерной постановке, симметрия — осевая. Численное интегрирование уравнений совместно с определяющими соотношениями для материалов при соответствующих начальных и граничных условиях выполняли на Эйлеровой сетке — для воздуха и взрывчатого вещества, а на Лагранжевой сетке — для цилиндра, диска и плиты. Пространственное разрешение составляло 1 ячейка на 1 мм. Взрывчатое вещество, так же как и материалы для цилиндра, диска и плиты, выбирали из стандартной библиотеки программ.

При создании компьютерной модели бризантного действия взрыва использовали следующие уравнения:

- для тротила (в программе *TNT*) — уравнение *Джонса—Уилкинса—Ли (JWL)*, которое описывает состояние и поведение продуктов детонации;
- для воздуха (в программе *AIR*) — уравнение *идеального газа (Ideal Gas)*, описывающее зависимость между давлением, объемом и абсолютной температурой газа. Внутреннюю энергию начального условия для воздуха задавали равной $2,068 \cdot 10^5$ Дж/кг, чтобы инициализировать его с давлением в 1 атм. Следует отметить, что для тротила и воздуха нет модели прочности;
- для стали 1006 (в программе *STEEL 1006*) и свинца (в программе *LEAD*) имеются одинаковые уравнения *ударной адиабаты (Shock)*, которые описывают моделирование взаимодействия жидкостей с объектом. Необходимо отметить, что модели прочности для этих материалов разные — модель *Джонсона—Кука (Johnson—Cook)*, являющаяся частным случаем изотропного упрочнения, и уравнение *Стейнберга — Гуинана (Steinberg—Guinan)*, описывающее прочность материала по модулю и пределу упругости соответственно.

На левой, правой и верхней границах модели устанавливали граничные условия *Flow Out* (в программе *Euler*), определяющие возможность свободного течения веществ изнутри за пределы расчетного поля без каких-либо отражений от границ.

Для материалов (взрывчатое вещество, свинец, сталь и воздух) устанавливали соответствующие ГОСТ 5984—99 параметры. Геометрические размеры Лагранжевых частей: свинцовый цилиндр ($d = 40$ мм, $h = 60$ мм); стальная пластина ($d = 41$ мм, $h = 10$ мм); стальная плита — подставка ($d = 200$ мм,

$h = 20$ мм). Настройку Лагранжевых частей выполняли по геометрическим размерам Эйлеровых частей: заряд тротила ($m = 50$ г, $d = 40$ мм, $h = 40$ мм при $\rho = 1$ г/см³); воздух (в соответствии с параметрами используемых материалов). Устанавливали и другие характеристики веществ, а также их уравнения состояния (с коэффициентами и параметрами) и др. На рис. 1 (см. вторую сторону обложки) представлены компьютерная модель (рис. 1, а) и реальная экспериментальная сборка (рис. 1, б), построенные в соответствии с ГОСТ 5984—99. Голубым цветом (1) показаны стальной диск и стальная плита, синим (4) — воздух, зеленым (2) — свинцовый цилиндр, красным (3) — заряд тротила. Иницируется заряд в желтой точке.

Результаты экспериментов и их обсуждение

Результатом данной работы стала разработанная и отработанная методика, позволяющая создать компьютерную модель бризантного действия взрыва и выполнить расчет бризантности за 18 основных этапов. Необходимо также отметить, что компьютерная модель позволяет наблюдать изменение высоты свинцового цилиндра в разные моменты времени при детонации тротила, что достаточно сложно при проведении натурального эксперимента. Следует отметить, что свинцовый цилиндр компьютерной модели (рис. 2, а, см. вторую сторону обложки) приобретал грибовидную форму, как и в реальном эксперименте (рис. 2, б).

Результат замеров обжатия свинцового цилиндра показал бризантность 16 мм (см. таблицу), что согласуется с экспериментальными данными [12] и составляет 3 %. Этот результат подтверждает возможность использования ППО *ANSYS AUTODYN* для моделирования бризантного действия взрыва.

Далее проводили апробацию разработанной методики для других бризантных (вторичных) взрывчатых веществ повышенной мощности (ПМ 3, ПМ 2, ПМ 1, смесь ПН 1 и ПМ 1) [8, 14], которые имеются в стандартной библиотеке ППО *ANSYS AUTODYN*. Результаты приведены в таблице, где $\Delta H_{\text{эксп}}$ — высота свинцового цилиндра из литературных данных; $\Delta H_{\text{комп}}$ — высота свинцового цилиндра, полученная с помощью компьютерной модели.

Результаты определения бризантности для ВВ и их составов

Наименование ВВВ	$\Delta H_{\text{эксп}}$, мм	$\Delta H_{\text{комп}}$, мм	Относительная погрешность, %
НМ 1	16,5 [12, 8, 14]	16	3
ПМ 3	19,0 [12, 8, 14]	18	5,2
ПМ 2	24,0 [12, 8, 14]	22,6	5,8
Смесь НМ 1 / ПМ 1 (40/60)	22,5 [13, 8, 14]	22,1	1,7
ПМ 1	23,0 [13, 8, 14]	21,3	7,3

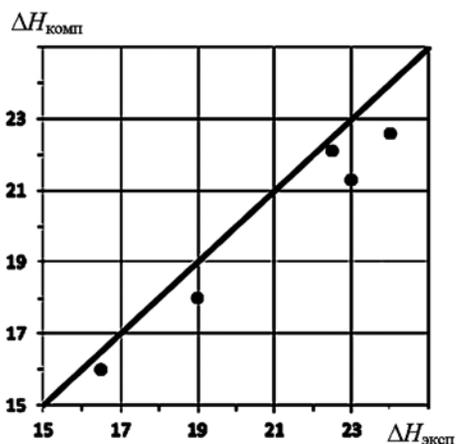


Рис. 3. Зависимость $\Delta H_{\text{комп}} = f(\Delta H_{\text{эксп}})$

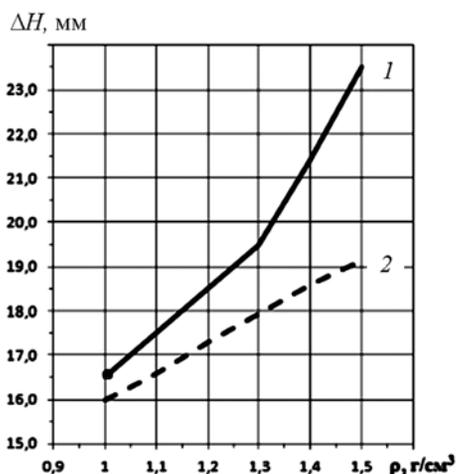


Рис. 4. Зависимость обжатия свинцовых цилиндров от плотности зарядов:

1 — кривая, полученная на основе литературных данных [1];
2 — кривая, полученная на основе компьютерной модели

Для наглядности табличные данные представлены в виде графика на рис. 3. Точками показаны данные бризантности, полученные с помощью компьютерной модели, сплошной линией показана тождественная функция $\Delta H_{\text{комп}} = f(\Delta H_{\text{эксп}})$. На основании полученной зависимости видно, что моделирование с использованием ППО занижает результат от 1,7 до 7,3 %. Это связано с неполным учетом основных свойств объекта.

Вторым этапом апробации является сравнительное изучение зависимости бризантности тротила от плотности его заряда на основе литературных данных [1] и результатов моделирования (рис. 4).

Зависимость ΔH от ρ (рис. 4), полученная на основе компьютерной модели (кривая 2), имеет линейный характер, где во всем изученном диапазоне данных (1,0... 1,5 г/см³) наблюдается рост бризантности на 16 %. При построении этой зависимости по экспериментальным данным [1] (кривая 1) картина меняется: зависимость имеет нелинейный характер, где в интервале плотности 1,3...1,5 г/см³ наблюдается значительный рост значения бри-

зантности, который составляет 17 %, в диапазоне 1,0...1,3 г/см³ — 15 %, а в общем (1,0...1,5 г/см³) на 30 %. В связи с этим характер кривой 2, полученной на основе компьютерной модели, отличается от кривой 1, построенной по экспериментальным данным, которая фиксирует резкие отклонения (на 17 %) в диапазоне плотности 1,3...1,5 г/см³.

Заключение

Использование компьютерного моделирования на основе современных информационных технологий для изучения бризантного действия взрыва является актуальным.

Разработана и отработана методика создания компьютерной модели для оценки бризантного действия взрыва по обжатию свинцового цилиндра (проба Гесса). Проведено сравнительное изучение результатов моделирования и эксперимента по бризантности. Показано, что компьютерная модель позволяет прогнозировать бризантное действие взрыва (проба Гесса) для различных ВВ с ошибкой до 7,3 %.

Сравнительное изучение результатов моделирования и эксперимента для зависимости бризантности тротила [8] от плотности его заряда показало, что компьютерная модель позволяет прогнозировать результат с ошибкой до 3 %.

Список литературы

1. Баум Ф. А., Станюкович К. П., Шехтер Б. И. Физика взрыва. М.: Наука, 1975. 752 с.
2. Андреев С. Г., Бабкин А. Б., Баум Ф. А. и др. Физика взрыва. Изд. 3-е, испр. Т. 1. М.: Наука, 2004. 832 с.
3. Миропольский Ф. П., Кузнецов В. В., Саркисян Р. С., Галущко Б. И. Авиационные средства поражения. М.: Военное издательство, 1995. 255 с.
4. Андреев К. К. Взрыв и взрывчатые вещества. М.: Военное Министерство обороны СССР, 1956. 112 с.
5. Андреев К. К., Беляев А. Ф. Теория взрывчатых веществ. М.: Оборонгиз, 1960. 597 с.
6. Дубнов Л. В., Бахаревич Н. С., Романов А. И. Промышленные взрывчатые вещества. М.: Недра, 1988. 358 с.
7. Шагов Ю. В. Взрывчатые вещества и пороха. М.: Воениздат, 1976. 120 с.
8. Пиротехника. Справочник по взрывчатым веществам, порохам и пиротехническим составам. Москва. 2012. С. 182.
9. Будников М. А., Левкович Н. Л., Быстров И. В., Сиротинский В. Ф., Шехтер Б. И. Взрывчатые вещества и пороха. М.: Государственное издательство оборонной промышленности. 1955, 109 с.
10. Мухутдинов А. Р., Вахидова З. Р., Ефимов М. Г. Моделирование процесса горения твердого топлива в топочном устройстве // Вестник Казан. технол. ун-та. 2014. Т. 17, № 20. С. 114—116.
11. Мухутдинов А. Р., Вахидова З. Р., Двоеносова М. В. Исследование особенности горения фрезерного торфа нейросетевым моделированием // Вестник Казан. технол. ун-та, 2014. Т. 17, № 22. С. 55—57.
12. Мухутдинов А. Р., Вахидова З. Р. Результаты изучения картины процесса горения твердого топлива с использованием информационных технологий // Вестник Казан. технол. ун-та, 2013. Т. 16, № 3. С. 69—72.
13. Хмельницкий Л. И. Справочник по бризантным взрывчатым веществам. Часть 1. Москва, 1962. С. 44.
14. Горст А. Г. Пороха и взрывчатые вещества. М.: Машиностроение, 1972. 208 с.
15. Взрывчатые вещества ВВ, классификация. <http://www.eragun.org/ind14.html>.

A. R. Mukhutdinov¹, Professor, e-mail: muhutdinov@rambler.ru,
Z. R. Vahidova², Associate Professor, e-mail: MRZulphiya@rambler.ru,
M. G. Efimov¹, Graduate Student, e-mail: jero07@bk.ru
¹KNRTU,
²KNRTU-KAI

Computer Modelling of Brisant Action of Explosion

In this article the developed and fulfilled technique of creation of computer model of brisant action of explosion for sinking of the lead cylinder (Ness's test) is presented to ANSYS AUTODYN. Comparative studying of results of modeling and experiment is carried out: shattering effects of secondary explosives of the increased power. It is shown that the computer model allows to predict brisant action of explosion with a mistake to 7%; dependences of shattering effect of trotyl on density of its charge. It is shown that the computer model allows to predict brisant action of explosion (Hess's test) with a mistake to 3%. It is established that character of the curve received on the basis of computer model differs from the curve constructed on experimental data which fixes sharp deviations (for 17%) in the range of density of 1,3...1,5 g/cm³.

Keywords: computer model, shattering effect, operational parameters, explosives.

References

1. Baum F. A., Stanjukovich K. P., Shehter B. I. *Fizika vzryva*, Moscow, Nauka, 1975, 752 p.
2. Andreev S. G., Babkin A. B., Baum F. A. et al. *Fizika vzryva*. Izd. 3-e, ispr. Vol. 1. Moskva, Nauka, 2004, 832 p.
3. Miropol'skij F. P., Kuznecov V. V., Sarkisjan R. S., Galushko B. I. *Aviacionnye sredstva porazheniya*, Moscow, Voennoe izdatel'stvo, 1995, 255 p.
4. Andreev K. K. *Vzryv i vzryvchatye veshhestva*, Moscow, Voennoe Ministerstvo Oborony SSSR, 1956, 112 p.
5. Andreev K. K., Beljaev A. F. *Teoriya vzryvchatykh veshhestv*, Moscow, Oborongiz, 1960, 597 p.
6. Dubnov L. V., Baharevich N. S., Romanov A. I. *Promyshlennye vzryvchatye veshhestva*. Moscow, Nedra, 1988, 358 p.
7. Shagov Ju. V. *Vzryvchatye veshhestva i poroha*. Moscow, Voeniadat, 1976, 120 p.
8. Piropravka. *Spravochnik po vzryvchatym veshhestvam, poroham i pirotehnicheskim sostavam*, Moscow, 2012, 182 p.
9. Budnikov M. A., Levkovich H. L., Bystrov I. V., Sirotinskij V. F., Shehter B. I. *Vzryvchatye veshhestva i poroha*. Moscow: Gosudarstvennoe izdatel'stvo oboronnoj promyshlennosti, 1955, 109 p.
10. Muhutdinov A. R., Vahidova Z. R., Efimov M. G. Modelirovanie processa gorenija verdnogo topliva v topochnom ustrojstve, *Vestnik Kazan, tehnol. un-ta*, 2014, vol. 17, no. 20, pp. 114–116.
11. Muhutdinov A. R., Vahidova Z. R., Dvoenosova M. V. Issledovanie osobennosti gorenija frezernogo torfa nejrosetevym modelirovanijem, *Vestnik Kazan. tehnol. un-ta*, 2014, vol. 17, no. 22, pp. 55–57.
12. Muhutdinov A. R., Vahidova Z. R. Rezul'taty izuchenija kartiny processa gorenija tverdogo topliva s ispol'zovaniem informacionnykh tehnologij, *Vestnik Kazan. tehnol. un-ta*, 2013, vol. 16, no. 3, pp. 69–72.
13. Hmel'nickij L. I. *Spravochnik po brizantnym vzryvchatym veshhestvam*, Chast' 1. Moscow, 1962, 44 p.
14. Gorst A. G. *Poroha i vzryvchatye veshhestva*, Moscow, Mashinostroenie, 1972, 208 p.
15. *Vzryvchatye veshhestva VV, klassifikacija*. URL: <http://www.eragun.org/ind14.html>.

УДК 004.36

С. А. Инютин, д-р техн. наук, проф., e-mail: inyutin_int@mail.ru,
Московский авиационный институт (национальный исследовательский университет) (МАИ)

Метод вычисления количественной характеристики модулярной величины

Разработан итерационный метод вычисления количественной характеристики отношения порядка для компьютерных модулярных форматов данных в параллельных реконфигурируемых вычислительных системах. Количественная характеристика предназначена для выполнения немодулярных операций в параллельной компьютерной арифметике. Итерационный метод позволяет получить характеристику с использованием дополнительного оборудования с минимальной разрядностью при квадратичной сложности вычислительного алгоритма.

Ключевые слова: многопроцессорные реконфигурируемые системы, вычислительный процесс, сложность вычисления, модулярные компьютерные форматы, числовые характеристики

Введение

Для программного инструментария, позволяющего выполнять вычисления над целочисленными данными, принадлежащими сверхбольшим компьютерным диапазонам, необходима разработка теоретической базы или специальной компьютерной арифметики, использующей множество-носитель с модулярными представлениями числовых данных [1, 2]. Такая арифметика над соответствующими типами данных позволяет организовать распараллеливание вычислительного процесса на множество реальных или моделируемых процессорных элементов (ядер), на которых выполняются вычисления по отдельным вычислительным трактам (модулям) в вычислительных SIMD- и SIMP-архитектурах [3]. В специализированной компьютерной арифметике в базисных или производных форматах отображаются векторы с компонентами, являющимися вычетами числовых величин по простым или взаимно простым модулям [4]. Вычеты можно считать модульными разрядами модулярного представления числовой величины.

Определим компьютерную модулярную арифметику как множество форматов модулярных данных, способ введения арифметического операционного диапазона, алгоритмы выполнения модульных операций, методы вычисления количественной характеристики отношения порядка для модулярных представлений числовых величин, а также методы и алгоритмы выполнения немодульных операций [5].

Достоинством модулярной арифметики является то, что аддитивные и мультипликативные операции в модулярной компьютерной арифметике (сложение, вычитание, умножение, целочисленное деление) выполняются независимо и параллельно по каждому из модульных разрядов. Для всех этих операций нет переносов между модульными разрядами, что позволяет выполнять параллельные вычисления в независимых вычислительных трактах на соответствующей вычислительной архитектуре.

Позиционные характеристики модулярных представлений

Для создания вычислительных средств (технических устройств или комплекса программ) SIMD-архитектуры на основе модулярных форматов данных и соответствующей арифметики необходимо иметь алгоритмы эффективного вычисления функций от компонент модулярного представления, называемых количественными характеристиками отношения порядка для модулярных представлений или позиционными характеристиками [6]. Под эффективностью вычислений в этом случае понимается достижение компромисса между алгоритмической сложностью и затратами (разрядностью) дополнительного операционного оборудования для вычисления таких функций, позволяющих определить порядок, знак в арифметическом диапазоне

на множестве модулярных представлений числовых величин. Эффективность выполнения немодульных операций в модулярной арифметике зависит от эффективности вычисления этих функций.

Рассмотрим метод уменьшения разрядности дополнительного оборудования для вычисления позиционных характеристик для модулярного вычислительного диапазона P .

Для компьютерной модулярной арифметики и соответствующих типов данных представления числовых величин $A(\text{mod } P) \leftrightarrow (\alpha_1 \text{ mod } p_1, \dots, \alpha_n \text{ mod } p_n)$ введены позиционные характеристики, названные классическим и нормированным рангом Z_A [1].

Система модулярных оснований (простых или взаимно простых чисел) может быть упорядочена $p_1 < p_2 < \dots < p_n, \forall i, j (p_i, p_j) = 1$ и храниться в КЭШ-памяти процессорных элементов [7].

Целую числовую величину можно разложить в сумму, связывающую значение числовой величины с нормированными компонентами вектора модулярного представления:

$$A = \sum_{i=1}^k \alpha_i \frac{P}{p_i} - Z_A P, \quad (1)$$

где $\alpha_i = \left\lfloor \tilde{\alpha}_i \frac{P}{p_i} \right\rfloor_{p_i}^{-1}$ — нормированная компонента

модулярного представления; $\tilde{\alpha}_i = |A|_{p_i}$ — компонента классического модулярного представления, p_i — основания (простые или взаимно простые

числа) модулярной системы; $P = \prod_{i=1}^k p_i$ — макси-

мум модулярного диапазона; $Z_A = \left\lfloor \sum_{i=1}^k \frac{\alpha_i}{p_i} \right\rfloor$ — пози-

ционная характеристика — нормированный ранг.

Для точного вычисления позиционной характеристики необходимо суммирование рациональных дробей со знаменателем P , т.е. требуются вычисления в компьютерном диапазоне с максимальным значением P , являющимся сверхбольшим числом при большом числе оснований, близких к верхней границе компьютерного целочисленного диапазона. Значение нормированного ранга — целочисленной функции от нормированных компонент модулярного представления — принадлежит отрезку $Z_A \in [0, k - 1]$, причем, как правило, $k \ll P$, что приводит к явному неравенству областей значений и определения нормированного ранга. Большинство известных методов вычисления позиционной характеристики модулярной величины имеют квадратичную сложность $O(k^2)$. Повысить эффективность вычисления позиционной характеристики модулярной величины — нормированного ранга Z_A возможно разработкой методов и функций с областями определения, близкими к областям значений [8].

Наименьшие неотрицательные вычеты рациональных чисел по модулю

Выведем математический аппарат для обоснования алгоритма вычисления позиционной характеристики по дополнительному модулю.

В теории чисел известны наименьшие неотрицательные вычеты a от целых чисел A по целому простому модулю p :

$$A \equiv a(\text{mod } p), \quad a \in \{0, 1, \dots, p-1\} \subset \mathbf{N}.$$

В дальнейшем, чтобы подчеркнуть бинарную операцию вычисления вычета относительно переменных A, p будем использовать обозначение операции:

$$a = |A|_p = A - \left[\frac{A}{p} \right] p \in \{0, 1, \dots, p-1\} \subset \mathbf{N},$$

где квадратными скобками обозначена разрывная функция — целая часть, не большая частного от деления.

Пусть $p > \max\{a, b\}$, определим вычет от рационального числа по простому целому модулю p следующим образом:

$$\left| \frac{a}{b} \right|_p = \left| a|b|_p^{-1} \right|_p = n \in \{0, 1, \dots, p-1\} \subset \mathbf{N}.$$

Это позволяет факторизовать множество рациональных чисел и определить p классов вычетов $\{i\}$ по модулю, причем одному классу $\{i\}$ принадлежат рациональные дроби и целые числа, для которых выполняется соотношение:

$$\left| \frac{a}{b} \right|_p = \left| \frac{a'}{b'} \right|_p = \left| a|b|_p^{-1} \right|_p = \left| a'|b'|_p^{-1} \right|_p = n \in \{0, 1, \dots, p-1\} \subset \mathbf{N}.$$

При условии $p > \max\{a, b\}$ однозначно восстанавливается числитель рациональной дроби с известным знаменателем, что обеспечивает биективность отображения.

Упорядочим полную систему наименьших неотрицательных вычетов по модулю:

$$\left\{ n = \frac{n^1}{1}, \frac{n^2}{2}, \dots, \frac{n^i}{i}, \dots, \frac{n^{p-1}}{p-1} \right\} = \left\{ \min \frac{n^i}{i}, \dots, \frac{1}{j}, \dots, n \right\},$$

где $n \equiv n^2 |2|_p^{-1} \equiv \dots n^i |i|_p^{-1} \equiv \dots n^{p-1} |p-1|_p^{-1} (\text{mod } p)$.

Полученной полной системе вычетов можно придать вероятностную интерпретацию:

$$\left\{ 1 = \frac{n^1}{1}, \frac{n^2}{2}, \dots, \frac{n^i}{in}, \dots, \frac{n^{p-1}}{(p-1)n} \right\} = \left\{ \min \frac{n^i}{in}, \dots, 1 \right\},$$

где $n^i \equiv in(\text{mod } p)$.

Аналогично строится полная система абсолютных наименьших вычетов рациональных чисел по модулю.

Определим свойства наименьших неотрицательных вычетов рациональных чисел по простому модулю p применительно к арифметическим опе-

рациям на множестве вычетов: умножение, сложение, вычитание, деление.

Свойство 1 (операция умножения). Условие изоморфизма отображения или восстановления рациональной дроби $p > \max\{ac, bd\}$:

$$\left| \frac{a}{b} \right|_p \left| \frac{c}{d} \right|_p = \left| ac|b|_p^{-1}|d|_p^{-1} \right|_p = \left| \frac{ac}{bd} \right|_p = |n_1 n_2|_p = n \in \{0, 1, \dots, p-1\} \subset \mathbf{N},$$

где n_1, n_2 — некоторые целые элементы полной системы наименьших неотрицательных вычетов по модулю.

Свойство 2 (операция сложения). Условие изоморфизма отображения $p > \max\{ad + cb, bd\}$:

$$\left| \frac{a}{b} \right|_p + \left| \frac{c}{d} \right|_p = \left| (ad + cb)|d|_p^{-1}|b|_p^{-1} \right|_p = \left| (a|b|_p^{-1} + c|d|_p^{-1}) \right|_p = \left| \frac{ad + cb}{bd} \right|_p = |n_1 + n_2|_p = n \in \{0, 1, \dots, p-1\} \subset \mathbf{N}.$$

Свойство 3 (операция вычитания). Условие изоморфизма отображения $p > \max\{ad - cb, bd\}$:

$$\left| \frac{a}{b} \right|_p - \left| \frac{c}{d} \right|_p = \left| (ad - cb)|d|_p^{-1}|b|_p^{-1} \right|_p = \left| (a|b|_p^{-1} - c|d|_p^{-1}) \right|_p = \left| \frac{ad - cb}{bd} \right|_p = |n_1 - n_2|_p = n \in \{0, 1, \dots, p-1\} \subset \mathbf{N}.$$

Свойство 4 (операция деления). Условие изоморфизма отображения $p > \max\{ac, bd\}$:

$$\left| \frac{a}{b} \right|_p \left| \frac{d}{c} \right|_p = \left| \frac{a}{b} \right|_p \left| \frac{c}{d} \right|_p = \left| ac|b|_p^{-1}|d|_p^{-1} \right|_p = \left| \frac{ac}{bd} \right|_p = |n_1|_p |n_2|_p^{-1} = n \in \{0, 1, \dots, p-1\} \subset \mathbf{N}.$$

Метод вычисления позиционной характеристики

Рассмотрим метод вычисления позиционной характеристики модулярной величины — нормированного ранга

$$Z_A = \left[\sum_{i=1}^k \frac{\alpha_i}{p_i} \right]. \quad (2)$$

Для разработки и обоснования алгоритма вычисления позиционной характеристики на основе вычетов рациональных чисел и оценки его сложности сформулируем две теоремы.

Теорема 1.

$$\left| \sum_{i=1}^k \alpha_i |p_i|_g^{-1} \right|_g \equiv \frac{1}{p} \sum_{i=1}^k \alpha_i P_i(\text{mod } g). \quad (3)$$

Доказательство. Последовательное применение k -раз свойства 2 вычета от рациональных чисел по целому простому модулю позволяет получить ве-

личину $\frac{1}{p} \sum_{i=1}^k \alpha_i P_i(\text{mod } g)$.

Следствие 1. Вычет первого слагаемого формулы (1) по модулю g вычисляется умножением (3) на P по модулю:

$$\left| \sum_{i=1}^k \left| \alpha_i |p_i|_g^{-1} \right|_g P \right|_g \equiv \sum_{i=1}^k \alpha_i P_i \pmod{g}. \quad (4)$$

Следствие 2. Сложность вычисления выражения (4) является линейной $O(k)$.

Для вычисления второго слагаемого формулы (1) используем полиадическое представление числовой величины $A = \alpha_1 + p_1(\beta_2 + p_2(\beta_3 + (\dots + p_{n-1}(\beta_n + p_n(0))\dots)))$ с тем же набором оснований, что и у модулярного представления.

Теорема 2. Компоненты модулярного и полиадического представлений, вычисляемых по простому модулю g , связывают соотношения:

$$|A^1|_g = \left| \frac{A - \alpha_1}{p_1} \right|_g = \left| (A - \alpha_1) |p_1|_g^{-1} \right|_g = \left| \beta_2 + p_2 \left[\frac{A^1}{p_2} \right]_g \right|_g;$$

$$|A^2|_g = \left| \frac{A^1 - \alpha_2}{p_2} \right|_g = \left| (A^1 - \alpha_2) |p_2|_g^{-1} \right|_g = \left| \beta_3 + p_3 \left[\frac{A^2}{p_3} \right]_g \right|_g;$$

$$|A^{k-1}|_g = \left| \frac{A^{k-2} - \alpha_{k-1}}{p_{k-1}} \right|_g = \dots = \left| (A^{k-2} - \alpha_{k-1}) |p_{k-1}|_g^{-1} \right|_g = \left| \beta_k + p_k \left[\frac{A^{k-1}}{p_k} \right]_g \right|_g = |\beta_k|_g.$$

Доказательство. Рассмотрим итерационные соотношения, позволяющие ввести биективное отображение модулярного в полиадическое представление для одной числовой величины:

$$A = \alpha_1 + p_1(\beta_2 + p_2(\beta_3 + (\dots + p_{n-1}(\beta_n + p_n(0))\dots))) \Leftrightarrow (\alpha_1, \alpha_2, \dots, \alpha_k).$$

Учтем следующие соотношения:

$$A = \alpha_1 + p_1(\beta_2 + p_2(\beta_3 + (\dots + p_{n-1}(\beta_n + p_n(0))\dots))) = \alpha_1 + p_1 \left[\frac{A}{p_1} \right] = \alpha_1 + p_1((A - \alpha_1) |p_1|_g^{-1}),$$

$$A = \alpha_1 + p_1 A^1 = \alpha_1 + p_1 \left[\frac{A}{p_1} \right],$$

$$A^1 = \frac{A - \alpha_1}{p_1} = \beta_2 + p_2 \left[\frac{A^1}{p_2} \right],$$

$$A^2 = \frac{A^1 - \alpha_2}{p_2} = \beta_3 + p_3 \left[\frac{A^2}{p_3} \right].$$

Для описания итерационного процесса введен верхний индекс, соответствующий номеру этапа

$$(\alpha_1, \dots, \alpha_k) = (\alpha_1^1, \alpha_2^1, \dots, \alpha_k^1).$$

На первом этапе выполнены преобразования в компонентах вектора модулярного представления:

$$(0, \alpha_2^2, \dots, \alpha_k^2) = (0, \left| (\alpha_2^1 - \alpha_1^1) |p_1|_{p_2}^{-1} \right|_{p_2}, \dots, \left| (\alpha_k^1 - \alpha_1^1) |p_1|_{p_k}^{-1} \right|_{p_k}).$$

В частности, во второй компоненте получено $\left| (A - \alpha_1) |p_1|_{p_2}^{-1} \right|_{p_2} = \beta_2$. Аналогично вычисляются остальные компоненты вектора $\left| (A - \alpha_1) |p_1|_{p_i}^{-1} \right|_{p_i}$. В результате получен вектор со следующими компонентами:

$$(0, \alpha_2^2, \dots, \alpha_k^2) = (0, \beta_2, \alpha_3^2, \dots, \alpha_k^2).$$

На следующих этапах итерационного процесса аналогично вычисляются компоненты $\forall j = i, \dots, k$ $\left| (\alpha_j^i - \alpha_i^i) |p_i|_{p_j}^{-1} \right|_{p_j}$ векторов $(0, \dots, 0, \alpha_i^i, \alpha_{i+1}^i, \dots, \alpha_k^i)$, что позволяет получить на основе компонент модулярного представления последовательно компоненты $\{\beta_2, \beta_3, \dots, \beta_n\}$ полиадического представления числовой величины.

Следствие 1. Обработанные компоненты последовательно получаемых векторов, например $x_1 = \left| (A - \alpha_1) |p_1|_{p_1}^{-1} \right|_{p_1} \equiv \left[\frac{A}{p_1} \right]_{p_1}$, могут быть приравнены нулю, так как они не используются для этапов в алгоритме $A-1$ для вычисления позиционной характеристики.

Следствие 2. Сложность вычисления последовательности компонент полиадического представления $\{\beta_2, \beta_3, \dots, \beta_n\}$ равна $O(k^2)$.

Вышеприведенные теоремы позволяют сформировать алгоритм вычисления позиционной характеристики.

Алгоритм $A-1$.

1. Вычисление вычета $\left| \sum_{i=1}^k \left| \alpha_i |p_i|_g^{-1} \right|_g \right|_g = N$.
2. Модификация вычета $\left| \sum_{i=1}^k \left| \alpha_i |p_i|_g^{-1} \right|_g |p_1|_g \dots |p_k|_g \right|_g = \left| \sum_{i=1}^k \left| \alpha_i |p_i|_g^{-1} \right|_g |P|_g \right|_g = \tilde{N}$.
3. Формирование вектора $(\tilde{N}, \alpha_1, \dots, \alpha_k)$ и приравнивание его вектору с введенным верхним индексом $(\tilde{N}, \alpha_1, \dots, \alpha_k) = (\tilde{N}^1, \alpha_1^1, \alpha_2^2, \dots, \alpha_k^1)$.
4. На первой итерации формируется вектор $\left(\left| (\tilde{N}^1 - \alpha_1^1) |p_1|_g^{-1} \right|_g, 0, \left| (\alpha_2^1 - \alpha_1^1) |p_1|_{p_2}^{-1} \right|_{p_2}, \dots, \left| (\alpha_k^1 - \alpha_1^1) |p_1|_{p_k}^{-1} \right|_{p_k} \right) = (\tilde{N}^2, 0, \alpha_2^2, \alpha_3^2, \dots, \alpha_k^2)$.

5. Аналогичные вычисления выполняются для всех остальных компонент (до k -й, включительно) вектора. В частности, для i -й итерации формируется вектор $\left(\left(\tilde{N}^{i-1} - \alpha_{i-1} \right) \left| p_{i-1} \right|_g^{-1}, 0, \dots, \dots, 0, \left(\alpha_i - \alpha_{i-1} \right) \left| p_{i-1} \right|_{p_2}^{-1}, \dots, \left(\alpha_k - \alpha_{i-1} \right) \left| p_{i-1} \right|_{p_k}^{-1} \right) = (\tilde{N}^i, 0, \dots, 0, \alpha_i, \alpha_{i+1}, \dots, \alpha_k)$.

6. Критерием останова алгоритма является выполнение вышеуказанных преобразований во всех компонентах (до k -й компоненты включительно). В результате получен вектор со всеми нулевыми компонентами, кроме первой, в которой сформировано числовое значение позиционной характеристики — нормированный ранг $Z_A = \left[\sum_{i=1}^k \frac{\alpha_i}{p_i} \right]$.

Теорема 1 служит для обоснования результатов этапов 1, 2, а теорема 2 предназначена для обоснования результаты этапов алгоритма 4—6. В следствиях из теорем показано, что суммарная сложность описанного алгоритма является квадратичной $O(k^2)$.

Закключение

Вычисление позиционной характеристики — нормированный ранг — необходимо для оценки значения модулярной величины и выполнения немодульных операций над данными в модулярных форматах.

Предлагаемый метод вычисления позиционной характеристики, в отличие от описанного в работе [8], не требует при представлении числовых вели-

чин введения избыточности в модулярный формат для устранения так называемой "критичности".

Областью значений позиционной характеристики — нормированный ранг является множество $[0, k)$, что позволяет выбрать простое значение дополнительного модуля $g \geq k$, это дает экономию в разрядности аппаратуры вычисления позиционной характеристики, так как, как правило, $k \ll p_i$.

Вышеприведенный метод, имеющий квадратичную сложность, позволяет организовать эффективное вычисление позиционной характеристики — нормированный ранг на аппаратуре с разрядностью $k \leq g \ll p_i$, дополняющей средства арифметических преобразований числовых величин в компьютерных модулярных форматах.

Список литературы

1. **Амербаев В. М.** Теоретические основы машинной арифметики. Алма-Ата: Наука, 1976. 320 с.
2. **Инютин С. А.** Основы модулярной алгоритмики. Ханты-Мансийск: Полиграфист, 2009. 237 с.
3. **Инютин С. А.** Анализ сложности многоразрядных вычислительных процессов // Научные труды МАТИ. 2014. Вып. 22 (94). С. 154—159.
4. **Inyutin S. A.** Parallel Square Modular Computer Algebra // Transaction of Parallel Processing and Applied Mathematics PPAM — 2003. Poland—Denmark: Springer, 2003. P. 117—123.
5. **Ноден П., Китте К.** Алгебраическая алгоритмика. М.: Мир, 1999. 720 с.
6. **Амербаев В. М., Тельпухов Д. В.** Обратный преобразователь модулярной арифметики с использованием неточного ранга // Известия высших учебных заведений. Электроника. 2013. № 1. С. 41—46.
7. **Шилов В. В., Столярский Е. З.** Организация и работа кэш-памяти // Информационные технологии. 2000. № 7. С. 2—8.
8. **Инютин С. А.** Особенности вычисления характеристик модулярной величины // Информационные технологии. 2014. № 5 (213). С. 22—27.

S. A. Inyutin, Professor, e-mail: inyutin_int@mail.ru,
Moscow Aviation Institute (Nation Research University) (MAI)

Method Calculation Quantitative Characteristic Computer Modular Value

Developed iterative method for computing quantitative characteristics from wearing for computer modular data formats in a parallel re-configurable computing systems. Quantitative characterization earmarked for implementation non-modular operations in a parallel computer arithmetic. The iterative method allows to obtain a characteristic with the use of additional equipment with a minimum width while the quadratic computational complexity of the calculation algorithm.

Keywords: reconfigurable multiprocessor systems, modular computing process, complexity of computation, figure characteristic for modular computer formats, numerical characteristics, cellular algorithm of the routing

References

1. **Amerbaev V. M.** *Teoreticheskie osnovy mashinnoy arifmetiki* (Theoretic base computer arithmetic). Alma-Ata, Nauka, 1976. 320 p.
2. **Inyutin S. A.** *Osnovy modul'arnoy algoritmiki* (Base at modular algorithmic). Hantyi-Mansiysk, Poligrafist, 2009. 237 p.
3. **Inyutin S. A.** Analiz slojnosti mnogorazryadnyh vyichislitel'nyh protsessov (Analyst many digital calculation process), *Nauchnyye trudy MATI*. 2014, vol. 22 (94), pp. 154—159.
4. **Inyutin S. A.** Parallel Square Modular Computer Algebra, *Transaction of Parallel Processing and Applied Mathematics PPAM — 2003*. Poland-Denmark: Springer, 2003, pp. 117—123.

5. **Noden P., Kitte K.** *Algebraicheskaya algoritmika* (Algebra algorithmic). Moscow, Mir, 1999. 720 p.
6. **Amerbaev V. M., Telpuhov D. V.** Obratnyiy preobrazovatel modul'arnoy arifmetiki s ispolzovaniem netochnogo ranga (Reverse drive modular arithmetic using precision pang), *Izvestiya vysshih uchebnyih zavedeniy. Electronic*, 2013, no. 1, pp. 41—46.
7. **Shilov V. V., Stolyarskiy E. Z.** Organizatsiya i rabota kesh-pamyati (Planning work Kech- memory), *Informatsionnyye tehnologii*, 2000, no. 7, pp. 2—8.
8. **Inyutin S. A.** Osobennosti vyichisleniya harakteristik modul'arnoy velichiny (Peculiarity calculation characteristics for computer modular value), *Informatsionnyye tehnologii*, 2014, no. 5 (213), pp. 22—27.

УДК 004.75

В. А. Богатырев, д-р техн. наук, проф., e-mail: vladimir.bogatyrev@gmail.com, **А. В. Богатырев**, аспирант, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО)

Модель резервированного обслуживания запросов реального времени в компьютерном кластере

Для вычислительных систем кластерной архитектуры, работающих в реальном времени, представляемых группой одноканальных систем массового обслуживания с локальными очередями, предложена модель оценки вероятности своевременного и безошибочного обслуживания запросов с выполнением копий этих запросов в нескольких узлах. Рассмотрены варианты диспетчеризации с уничтожением и без уничтожения резервных копий запросов, ожидающих в локальных очередях узлов сверх допустимого времени.

Определена зависимость вероятности своевременного и безошибочного обслуживания запросов от кратности резервирования копий запросов, направляемых на обслуживание в разные узлы.

Ключевые слова: модель, надежность, резервированное обслуживание запросов, реальное время, кластер, диспетчеризация

Введение

К компьютерным системам, решающим ответственные задачи, предъявляют высокие требования по отказоустойчивости, надежности вычислительного процесса, достигаемые в результате резервирования и консолидации ресурсов системы при объединении компьютерных узлов в кластеры [1–6].

Для компьютерных систем реального времени при обеспечении функциональной надежности критичны безошибочность и своевременность обслуживания запросов [1–5]. Под функциональной надежностью понимается надежность (устойчивость) вычислительного процесса по своевременному безошибочному выполнению поступающих в систему запросов в условиях отказов, сбоев, ошибок и деструктивных воздействий [6, 7].

Для структурно избыточных компьютерных систем, включая кластеры, устойчивость (надежность) вычислительного процесса повышается при динамическом распределении запросов [8–15], в том числе при направлении резервных копий запросов, поступающих в систему на обслуживание в несколько узлов (каждая копия выполняется отдельным узлом).

Модели многоканальных систем обслуживания с общей очередью при резервированном выполнении запросов предложены и исследованы в работах [16, 17]. Для организации резервированного обслуживания, названного в работах [16, 17] "широковещательное обслуживание с копированием запроса",

запрос направляется в свободные в момент его поступления узлы (приборов), причем в каждом узле обслуживание резервной копии запроса выполняется независимо. В работе [17] рассматривается адаптивное широковещательное (резервированное) обслуживание запросов, при котором, если в момент поступления запроса число занятых приборов меньше некоторого порогового значения, то резервные копии запроса направляются для выполнения во все свободные узлы, иначе запрос обслуживается только в одном из узлов. Резервированный запрос считается успешно выполненным при его обслуживании хотя бы в одном из узлов [16, 17]. Эффективность обслуживания запросов в работах [16, 17] оценивается средним временем ожидания, вместе с тем для систем реального времени более важна своевременность вычислений, определяемая по вероятности выполнения запросов за время, меньшее предельно допустимого значения или по вероятности ожидания запросов меньше предельно допустимого времени t [18].

В кластерных системах, консолидирующих ресурсы нескольких серверов, особенность выполнения запросов (в том числе с их резервированным обслуживанием в разных узлах) заключается в организации очередей в каждом сервере, который, таким образом, соответствует одноканальной системе массового обслуживания [18] с локальной бесконечной очередью. Для резервированного выполне-

ния запроса его резервные копии заносятся в несколько очередей разных серверов.

Резервированное выполнение запросов несколькими узлами, представляемыми одноканальными системами массового обслуживания с локальными очередями, приводит к повышению функциональной надежности вычислительного процесса при отказах и ошибках узлов. Вместе с тем резервирование запросов вызывает возрастание загрузки узлов, а это может привести к увеличению среднего времени ожидания и вероятности недопустимой задержки запросов в очередях. Задача оптимизации в кластерных системах с резервированным выполнением запросов в реальном времени поставлена и решена в работе [19].

В то же время резервированное выполнение запросов несколькими узлами с учетом стохастичности обслуживания потенциально может привести к увеличению вероятности своевременного выполнения запроса хотя бы от одного из узлов. Следует также отметить, что резервированное обслуживание запросов при увеличении загрузки узлов приводит к снижению максимально возможной интенсивности обслуживаемого потока запросов, не вызывающей нарушения стационарности процесса обслуживания в узлах.

1. Направления исследований

Для кластерных систем реального времени при направлении резервных копий запроса в очереди нескольких узлов (серверов) рассмотрим вычислительный процесс, при котором считывание результатов вычислений проводится в момент времени t , отсчитываемый после занесения запроса в очередь узлов с учетом времени обслуживания. Такая организация вычислительного процесса обуславливается, например, особенностью систем реального времени (в частности, управляющих систем), в которых результаты вычислений требуются к определенным моментам времени, и не стоит задача минимизации времени ожидания запросов, главное — их получение к заданному моменту времени.

Особенность рассматриваемой организации вычислительного процесса заключается в необходимости начала обслуживания запроса за время, не превышающее предельно допустимый порог ожидания t , после этого времени (с учетом задержки вычислений) начинается считывание результатов вычислений, которые должны быть получены хотя бы в одном из k узлов, задействованных в резервированных вычислениях. После превышения порога допустимого времени ожидания t резервные копии запроса, еще находящиеся в очередях, теряют свою актуальность и могут быть уничтожены. Уничтожение резервных копий запросов с просроченным временем ожидания позволит исключить непродуктивное выполнение резервных копий запросов, актуальность которых при ожидании в очереди

сверх времени t теряется, позволяет снизить загрузку узлов и, соответственно, задержки в их очередях.

Выбор варианта диспетчеризации с уничтожением и без уничтожения резервных копий запросов, ожидающих обслуживания сверх допустимого времени, должен сопровождаться соответствующими расчетами средних задержек и вероятностей своевременного обслуживания резервированных запросов.

Модели массового обслуживания кластеров реального времени с резервированием вычислительного процесса без уничтожения резервных копий запросов, ожидающих сверх допустимого времени, предложены в работе [18]. При построении модели обслуживания с k -кратным резервированием выполнения запросов в работе [18] предполагается увеличение интенсивности потока обслуженных запросов в k раз относительно исходного входного потока запросов. Применение моделей резервированного обслуживания в системах с уничтожением резервных копий запросов, ожидающих в очереди сверх допустимого времени, как показано в работе [18], приводит к нижней оценке вероятности своевременного получения результатов. Однако оценка погрешности такого нижнего приближения в этой работе не приведена.

Цель исследований — повышение вероятности своевременного и безошибочного обслуживания в результате направления резервных копий запросов на обслуживание в несколько узлов кластера.

Для достижения поставленной цели ставится задача разработать модели, отражающие влияние уничтожения просроченных резервных копий запросов (для которых ожидание превышает допустимое время) на вероятность своевременного и безошибочного обслуживания запросов в кластере.

2. Объект и задачи исследования

В качестве объекта исследований рассматривается вычислительный кластер, объединяющий n идентичных компьютерных узлов (серверов), в каждом из которых организуется собственная очередь запросов, таким образом, каждый узел кластера соответствует одноканальной системе массового обслуживания [20] с локальной очередью.

Будем считать известными среднее время выполнения узлом запроса ν , а также интенсивности входного потока Λ , потока отказов λ_0 и ошибок вычислений λ_1 узлов.

Поступающий в кластер запрос может быть распределен на обслуживание в любой компьютерный узел, для повышения надежности вычислительного процесса резервные копии запроса могут быть распределены на обслуживание в k узлов.

Для успешного обслуживания запроса необходимо, чтобы к моменту времени t было начато обслуживание его резервной копии хотя бы в одном из k узлов, задействованных в резервированных вычислениях. Резервные копии запроса, находя-

щиеся в очередях узлов к моменту времени t , теряют свою актуальность (не отвечают условиям своевременности) и могут уничтожаться. Для организации уничтожения просроченных копий резервных запросов в момент их поступления в очередь узла запускается таймер, отсчитывающий предельное время t нахождения запроса в очереди. Если к моменту срабатывания таймера запрос не начал обслуживаться (находится в очереди), то он уничтожается. В результате уничтожения просроченных резервных копий запросов удается снизить загруженность узлов, но при этом несколько усложняется диспетчеризация запросов.

В работе ставится задача построения моделей и сравнения эффективности кластеров при организации в узлах очередей с уничтожением и без уничтожения резервных копий запросов, ожидающих в очередях дольше предельно допустимого времени t .

Разрабатываемые модели направлены на оценку вероятности превышения допустимого времени ожидания хотя бы в одном из узлов кластера, выделенных для обслуживания резервных копий запросов.

Для решения поставленной задачи сначала исследуем влияние предлагаемой организации резервированного обслуживания запросов на своевременность результатов без учета ошибок вычислений, а затем проанализируем дополнительное влияние возможности возникновения ошибок на своевременность резервированного обслуживания запросов.

3. Модели резервированного обслуживания запросов

При построении модели обслуживания будем пренебрегать потерями на диспетчеризацию, в том числе на уничтожение резервных копий запросов, ожидание которых в очередях превышает лимит времени t . Такое приближение допустимо при несущественном влиянии на замедление вычислительного процесса в сервере процесса уничтожения просроченных запросов в очереди, проводимых при диспетчеризации по таймеру без реализации межмашинного обмена между серверами после завершения выполнения копии запроса одним из них.

Резервированное обслуживание запроса считается успешно выполненным, если результаты требуемых вычислений получены к моменту времени t хотя бы в одном из k узлов, принимающих резервные копии запроса к обслуживанию.

Как показано в работе [18], резервированное обслуживание запросов без уничтожения просроченных в очереди резервных копий запросов сверх допустимого времени ожидания t приводит к увеличению интенсивности запросов в k раз.

В предположении независимости вычислительных процессов в разных узлах, представляемых системами массового обслуживания (СМО) типа М/М/1 [20] с бесконечными локальными очередями,

вероятность того, что хотя бы в одном из k узлов, принимающих в очередь резервированный запрос, его ожидание меньше предельно допустимой задержки t , в соответствии с [18] вычислим как

$$R(t) = 1 - (1 - r)^k = 1 - \left(\frac{\Lambda v k}{n} \exp\left(-t\left(v^{-1} - \frac{\Lambda v k}{n}\right)\right) \right)^k, \quad (1)$$

где r — вероятность не превышения времени ожидания в некотором узле установленного предела t , $r = 1 - (v\Lambda k/n) \exp(-t(v^{-1} - \Lambda k/n))$, а среднее время ожидания запросов вычисляется как

$$w = \int_0^{\infty} (1 - R(t)) dt = \int_0^{\infty} \left(\frac{\Lambda v k}{n} \exp\left(-t\left(v^{-1} - \frac{\Lambda v k}{n}\right)\right) \right)^k dt. \quad (2)$$

Формула (2) дает верхнюю (пессимистическую) оценку среднего времени ожидания, так как не учитывает возможность уменьшения загрузки в результате удаления из очередей резервных копий запросов, выполнение которых при ожидании сверх времени t становится неактуальным.

Для кластерной системы без резервирования запросов, узлы которой представляются СМО типа М/М/1 с бесконечной очередью, вероятность того, что время ожидания запросов в узле меньше предельно допустимого времени t , вычисляется как

$$r = 1 - \frac{\Lambda}{n} v \exp\left(-t_0\left(v^{-1} - \frac{\Lambda}{n}\right)\right), \quad (3)$$

а среднее время ожидания запросов [20] как $w = (\Lambda v^2/n) / [1 - (\Lambda v/n)]$.

Рассмотрим вариант управления очередями, при котором реализуется уничтожение резервных копий запросов, находящихся в очередях сверх времени t . В результате такого уничтожения просроченных резервных копий интенсивность обслуживаемых запросов (загрузка узлов) вследствие k -кратного резервирования увеличивается не в k раз, как для варианта без уничтожения просроченных копий запросов, а в $I \leq k$ раз относительно обслуживания без резервирования ($I \geq 1$).

Коэффициент I определяется как математическое ожидание числа узлов, принимающих запрос к резервированному обслуживанию, для которых время ожидания запросов меньше допустимого значения t (в этих узлах копии ожидающих запросов не уничтожаются).

Значение предельно допустимого времени ожидания запроса в очереди t обусловлено требованиями реального времени и задается как константа в зависимости от особенностей прикладного процесса. Очевидно, что величина I зависит от интенсивности запросов и от ограничения допустимого времени ожидания t . Действительно, если время t велико (нет жестких требований, вызванных реальным временем обслуживания запросов), то вероятность

своевременного выполнения запросов в узлах повышается, в результате повышается и значение I . С уменьшением предельно допустимого времени ожидания t (ужесточение требований реального времени) вероятность своевременного вычисления в каждом узле и, соответственно, математическое ожидание числа узлов, своевременно выполнивших запрос за время t , уменьшается.

Таким образом, с уменьшением предельно допустимого времени ожидания t потенциально должна расти эффективность от уничтожения просроченных запросов в очередях.

Составим уравнение для вычисления коэффициента I увеличения интенсивности потока обслуженных запросов с учетом уничтожения просроченных копий запросов в очередях как

$$I = kr = k \left(1 - \frac{\Lambda v I}{n} \exp \left(-t \left(v^{-1} - \frac{\Lambda I}{n} \right) \right) \right),$$

где kr — математическое ожидание числа узлов, принимающих копии запроса к резервированному обслуживанию, при котором время ожидания в очереди меньше предельно допустимого значения t , а r — вероятность того, что ожидание в очереди узла меньше предельно допустимого значения t .

$$r = 1 - \frac{\Lambda v I}{n} \exp \left(-t \left(v^{-1} - \frac{\Lambda I}{n} \right) \right), \quad (4)$$

при этом загрузка узла с учетом уничтожения в очередях просроченных запросов (в предположении сбалансированности загрузки всех n узлов) $\rho = \Lambda v I / n$.

Уравнение легко решается, например, в системе компьютерной математики MathCAD-15 с использованием встроенной функции *root*:

$$s := \text{root} \left(-I + \left[1 - \frac{\Lambda v I}{n} \exp \left(-t \left(v^{-1} - \frac{\Lambda I}{n} \right) \right) \right], I \right),$$

где s принимает значение искомой величины I .

После определения коэффициента увеличения загрузки I при k -кратном резервировании запросов в очередях узлов кластера вероятность непревышения допустимой задержки ожидания t хотя бы одним из k узлов, принимающих запрос в очереди, находим как

$$R(t) = 1 - \left(\left(\frac{\Lambda v I}{n} \right) \exp \left(-t \left(v^{-1} - \frac{\Lambda v I}{n} \right) \right) \right)^k. \quad (5)$$

Среднее время ожидания резервированных запросов определим по следующей формуле:

$$w = \int_0^{\infty} (1 - R(t)) dt = \int_0^{\infty} \left(\frac{\Lambda v I}{n} \exp \left(-t \left(v^{-1} - \frac{\Lambda v I}{n} \right) \right) \right)^k dt. \quad (6)$$

В кластере, содержащем n одинаковых компьютерных узлов (серверов), вероятность своевременности, безошибочности и надежности выполнения

запроса в некотором узле кластера определим как $b = rp$, где r — вероятность того, что время ожидания запросов в очереди некоторого узла меньше предельно допустимого значения t ; $p = \exp(-\lambda(t + v))$ — вероятность того, что за период $t + v$ до считывания результатов, включающий время вычислений, ожидания запроса в очереди и его результата в выходном буфере, отказы и ошибки вычислений в рассматриваемом узле не возникают, при этом $\lambda = \lambda_0 + \lambda_1$ — суммарная интенсивность сбоев, отказов и ошибок узла.

При диспетчеризации очередей без уничтожения резервных копий запросов, ожидающих сверх допустимого времени t , вероятность r вычисляется по формуле (3), а с уничтожением копий — по формуле (4).

Вероятность своевременного получения безошибочных результатов хотя бы от одного из k узлов кластера, задействованных в резервированном выполнении запроса, определим как $P = 1 - (1 - b)^k$, где $b = rp$.

4. Расчет вероятности своевременного обслуживания резервированных запросов

Приведем расчет вероятности своевременного обслуживания запросов для вариантов с уничтожением и без уничтожения резервных копий запросов, ожидающих в очередях сверх установленного предельно допустимого срока t .

При расчетах будем считать, что $n = 20$ шт., $v = 0,1$ с. Расчет проведем для идеального случая безошибочности вычислений и безотказности узлов.

Зависимость увеличения загрузки узлов I от интенсивности запросов для варианта с уничтожением резервных копий запросов, находящихся в очередях сверх времени t , представлена на рис. 1. На рис. 1 при кратности резервирования запросов $k = 3$

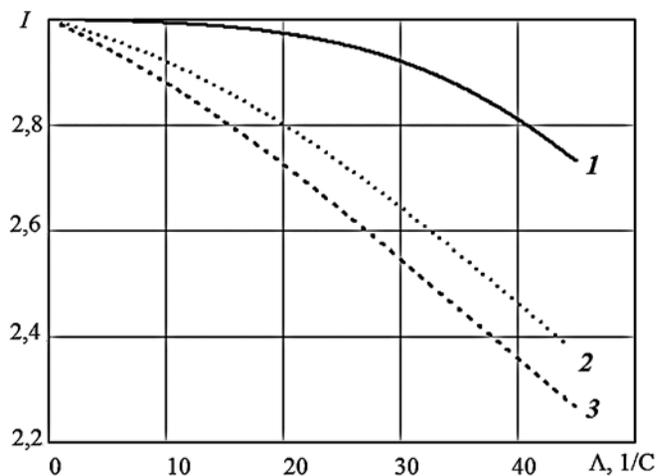


Рис. 1. Зависимость увеличения загрузки узлов I при уничтожении просроченных в очередях запросов

кривые 1, 2, 3 соответствуют предельно допустимому времени ожидания запросов $t = 0,5; 0,2; 0,15$ с.

Из представленных зависимостей видна эффективность снижения загрузки узлов в результате уничтожения просроченных резервных копий запросов. Анализируемая эффективность растет с увеличе-

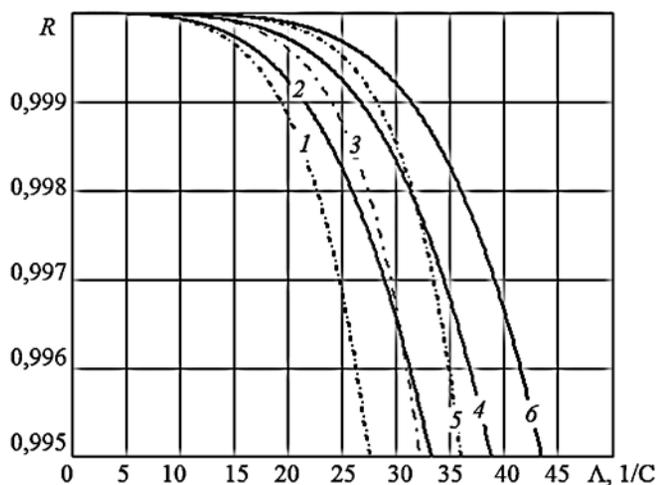


Рис. 2. Зависимость вероятности неперевышения допустимой задержки ожидания от интенсивности входного потока

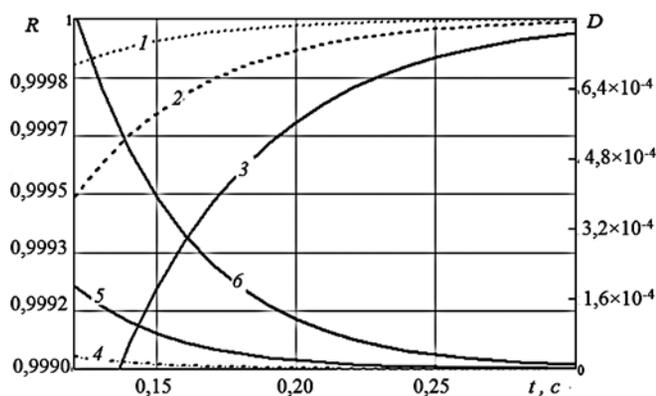


Рис. 3. Зависимость вероятности неперевышения допустимой задержки ожидания от значения этой задержки

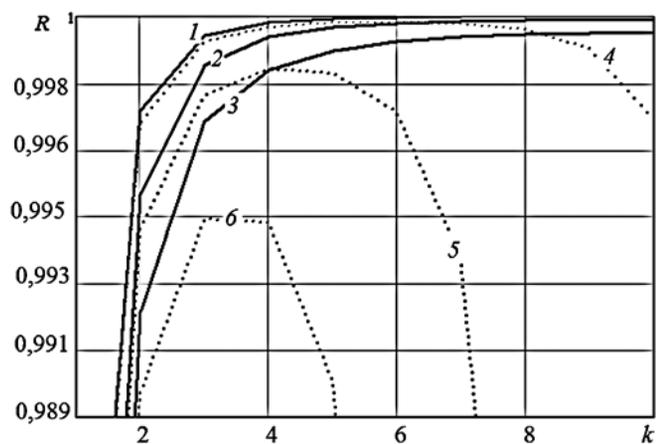


Рис. 4. Вероятность неперевышения допустимой задержки ожидания для диспетчеризации с уничтожением и без уничтожения просроченных в очередях запросов

нием интенсивности запросов и с уменьшением предельно допустимого времени ожидания t .

Зависимость вероятности неперевышения допустимой задержки ожидания t резервированных вычислений хотя бы одним из k узлов, выполняющих запрос, от интенсивности входного потока Λ представлена на рис. 2, а от значения допустимой задержки t — на рис. 3. Расчеты выполнены при кратности резервирования вычислений $k = 3$. На рис. 2 кривые 1 и 2 соответствуют предельно допустимой задержке $t = 0,15$ с в случае управления очередями при уничтожении просроченных к моменту времени t резервных копий запросов и без такого уничтожения. Кривые 3 и 4 соответствуют вариантам диспетчеризации с уничтожением и без уничтожения просроченных копий запросов при $t = 0,2$ с, а кривые 5 и 6 — при $t = 0,25$ с.

При организации управления очередями с уничтожением просроченных резервных копий запросов на рис. 3 кривыми 1—3 представлена зависимость вероятности своевременного обслуживания запросов R , вычисляемой по формуле (5), от значения допустимой задержки t при интенсивности запросов $\Lambda = 10; 15; 20$ 1/с соответственно. На рис. 3 кривые 4—6 соответствуют разнице D вероятностей своевременности резервированного обслуживания запросов R с уничтожением и без уничтожения просроченных к моменту времени t резервных копий запросов (вычисляемых соответственно по формулам (5) и (1)).

Из графиков видна высокая эффективность управления очередями с уничтожением просроченных резервных копий запросов. Причем эффективность уничтожения запросов в очередях после потери их актуальности возрастает при уменьшении допустимого предела времени t ожидания. При этом с ростом интенсивности потока запросов эффективность уничтожения резервных копий запросов, ожидающих в очереди сверх допустимого времени, повышается.

На рис. 4 приведены результаты расчетов вероятностей неперевышения допустимого времени ожидания t в зависимости от кратности резервирования запросов k . Расчеты выполнены для вариантов диспетчеризации с уничтожением и без уничтожения просроченных в очередях резервных копий запросов. Расчеты проведены для $\nu = 0,1$ с, $n = 20$ шт., $t = 0,12$ с. Кривые 1, 2, 3 соответствуют $\Lambda = 15, 20, 25$ 1/с в случае уничтожения просроченных резервных копий запросов в очередях, а кривые 4, 5, 6 — без их уничтожения. Результаты расчетов показывают влияние уничтожения просроченных в очередях резервных копий запросов реального времени на повышение эффективности их обслуживания.

На рис. 5 приведены графики вероятностей неперевышения допустимого времени ожидания t для кластеров без резервирования и с резервированием вычислений в зависимости от кратности их резер-

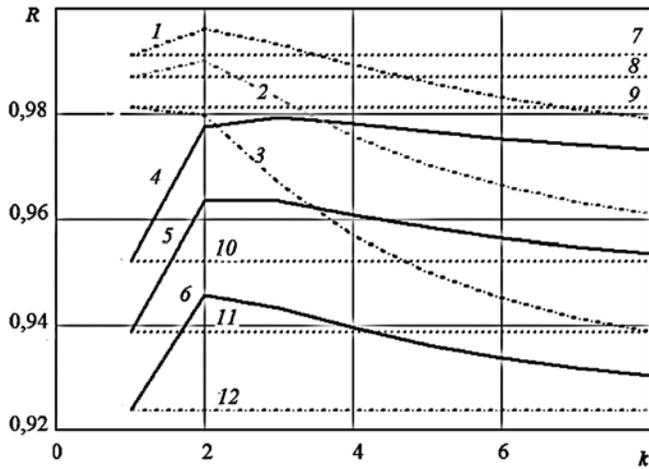


Рис. 5. Вероятности не превышения допустимого времени ожидания t для кластеров без резервирования и с резервированием вычислений

вирования k . Расчеты выполнены с учетом уничтожения просроченных в очереди резервных копий запросов для различных значений интенсивности запросов Λ и допустимого времени ожидания в очереди t .

При интенсивности входного потока $\Lambda = 60, 70, 80$ 1/с кривые 1–3 соответствуют резервированному вычислениям для $t = 0,25$ с, а кривые 4–6 — для $t = 0,5$ с. Кривые 7–9 на рис. 5 соответствуют нерезервированному выполнению запросов для $t = 0,25$ с, а кривые 10–12 — для $t = 0,5$ с.

Приведенные зависимости показывают существование области эффективности резервирования вычислений (обслуживания с резервированием запросов).

Таким образом, для систем реального масштаба времени, предусматривающих уничтожение просроченных в очереди резервных копий запросов, существует область эффективности резервированного обслуживания запросов. При этом оптимальная кратность резервирования и целесообразность резервирования запросов зависит от их интенсивности и от ограничения допустимого времени ожидания t . Таким образом, возникает необходимость оптимизации резервированных вычислений в кластерных системах реального времени.

5. Расчет среднего времени ожидания при резервировании вычислений

В предыдущем разделе показано, что резервированное выполнение запросов в нескольких узлах кластера позволяет существенно повысить вероятность своевременности вычислений, когда требуется завершение вычислений к установленному сроку хотя бы в одном из k узлов, выполняющих резервные копии запросов. Проанализируем теперь влияние резервирования выполнения запросов на среднее время их ожидания в очередях с уничто-

жением и без уничтожения резервных копий запросов, ожидающих в очередях сверх установленного предельно допустимого срока.

Для идеального случая безошибочности вычислений и безотказности узлов расчет среднего времени ожидания в очередях при резервированном обслуживании запросов w проведем по формулам (6), (2) соответственно, когда просроченные резервные копии запросов в очередях уничтожаются и не уничтожаются. При расчетах примем число узлов в кластере $n = 20$ шт., а среднее время выполнения запроса $v = 0,1$ с.

Зависимости среднего времени ожидания запросов w (рис. 6) от интенсивности входного потока Λ для диспетчеризации очередей с уничтожением и без уничтожения просроченных резервных копий запросов при кратности резервирования вычислений $k = 4, 3$ и 2 представлены соответственно парами кривых 1, 2, 3, 4 и 5, 6. Среднему времени ожидания в системах без резервирования обслужи-

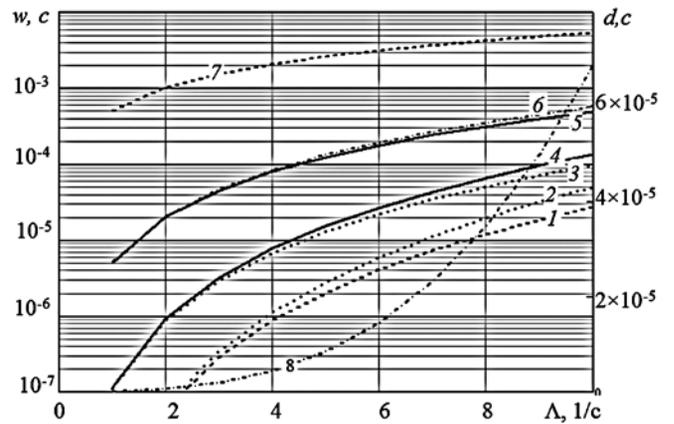


Рис. 6. Зависимости среднего времени ожидания запросов w от интенсивности входного потока Λ

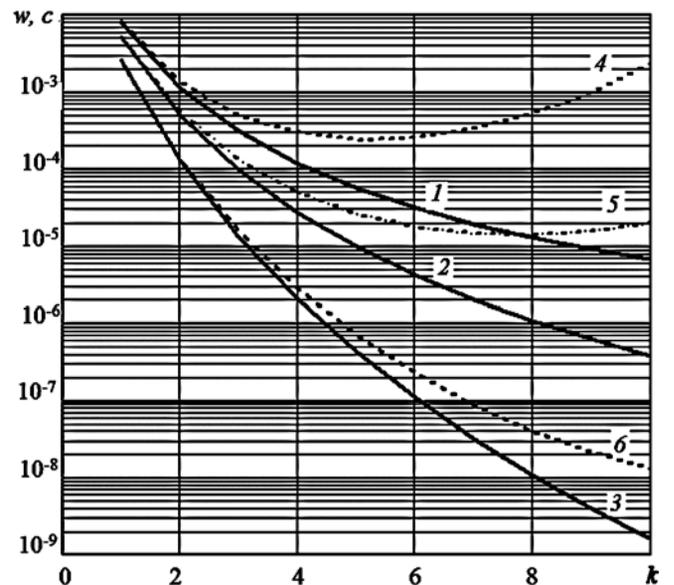


Рис. 7. Зависимости среднего времени ожидания запросов w от кратности резервирования вычислений k

вания запросов соответствует кривая 7. Кривая 8 уточняет разницу d среднего времени ожидания для диспетчеризации без уничтожения и с уничтожением просроченных резервных копий запросов в очередях при кратности резервирования вычислений $k = 2$ (уточнение вызвано близостью кривых 5, 6).

Зависимости среднего времени ожидания запросов w от кратности резервирования вычислений представлены на рис. 7 кривыми 1–3 для интенсивности входного потока $\Lambda = 15, 10, 5$ 1/с в случае диспетчеризации с уничтожением просроченных копий запросов, а кривыми 4–6 — без уничтожения копий.

Представленные на рис. 6 и 7 зависимости показывают эффективность снижения среднего времени ожидания в результате резервированного выполнения запросов несколькими (k) узлами кластера, причем эта эффективность повышается при уничтожении в очередях резервных копий запросов, ожидающих сверх допустимого лимита времени t .

Заключение

Для вычислительной системы кластерной архитектуры предложены модели обслуживания при диспетчеризации очередей с уничтожением и без уничтожения резервных копий запросов, срок нахождения которых в очередях превысил допустимое ожидание.

Проанализировано влияние кратности резервирования запросов на вероятность их своевременного и безошибочного обслуживания с уничтожением и без уничтожения просроченных в очередях резервных копий запросов.

Показано значительное повышение вероятности своевременного обслуживания в результате резервирования запросов. Сделан вывод, что диспетчеризация очередей с уничтожением просроченных в очереди резервных копий запросов позволяет существенно повысить эффективность обслуживания запросов.

Доказано существование области эффективного резервирования запросов.

Установлено существование оптимальной кратности резервирования запросов, при котором достигается максимум вероятности своевременного выполнения запросов, причем оптимальная кратность резервирования зависит от интенсивности запросов и их допустимой задержки в очереди.

Проведенные исследования эффективности предлагаемой организации резервированного обслуживания запросов ограничены аналитическим моделированием при представлении процессов обслуживания запросов СМО типа М/М/1. Подтверждение эффективности предлагаемых решений в более общих случаях требует дальнейших исследований, в том числе основанных на натурных испытаниях в кластерах, выполняющих реальные прикладные задачи.

1. **Aysan H.** Fault-tolerance strategies and probabilistic guarantees for real-time systems Mälardalen University, Västerås, Sweden, 2012. 190 p.

2. **Kopetz H.** Real-Time Systems: Design Principles for Distributed Embedded Applications. Springer, 2011. 396 p.

3. **Koren I.** Fault tolerant systems. San Francisco: Morgan Kaufmann publications. 2009. 378 p.

4. **Черкесов Г. Н.** Живучесть и отказобезопасность ответственных технических систем // Проектирование и технология электронных средств. 2015. № 1. С. 15–24.

5. **Алиев Т. И., Муравьева-Витковская Л. А.** Приоритетные стратегии управления трафиком в мультисервисных компьютерных сетях // Известия вузов. Приборостроение. 2011. Т. 54, № 6. С. 44–48.

6. **Щеглов А. Ю., Щеглов К. А.** Возможности методов резервирования для повышения уровня интегрированной информационно-эксплуатационной безопасности современных информационных систем // Информационные технологии. Том 21, № 7. 2015. С. 521–527.

7. **Гатчин Ю. А., Жаринов И. О., Коробейников А. Г.** Математические модели оценки инфраструктуры системы защиты информации на предприятии // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 2 (78). С. 92–95.

8. **Богатырев В. А.** Надежность вариантов размещения функциональных ресурсов в однородных вычислительных сетях // Электронное моделирование. 1997. № 3. С. 21–29.

9. **Богатырев В. А.** К распределению функциональных ресурсов в отказоустойчивых многомашинных вычислительных системах // Приборы и системы. Управление, контроль, диагностика. 2001. № 12. С. 1–5.

10. **Богатырев В. А., Богатырев С. В.** Объединение резервированных серверов в кластеры высоконадежной компьютерной системы // Информационные технологии. 2009. № 6. С. 41–47.

11. **Богатырев В. А.** Мультипроцессорные системы с динамическим перераспределением запросов через общую магистраль // Известия вузов. Приборостроение. 1985. № 3. С. 33–38.

12. **Богатырев В. А., Богатырев А. В., Богатырев С. В.** Перераспределение запросов между вычислительными кластерами при их деградации // Известия вузов. Приборостроение. 2014. Т. 57, № 9. С. 54–58.

13. **Богатырев В. А., Богатырев А. В., Богатырев С. В.** Оценка надежности выполнения кластерами запросов реального времени // Известия вузов. Приборостроение. 2014. Т. 57, № 4. С. 46–48.

14. **Богатырев В. А., Богатырев А. В., Богатырев С. В.** Оценка своевременности выполнения критических запросов в двухуровневых кластерах // Научно-технический вестник информационных технологий, механики и оптики. 2014. № 2 (90). С. 177–179.

15. **Богатырев В. А., Богатырев А. В., Богатырев С. В.** Оптимизация перераспределения нагрузки в кластерах при изменяющейся активности источников запросов // Известия вузов. Приборостроение. 2014. Т. 57, № 4. С. 41–45.

16. **Lee M. H., Dudin A. N., Klimenok V. I.** The SM/V/N queueing system with broadcasting service // Math. Probl. in Engineer. 2006. V. 2006. Article ID 98171. 18 p.

17. **Дудин А. Н., Сунь Б.** Многолинейная ненадежная система с управляемым широкополосным обслуживанием // Автоматика и телемеханика. 2009. Т. 70, № 12. С. 147–160.

18. **Bogatyrev V. A., Bogatyrev A. V.** Functional Reliability of a Real-Time Redundant Computational Process in Cluster Architecture Systems // Automatic Control and Computer Sciences. 2015. Vol. 49, N. 1. P. 46–56.

19. **Богатырев В. А., Богатырев А. В.** Оптимизация резервированного распределения запросов в кластерных системах реального времени // Информационные технологии. 2015. Том 21, № 7. С. 495–502.

20. **Вишневецкий В. М.** Теоретические основы проектирования компьютерных сетей. М.: Техносфера, 2003. 512 с.

The Model of Redundant Service Requests Real-Time in a Computer Cluster

Considered computing cluster architecture, operating in real time. The aim of the research is increasing the probability of timely and error-free service. This goal is achieved due to the fact that the system provides the direction of backup service requests to multiple cluster nodes with the possibility of destruction after exceeding a certain threshold expectations. Consideration of options for dispatching with destruction and without destruction of backup copies of requests waiting in the local queues of nodes in excess of the allowable time. The dependence of the probability of timely and accurate service requests from the multiplicity of backup copies of requests to the service at different nodes. Shown a significant increase in the probability of timely services as a result of reservation requests. It is shown that the scheduling queue with the destruction of expired queued backup requests can significantly improve the efficiency of service requests. The existence of effective reservation requests depending on scheduling the rate and permissible delay in queue.

Keywords: model, reliability, a redundant service requests, real-time cluster scheduling

References

1. Aysan H. *Fault-tolerance strategies and probabilistic guarantees for realtime systems*, Västerås, Sweden, Mårådalens University, 2012, 190 p.
2. Kopetz H. *Real-Time Systems: Design Principles for Distributed Embedded*, Applications Springer, 2011, 396 p.
3. Koren I. *Fault tolerant systems*, San Francisco, Morgan Kaufmann publications, 2009, 378 p.
4. Cherkosov G. N. Zhivuchest' i otkazobezopasnost' otvetstvennykh tekhnicheskikh sistem, *Proektirovanie i tekhnologiya jelektronnykh sredstv*, 2015, no. 1, pp. 15–24 (Cherkosov G. N. The durability and fail-safety critical technical systems, Engineering and technology of electronic means 2015, no. 1, pp. 15–24).
5. Aliev T. I., Murav'eva-Vitkovskaya L. A. Prioritetnye strategii upravleniya trafikom v mul'tiservisnykh komp'yuternykh setjah, *Izvestiya Vuzov. Priborostroenie*, 2011, vol. 54, no. 6, pp. 44–48. (Aliev T. I., Muravyova-Vitkovskaya L. A. Priority strategies traffic management in multi-service computer networks, *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 2011, vol. 54, no. 6, pp. 44–48).
6. Shheglov A. Ju., Shheglov K. A. Vozmozhnosti metodov rezervirovaniya dlja povysheniya urovnya integrirovannoy informacionno-jekspluatacionnoj bezopasnosti sovremennykh informacionnykh sistem, *Informacionnye tekhnologii*, 2015, no. 7, vol. 21, pp. 521–527 (Shcheglov A. Yu., Shcheglov K. A., Possible methods of redundancy to increase the level of integrated information and operational security of modern information systems, *Information technology*, 2015, no. 7, vol. 21, pp. 521–527).
7. Gatchin Ju. A., Zharinov I. O., Korobejnikov A. G. Matematicheskie modeli ocenki infrastruktury sistemy zashhity informacii na predpriyatii, *Nauchno-tekhnicheskij vestnik informacionnykh tekhnologii, mehaniki i optiki*, 2012, no. 2 (78), pp. 92–95 (Hatchin J. A., Zharinov I. O., Korobeinikov A. G. Mathematical models for estimating system infrastructure information protection at the enterprise, *Scientific and technical Bulletin of information technologies, mechanics and optic*, 2012, no. 2 (78), pp. 92–95).
8. Bogatyrev V. A. Nadezhnost' variantov razmeshheniya funktsional'nykh resursov v odnorodnykh vychislitel'nykh setjah, *Jelektronnoe modelirovanie*, 1997, no. 3, pp. 21–29 (Bogatyrev V. A. The reliability properties of the functional resources in a homogeneous computing networks, *Engineering Simulation*, 1997, no. 3, pp. 21–29).
9. Bogatyrev V. A. Karaspredeleniju funktsional'nykh resursov v otkazoustojchivykh mnogomashinnykh vychislitel'nykh sistemah, *Pribory i sistemy. Upravlenie, kontrol', diagnostika*, 2001, no. 12, pp. 1–5 (Bogatyrev V. A. the distribution of functional resources in a fault-tolerant multicomputer systems, *Devices and systems. Management, control, diagnostics*, 2001, no. 12, pp. 1–5).
10. Bogatyrev V. A., Bogatyrev S. V. Ob#edinenie rezervirovannykh serverov v klasteriy vysokonadezhnoy komp'yuternoy sistemy, *Informacionnye tekhnologii*, 2009, no. 6, pp. 41–47. (Bogatyrev V. A., Bogatyrev S. V. Association of the redundant servers in the cluster of highly reliable computer systems, *Information technologies*, 2009, no. 6, pp. 41–47.)
11. Bogatyrev V. A. Mul'tiprocessornyye sistemy s dinamicheskim pereraspredeleniem zaprosov cherez obshhuyu magistral, *Izvestiya*

Vysshikh Uchebnykh Zavedeniy. Priborostroenie, 1985, no. 3, pp. 33–38 (Bogatyrev V. A. Multiprocessor system with dynamic reallocation requests through a common line, *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 1985, no. 3, pp. 33–38.)

12. Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Pereraspredelenie zaprosov mezhdru vychislitel'nykh klasterami pri ih degradacii, *Izvestiya Vuzov Priborostroenie*, 2014, vol. 57, no. 9, pp. 54–58 (Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. the Redistribution of requests between the compute clusters at their degradation, *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 2014, vol. 57, no. 9, pp. 54–58).

13. Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Ocenka nadezhnosti vypolneniya klasterami zaprosov real'nogo vremeni, *Izvestiya Vuzov Priborostroenie*, 2014, vol. 57, no. 4, pp. 46–48 (Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. the Estimation of the reliability of clusters perform real-time queries, *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 2014, vol. 57, no. 4, pp. 46–48).

14. Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Ocenka svoevremennosti vypolneniya kriticheskikh zaprosov v dvuhurovnevnykh klasterah, *Nauchno-tekhnicheskij vestnik informacionnykh tekhnologii, mehaniki i optiki*, 2014, no. 2 (90), pp. 177–179, (Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V., Assessment of the timeliness of implementation of the critical queries in two-level clusters, *Scientific-technical Bulletin of information technologies, mechanics and optics*, 2014, no. 2 (90), pp. 177–179).

15. Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Optimizatsiya pereraspredeleniya nagruzki v klasterah pri izmenyajushhejsja aktivnosti istochnikov zaprosov, *Izvestiya Vuzov Priborostroenie*, 2014, vol. 57, no. 4, pp. 41–45 (Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Optimization of load redistribution in clusters of variable activity of the source of the query, *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 2014, vol. 57, no. 4, pp. 41–45).

16. Lee M. H., Dudin A. N., Klimenok V. I. The SM/V/N queueing system with broadcasting service, *Math. Probl. in Engineer*, 2006, vol. 2006, Article ID 98171, 18 p.

17. Dudin A. N., Sun' B. Mnogolinejnaya nenadezhnaya sistema s upravljatel'nykh shirokoveshhatel'nykh obsluzhivaniem, *Avtomatika i telemechanika*, 2009, vol. 70, no. 12, pp. 147–160. (Dudin A. N., Sun B. unreliable Multi-line system with controllable broadcasting service, *Automation and remote control*, 2009, vol. 70, no. 12, pp. 147–160).

18. Bogatyrev V. A., Bogatyrev A. V. Funktsional'nykh Reliability of a Real-Time Redundant Computational Process in Cluster Architecture Systems, *Automatic Control and Computer Sciences*, 2015, vol. 49, no. 1, pp. 46–56.

19. Bogatyrev V. A., Bogatyrev A. V. Optimizatsiya rezervirovannogo raspredeleniya zaprosov v klasternykh sistemah real'nogo vremeni, *Informacionnye tekhnologii*, 2015, no. 7, vol. 21, pp. 495–502 (Bogatyrev V. A., Bogatyrev A. V., Optimization of the redundant distribution of requests in a clustered real-time systems, *Information technology*, 2015, vol. 21, no. 7, pp. 495–502).

20. Vishnevskij V. M. *Teoreticheskie osnovy proektirovaniya komp'yuternykh setej*, Moscow, Tehnosfera, 2003, 512 p. (Vishnevsky V. M. Theoretical bases of design of computer networks. Moscow, Technosphere, 2003, 512 p.)

С. Ю. Микова, студент, e-mail: sofyamikova@mail.ru,
 В. С. Оладько, канд. техн. наук, доц., e-mail: oladko.vs@yandex.ru
 Волгоградский государственный университет, г. Волгоград

Сравнение алгоритмов выявления сетевых аномалий с помощью меры Ван Ризбергена

Предложена функциональная модель исследования качества алгоритмов обнаружения сетевых аномалий, использующая в качестве базовой метрики качества меру Ван Ризбергена. Разработан программный комплекс, автоматизирующий предложенную модель. Проведена экспериментальная оценка качества двух алгоритмов — Бродского—Дарховского и алгоритма обнаружения сетевых аномалий на основе дискретного вейвлет-преобразования. Проведенный анализ показал, что лучшим качеством классификации обладает алгоритм Бродского—Дарховского.

Ключевые слова: точность, полнота, ошибка первого рода, ошибка второго рода, алгоритм Бродского—Дарховского, дискретное вейвлет-преобразование, атака

Введение

На сегодняшний день важным атрибутом является глобальная информационная интеграция, которая состоит из построения компьютерных сетей и их объединения с помощью глобальной сети Интернет. В соответствии с работами [1, 2] в процессе пользования компьютерными сетями возникает ряд проблем, связанных с эксплуатацией злоумышленниками уязвимостей в приложениях, отказом в обслуживании сетевых ресурсов, утечкой информации, нарушением целостности и доступности данных. Поэтому администраторы должны диагностировать работу сети, подключенных к ней серверов и защищать информационные ресурсы сети от несанкционированной деятельности злоумышленников, воздействий вредоносного программного обеспечения и других видов атак. Одним из подходов к контролю над состоянием сети является регулярный мониторинг аномалий, возникающих в сетевом трафике. Своевременное выявление и подробный анализ сетевой аномалии позволяют администраторам обнаружить атаку злоумышленника на ранней стадии ее проведения. Следовательно, актуальной задачей является поиск наиболее качественного алгоритма обнаружения сетевых аномалий, который с наибольшей точностью и полнотой способен определять их, сводя к минимуму число пропусков и/или ложных срабатываний. В данной работе представлена модель исследования характеристик алгоритмов обнаружения сетевых аномалий, где в качестве основного показателя качества используется мера Ван Ризбергена. На основании разработанной модели проведено сравнение качества двух алгоритмов обнаружения сетевых аномалий — алгоритма Бродского—Дарховского (БД) и алгоритма обнаружения аномалий на основе дискретного вейвлет-преобразования (ДВП).

Показатели качества алгоритма обнаружения сетевых аномалий

Анализ литературных источников [3—7] показывает, что одним из возможных подходов к оценке качества алгоритмов обнаружения сетевых аномалий является вычисление полноты и точности алгоритма, которые напрямую зависят от ошибок первого и второго рода, возникающих в процессе работы алгоритма при решении задач классификации. Мера Ван Ризбергена является функцией от этих показателей и представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремятся к нулю, и вычисляется по следующей формуле:

$$F = 2 \frac{Precision \cdot Recall}{Precision + Recall}, \quad (1)$$

где значения полноты $Recall$ и точности $Precision$ вычисляется по формулам

$$Recall = \frac{TP}{TP + FN}; \quad (2)$$

$$Precision = \frac{TP}{TP + FP}, \quad (3)$$

где TP — истинно положительное решение (правильно обнаруженные аномалии); FP — ложно положительное решение (ошибки второго рода); FN — ложно отрицательное решение (ошибки первого рода).

Таким образом, данная мера зависит от числа ошибок первого и второго рода. Чем меньше будет ошибок первого и второго рода при обнаружении аномалий, тем более полным и точным будет алгоритм по их обнаружению.

Модель исследования качества алгоритмов обнаружения сетевых аномалий на основе меры Ван Ризбергена

Задача исследования состоит в том, чтобы, генерируя сетевой трафик с различными характеристиками как условно нормальными, так и аномальными (превышение допустимого объема трафика, частоты передачи пакетов, размеров сетевых пакетов), оценить качество существующих алгоритмов обнаружения сетевых аномалий. Полученные в результате исследования данные не только позволят выбрать наилучший алгоритм, но и могут быть использованы в дальнейшем при разработке новых алгоритмов, учитывающих недостатки и ограничения существующих.

Функциональная модель исследования качества алгоритмов обнаружения сетевых аномалий представлена в виде схемы в нотации IDEF0 на рис. 1. На вход модели подается следующая информация:

- о выбранных для исследования алгоритмах обнаружения сетевых аномалий;
- о параметрах трафика тестовой сети (интенсивность трафика от различных источников (хостов), размер сетевого пакета, число передаваемых за единицу времени сообщений), которые затем будут использованы для формирования "нормального трафика", имитирующего штатный режим работы сети, и трафика, содержащего аномалии, которые в идеальном случае должны быть полностью обнаружены алгоритмом.

На выходе модели должны быть данные о значениях показателей качества исследуемых алгоритмов

(точности, полноты и меры Ван Ризбергена), результаты сравнения исследуемых алгоритмов по мере Ван Ризбергена и решение о том, какой алгоритм считается наилучшим по данным показателям.

В обобщенном виде процесс исследования состоит из четырех основных шагов:

1) установка конфигурационных параметров, выбранных для исследования алгоритмов обнаружения сетевых аномалий;

2) генерация нормального и аномального тестового сетевого трафика, который должен быть проанализирован и классифицирован выбранными алгоритмами;

3) классификация алгоритмом поступившего сетевого трафика на нормальный трафик и на трафик, содержащий аномалии;

4) оценка качества алгоритмов на основании данных об обнаруженных алгоритмами аномалиях сетевого трафика и ошибках, совершенных в процессе классификации; сравнение алгоритмов по мере Ван Ризенбергена и принятие решения о лучшем по данному показателю алгоритме.

На рис. 2 представлена декомпозиция процесса оценки качества алгоритмов обнаружения сетевых аномалий, описанного на шаге 4.

Для проведения исследований и оценки качества алгоритмов в соответствии с разработанной моделью был разработан программный комплекс (свидетельство о государственной регистрации программы для ЭВМ № 2015615571 от 19.08.2015), подробно описанный авторами в работе [8]. Программный комплекс имеет клиент-серверную архитектуру,



Рис. 1. Модель исследования качества алгоритмов обнаружения сетевых аномалий

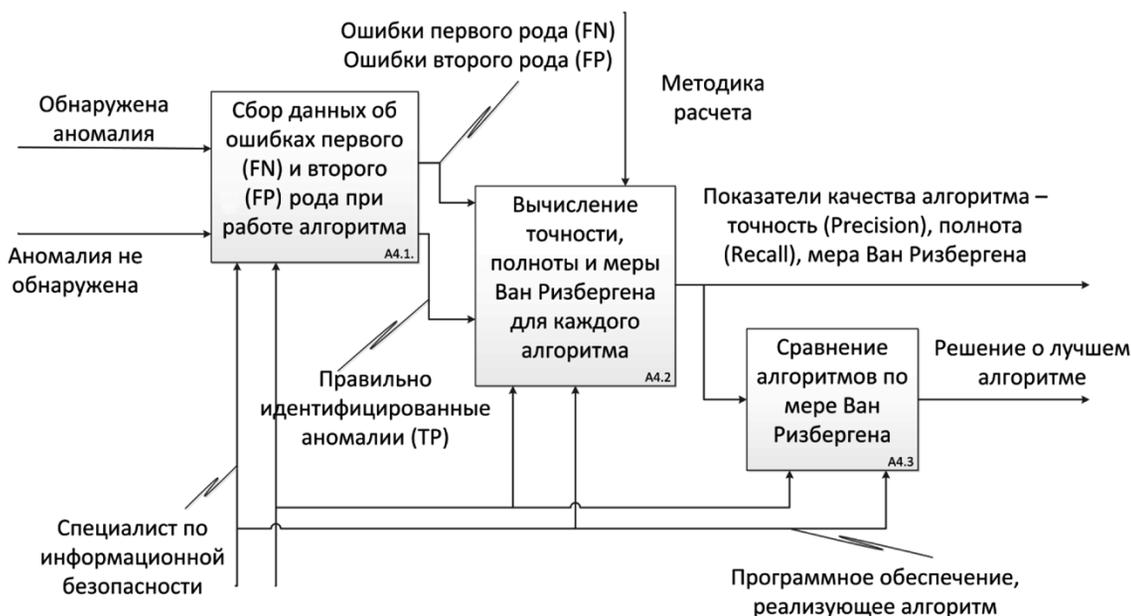


Рис. 2. Декомпозиция процесса оценки качества алгоритмов обнаружения сетевых аномалий

где клиентская часть отвечает за настройку параметров анализа качества алгоритмов и генерацию тестового сетевого трафика с аномальными и нормальными характеристиками. Серверная часть реализует алгоритмы обнаружения сетевых аномалий и собирает информацию о результатах классификации, а также данные, характеризующие алгоритм:

- 1) ошибки первого рода — FN ;
- 2) ошибки второго рода — FP ;
- 3) число правильно идентифицированных аномалий — TP .

На основании собранных данных программой проводится оценка полноты и точности алгоритма. Затем аналитически вычисляется значение меры Ван Ризбергена для каждого из анализируемых алгоритмов и на основании анализа полученных данных выбирается наилучший.

В качестве объекта исследования авторами были выбраны два наиболее известных и эффективных с вычислительной точки зрения алгоритма обнаружения сетевых аномалий — алгоритм ДВП [7] и алгоритм БД [9].

Обзор алгоритма Бродского—Дарховского и алгоритма на основе дискретного вейвлет-преобразования

Алгоритм обнаружения аномалий БД работает в стандартном режиме анализа сетевого трафика и режиме скользящего окна. При выборе стандартного режима особое влияние оказывают шумы. При выборе алгоритма в режиме скользящего окна совокупное действие помех уменьшается и выбросы, характеризующие начало и конец воздействия, представляются в более явном виде. Для практиче-

ской реализации лучше использовать алгоритм в режиме скользящего окна. Решение о наличии или отсутствии номинального выброса в трафике принимается в онлайн-режиме, используется скользящее окно WI , смещающееся слева направо по мере поступления данных.

Алгоритм ДВП основан на дискретном вейвлет-преобразовании и байесовском анализе. При исследовании сетевого трафика с применением статистических критериев используется техника скользящих окон,двигающихся во времени с определенным шагом, с фиксацией значений трафика в реальном времени, который находится во временных границах каждого окна. Применение данной техники позволяет увеличить надежность обнаружения незначительных аномалий. Конечным критерием обнаружения аномалии является превышение красного порога одним из критериев на этапе реконструкции коэффициентов. Достоинством данного алгоритма является то, что аномалия хорошо обнаруживается на каждом уровне байесовского вейвлет-преобразования декомпозиции. Недостатком данного алгоритма является то, что при начальном уровне разложения он обнаруживает наибольшее число аномалий, но некоторые аномалии могут быть пропущены, если начать разложение с более старших уровней. Более того, на старших уровнях повышается число возникновения ложных тревог, что можно объяснить низкой разрешающей способностью дискретного вейвлет-преобразования во времени. Особенно хорошо этот алгоритм подходит для обнаружения аномалий, связанных с DDos-атаками [7].

Вычисление меры Ван Ризбергена для алгоритма Бродского—Дарховского и алгоритма на основе дискретного вейвлет-преобразования

Входными данными при проведении экспериментов являются размер скользящего окна $W \in [10, 1000]$, число интервалов $I = 2200$ и число поданных аномалий в сгенерированном тестовом сетевом трафике $A \in [1, 512]$. При проведении экспериментальных исследований использовали 70 выборок комбинаций входных параметров W и A .

Затем для каждой выборки входных значений N раз был проведен прогон двух анализируемых алгоритмов. В результате была получена информация о среднем числе ошибок первого и второго рода, а также о числе правильно обнаруженных аномалий и рассчитаны по формулам (2) и (3) значения точности и полноты. В табл. 1 представлен пример результирующей оценки частных показателей качества работы алгоритма БД для входного параметра размера окна $W = 10$.

Далее была рассчитана мера Ван Ризбергена [формула (1)] для результирующей оценки двух алгоритмов. Результаты расчета приведены в табл. 2.

Для удобства сравнительного анализа качества работы алгоритмов для каждого размера окна W было рассчитано среднее значение меры Ван Ризбергена (табл. 3).

Проведенные экспериментальные исследования алгоритмов обнаружения сетевых аномалий показали следующее:

- наилучшим качеством классификации по мере Ван Ризбергена на всем выбранном диапазоне значений скользящего окна W обладает алгоритм обнаружения сетевых аномалий БД;
- на краевых значениях диапазона изменения скользящего окна $W \in [10, 1000]$ полученные значения меры Ван Ризбергена алгоритма БД выше значений меры Ван Ризбергена алгоритма ДВП на 30 и 28 % соответственно;
- качество классификации сетевого трафика алгоритмами обнаружения аномалий зависит от размера скользящего окна;
- при увеличении размера скользящего окна W качество классификации обоих алгоритмов снижается, для алгоритма БД значение меры Ван Ризбергена на правой границе диапазона изменения скользящего окна ($W = 1000$) уменьшилось в 5 раз, а для алгоритма ДВП — в 6 раз (рис. 3). Таким образом, можно выдвинуть гипотезу, что при дальнейших углубленных исследованиях влияния размера скользящего окна на качество классификации будет наблюдаться аналогичная тенденция снижения качества при увеличении размера скользящего окна. Это может быть в первую очередь связано с тем, что аналогично скользящему

Таблица 1
Результирующая оценка частных показателей качества работы алгоритма БД для размера окна $W = 10$ и интервала $I = 2200$

A	1	2	4	8	16	32	64	128	256	512
$TP_{\text{средн}}$	0,4	0,6	2,2	3	6,2	14	28,4	54	100	106
$FN_{\text{средн}}$	3,2	4,4	4,4	7	11,4	18,6	35,4	66	95	112
$FP_{\text{средн}}$	0,6	1,4	1,8	5	9,8	18	35,6	74	156	406
$Precision$	0,4	0,3	0,55	0,38	0,39	0,44	0,44	0,42	0,39	0,21
$Recall$	0,11	0,12	0,33	0,3	0,35	0,43	0,45	0,45	0,51	0,49

Таблица 2
Значения меры Ван Ризбергена для алгоритмов ДВП и БД

W	Число аномалий A									
	1	2	4	8	16	32	64	128	256	512
Значение меры Ван Ризбергена алгоритма БД										
10	0,173	0,171	0,415	0,33	0,36	0,43	0,44	0,43	0,44	0,28
15	0,1	0,22	0,32	0,46	0,31	0,38	0,32	0,35	0,32	0,202
20	0,4	0,17	0,1	0,32	0,32	0,31	0,32	0,31	0,24	0,17
25	0,18	0,2	0,48	0,18	0,29	0,32	0,27	0,27	0,21	0,14
30	0,25	0,08	0,139	0,275	0,28	0,22	0,24	0,24	0,21	0,11
500	0,33	0,16	0,09	0,04	0,02	0,012	0,006	0,021	0,004	0,003
1000	0,28	0,16	0,09	0,047	0,02	0,012	0,012	0,018	0,001	0,0007
Значение меры Ван Ризбергена алгоритма ДВП										
10	0,15	0,12	0,15	0,16	0,17	0,16	0,13	0,06	0,01	0,0007
15	0,09	0,02	0,008	0,01	0,002	0,001	0,008	0,005	0,001	0,0007
20	0,73	0,06	0,09	0,08	0,09	0,07	0,05	0,05	0,009	0,0007
25	0,035	0,01	0,006	0,003	0,001	0,001	0,029	0,008	0,001	0,0007
30	0,064	0,06	0,06	0,065	0,05	0,042	0,075	0,035	0,001	0,0007
500	0,003	0,004	0,015	0,02	0,008	0,037	0,019	0,001	0,0015	0,00078
1000	0,004	0,006	0,083	0,041	0,02	0,0123	0,006	0,003	0,0015	0,00078

Таблица 3

Среднее значение меры Ван Ризбергера для исследуемых алгоритмов ДВП и БД

Размер скользящего окна W	Мера Ван Ризбергера	
	Алгоритм ДВП	Алгоритм БД
10	0,114803	0,35098
15	0,016187	0,30158
20	0,061216	0,269707
25	0,010735	0,256468
30	0,046203	0,208267
500	0,011936	0,07118
1000	0,018441	0,066088

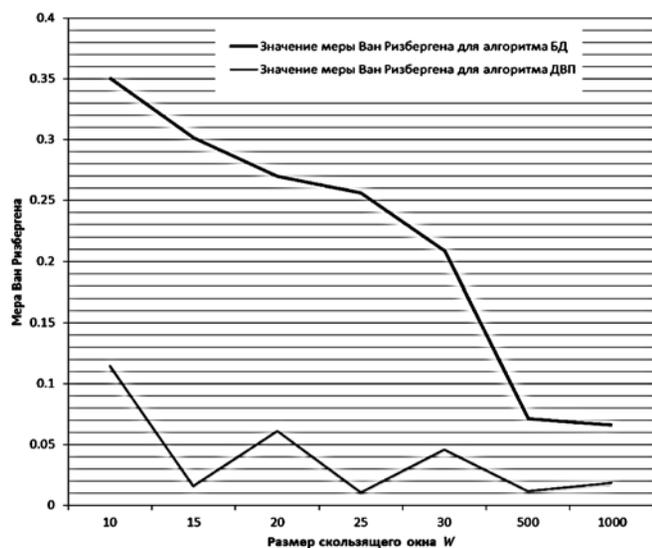


Рис. 3. Зависимость меры Ван Ризбергера от размера окна

окну протокола TCP скользящее окно W алгоритма обнаружения аномалий определяет диапазон подлежащих анализу байтов трафика, которые алгоритм готов принять в настоящее время. Чем больше размер окна, тем больший объем данных должен проанализировать и классифицировать алгоритм. В результате увеличивается неопределенность при классификации, а следовательно, и вероятность ошибок первого и второго рода, что оказывает непосредственное негативное влияние на качество классификации.

Сравнение полученных авторами результатов исследования алгоритмов с результатами, представленными в работах [6–9], показывает следующее.

1. В работе [6] представлен обзор качественных и количественных критериев, которые можно применять при многокритериальном анализе качества алгоритмов обнаружения сетевых аномалий. Однако сама оценка не проводится, поэтому не представляется возможным сравнить результаты данной работы с результатами, представленными в работе [6].

2. В работе [7, с. 114] показано, что при увеличении ширины окна W до 1500 наблюдается увеличение ошибок первого рода, что, в свою очередь, оказывает влияние на такой показатель качества,

как полнота классификации, и, опираясь на формулу (2), позволяет сделать вывод о его снижении. А так как полнота участвует в вычислении меры Ван Ризбергера (1), то чем показатель полноты ниже, тем ниже будет и общее качество классификации. Таким образом, можно сделать вывод, что полученные авторами результаты не противоречат результатам исследования, представленным в работе [7].

3. В работе [8] представлено описание архитектуры, алгоритмов функциональных модулей и технических особенностей реализации разработанного авторами программного комплекса, предназначенного для исследования качества алгоритмов обнаружения сетевых аномалий. При этом описание экспериментальных исследований или методики их проведения не приведено, соответственно, по объективным причинам сравнение результатов работ невозможно.

4. В работе [9, с. 116] сделан вывод, что алгоритм разладки Бродского—Дарховского по сравнению с другими алгоритмами анализа сетевого трафика имеет меньшее число ложных тревог (ошибок первого рода) и малое запаздывание в обнаружении изменений сетевого трафика, при этом лучшими показателями обладают рекуррентные версии алгоритма. А это соответствует результатам, полученным авторами при сравнении качества классификации двух алгоритмов — БД и ДВП.

Заключение

Была разработана функциональная модель исследования качества алгоритмов обнаружения сетевых аномалий, где базовой метрикой взята гармоническая мера Ван Ризбергера. Данная модель может быть использована для решения задач, связанных с выбором наиболее эффективного алгоритма или средства обнаружения аномалий в сети, что позволило бы с меньшим числом ошибок и ложных срабатываний выявлять признаки возможных атак злоумышленника на сетевые ресурсы организации. Реализован программный комплекс, автоматизирующий предложенную модель, и проведены экспериментальные исследования двух алгоритмов обнаружения сетевых аномалий — алгоритма ДВП и алгоритма БД.

Данные исследования показали, что при сравнении двух алгоритмов по мере Ван Ризбергера лучшим качеством классификации обладает алгоритм обнаружения сетевых аномалий БД. Также анализ результатов экспериментальных исследований позволяет сделать вывод, что для получения лучшего качества классификации необходимо при возможности минимизировать значение такого входного параметра, характерного для обоих алгоритмов, как размер скользящего окна W . А поскольку мера Ван Ризбергера — это метрика, которая гармонически объединяет информацию о точности и полноте анализируемого алгоритма, то при использовании данных алгоритмов для обнаружения

сетевых аномалий необходимо еще в процессе обучения провести калибровку точности и полноты, варьируя значения окна W , так как увеличение данных оценок влияет на повышение качества классификации.

Разработанный программный комплекс может быть применен на практике для оценки качества и выбора наилучшего алгоритма обнаружения аномалий при организации защиты корпоративной сети или в учебном процессе, на лабораторном практикуме, в качестве стенда макета при обучении студентов направления информационная безопасность.

Список литературы

1. Шелухин О. И., Сакалема Д. Ж., Филинова А. С. Обнаружение вторжений и компьютерные сети. М.: Горячая линия-Телеком, 2013. 220 с.
2. Щеглов К. А., Щеглов А. Ю. Защита от атак на уязвимость приложений // Информационные технологии. 2014. № 9. С. 34—39.
3. Никишова А. В., Чурилина А. Е. Обнаружение распределенных атак на информационную систему предприятия // Из-

вестия ЮФУ. Технические науки. Тематический выпуск "Информационная безопасность". 2013. № 12 (149). С. 135—143.

4. Зеленков Ю. Г., Сегалович И. В. Сравнительный анализ методов обнаружения нечетких дубликатов для Web-документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 9-й Всероссийской научной конференции. Переславль-Залесский, Россия, 2007. С. 84—89.

5. Никишова А. В., Попов И. С. Обнаружение сетевых аномалий // Приоритетные технологии: актуальные вопросы теории и практики: Сб. науч. докл. Первого Всерос. конгресса, г. Волгограда, 24—25 апр. 2014 г. — В.: Изд-во ВолГУ, 2014. С. 67—69.

6. Микова С. Ю., Оладько В. С., Нестеренко М. А., Кузнецов И. А. Критерии оценки качества алгоритмов обнаружения сетевых аномалий // Международный научно-исследовательский журнал. 2015. № 4 (35). С. 87—88.

7. Шелухин О. И., Панкрушин А. П. Оценка достоверности обнаружения аномалий сетевого трафика методами дискретного вейвлет-преобразования // T-Comm — Телекоммуникации и Транспорт. — 2013. Т. 7, № 10. С. 110—113.

8. Микова С. Ю., Оладько В. С. Программный комплекс для исследования качества алгоритмов обнаружения сетевых аномалий // Информационные системы и технологии. 2015. № 5 (91). С. 130—138.

9. Шелухин О. И., Филинова А. С. Обнаружение сетевых аномальных выбросов трафика методом разладки Бродского—Дарховского // T-Comm — Телекоммуникации и Транспорт. 2013. Т. 7, № 10. С. 116—118.

S. Yu. Mikova, student, e-mail: sofya_mikova@mail.ru,

V. S. Oladko, Ph. D., Associate Professor, e-mail: oladko.vs@yandex.ru,

Volgograd State University, Volgograd

Comparison of Algorithms to Identify Network Anomalies Using Measures Van Rizbergena

The problem of selecting algorithms for detecting network anomalies viewed in the article. The aim of this study is to develop an approach to the selection of the best algorithm to detect anomalies in network traffic; this approach will reduce the risk of possible attacks. The authors as a criterion for the selection of the algorithm used the criterion of the quality of the classification of network traffic. Functional model of quality research algorithms to detect network anomalies offered. As a basic measure of quality used Van Rizbergena. Program complex that automates the proposed model developed. The objective of the study is to generate network traffic with different characteristics both normal and abnormal, to assess the quality of existing algorithms for detecting network anomalies. Experimental evaluation of the quality of the two algorithms — Brodsky-Darhovsky algorithm and network anomalies detection algorithm based on discrete wavelet transform performed. These estimate algorithms showed that the best quality classification algorithm has Brodsky — Darhovsky algorithm.

Keywords: accuracy, completeness, false positive, false negative, Brodsky—Darhovsky algorithm, discrete wavelet transform, attack

References

1. Shelukhin O. I., Sakalema D. Zh., Filinova A. S. *Obnaruzheniye vtorzheniy i komp'yuternyye seti*, Moscow, Goryachaya liniya-Telekom, 2013, 220 p.
2. Shcheglov K. A., Shcheglov A. Yu. Zashchita ot atak na uyazvимость prilozheniy, *Informatsionnyye tekhnologii*, 2014, no. 9, pp. 34—39.
3. Nikishova A. V., Churilina A. Ye. Obnaruzheniye raspredelennykh atak na informatsionnyuyu sistemu predpriyatiya, *Izvestiya YUFU, Tekhnicheskkiye nauki, Tematicheskyy vypusk "Informatsionnaya bezopasnost"*, 2013, no. 12 (149), pp. 135—143.
4. Zelenkov Yu. G., Segalovich I. V. Sravnitel'nyy analiz metodov obnamzheniya nechetkikh dublikatov dlya Web-dokumentov, *Elektronnyye biblioteki: perspektivnyye metody i tekhnologii, elektronnyye kolleksii: trudy 9-oy Vserossiyskoy nauchnoy konferentsii*. — Pereslavl' — Zalesskiy, Rossiya, 2007, pp. 84—89.
5. Nikishova A. V., Popov I. S. Obnaruzheniye setevykh anomalii, *Priortetnyye tekhnologii: aktual'nyye voprosy teorii i praktiki: sb.*

nauch. dokl. Pervogo Vseros. kongressa, g. Volgograda, 24—25 apr. 2014 g. Volgograd, Izd-vo VolGU, 2014, pp. 67—69.

6. Mikova S. Yu., Olad'ko V. S., Nesterenko M. A., Kuznetsov I. A. Kriterii otsenki kachestva algoritmov obnaruzheniya setevykh anomaliiy, *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal*, 2015, no. 4 (35), pp. 87—88.

7. Shelukhin O. I., Pankrushin A. P. Otsenka dostovernosti obnaruzheniya anomaliiy setevogo trafika metodami diskretnogo veyvlet-preobrazovaniya, *T-Comm — Telekommunikatsii i Transport*, 2013, vol. 7, no. 10, pp. 110—113.

8. Mikova S. Yu., Oladko V. S. Programmnyy kompleks dlya issledovaniya kachestva algoritmov obnaruzheniya setevykh anomaliiy, *Informatsionnyye sistemy i tekhnologii*, 2015, no. 4 (90), sentyabr'—oktyabr' 2015, pp. 130—138.

7. Shelukhin O. I., Filinova A. S. Obnaruzheniye setevykh anomal'nykh vybrosov trafika metodom razladki Brodskogo—Darhovskogo, *T-Comm — Telekommunikatsii i Transport*, 2013, vol. 7, no. 10, pp. 116—118.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.7

С. М. Авдошин, канд. техн. наук, проф., savdoshin@hse.ru,

А. В. Лазаренко, студент, avlazarenko@edu.hse.ru

Национальный исследовательский университет "Высшая школа экономики" (НИУ ВШЭ)

Методы деанонимизации пользователей Tor

Выполнен обзор методов деанонимизации пользователей Tor. Рассмотрены пассивные и активные методы деанонимизации, приведены их ключевые характеристики.

Ключевые слова: DoS; Tor; анализ трафика; анонимная сеть; деанонимизация; тайминг-атаки

Введение

На сегодняшний день сеть Tor [1] является самой большой в мире развернутой анонимной сетью. Ежемесячное число активных пользователей сети превышает 2 млн человек, а число волонтерских серверов, используемых в качестве узлов сети, превосходит 6 тыс. [2].

Поскольку помимо обычных пользователей преимуществами анонимизации трафика пользуются террористы, продавцы наркотиков и оружия, а также прочие нарушители закона, то деанонимизация пользователей является достаточно актуальной и важной задачей для специальных служб различных государств [3, 4]. Так, например, МВД РФ объявляло тендер на разработку способов деанонимизации пользователей сети Tor [5].

Прогресс, достигнутый в разработке методов деанонимизации, позволил американским специальным службам осуществить ряд успешных операций по борьбе с наркоторговлей. Так, например, был закрыт доступ к самому большому в теневом Интернете магазину наркотиков Silk Road [6].

Сеть Tor состоит из волонтерских серверов, являющихся ее узлами. Пользователи через луковый прокси (далее — ЛП) загружают список узлов из сервера каталогов и строят анонимные туннели (цепи), используя луковую маршрутизацию. ЛП строит цепь, как правило, состоящую из трех узлов: входного (*guard*), промежуточного (*middle*), выходного (*exit*). Время жизни цепи составляет 10 мин (по умолчанию). Входной узел обычно выбирается из фиксированного набора из трех узлов, уникального для каждого ЛП.

Более детальное описание Tor можно найти в работе [1].

Для организации атак необходимо обладать некоторыми ресурсами, например, коррумпированными узлами Tor, или серверами, доступ к которым пытаются получить пользователи.

Поскольку сеть Tor является оверлейной сетью, она работает на основе транспортного слоя. Основными организациями, управляющими интернет-маршрутизацией, являются автономные системы (далее — АС). Атакующий может контролировать одну или несколько АС и предполагается, что он наблюдает трафик, проходящий через АС. Анализ влияния АС на деанонимизацию пользователей Tor можно найти в работе [7].

Как правило, атакующий преследует цель скомпрометировать как можно больше цепей, относящихся к конкретному пользователю или группе пользователей, поскольку компрометирование цепей влечет за собой деанонимизацию пользователей.

Ниже приведены термины и сокращения, использованные в работе.

Термины и сокращения

BGP (*Border Gateway Protocol*) — это основной протокол динамической маршрутизации, который используется в Интернете. Маршрутизаторы, использующие протокол BGP, обмениваются информацией о доступности сетей.

BGP Hijack (BGP-похищение) — незаконное поглощение группы IP-адресов.

BGP Interception (BGP-прослушивание) — незаконное прослушивание группы IP-адресов.

CREATE-сообщение — сообщение управляющего типа, посылается для установления новой цепи через сеть.

DESTROY-сообщение — сообщение управляющего типа, посылается для разрыва существующей цепи.

DoS (Denial of Service) — хакерская атака на вычислительную систему с целью довести ее до отказа, т. е. создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам) либо этот доступ затруднен.

HTTP (HyperText Transfer Protocol) — протокол прикладного уровня передачи данных (изначально в виде гипертекстовых документов в формате HTML, в настоящий момент используется для передачи произвольных данных).

ID — уникальный идентификатор.

Iframe — html-контейнер, содержание которого игнорируется браузерами, не поддерживающими данный тег.

IP-адрес (Internet Protocol Address) — уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP.

ISP (Internet Service Provider) — организация, предоставляющая пользователям доступ к сети Интернет и связанные с этим услуги.

JavaScript — прототипно-ориентированный сценарный язык программирования.

RAPTOR (Routing Attacks on Privacy in Tor) — маршрутизационные атаки на приватность в Tor.

RELAY DATA-сообщение — сообщение управляющего типа, посылается для выполнения команды к ретранслированию данных.

Sybil-атака — атака на безопасность компьютерной системы, где репутационная система подрывается с помощью нелегитимных сущностей в пиринговых сетях.

Tagging attack — атака на приватность в сети Tor, при которой помечается какая-то ячейка и затем происходит ее поиск на другом конце соединения.

TCP (Transmission Control Protocol) — протокол управления передачей — один из основных протоколов передачи данных Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.

Timing-атака — атака по сторонним каналам, в которой атакующий пытается скомпрометировать систему с помощью анализа времени, затрачиваемого на исполнение операций.

Tor (The Onion Router) — анонимная сеть и открытое программное обеспечение, позволяющее сохранять пользователям свою анонимность.

WF (Website Fingerprinting) — класс пассивных атак, позволяющий определить посещенный пользователем портал или скрытую службу.

Автономная система (АС) — это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом.

Конвейерная обработка HTTP — технология, которая позволяет передавать на сервер сразу несколько запросов в одном соединении, не ожидая соответствующих ответов.

Коррупцированная автономная система — автономная система, контролируемая наблюдателем.

Коррупцированный сервер — конечная точка назначения, контролируемая наблюдателем.

Коррупцированный узел — узел, трафик которого может модифицировать и просматривать атакующий.

ЛП — луковый прокси.

Метаданные сети — субканальная информация об используемых данных.

Микс Чаума — устройство для передачи и хранения, принимающее какое-то число сообщений фиксированной длины от нескольких источников, совершающее криптографическую трансформацию сообщений и затем передающее сообщение к следующему пункту назначения в случайном порядке.

Скрытая служба — портал/сайт, доступный только внутри сети Tor.

УМПС — узел с минимальной пропускной способностью.

ЦП — целевой поток.

1. Пассивные методы

Пассивные методы лежат в основе атак, при которых атакующий отслеживает сообщения внутри системы, не делая попыток изменения данных или вторжения в процесс взаимодействия.

1.1. Атаки анализа трафика

Анализ трафика — это извлечение информации из метаданных сети, включая объем и временные характеристики сетевых пакетов. Наблюдатель использует эти данные для выявления связи между инициатором сообщения и конечной точкой назначения. Ввиду того что внутренние механизмы Tor скрывают битовые шаблоны данных, передающихся через цепь, атакующий не может использовать информацию, содержащуюся в сообщении.

Традиционные атаки анализа трафика делятся на два класса.

В первом классе атак анонимная сеть представляется в виде черного ящика и рассматриваются временные связи между инициацией пользователем соединения и соединениями, установленными вне сети. Данный класс атак подробно не рассматривается в настоящей статье, поскольку он неприменим к глобальной сети ввиду ее размеров и огромного количества ресурсов, необходимых для проведения атаки. В работе [8] представлено подробное описание атак этого класса. Статистический вариант атак представлен в работе [9], в работе [10] подтвержден экспериментально. Такие атаки эффективны, если атакующий обзревает большую часть сети и есть возможность зарегистрировать пользова-

телей, заходящих в сеть и на внешние сервисы. Временной анализ атак рассмотрен в работах [10, 11].

Второй класс атак рассматривает трафик и нагрузку на каждом узле только внутри анонимной сети. В работе [12] представлено детальное описание данных атак. В действительности атакующий просматривает поток данных на каждом узле, например ответ сервера инициатору запроса. Поток в узле представляется в виде зависимости объема трафика от времени, затем проводится сглаживание этой функции за счет свертки ее с экспоненциально-убывающей функцией в целях получения шаблона, предсказывающего вид трафика в анонимной сети. Далее все фрагменты трафика в сети, преобразованные таким же образом, сравниваются с шаблоном. Степень сходства с шаблонами позволяет определить его принадлежность к конкретному узлу в цепочке соединения. Похожие атаки представлены в работах [13, 14].

Очевидная проблема таких атак заключается в том, что атакующий должен просматривать все узлы в сети и иметь возможность регистрировать метаданные трафика. В результате данные атаки используют модель пассивного глобального наблюдателя, который не рассматривается в модели угроз Тог. Заметим, что если наблюдатель обозревает не всю сеть, а какую-то ее часть, то он имеет возможность отследить случайные соединения. Исследование эффективности атак анализа трафика в зависимости от реальных моделей атакующего представлены в работе [15].

Малозатратная атака анализа трафика сети Tor [16] базируется на следующем наблюдении: высокая нагрузка на один из узлов в цепи влияет на задержку всех остальных узлов. Путем перенаправления соединения через специфические узлы и замера возникающих задержек наблюдатель может получить примерную нагрузку на узел. Эту нагрузку можно сравнить с известным шаблоном трафика с помощью традиционных техник анализа трафика [12].

Любой пользователь Тог может провести такие измерения и определить нагрузку на узел. Предполагается, что атакующий контролирует коррумпированный узел. Этот узел инициирует соединение, проходящее через другие узлы, нагрузка на которых должна быть измерена. Затем коррумпированный узел наполняет соединение пробным трафиком, который позволяет измерить задержки и вычислить нагрузку на узел, являющийся целью атакующего.

Просматривая соединение пользователя с сетью Тог или соединение сети Тог с конечной точкой назначения, наблюдатель может использовать эту технику для определения узлов, через которые ретранслировалось соединение. Существует более мощный вариант данной атаки: наблюдатель контролирует сервер, к которому подключается отсле-

живаемый пользователь. Сервер посылает пользователю через Тог данные, имеющие специфический шаблон трафика, что позволяет идентифицировать узлы пользователя.

Малозатратная атака на основе маршрутизации [17] предполагает, что анонимность может быть скомпрометирована с помощью локального набора коррумпированных узлов. Предполагается, что коррумпированные узлы этого набора имеют высокую пропускную способность.

Для применения атаки наблюдатель должен контролировать набор, состоящий из более чем одного узла из списка активных узлов Тог.

На первом шаге необходимо внедрить коррумпированные узлы в качестве входных и выходных узлов в цепи пользователя. Существенное сокращение необходимых ресурсов происходит за счет использования узлов с низкой пропускной способностью. Данное улучшение основывается на том, что узел может предоставить недостоверные данные о своих ресурсах службе каталогов, представляясь мощным узлом для всей системы. Однако вычислительных ресурсов узла должно хватать для установки новых соединений. Таким образом, все ресурсы узла направляются на прием новых соединений от ЛП.

Если коррумпированные узлы находятся в промежуточных позициях (между входным и выходным узлами), они могут разорвать цепь, поскольку такая конфигурация несовместима с атакой. Это вызовет перестройку цепи, и, возможно, после этого установится конфигурация, при которой коррумпированные узлы станут концевыми.

Наличие коррумпированных входного и выходного узлов является обязательным условием для успешного применения атаки, поскольку иная конфигурация не позволит проводить корреляцию трафика.

Если цепь полностью состоит из коррумпированных узлов, то компрометирование анонимности является тривиальной задачей. В наиболее вероятном варианте компрометированные узлы будут только входными и выходными для данного пользователя. Данный тип атаки позволяет компрометировать цепь до того, как пользователь начинает пересылать данные.

Для того чтобы собрать достаточно информации для корреляции клиентских запросов с ответами сервера через Тог, каждый коррумпированный узел регистрирует следующую информацию для каждой ячейки с данными: ее позицию в цепи (на входном, выходном узлах или в середине); местное время; ID прошлой цепи; прошлый IP-адрес; прошлый порт соединения; IP-адрес следующего промежуточного соединения; порт следующего промежуточного соединения; ID цепи следующего промежуточного соединения. Как только эти данные будут собраны, атакующий может связать пути,

в которых содержатся выходной и входной коррумпированные узлы, с ЛП, от которых идут запросы построения цепей. Обладая этой информацией, атакующий может связать адресанта с адресатом, компрометируя анонимность системы.

Для использования атаки данного типа коррумпированные узлы должны быть координированы. Самый простой способ координации — использовать централизованный сервер для сбора логов с узлов, что позволит наблюдателю выполнить алгоритм компрометирования цепи в реальном времени.

Алгоритм компрометирования цепи работает следующим образом: коррумпированный узел подтверждает, что запрос на создание цепи был отправлен ЛП, а не узлом. Затем проверяются хронологический порядок шагов создания цепи и совпадение промежуточного соединения для входного узла с промежуточным соединением выходного. После этого проверяется, получено ли для ответа от выходного узла сообщение, посланное от входного узла выходному. Если все проверки выполнены успешно, то цепь компрометирована.

В работе [17] приведены результаты экспериментов, которые проводили с использованием закрытой развернутой сети Tor. Она состояла из 60 легитимных и шести коррумпированных узлов. Атака успешно скомпрометировала 46 % цепей.

Атаки на основе пропускной способности канала [18] являются незаметными как для пользователя, так и для узла. Пропускная способность канала Tor может быть использована в качестве следа для обнаружения узла с минимальной пропускной способностью (УМПС). Через подробное рассмотрение динамики пропускной способности канала можно выявить, что два потока используют один и тот же набор узлов, это позволяет атакующему связать потоки с цепями и скомпрометировать приватность пользователя.

Фундаментальное наблюдение, используемое в атаке, — разнородность пропускных способностей узлов Tor [18]. Существуют три ключевых фактора, определяющих пропускную способность цепи: а) пропускная способность УМПС; б) число активных ТСР-потоков между УМПС и следующим узлом; в) число других активных цепей, размноженных через ТСР-соединение. То есть основная идея атаки — корреляционный анализ пропускных способностей цепей.

Первый вариант атаки позволяет определить, что две цепи используют общий набор узлов. На основе мониторинга пропускных способностей узлов проводится простой статистический тест для выявления корреляции между ними. Существуют три возможных варианта.

- Обе цепи используют одинаковый набор узлов. При таком варианте коэффициент корреляции пропускных способностей цепей будет высок. Это объясняется тем, что любые вариации по-

токов или пропускных способностей узлов будут влиять на пропускную способность цепей одинаково.

- Обе цепи не имеют общих узлов. В этом случае степень корреляции будет близка к нулю.
- Обе цепи имеют как минимум один общий узел. Если общий узел является УМПС в обеих цепях, тогда пропускные способности цепей будут иметь высокий коэффициент корреляции. Иначе изменения потоков или пропускных способностей узлов при пересылке данных не будут влиять на пропускную способность обеих цепей и их пропускные способности будут зависеть от их УМПС.

Второй вариант атаки — идентификация одного или нескольких узлов в качестве участника какого-либо целевого потока (ЦП) [18]. Этот поток может быть любым потоком, инициированным ЛП через сеть. Предполагается, что атакующий может сформировать пробный поток, просматривать трафик ЦП и имеет коррумпированный выходной узел. Кроме этого, у наблюдателя должен быть либо веб-сервер, к которому пользователь пытается получить доступ, либо ISP, ретранслирующий данные. Атакующему не нужно модифицировать трафик. Он строит цепь с одним узлом, через эти узлы вычисляет корреляцию между ЦП и пробным потоком. Если пропускная способность имеет высокий коэффициент корреляции с пропускной способностью ЦП, тогда сервер может допустить, что оба потока проходят через общий узел.

Website fingerprinting (WF) — класс атак, получивший большую популярность среди исследователей [19—21]. Он демонстрирует, что наблюдатель, способный просматривать зашифрованный трафик какой-то части сети, имеет возможность, при некоторых условиях, скомпрометировать портал, посещенный пользователем в сети.

Атакующий должен иметь доступ к входному узлу пользователя, чтобы просматривать его трафик и видеть IP-адрес пользователя.

Стратегия атакующего следующая: он пытается смоделировать сетевые условия пользователей путем создания собственного ЛП, посещающего те порталы, связь пользователя с которыми необходимо подтвердить. Затем обучается классификатор с учителем и использованием большого количества сетевых свойств портала (цепочки пакетов, их размер и временные интервалы между ними). Используя построенную модель, атакующий классифицирует трафик пользователей в сети.

Атака начинается со сбора данных. В сети Tor есть несколько факторов, значительно усложняющих качество собранных данных: процесс конструирования цепей; пропускная способность; загруженность. Например, образцы трафика, собранные через одну цепь, могут отличаться от образцов, собранных через другую цепь. Контент порталов может со-

временем меняться, что сильно влияет на образцы трафика.

WF-атаку можно представить в виде задачи классификации [19]. Каждый класс может быть группой сайтов, например "Специфический сайт", "Остальные сайты". Таким образом, после сбора данных нужно обучить классификатор. После тренировки на полученных образцах классификатор сможет идентифицировать неизвестные образцы.

Наиболее продвинутой модификацией WF-атаки на Tor позволяет деанонимизировать пользователя с 95 %-ной вероятностью в закрытой сети Tor, и с 91 %-ной вероятностью в сети, открытой для узкой группы сайтов.

В ответ на успешную попытку применения WF-атаки [22] разработчики Tor сделали экспериментальную защиту [23]. Защита содержит три компонента: конвейерная обработка HTTP, размер конвейера и порядок запросов, задаваемые случайно. В работе [19] показали, что защита неэффективна. Более подробную техническую информацию о WF можно найти в работе [24].

1.2. *Timing-атака*

Timing-атаки являются одними из самых ранних методов деанонимизации. Их наблюдали в самых старых анонимных сетях, включая ранние версии луковой маршрутизации [25]. Данные атаки очень похожи на атаки анализа трафика в сетях, основанных на миксах Чаума [26].

Классическая Timing-атака [27] использует метод, при котором атакующий наблюдает временные шаблоны в сетевом потоке и затем проводит корреляцию выявленных шаблонов с другими, найденными в трафике. Если атакующий имеет возможность наблюдать как пользовательский трафик, так и трафик в конечной точке соединения, то он может установить связь между ними.

Идея основывается на том, что в Tor задержка не может быть большой, т. е. временной шаблон пакетов данных должен сохраняться при продвижении через цепочку соединения.

Для применения данной атаки атакующему необходимо иметь коррумпированный узел.

Модель атаки следующая.

1. Коррумпированный узел устанавливает соединение с другими узлами Tor, чтобы измерить задержки соединений.

2. Коррумпированный узел продолжает проводить мониторинг задержек всех установленных соединений на протяжении определенного временного промежутка.

3. Значения задержек используют для расчета транспортной нагрузки тех узлов Tor, с которыми установил соединение злокачественный узел.

4. Вычисляют шаблоны трафика, зависящие от транспортной нагрузки.

5. Когда атакующий получит шаблоны трафика всех узлов, он может воспроизвести атаку по сценарию атаки анализа трафика.

Для того чтобы сделать атаку более эффективной, необходимо сделать коррумпированный сервер, к которому будет подключаться пользователь. Благодаря этому нет необходимости просматривать соединение для извлечения шаблона трафика. Наблюдатель может выбрать шаблон трафика, который легко обнаруживается и посылает свои потоки через коррумпированный сервер. Цель такого улучшения — найти цепь между узлом пользователя и коррумпированным сервером. С таким улучшением анонимность системы падает до уровня простого прокси.

1.3. *Circuit fingerprinting-атака* [28]

Данная атака является одной из самых современных комбинированных атак.

Атака спроектирована для компрометирования приватности пользователя, контактирующего со скрытыми службами Tor, что позволяет с высокой степенью точности определить взаимосвязь пользователя со скрытой службой. Как только активность пользователя и скрытой службы идентифицирована, используется WF-атака. Предполагается, что атакующий наблюдает трафик между пользователем и сетью Tor.

Трафик, участвующий во взаимодействии со скрытой службой, оставляет четкий след, поэтому легко выявить соответствующий шаблон трафика.

Первый шаг атакующего — найти цепи, участвующие во взаимодействии со скрытой службой. Для классификации таких цепей используют следующие характеристики: длительность активности; число входящих и исходящих сообщений; последовательность первых 10 сообщений. При этом для классификации цепочки используется дерево решения.

Определив цепочку, атакующий должен получить доступ к входному узлу пользователя, которого он пытается деанонимизировать. Для того чтобы понять, пытается ли пользователь получить доступ к анонимной скрытой службе или к обычному portalу, используется классификатор. Второй классификатор используется, чтобы определить, какую скрытую службу пользователь посетил.

Результаты эксперимента, представленные в работе [28], показали, что данный метод позволяет выявить связь пользователя со скрытым сервисом в 98 % случаев при первой атаке, в 99 % случаев при второй атаке. Кроме того, в 88 % случаев корректно определяется, какую из 50 страниц, за которыми ведется наблюдение, пользователь посетил.

2. Активные методы

Активные методы лежат в основе атак, во время которых атакующий пытается изменить данные или каким-либо другим образом вторгнуться в процессы системы.

2.1. *Timing-атака*

Timing-атака с использованием браузера [29] позволяет наблюдателю выявить часть пользователей Tor, использующих коррумпированный узел и оставивших открытым окно браузера не менее чем на час. Для реализации атаки необходим коррумпированный сервер, входной и выходной коррумпированные узлы.

Выходной узел модифицирует HTTP-трафик, проходящий через него, вставляя невидимый контейнер *iframe*, содержащий JavaScript-код, в запрашиваемые веб-странички. JavaScript-код итеративно контактирует с сервером, посылая уникальный идентификатор, и продолжает работать до тех пор, пока человек оставляет открытой вкладку с зараженной страничкой в браузере. Полная атака работает следующим образом.

1. Атакующий разворачивает необходимые ресурсы: а) вставляет два коррумпированных узла в сеть Tor (входной и выходной); б) разворачивает веб-сервер, который получает и записывает данные, посылаемые JavaScript-кодом.

2. Коррумпированный выходной узел модифицирует весь HTTP-трафик, вставляя туда невидимый JavaScript-код — генератор сигнала, который генерирует уникальный сигнал для каждого клиента Tor.

3. Веб-браузер клиента запускает JavaScript-код, посылая сигнал на сервер.

Этот трафик поступает через клиента Tor, но клиент все еще остается анонимным.

4. Каждые 10 мин ЛП строит новую цепь. ЛП выбирает коррумпированный входной узел (случайно).

5. Атакующий проводит анализ трафика для того, чтобы сравнить сигналы на каждой цепи, поступающие через его входной узел, с разными сигналами, которые принимает веб-сервер. Совпадение сопоставляет клиент Tor с его историей трафика, записанной во время использования коррумпированного выходного узла.

Входному узлу нужно только зарегистрировать проходящий шаблон трафика на каждой цепи, выходной узел нужен только для вставок JavaScript-кода. Существует модификация атаки, использующая только HTML, описание можно найти в работе [29].

Timing-атаки с использованием BGP на уровне AS [30] бывают двух типов.

Анализ трафика через BGP-похищение. Для деанонимизации пользователя наблюдатель может, в первую очередь, применить известные атаки для компрометирования входного узла [18]. Далее наблюдатель может начать атаку префиксного пере-

хвата против префикса, соответствующего найденному входному узлу. Атака позволяет коррумпированной AS увидеть трафик, предназначенный входному узлу, за счет поглощения всего трафика входного узла. Поэтому соединение будет активным только какое-то время, а потом оно будет сброшено. Коррумпированная AS может узнать набор клиентов, ассоциированных с входным узлом для продолжительности времени соединения, через инспекцию IP-заголовков.

Анализ трафика через BGP-прослушивание. Для того чтобы совершить точную деанонимизацию пользователя через анализ трафика, коррумпированная AS может запустить атаку BGP-подслушивания [31]. Эта атака позволяет AS стать промежуточной на пути по направлению к входному узлу, т. е. после перехвата трафик возвращается обратно к нужной точке назначения. Атака позволяет сохранить соединения, оставляя возможность AS точно деанонимизировать клиента через тайминг-анализ.

2.2. *Атаки анализа трафика*

Атака с пометкой ячеек (подтверждающая атака) [32] заключается в том, что атакующий имеет контроль над входным и выходным узлами пользователя. Атака начинается с коррумпированного входного узла. Входной узел выбирает сообщение в TCP-потоке данных и дублирует это сообщение. Исходный IP-адрес сообщения и момент времени дублирования регистрируются. Дублированное сообщение проходит весь путь через цепь и прибывает в выходной узел. Атакующий, управляя выходным узлом, должен засечь дублированное сообщение и записать время, IP-адрес назначения сообщения и порт. Тем самым он подтверждает, что ячейка использует коррумпированные узлы. Таким образом, атакующий устанавливает входные и выходные узлы.

В оригинальной статье про Tor [1] такой класс атак называют *"tagging attack"* (помечающие атаки). Суть в том, что помечается какое-то сообщение, которое потом ищется в потоке данных.

RAPTOR-атака [33] является совершенно новой техникой деанонимизации пользователя с помощью анализа трафика. В качестве наблюдателя здесь используется AS. RAPTOR-атака использует динамические аспекты протокола BGP.

Атака состоит из трех компонент, совместное применение которых дает синергетический эффект. RAPTOR использует асимметричную природу маршрутизации Интернета (BGP-путь от посылающего к приемнику может отличаться от BGP-пути от приемника к посылающему). Эта асимметрия повышает шансы атакующего, обладающего коррумпированной AS, просмотреть хотя бы одно из направлений.

Первый компонент — асимметричный анализ трафика. Данная форма является новой формой ана-

лиза трафика, позволяющей коррумпированной АС деанонимизировать пользователей. Традиционный анализ трафика рассматривает только один сценарий: наблюдатели обозревают трафик от клиента к входному узлу и от выходного узла к веб-серверу. Как правило, пути через Интернет асимметричны, так что путь от выходного узла к веб-серверу может отличаться от пути от веб-сервера к выходному узлу. Возможен и такой вариант: наблюдатель не имеет возможности просматривать трафик на пути от выходного узла к серверу, но может просматривать трафик ТСП-подтверждения доставки на пути от сервера к выходному узлу.

Асимметричный анализ трафика позволяет наблюдателю деанонимизировать пользователей до тех пор, пока наблюдатель может просматривать любое направление трафика на обоих концах соединения. Анализ работает для четырех сценариев: а) трафик данных от клиента к входному узлу и трафик от выходного узла к серверу; б) трафик данных от клиента к входному узлу, трафик ТСП-подтверждения доставки от сервера к выходному узлу; в) трафик ТСП-подтверждения доставки от входного узла к клиенту и трафик данных от выходного узла к серверу; г) трафик ТСП-подтверждения доставки от входного узла к клиенту, трафик ТСП-подтверждения доставки от сервера к выходному узлу. При анализе исследуются поля ТСП-заголовков в наблюдаемом трафике для выявления номера ТСП-последовательности и номер ТСП-подтверждения доставки. Далее вычисляется корреляция между этими полями.

Второй компонент — анализ натуральных перебоев. Путь между клиентом и входным узлом изменяется во времени вследствие физической топологии и политик АС. Такие изменения увеличивают со временем вероятность попадания пользовательских цепей в коррумпированные АС.

Третий компонент — атаки BGP-похищения и BCP-прослушивания. Подробно данные атаки были описаны в работе [31]. Атака BGP-прослушивания позволяет коррумпированной АС стать на пути перед входным узлом, т. е. после перехвата трафик будет возвращаться обратно через входной узел. Такой перехват сохранит соединение и позволит АС провести асимметричный анализ трафика. Кроме того, такая атака позволяет коррумпированной АС деанонимизировать пользователя, посещающего какой-то конкретный сайт. АС видит трафик на клиенте и может запустить атаку BGP-прослушивания против выходного узла.

Результаты экспериментов, приведенные в работе [33] показывают, что атака успешна в 90 % случаев.

2.3. DoS-атаки

Обычные DoS-атаки не требуют глубокого знания Тог и могут быть осуществлены с использованием простых, известных техник [34]. Кроме того,

детальный анализ обычных DoS-атак на Тог можно найти в работе [35]. В 2014 г. вышла работа [36], посвященная атаке, позволяющей израсходовать всю доступную память узла, однако вследствие кооперации авторов с разработчиками Тог атака более не работоспособна, поэтому в настоящей статье не рассматривается.

Атака packet spinning [37] заставляет пользователя выбирать коррумпированные узлы через вывод из строя легитимных узлов. Атакующий строит циклические цепи через сеть Тог и посылает большой объем данных через этот путь, чтобы легитимные узлы были максимально загружены. Атакующий запускает другой набор коррумпированных узлов, которые когда-нибудь будут выбраны пользователями, потому что атакующий перегрузил все легитимные. Атака будет успешной, если инициатор выбирает только коррумпированные узлы для своих цепей, в результате чего деанонимизация становится тривиальной.

Атака перегрузки с использованием длинных путей [38] основана на следующих свойствах Тог: а) маршрутизаторы Тог не вставляют искусственные задержки между запросами; б) IP-адреса всех узлов Тог публично известны и доступны.

Атака предполагает, что атакующий контролирует выходной узел. Узел используется для вставки JavaScript-кода в HTML-ответ на запрос. Код заставляет браузер посылать HTTP-запросы каждую секунду и в ответ на каждый запрос выходной узел посылает пустой ответ, который отвергается браузером. Атакующий записывает интервалы времени периодических запросов, производимых браузером. Так как запросы маленькие, возникает задержка, равная примерно разнице во времени доставки сигнала кодом.

У владельца коррумпированного выходного узла теперь стоит задача вызвать перегрузку узлов, подозреваемых в участии в цепи. Предполагаем, что все узлы являются подозреваемыми, и в самом простом случае атакующий будет итеративно через все узлы проверять, является ли данный узел входным узлом цепи.

Для каждого узла X атакующий конструирует длинную цепь, которая с повторениями включает X в цепь. Вследствие того что Тог сбрасывает цепь при попытке расширения цепи через предыдущий узел, нужно использовать два или более промежуточных узлов для замыкания цепи на X .

Как только цепь становится достаточно длинной (авторы статьи [38] предлагают 24 узла), атакующий использует цепь для передачи данных. Цепь длины m позволит атакующему с пропускной способностью p уменьшить пропускную способность сети Тог на значение $m \cdot p$. Узлу X придется участвовать в $m/3$ дополнительных цепях, что позволит атакующему встраивать большие задержки в конкретный узел.

Если узел X не относится к компрометирующей цепи, измеримые задержки не вызовут заметных изменений во время выполнения атаки. Если X — входной узел, атакующий будет просматривать внедренный шаблон задержки и выявит искомую цель.

CellFlood DoS-атака [39] вместо создания большого числа запросов, на обработку которых требуется мало вычислительных ресурсов, как в обычной DoS-атаке, использует несколько тяжелых запросов создания цепи, которые быстро генерируются атакующим с минимальным количеством ресурсов, однако будут требовать большое количество вычислительных ресурсов от узла, на который идет атака.

Обработка CREATE сообщения занимает в 4 раза больше времени, чем его генерация [40], вследствие криптографических операций, основанных на паре открытый-закрытый ключ, совершаемых во время расширения цепи. Это может быть использовано для всех доступных ресурсов атакуемого узла.

Благодаря архитектуре Тор, узел, на который приходит огромное число CREATE-сообщений, не теряет возможности пересылать сообщения типа RELAY_DATA. Узел, получающий CREATE-сообщения быстрее, чем его процессор может обработать, отвечает на них, посылая DESTROY-сообщения в ответ. Следовательно, узел, находящийся под атакой, будет отклонять запросы от легитимных узлов. Если атака выполняется стратегически, возможна перегрузка большей части сети и выявление цепи, проходящей через конкретные узлы.

Если атакующий заинтересован в том, чтобы исключить узел или набор узлов из сети Тор, ему выгоднее использовать поток CREATE-сообщений, чем обычную и более дорогую DoS-атаку.

Описание некоторых методов не вошло в настоящую работу ввиду большой степени схожести с уже рассмотренными. Другие методы в рамках тематики Тор описывать не имеет особого смысла. К числу таких атак относятся Sybil-атаки [41], которые возможны при внедрении в Тор большого числа коррумпированных узлов, составляющих ощутимую долю узлов Тор. В работе [42] описана комбинированная атака, основанная на анализе трафика, использующая совместно с Sybil-атакой. В работах [42, 43] можно найти атаки на основе анализа трафика с использованием исторических данных о TCP. Ресурс [44] содержит исчерпывающую библиографическую подборку по атакам и защите анонимных систем.

Заключение

В работе были рассмотрены методы, позволяющие успешно провести деанонимизацию пользователей сети Тор. Из рассмотрения методов можно сделать однозначный вывод: атакующему, имеющему большой запас ресурсов, например государственным службам, частным корпорациям и т. д., не

составит большого труда деанонимизировать большое число пользователей сети Тор. Различные комбинации рассмотренных атак можно применить и к деанонимизации пользователей анонимных сетей другого типа, описанных, например, в работе [45].

Список литературы

1. **Dingledine R., Mathewson N., Syverson P.** Tor: The Second-Generation Onion Router [Electronic resource] // Tor project [Official website]. URL: <https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf> (accessed: 8.09.2015).
2. **TorMetrics** [Electronic resource] // Tor project [Official website]. URL: <https://metrics.torproject.org> (accessed: 28.09.2015).
3. **The Russian** government hired people to hack the Tor browser, but they failed and now they're quitting [Electronic resource] // Meduza [Official website]. URL: <https://ineduza.io/en/news/2015/09/09/the-russian-government-hired-people-hack-the-tor-browser-but-they-failed-and-now-they-re-quitting> (accessed: 28.09.2015).
4. **The NSA's** Been Trying to Hack into Tor's Anonymous Internet For Years [Electronic resource] // Gizmodo [Official website]. URL: <http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819> (accessed: 28.09.2015).
5. **Закупка** № 0373100088714000008 [Электронный ресурс] // Государственные закупки [Официальный сайт]. URL: <http://zakupki.gov.ru/epz/order/notice/zkk44/view/comrnnon-intb.html?regNumber=0373100088714000008> (дата обращения: 2.10.2015).
6. **The Rise & Fall of Silk Road** [Electronic resource] // Wired [Official website]. URL: <http://www.wired.com/2015/04/silk-road-1/> (accessed: 28.09.2015).
7. **Danezis G.** Statistical disclosure attacks [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/statistical-disclosure.pdf> (accessed: 1.10.2015).
8. **Kedogan D., Agrawal D., Penz S.** Limits Of Anonymity in Open Environment [Electronic resource] // Springer Link [Official website]. URL: http://link.springer.com/chapter/10.1007%2F3-540-36415-3_4 (accessed: 1.10.2015).
9. **Danezis G.** Statistical disclosure attacks [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/statistical-disclosure.pdf> (accessed: 1.10.2015).
10. **Mathewson N., Dingledine R.** Practical traffic analysis: Extending and resisting statistical disclosure [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/doc/e2e-traffic/e2e-traffic.pdf> (accessed: 1.10.2015).
11. **Agrawal D., Kesdogan D., Penz S.** Probabilistic Treatment of Mixes to Hamper Traffic Analysis [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/agrawal03.pdf> (accessed: 1.10.2015).
12. **Danezis G.** The Traffic Analysis of Continuous-Time Mixes [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/danezis:pet2004.pdf> (accessed: 1.10.2015).
13. **Zhu Y., Fu X., Graham B., Bettati R., Zhao W.** On flow correlation attacks and countermeasures in mix networks [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/flow-correlation04.pdf> (accessed: 1.10.2015).
14. **Levine B. N., Reiter M. K., Wang C., Wright M. K.** Timing attacks in low-latency mix-based systems [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/flow-correlation04.pdf> (accessed: 1.10.2015).
15. **Johnson A., Wacek C., Jansen R., Sherr M., Syverson P.** Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries [Electronic resource] // Aaron Michael Johnson [Official website]. URL: <http://www.ohmygodel.com/publications/usersrouted-ccs13.pdf> (accessed: 2.10.2015).
16. **Murdoch S. J., Danezis G.** Low-Cost Traffic Analysis of Tor [Electronic resource] // UCL-CS [Official website]. URL: <http://sec.cs.ucl.ac.uk/users/smurdoch/papers/oakland05torta.pdf> (accessed: 1.10.2015).

17. **Bauer K., McCoy D., Grunwald D., Kohno T., Sicker D.** Low-Resource Routing Attacks Against Tor [Electronic resource] // University of Washington [Official website]. URL: <https://homes.cs.washington.edu/~yoshi/papers/Tor/wpes25-bauer.pdf> (accessed: 1.10.2015).
18. **Mittal P., Khurshid A., Juen J., Caesar M., Borisov M.** Stealthy Traffic Analysis of Low-Latency Anonymous Communication Using Throughput Fingerprinting [Electronic resource] // Princeton University [Official website]. URL: <http://www.princeton.edu/~pmital/publications/throughput-fingerprinting-ccs11.pdf> (accessed: 1.10.2015).
19. **Cai X., Zhang X., Joshi B., Johnson R.** Touching from a Distance: Website Fingerprinting Attacks and Defenses [Electronic resource] // Stony Brook University [Official website]. URL: <http://www3.cs.stonybrook.edu/~xcai/fp.pdf> (accessed: 1.10.2015).
20. **Wang T., Cai X., Nithyanand R., Johnson R., Goldberg I.** Effective Attacks and Provable Defenses for Website Fingerprinting [Electronic resource] // Centre for Applied Cryptographic Research The University of Waterloo [Official website]. URL: <http://cacr.uwaterloo.ca/techreports/2014/cacr2014-05.pdf> (accessed: 1.10.2015).
21. **Cai X., Nithyanand R., Wang T., Johnson R., Goldberg I.** A Systematic Approach to Developing and Evaluating Website Fingerprinting Defences [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/ccs2014-fingerprinting.pdf> (accessed: 1.10.2015).
22. **Panchenko A., Niessen L., Zinnen A., Engel A.** Website fingerprinting in onion routing based anonymization networks [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/wpes11-panchenko.pdf> (accessed: 1.10.2015).
23. **Experimental Defense for Website Traffic Fingerprinting** [Electronic resource] // Tor project [Official website]. URL: <https://blog.torproject.org/blog/experimental-defense-website-traffic-fingerprinting> (accessed: 1.10.2015).
24. **Wang T., Goldberg I.** Improved Website Fingerprinting on Tor [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/wpes13-fingerprinting.pdf> (accessed: 1.10.2015).
25. **Goldschlag D., Reed M., Syverson P.** Onion Routing for Anonymous and Private Internet Connections. January 28, 1999 [Electronic resource] // Onion Routing [Official website]. URL: <http://www.onion-router.net/Publications/CACM-1999.pdf> (accessed: 8.09.2015).
26. **Chaum D.** Untraceable Electronic Mail, Return Addressed, and Pigital Pseudonyms [Electronic resource] // Free Haven [Official website]. URL: <http://www.freehaven.net/anonbib/cache/chaum-mix.pdf> (accessed: 28.09.2015).
27. **Wiangripanawan R., Susilo W., Safavi-Naini R.** Pesign principles for low latency anonymous network systems secure against timing attacks [Electronic resource] // ACM Pigital Library [Official website]. URL: <http://dl.acm.org/citation.cfm?icH1274553> (accessed: 28.09.2015).
28. **Kwon A., AlSabah M., Lazar D., Dacler M., Devadas S.** Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services [Electronic resource] // USENIX [Official website]. URL: <http://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/kwon> (accessed: 2.10.2015).
29. **Abbot T., Lai K., Lieberman M., Price E.** Browser-Based Attacks on Tor [Electronic resource] // Privacy Enhancing Technologies [Official website]. URL: https://www.petsymposium.org/2007/papers/PET2007_preproc_Browser_based.pdf (accessed: 1.10.2015).
30. **Vanbever L., Li O., Rexford J., Mittal P.** Anonymity on QuickSand: Using BGP to Compromise Tor [Electronic resource] // ACM SIGCOMM [Official website]. URL: <http://conferences.sigcomm.org/hotnets/2014/papers/hotnets-XIII-final80.pdf> (accessed: 1.10.2015).
31. **Ballani H., Francis P., Zhang X.** A study of prefix hijacking and interception in the Internet [Electronic resource] // ACM Pigital Library [Official website]. URL: <http://dl.acm.org/citation.cfm?id=1282411> (accessed: 1.10.2015).
32. **Pries R., Yu W., Fu X., Zhao W.** A New Replay Attack Against Anonymous Communication Networks [Electronic resource] // IEEE Xplore [Official website]. URL: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4533341&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabsall.jsp%3Farnumber%3D4533341> (accessed: 2.10.2015).
33. **Sun Y., Edmundson A., Vanbever L., Li O., Rexford J., Chiang M., Mittal P.** RAPTOR: Routing Attack on Privacy in Tor [Electronic resource] // USENIX [Official website]. URL: <https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-sun.pdf> (accessed: 5.10.2015).
34. **Low orbit ion cannon.** [Electronic resource] // Wikipedia [Official website]. URL: http://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon (accessed: 1.10.2015).
35. **Borisov N., Mittal P., Danezis G., Tabriz P.** Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity [Electronic resource] // Princeton University [Official website]. URL: <http://wfww.princeton.edu/~pmital/publications/dos-ccs07.pdf> (accessed: 1.10.2015).
36. **Jansen R., Tschorsch F., Johnson A., Scheuermann B.** The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network [Electronic resource] // Center for High Assurance Computer Systems [Official website]. URL: <http://www.nrl.navy.mil/itd/chaos/biblio/sniper-attack-anonymously-deanonymizing-and-disabling-tor-network> (accessed: 1.10.2015).
37. **Pappas V., Athanopoulos E., Ioannidis S., Markatos E. P.** Compromising Anonymity Using Packet Spinning [Electronic resource] // FORTH-ICS [Official website]. URL: <http://www.ics.forth.gr/dcs/Activities/papers/torspin.isc08.pdf> (accessed: 2.10.2015).
38. **Evans N. S., Dingleline R., Grothoff C.** A practical congestion attack on tor using long paths [Electronic resource] // ACM Digital Library [Official website]. URL: <http://dl.acm.org/citation.cfm?id=1855771> (accessed: 2.10.2015).
39. **Barbera M. V., Kemerlis V. P., Pappas V., Keromytis A. D.** CellFlood: Attacking Tor Onion Routers on the Cheap [Electronic resource] // Springer Link [Official website]. URL: http://link.springer.com/chapter/10.1007%2F978-3-642-40203-6_37 (accessed: 1.10.2015).
40. **How fast is the RSA algorithm** [Electronic resource] // RSA Laboratories [Official website]. URL: <http://www.rsa.com/rsalabs/node.asp?id=2212> (accessed: 1.10.2015).
41. **Douceur J. R.** The Sybil Attack [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/cache/sybil.pdf> (accessed: 2.10.2015).
42. **Chakravarty S., Barbera M. V., Portokalidis G., Polychronakis M., Keromytis A. D.** On the Effectiveness of Traffic Analysis Against Anonymity Networks Using Flow Records [Electronic resource] // Columbia University [Official website]. URL: <https://mice.cs.columbia.edu/getTechreport.php?techreportID=1545> (accessed: 2.10.2015).
43. **Gilad Y., Herzberg A.** Spying in the Dark: TCP and Tor Traffic Analysis [Electronic resource] // Freenet [Official website]. URL: <http://freehaven.net/anonbib/cache/tcp-tor-pets12.pdf> (accessed: 2.10.2015).
44. **Selected Papers in Anonymity** [Electronic resource] // Freehaven [Official website]. URL: <http://freehaven.net/anonbib/> (accessed: 2.10.2015).
45. **Авдошин С. М., Лазаренко А. В.** Технология анонимных сетей // Информационные технологии. 2016. Т. 22, № 4. С. 284—291.

S. M. Avdoshin, Ph. D., professor, Head of Software Engineering School,
Faculty of Computer Science, HSE, e-mail: savdoshin@hse.ru
A. V. Lazarenko, Undergraduate Student, Software Engineering School,
Faculty of Computer Science, HSE, e-mail: avlazarenko@edu.hse.ru
National Research University Higher School of Economics (HSE)

Tor Users Deanonimization Methods

Tor currently is the biggest anonymous network in the world, which is not only the biggest but the safest one. That is why it is widely used by different criminals such as: killers, drug dillers, carders, etc.. The problem of users deanonimizaiton is considered as a very hard and sophisticated task, because of Tors' design and anonymity-features. However, it is possible to solve this problem. This paper is an overview of Tor users deanonimization methods. It describes passive and active methods, their key features. Passive methods are methods, where an adversary can only analyze traffic in the network, but modifications of it are restricted. Active methods actively use traffic modification for its own purposes. The majority of attacks uses Tor vulnerability, related to end-to-end traffic eavesdropping. In the modern Tor network such an attacks could be successful with a very small probability, because an attacker must control both entry and exit tor nodes of a victim. The probability of such combination is very small. On the other hand, there is a class of attacks, known as website fingerprinting attacks. This attacks are very powerful, if an attacker is smart enough in the field of data science. Moreover, WF attacks requires a particularly small amount of resources for successful usage in the real network. But if an attacker is a global adversary of the network, simple statistical tests will help him to successfully deanonimize all users in the network.

Keywords: anonymous network; deanonimization; PoS; timing attacks; Tor; traffic analysis

References

1. **Pingledine R., Mathewson N., Syverson P.** Tor: The Second-Generation Onion Router [Electronic resource], *Tor project* [Official website], URL: <https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf> (accessed: 8.09.2015).
2. **TorMetrics** [Electronic resource], *Tor project* [Official website], URL: <https://metrics.torproject.org> (accessed: 28.09.2015).
3. **The Russian government hired people to hack the Tor browser, but they failed and now they're quitting** [Electronic resource], *Meduza* [Official website]. URL: <https://meduza.io/en/news/2015/09/09/the-russian-government-hirecl-people-hack-the-tor-browser-but-they-failed-and-now-they-re-quitting> (accessed: 28.09.2015).
4. **The NSA's Been Trying to Hack into Tor's Anonymous Internet For Years** [Electronic resource], *Gizmodo* [Official website], URL: <http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819> (accessed: 28.09.2015).
5. **Zakupka № 0373100088714000008** [Electronic resource], *Gosudarstvennie zakupki* [Official website], URL: <http://zakupki.gov.ru/epz/order/notice/zkk44/view/common-info.html?regNumber=0373100088714000008> (accessed: 2.10.2015).
6. **The Rise & Fall of Silk Road** [Electronic resource], *Wired* [Official website]. URL: <http://www.wired.com/2015/04/silk-road-1/> (accessed: 28.09.2015).
7. **Panezis G.** Statistical disclosure attacks [Electronic resource], *Freehaven* [Official website]. URL: <http://freehaven.net/anonbib/cache/statistical-disclosure.pdf> (accessed: 1.10.2015).
8. **Kedogan D., Agrawal D., Penz S.** Limits Of Anonymity in Open Environment [Electronic resource], *Springer Link* [Official website], URL: http://link.springer.com/chapter/10.1007%2F3-540-36415-3_4 (accessed: 1.10.2015).
9. **Danezis G.** Statistical disclosure attacks [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/cache/statistical-disclosure.pdf> (accessed: 1.10.2015).
10. **Mathewson N., Pingledine R.** Practical traffic analysis: Extending and resisting statistical disclosure [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/doc/e2e-traffic/e2e-traffic.pdf> (accessed: 1.10.2015).
11. **Agrawal D., Kedogan D., Penz S.** Probabilistic Treatment of Mixes to Hamper Traffic Analysis [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/cache/agrawa103.pdf> (accessed: 1.10.2015).
12. **Panezis G.** The Traffic Analysis of Continuous-Time Mixes [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/cache/danezis:pet2004.pdf> (accessed: 1.10.2015).
13. **Zhu Y., Fu X., Graham B., Bettati R., Zhao W.** On flow correlation attacks and countermeasures in mix networks [Electronic resource], *Freehaven* [Official website]. URL: <http://freehaven.net/anonbib/cache/flow-correlation04.pdf> (accessed: 1.10.2015).
14. **Levine B. N., Reiter M. K., Wang C., Wright M. K.** Timing attacks in low-latency mix-based systems [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/cache/flow-correlation04.pdf> (accessed: 1.10.2015).
15. **Johnson A., Wacek C., Jansen R., Sherr M., Syverson P.** Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries [Electronic resource], *Aaron Michael Johnson* [Official website], URL: <http://www.ohmygodel.com/publications/usersrouted-ccs13.pdf> (accessed: 2.10.2015).
16. **Murdoch S. J., Panezis G.** Low-Cost Traffic Analysis of Tor [Electronic resource], *UCL-CS* [Official website], URL: <http://sec.cs.ucl.ac.uk/users/smurdoch/papers/oakland05torta.pdf> (accessed: 1.10.2015).
17. **Bauer K., McCoy D., Grunwald D., Kohno T., Sicker D.** Low-Resource Routing Attacks Against Tor [Electronic resource], *University of Washington* [Official website], URL: <https://homes.cs.washington.edu/~yoshi/papers/Tor/wpes25-bauer.pdf> (accessed: 1.10.2015).
18. **Mittal P., Khurshid A., Juen J., Caesar M., Borisov M.** Stealthy Traffic Analysis of Low-Latency Anonymous Communication Using Throughput Fingerprinting [Electronic resource], *Princeton University* [Official website]. URL: <http://www.princeton.edu/~pmital/publications/throughput-fingerprinting-ccs11.pdf> (accessed: 1.10.2015).
19. **Cai X., Zhang X., Joshi B., Johnson R.** Touching from a Pistance: Website Fingerprinting Attacks and Pefenses [Electronic resource], *Stony Brook University* [Official website], URL: <http://www3.cs.stonybrook.edu/~xcai/fp.pdf> (accessed: 1.10.2015).
20. **Wang T., Cai X., Nithyanand R., Johnson R., Goldberg I.** Effective Attacks and Provable Defenses for Website Fingerprinting [Electronic resource], *Centre for Applied Cryptographic Research The Umyersity of Waterloo* [Official website], URL: <http://cacr.uwaterloo.ca/techreports/2014/cacr2014-05.pdf> (accessed: 1.10.2015).
21. **Cai X., Nithyanand R., Wang T., Johnson R., Goldberg I.** A Systematic Approach to Developing and Evaluating Website Fingerprinting Defences [Electronic resource], *Freehaven* [Official website]. URL: <http://freehaven.net/anonbiD/cache/ccs2014-fingerpnnning.pdf> (accessed: 1.10.2015).

22. **Panchenko A., Niessen L., Zinnen A., Enggl A.** Website fingerprinting in onion routing based anonymization networks [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/cache/wpes11-panchenko.pdf> (accessed: 1.10.2015).
23. **Experimental** Defense for Website Traffic Fingerprinting [Electronic resource], *Tor project* [Official website], URL: <https://blog.torproject.org/blog/experimental-defense-vwebsite-traffic-fingerprinting> (accessed: 1.10.2015).
24. **Wang T., Goldberg I.** Improved Website Fingerprinting on Tor [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/cache/wpes13-fingerprinting.pdf> (accessed: 1.10.2015).
25. **Goldschlag D., Reed M., Syverson P.** Onion Routing for Anonymous and Private Internet Connections, January 28, 1999 [Electronic resource], *Onion Routing* [Official website], URL: <http://www.omon-router.net/Publications/CACM-1999.pdf> (accessed: 8.09.2015).
26. **Chaum D.** Untraceable Electronic Mail, Return Addressed, and Digital Pseudonyms [Electronic resource], *Free Haven* [Official website], URL: <http://www.treehaven.net/anonbib/caclie/chaum-mix.pdf> (accessed: 28.09.2015).
27. **Wiangsripanawan R., Susilo W., Safavi-Naini R.** Design principles for low latency anonymous network systems secure against timing attacks [Electronic resource], *ACM Digital Library* [Official website], URL: <http://dl.acm.org/citation.cfm?id=1274553> (accessed: 28.09.2015).
28. **Kwon A., AlSabah M., Lazar D., Dacler M., Devadas S.** Circuit Fingerprinting Attacks: Passive Deanonimization of Tor Hidden Services [Electronic resource], *USENIX* [Official website]. URL: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/kwon> (accessed: 2.10.2015).
29. **Abbot T., Lai K., Lieberman M., Price E.** Browser-Based Attacks on Tor [Electronic resource], *Privacy Enhancing Technologies* [Official website]. URL: https://www.petsymposium.org/2007/papers/PET2007_preproc_Browser_based.pdf (accessed: 1.10.2015).
30. **Vanbever L., Li P., Rexford J., Mittal P.** Anonymity on QuickSand: Using BGP to Compromise Tor [Electronic resource], *ACM SIGCOMM* [Official website]. URL: <http://conferences.sigcomm.org/hotnets/2014/papers/hotnets-XIII-final80.pdf> (accessed: 1.10.2015).
31. **Ballani H., Francis P., Zhang X.** A study of prefix hijacking and interception in the Internet [Electronic resource], *ACM Digital Library* [Official website]. URL: <http://dl.acm.org/citation.cfm?id=1282411> (accessed: 1.10.2015).
32. **Pries R., Yu W., Fu X., Zhao W.** A New Replay Attack Against Anonymous Communication Networks [Electronic resource], *IEEE Xplore* [Official website], URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4533341&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D453334 1 (accessed: 2.10.2015).
33. **Sun Y., Edmundson A., Vanbever L., Li O., Rexford J., Chiang M., Mittal P.** RAPTOR: Routing Attack on Privacy in Tor [Electronic resource], *USENIX* [Official website]. URL: <https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-sun.pdf> (accessed: 5.10.2015).
34. **Low orbit** ion cannon. [Electronic resource], *Wikipedia* [Official website]. URL: http://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon (accessed: 1.10.2015).
35. **Borisov N., Mittal P., Danezis G., Tabriz P.** Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity [Electronic resource], *Princeton University* [Official website], URL: <http://www.princeton.edu/~pmittal/publications/dos-ccs07.pdf> (accessed: 1.10.2015).
36. **Jansen R., Tschorsch F., Johnson A., Scheuermann B.** The Sniper Attack: Anonymously Deanonimizing and Disabling the Tor Network [Electronic resource], *Center for High Assurance Computer Systems* [Official website], URL: <http://www.nrl.navy.mil/itd/chacs/biblio/sni-per-attak-anonymously-deanonimizing-and-disabling-tor-network> (accessed: 1.10.2015).
37. **Pappas V., Athanasopoulos E., Ioannidis S., Markatos E. P.** Compromising Anonymity Using Packet Spinning [Electronic resource], *FORTH-ICS* [Official website], URL: <http://www.ics.forth.gr/dcs/Activities/papers/torspin.isc08.pdf> (accessed: 2.10.2015).
38. **Evans N. S., Dingledine R., Grothoff C.** A practical congestion attack on tor using long paths [Electronic resource], *ACM Digital Library* [Official website], URL: <http://dl.acm.org/citation.cfm?id=1855771> (accessed: 2.10.2015).
39. **Barbera M. V., Kemerlis V. P., Pappas V., Keromytis A. D.** CellFlood: Attacking Tor Onion Routers on the Cheap [Electronic resource], *Springer Link* [Official website], URL: <http://link.springer.com/chapter/10.1007%2F978-3-642-40203-637> (accessed: 1.10.2015).
40. **How fast** is the RSA algorithm [Electronic resource], *RSA Laboratories* [Official website]. URL: <http://www.rsa.com/rsalabs/node.asp?id=2212> (accessed: 1.10.2015).
41. **Douceur J. R.** The Sybil Attack [Electronic resource], *Freehaven* [Official website]. URL: <http://freehaven.net/anonbib/cache/sybil.pdf> (accessed: 2.10.2015).
42. **Chakravarty S., Barbera M. V., Portokalidis G., Polychronakis M., Keromytis A. D.** On the Effectiveness of Traffic Analysis Against Anonymity Networks Using Flow Records [Electronic resource], *Columbia University* [Official website], URL: <https://mice.cs.columbia.edu/getTechreport.php?techreportID=1545> (accessed: 2.10.2015).
43. **Gilad Y., Herzberg A.** Spying in the Dark: TCP and Tor Traffic Analysis [Electronic resource], *Freener* [Official website]. URL: <http://freehaven.net/anonbib/cache/tcp-tor-pets12.pdf> (accessed: 2.10.2015).
44. **Selected** Papers in Anonymity [Electronic resource], *Freehaven* [Official website], URL: <http://freehaven.net/anonbib/> (accessed: 2.10.2015).
45. **Avdoshin S., Lazarenko A.** Technology of Anonymous Networks, *Informational Technologies*, 2016, vol. 22, no. 4, pp. 284–291.

БАЗЫ ДАННЫХ DATABASE

УДК 004.5

А. С. Коровин, студент, e-mail: akorovin00@gmail.com,

И. П. Скирневский, ассистент, e-mail: igorskir@yandex.ru

Национальный Исследовательский Томский Политехнический Университет, Томск

Система динамической визуализации больших массивов данных сложных физических экспериментов

Описан один из подходов реализации инструмента, позволяющего отображать статус физических экспериментов с помощью различных видов визуализации. Сложность работы системы заключается в обработке большого объема разнородных данных в реальном времени. Для реализации предложено использовать стек веб-технологии, что позволяет применять инструмент удаленно среди широкого круга пользователей, имеющих разные типы устройств. Разработка такой системы была оправдана, так как на сегодняшний день отсутствуют проекты подобного вида, в то время как существующие приложения справляются с задачами неэффективно.

Ключевые слова: система визуализации, большие данные, мониторинг данных, клиент-серверная архитектура, гибкий интерфейс

Введение

В настоящее время сложность организации информационных систем переходит на уровень, позволяющий хранить и обрабатывать все большие объемы данных. В эру достижения огромной производительности и асинхронных вычислений увеличивается актуальность задачи обработки, интерпретации и предоставления конечному пользователю полученных данных.

Зачастую сложность обработки данных возрастает в силу их разнородности в рамках одного проектного решения и динамики обновления. За последнее время проблема больших данных оказалась в фокусе многих крупных компаний, большинство из которых инвестируют средства для проведения исследований проблем, связанных с большими объемами информации. [1]

В работе рассмотрена проблема визуализации данных, полученных в ходе сложных физических экспериментов. В исследовании [2] говорится, что эффективность визуализации данных оказывает прямое влияние на результаты работы конечных пользователей. Интенсивность работы исследовательской группы значительно увеличивается в силу использования актуальных, динамически обновляющихся, структурированных массивов данных с возможностью гибкого изменения визуальных элементов и вывода дополнительной статистики.

Здесь описан один из подходов реализации инструмента, позволяющего создавать и отображать статус физических экспериментов с помощью различных типов визуализации, использующих не только реальные расчетные данные, изменяющиеся с течением времени, но и архивные данные. Решение, приведенное в работе, позволяет в наглядной форме отобразить существенные для исследователя аспекты изучаемых процессов в эксперименте.

Актуальность разработки системы характеризуется несоответствием существующих приложений к требованиям, описанным в данной работе. Все существующие решения не позволяют выполнять работу удаленно, так как являются настольными приложениями, что не соответствует требованию кроссплатформенности и удаленного доступа с любой аппаратной платформы. В свою очередь, веб-приложения, выполняющие схожие функции, не адаптированы под мобильные платформы. Гибкий интерфейс системы визуализации обеспечивает быстрое преобразование (добавление, изменение) экранов отображения, что позволяет добавлять новые модули отображения данных в динамике, адаптируясь к меняющимся условиям физических экспериментов. Таким образом, описанный в работе подход может быть легко масштабируемым для визуализации данных экспериментов любой сложности, что делает его универсальным, а возможность гибкой настройки адаптирует графический интерфейс под конечного пользователя.

1. Область применения системы

В технологическом институте Карлсруэ разрабатывают системы автоматизации для различных физических экспериментов, одним из которых является эксперимент KATRIN (Karlsruhe Tritium Neutrino Experiment) [3], проводимый в целях измерения массы частицы нейтрино, а также эксперимент KITSUBE [4], который направлен на изучение атмосферы, и другие эксперименты. На рис. 1, представлена секция спектрометров, где находится множество сенсоров, с которых собираются данные.

Каждая система автоматизации физического эксперимента имеет набор характеристик, отражающих ее состояние. Подобные характеристики в подавляющем большинстве случаев представляются в различных видах и формах. Данные от эксперимента KATRIN отображаются для конечного пользователя в виде временных рядов, а данные от эксперимента KITSUBE могут быть представлены в виде временных рядов, состоящих из двумерных погодных карт. На рис. 2 (см. четвертую сторону обложки) представлена погодная карта за пятидневный промежуток времени.

Таким образом, выходные массивы данных от подобного рода экспериментов представляют собой трудночитаемые человеком структуры большого объема, поэтому следует выдавать параметры в сжатом виде. Задача визуализации осложняется постоянным изменением данных с течением времени. Неточность в показаниях может привести к неверным вычислениям при дальнейшей аналитике, следовательно, возникает необходимость в постоянной актуализации состояния системы для конечного пользователя.

Доступ к результатам визуализации играет большую роль, так как разные пользователи и группы могут удаленно анализировать данные. Таким образом, необходимо обеспечить доступ к инструменту визуализации всем членам группы: ученым, участвующим в эксперименте, операторам, следящим за ходом работ, и другим участникам команды, в том числе и разработчикам инструмента. Системы с подобным уровнем доступа на сегодняшний день

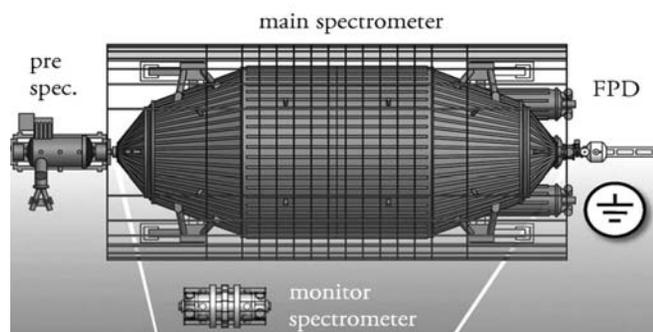


Рис. 1. Секция спектрометров эксперимента KATRIN

реализуются на базе веб-технологий и позволяют удаленно работать с данными с различных платформ, как настольных, так и мобильных.

Растущее число обладателей мобильных устройств порождает спрос на мобильные приложения, в том числе и уже существующих программных продуктов. Специфика таких устройств диктует особые условия визуализации данных, связанные с дизайном, эргономичностью, контентом и другими характеристиками отображения контента. Следовательно, приложение, обеспечивающее визуализацию данных, должно отвечать таким требованиям, как:

- кроссплатформенность;
- оптимизация под мобильные экраны;
- упрощенная и удобная навигация;
- возможность настройки графического интерфейса;
- низкая загрузка передающего канала.

Использование стека веб-технологий для передачи результатов и представления последних пользователям имеет несколько недостатков, относящихся к проблемам производительности, возникающим вследствие ограниченности пропускной способности и задержек в сети. Подробно данная проблема описана в работе [5]. Второй проблемой, связанной с использованием клиент-серверной архитектуры, является уязвимость конфиденциальных данных, хранящихся на серверной стороне, а также возможность перехвата пакетов злоумышленниками во время передачи между клиентом и сервером. Большинство существующих решений, использующихся в Технологическом институте Карлсруэ, не соответствуют требованиям, предъявляемым к гибким системам визуализации данных физических экспериментов, приведенных выше. Следовательно, существующие системы визуализации не способны показывать данные в динамике и не имеют универсальных интерфейсов для адаптации к новым условиям экспериментов. Такие системы недостаточно гибки, что делает процесс внедрения в другие эксперименты ресурсозатратным, а компоненты и визуальные элементы таких систем не оптимизированными к мобильным устройствам.

Следовательно, существует потребность в инструменте, способном хранить, регистрировать и отображать данные с разных экспериментов в различных — как графических, так и текстовых — формах.

Для решения поставленных задач возникает потребность в создании большой библиотеки визуальных элементов отображения. Изменения экранов визуализации и быстрая адаптация к новым условиям экспериментов должны обеспечиваться редактором с графическим интерфейсом.

2. Архитектура приложения

Благодаря увеличению скорости и улучшению качества интернет-соединения, появлению стандарта HTML5 и расширению функциональных возмож-

ностей браузеров, современные веб-приложения по возможностям сравнимы с приложениями для настольных компьютеров. Таким образом, для решения поставленной задачи инструмент реализован с помощью стека веб-технологий.

Такое приложение может быть запущено в браузере любой операционной системы, это уменьшает требовательность к ресурсам аппаратной платформы. Помимо этого, отсутствуют сложности, связанные с обратной совместимостью и поддержкой прошлых версий программ. Пользователям часто приходится решать проблемы обновления установленных копий при появлении новой версия настольного приложения. При использовании веб-приложений такие проблемы отсутствуют, так как существует только одна версия, в которой работают все пользователи.

Другим преимуществом является то, что в веб-приложении используется централизованное хранилище данных. Один из важных аспектов хранения данных — защита от потерь. Это реализуемо только при регулярной репликации баз данных, что неудобно, когда данные распределены среди большого числа компьютеров, так как очень затруднительно создавать и поддерживать все резервные копии проекта. Хранение информации в одном месте позволяет решить эту проблему [6]. Особенно это выгодно в том случае, когда пользоваться приложением будет большое число людей.

Для взаимодействия компонентов распределенной программной системы использовали архитектурный стиль REST. Для его внедрения необходимо наличие клиент-серверной архитектуры, кэширование ответов и унифицированный программный интерфейс. Подробно данный стиль описан в работе [7]. REST-сервис не требует больших затрат при реализации, так как он базируется на уже существующих протоколах HTTP, URI, JSON и RDF. Данный стиль легко масштабируется ввиду возможности включения в каждый ресурс всех состояний, необходимых для обработки запроса. Хорошая производительность достигается за счет кэширования данных и страниц приложения [7].

Для построения интерактивных пользовательских интерфейсов был использован набор технологий AJAX. В работе [8] авторы пришли к выводу, что применение данной технологии приводит к существенной экономии трафика, уменьшению нагрузки на сервер и ускорению реакции интерфейса. Схема взаимодействия клиентской части с серверной с помощью технологии AJAX представлена на рис. 3.

Наличие широкого спектра экспериментов и множества разноплановых групп пользователей системы выдвигает на передний план задачу гибкой настройки инструментов визуализации системы. В рамках решения реализовано отдельное приложение с графическим интерфейсом — редактор страниц визуализации. Этот способ существенно

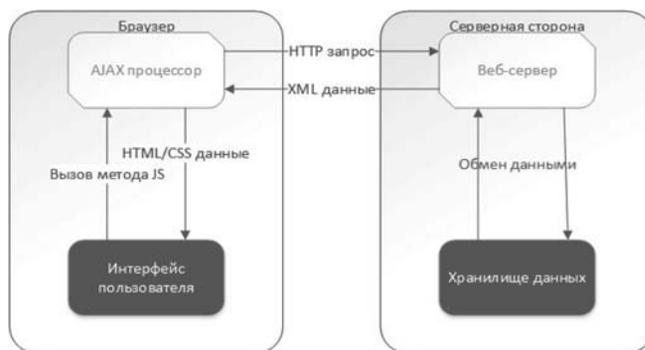


Рис. 3. Схема взаимодействия клиентской и серверной сторон с помощью AJAX

ускоряет разработку и изменение существующих конфигураций страниц без вмешательства специалиста и разработчика продукта.

3. Реализация веб-приложения

Веб-система визуализации реализована как "толстый клиент", где логика приложения перенесена на сторону клиента. В работе [9] говорится, что такой подход может эффективно снизить нагрузку на серверную часть и уменьшить вероятность возникновения задержек сети. Схема работы представлена на рис. 4.

Все сущности элементов визуализации в веб-приложении основаны на моделях и представлениях. Состояние интерфейса меняется с помощью компонента контроллера. Пример схемы работы с элементом визуализации показан на рис. 5, где каждая модель и представление элемента наследуются от базового класса для повышения уровня абстракции и уменьшения дублирования кода. Для хранения,

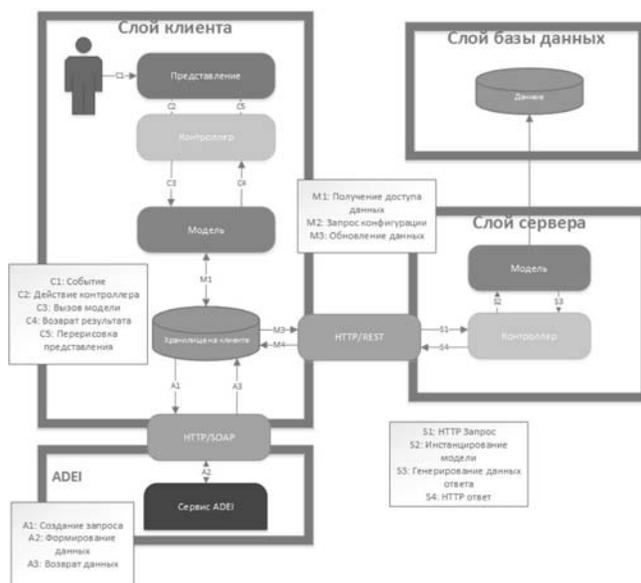


Рис. 4. Архитектура приложения

создания, удаления и управления всеми визуальными элементами использован объект `WidgetManager`.

Контроллер управляет запросами пользователей. В зависимости от типа запроса и информации, содержащейся в нем, контроллер выполняет необходимые действия, управляя ресурсами и объектами. Представление состоит из шаблона, данных сенсоров и кода, который генерирует из шаблона код HTML с помощью элемента конфигурации. Модель — хранилище данных и правила для работы с ними. Модель реагирует на определенные запросы и изменяет свое состояние в зависимости от типа запроса и может содержать такие данные, как размер инструмента визуализации или координаты на экране.

Для хранения и получения данных используется система ADEI. Она предоставляет сервисы для получения многомерных временных рядов с раз-

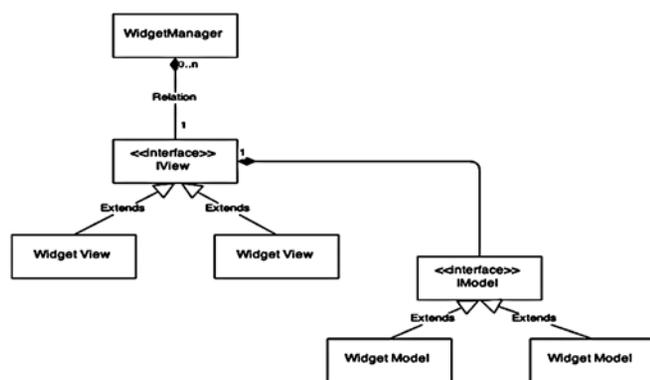


Рис. 5. Схема работы с элементом визуализации

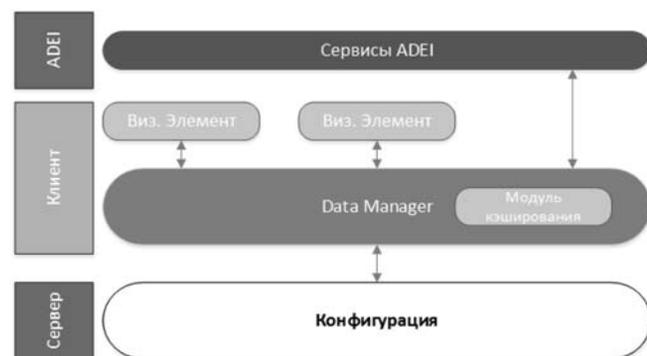


Рис. 6. Схема элементов взаимодействия с данными

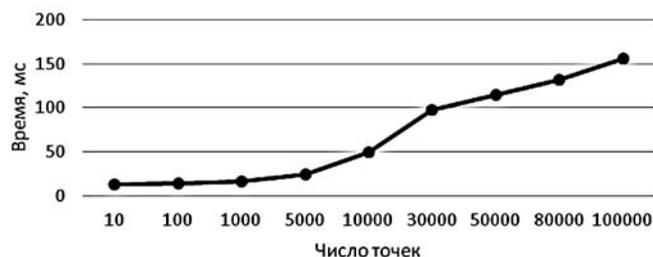


Рис. 7. Время вставки значений в график

личных сенсоров, присутствующих в физических экспериментах. Для получения данных необходимо сформировать запрос для системы ADEI, которая передаст нужные данные веб-приложению [10]. Далее данные кэшируются на клиентской части, чтобы уменьшить нагрузку на сервер. Схема работы с данными представлена на рис. 6, где объект `DataManager` используется для хранения информации об источниках данных и получения данных от сервисов ADEI. При получении данных происходит обновление элементов визуализации.

Одной из главных метрик эффективности приложения является время вставки данных в график, поскольку оно напрямую влияет на время загрузки приложения. На рис. 7 представлены результаты теста времени вставки разного числа точек.

Конфигурация широко используется для описания настроек экранов визуализации. С ее помощью можно систематически идентифицировать, устанавливать связи, сопровождать и управлять различными компонентами системы. На рис. 8 представлен пример структуры конфигурации, содержащей список элементов, описание источников данных и глобальные настройки приложения.

Для передачи данных на сторону клиента используется формат CSV (*comma-separated values*), где значения разделены запятыми. Данный формат выбран ввиду компактности и легкости обработки, поскольку имеет простую структуру. На рис. 9 (см.

```

1 {
2   "screen": {
3     "fluid": false,
4     "glb_sensor_size": 3,
5     "blocksize": 50,
6     "boardsize": 50,
7     "boardsizey": 21
8   },
9   "datasource": {
10    "host": "http://katrin.kit.edu/adei/",
11    "dbgroup": "0",
12    "dbname": "katrin_rep",
13    "server": "fpd"
14  },
15  "elements": {
16    "sensorgroup1": {
17      "type": "sensorgroup",
18      "name": "Magnet Pinch",
19      "size": [6, 5],
20      "coords": [17, 13],
21      "diffsensors": false,
22      "diffgroups": false,
23      "sensors": [{
24        "id": "sensid1",
25        "mask": "102",
26        "unit": "A",
27        "name": "Current",
28        "min": 1.0,
29        "max": 2.0,
30        "precision": 0
31      }, {
32        "id": "sensid2",
33        "mask": "106",
34        "unit": "K",
35        "name": "T1",
36        "precision": 3
37      }
38    ]
39  }
40 }
41

```

Рис. 8. Пример конфигурации экрана визуализации

четвертую сторону обложки) представлена диаграмма сравнения скорости обработки данных объемом в 50 тыс. записей на клиентской части с использованием форматов JSON (*Javascript Object Notation*), XML (*Extensible Markup Language*) и CSV. Результаты показали, что формат CSV обрабатывается быстрее, чем все другие представленные форматы.

4. Графический интерфейс системы

Реализован интерфейс главного экрана эксперимента KATRIN, представленный на рис. 10 (см. четвертую сторону обложки), где отображены результаты визуализации таких данных, как температура главного спектрометра, сила тока, вольтаж детекторов и другие характеристики в реальном времени. Для реализации этого интерфейса была использована идея дизайнерского языка компании Microsoft — Metro, основанная на принципах дизайна швейцарского стиля в типографике.

Разработан интерфейс графического редактора, где пользователь может создать или отредактировать уже созданный экран визуализации эксперимента (рис. 11). Это возможно сделать с помощью как встроенного текстового редактора, так и графического интерфейса, поддерживающего технологию Drag and Drop для удобного изменения координат визуального элемента.

Заключение

Спроектированная система позволяет визуализировать разные виды данных и является важным шагом в процессе разработки прикладных информационно-телекоммуникационных систем, предоставляющим специалистам из разных областей науки возможности надежного и гибкого анализа разнородных данных физических экспериментов. Доступность созданной системы позволяет ученым сконцентрироваться на исследованиях, обеспечивая последних актуальной информацией в удобном формате.

Предложенные архитектура и модель взаимодействия данных могут помочь разработчикам подобных программ реализовывать собственные системы визуализации. В новой версии разрабатываемой системы получения данных ADEI также планируется использовать подобную технологию построения интерфейса и принцип модульности в архитектуре. Разработанное решение будет внедрено в виде основного инструмента визуализации данных таких физических экспериментов, как KATRIN и KITCUBE.

В будущем планируется адаптировать систему к ряду крупных экспериментов, в числе которых

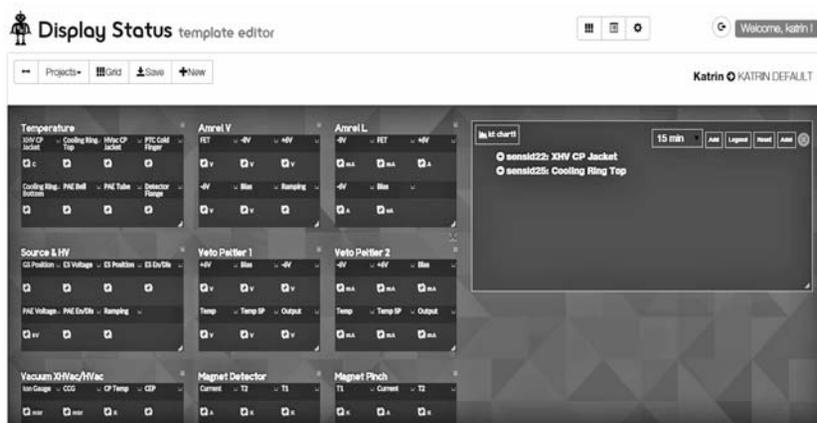


Рис. 11. Интерфейс графического редактора

TOSKA (Toroidal Solenoid Test Facility Karlsruhe) [12], тестирующий сверхпроводимые соленоиды, используемые для исследования реакций термоядерного синтеза, и Edelweiss [13], направленный на изучение частиц темной материи. Кроме того, планируется расширение набора визуальных элементов для обеспечения трехмерной визуализации данных.

Список литературы

1. Riviera J., Van der Meulen R. Gartner Survey Reveals That 73 Percent of Organizations Have Invested or Plan to Invest in Big Data in the Next Two Years [Electronic resource]. URL: <http://www.gartner.com/newsroom/id/2848718> (accessed: 14.03.2015).
2. Plaisant C. The challenge of information visualization evaluation // OSDI Oper. Syst. Des. Implement. 2004. P. 109–116.
3. Wolf J. The KATRIN neutrino mass experiment, Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers // Detect. Assoc. Equip. 2010. Vol. 623. N 1. P. 442–444.
4. Kalthoff N., Bianca A., Andreas W., Martin K. et al. KITcube, A mobile observation platform for convection studies deployed during HyMeX // Meteorol. Zeitschrift. 2013. Vol. 22. N 6. P. 633–647.
5. Andersen D., Bansal D., Curtis D., Seshan S., Balakrishnan H. System Support for Bandwidth Management and Content Adaptation in Internet Applications // Proceeding of the 4th conference on Symposium on Operating System Design & Implementation. 2001. Vol. 4. P. 14.
6. Why Should You Have a Centralized System? | Effective Database Management [Electronic resource]. URL: <http://effectivedatabase.com/resources/why-should-you-have-a-centralized-system/> (accessed: 14.03.2015).
7. Jaki M. Representational State Transfer (REST) // Pro PHP XML and Web Services SE — 17. 2006. P. 633–672.
8. Sghneider F., Agarwal S., Alpan T., Feldmann A. The New Web: Characterizing AJAX Traffic // Lect. Notes Comput. Sci. 2008. Vol. 4979. P. 31–40.
9. Doyle M., Cunningham P. On balancing client-server load in intelligent web-based applications involving dialog. 1999. N 25. P. 1–28.
10. Chilingaryan S., Beglarian A., Kopmann A., Vöcking S. Advanced data extraction infrastructure: Web based system for management of time series data // J. Phys. Conf Ser 2010. Vol. 219, N 4. P. 1–6.
11. Zyp K. AMD: The Definitive Source [Electronic resource]. URL: <http://www.sitepen.com/blog/2012/06/25/amd-the-definitive-source/> (accessed: 14.03.2015).
12. ICPS 2014: Experiments at Campus North [Electronic resource]. URL: http://www.icps2014.com/kit_trip/experiments/ (accessed: 14.03.2015).
13. Edelweiss-III [Electronic resource]. URL: <http://edelweiss.in2p3.fr/> (accessed: 14.03.2015).

A. S. Korovin, Student, e-mail: akorovin00@gmail.com,
I. P. Skirnevskij, Assistant, e-mail: igorskir@yandex.ru,
National Research Tomsk Polytechnic University, Tomsk

Web-Application for Real-Time Big Data Visualization of Complex Physical Experiments

This article examines a tool that provides the flexible user interface for representing status of various physical experiments using different visualization approaches. These experiments produce large volumes of heterogeneous data in real-time, therefore the data processing procedure is complicated. We implemented the tool as a web-based application due to capabilities of modern browsers for interactive visualization, regardless of the operating system and device type (mobile, desktop). Using the architectural style REST simplified component implementation, improved the effectiveness of performance tuning. Frequent changes in experiment conditions require the method for adapting the visualization screens, so we created the graphical user interface builder to arrange graphical control elements using a drag-and-drop technology and WYSIWYG editor. The designed system is a step in the development of applied information technology systems and allows specialists from different fields of science to analyze heterogeneous data of physical experiments effectively.

Keywords: visualization system, big data, data monitoring, client-server architecture, flexible interface

References

1. **Riviera J., Van der Meulen R.** *Gartner Survey Reveals That 73 Percent of Organizations Have Invested or Plan to Invest in Big Data in the Next Two Years* [Electronic resource]. URL: <http://www.gartner.com/newsroom/id/2848718> (accessed: 14.03.2015).
2. **Plaisant C.** The challenge of information visualization evaluation, *OSDI Oper. Syst. Des. Implement*, 2004, pp. 109–116.
3. **Wolf J.** The KATRIN neutrino mass experiment, *Nucl. Instalments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip*, 2010, vol. 623, no. 1, pp. 442–444.
4. **Kalthoff N., Bianca A., Andreas W., Martin K.** et al. KITcube, A mobile observation platform for convection studies deployed during HyMeX, *Meteorol. Zeitschrift*, 2013, vol. 22, no 6, pp. 633–647.
5. **Andersen D., Bansal D., Curtis D., Seshan S. Balakrishnan H.** System Support for Bandwidth Management and Content Adaptation in Internet Applications, *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*, 2001, vol. 4, pp. 14.
6. **Why Should You Have a Centralized System?** [Effective Database Management [Electronic resource]. URL: <http://effectivedata-base.com/resources/why-should-you-have-a-centralized-system/> (accessed: 14.03.2015).
7. **Jakl M.** Representational State Transfer (REST), *Pro PHP XML and Web Services SE — 17*, 2006, pp. 633–672.
8. **Schneider F., Agarwal S., Alpan T., Feldmann A.** The New Web: Characterizing AJAX Traffic, *Lect. Notes Comput. Sci.*, 2008, vol. 4979, pp. 31–40.
9. **Doyle M., Cunningham P.** On balancing client-server load in intelligent web-based applications involving dialog. 1999, no. 25, pp. 1–28.
10. **Chilingaryan S., Beglarian A., Kopmann A., Vöcking S.** Advanced data extraction infrastructure: Web based system for management of time series data, *J. Phys. Conf. Ser.* 2010, vol. 219, no. 4, pp. 1–6.
11. **Zyp K.** AMD: The Definitive Source [Electronic resource]. URL: <http://www.sitepen.com/blog/2012/06/25/amd-the-definitive-source/> (accessed: 14.03.2015).
12. **ICPS 2014: Experiments at Campus North** [Electronic resource]. URL: http://www.icps2014.com/kit_trip/experiments/ (accessed: 14.03.2015).
13. **Edelweiss-III** [Electronic resource]. URL: <http://edelweiss.in2p3.fr/> (accessed: 14.03.2015).

ИНФОРМАЦИЯ

Двенадцатая Конференция CEE-SECR / Разработка ПО Digital October, Москва, 27–28 (29) октября 2016 г.



Разработка ПО/CEE-SECR (www.secr.ru) — ежегодная независимая конференция, собирающая около 1000 участников, среди которых разработчики ПО, исследователи, инженеры-практики, лидеры общественного мнения, предприниматели и инвесторы. За годы работы конференция CEE-SECR завоевала заслуженный авторитет главного профессионального события в индустрии разработки программного обеспечения в России и странах Восточной Европы.

В состав программного комитета конференции CEE-SECR входят эксперты мирового уровня, представляющие как ИТ-индустрию, так и научно-исследовательские организации.

Подробности: <http://2016.secr.ru/>

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИОМЕДИЦИНСКИХ СИСТЕМАХ INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

УДК 004.03

У. О. Агеева, ученица средней школы, **В. Г. Агеева**, руководитель, учитель начальных классов высшей квалификационной категории, МБОУ-СОШ № 1 р. п. Степное Советского района Саратовской области, **А. Б. Барский**, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru, МНИТ, Москва

Бионическое интеллектуальное протезирование конечностей и логические нейронные сети

Анализируются история протезирования и достижения в области протезирования конечностей на базе биомехатроники. Указывается на важность решения задачи непосредственного подключения протеза к мозгу. Для этого предлагается использовать сигналы, формирующиеся на усеченных нервных окончаниях. Распознавание и преобразование этих сигналов в движение протеза предлагается проводить с помощью искусственной однослойной логической сети. Рассматриваются примерный вид такой сети и функция активации нейрона. Нейронная сеть описывается матрицей следования, что обеспечивает малую трудоемкость обработки — расчета величины возбуждения нейронов, а также возможность применения встроенных сигнальных микропроцессоров в качестве нейрокомпьютеров. Эти же микропроцессоры выполняют функции усиления сигналов, настройки параметров и контроля. Значение возбуждения нейрона является параметром, определяющим усилие, с которым выполняется указываемое нейроном движение. Композиция движений, инициируемых нейронами, определяет результирующее действие. Приводится возможная компоновка протеза с учетом считывания сигналов с культи, его энергопитания и настройки параметров нейронной сети.

Ключевые слова: протезирование, биомехатроника, логическая нейронная сеть, функция активации, нейрокомпьютер, интеллектуальный протез

Введение

В 2014 г. в Саратове проводился конкурс "Межрегиональные Мартыновские чтения" [1]. К сожалению, публикация работ не проводилась. Ученица 3-го класса средней школы рабочего поселка Степное Ульяна Агеева, под руководством мамы, представила доклад "Бионические протезы". Работа заняла второе место. В ней детально изложены история и состояние вопроса, а также отражены интенсивные разработки в области интеллектуализации протезов конечностей. Наряду с использованием сигналов от живых мышц культи привлекает внимание идея использования сигналов, приходящих от головного мозга. Распознавание этих сигналов на усеченных нервных окончаниях и их преобразование в команды адекватного движения искусственных механизмов следует считать главной задачей интеллектуального протезирования. При этом сам протез реализует недостающую часть тела: он является как косметическим, так и функциональным дополнением, а не роботом, управляемым на расстоянии. В этом смысл идеального бионического протезирования, и внешне, и функционально воспроизводящего орган в натуральной форме. Сле-

дует отметить, что природа распорядилась о гармонии функционального назначения, принципов работы и внешнего вида. Вскрытием этой гармонии и занимается бионика.

Бионикой называют прикладное направление науки о применении в технических устройствах и системах принципов организации, свойств, функций и структур живой природы.

Люди с давних пор мечтали о возможности восстановления утраченных конечностей подобно некоторым представителям животного мира, способным к удивительному процессу регенерации. К сожалению, во все времена это оставалось лишь мечтой, поэтому взамен утраченных органов люди пытались сделать что-либо похожее на части человеческого тела, их искусственные имитаторы. С древних времен их изготавливали из костей и рогов животных, ветвей и стволов деревьев, тканей и железа. В нынешнем веке протезы превратились в высокотехнологичные устройства, которые дают своему обладателю способности, порой превосходящие возможности обычного человека.

Ниже рассматриваются предположения о возможности применения искусственных логических нейронных сетей для реализации ограниченного

автономного управления протезом по усиленной (при необходимости) совокупности сигналов, образующихся на нервных окончаниях культи.

Протезирование в истории и история протезирования

Первое упоминание о протезировании в истории — побег из плена грека Фемистокла. Посаженный на цепь, он был вынужден отпилить себе ногу, а затем попросить знакомого плотника сделать ему протез.

В декабре 2006 г. археологи обнаружили в Шахри — Сухте древнейший протез глазного яблока, полусферической формы, диаметром чуть более 2,5 см (рис. 1, см. третью сторону обложки). Он выполнен из очень легкого материала, предположительно, битумной пасты. Поверхность искусственного глаза покрыта тонким слоем золота, в центре его выгравирована окружность (изображающая радужку глаза) с золотыми линиями, расходящимися в виде лучей. Женские останки, рядом с которыми обнаружен искусственный глаз, имели рост 1,82 м — намного выше, чем для средней женщины того времени.

С обеих сторон в искусственном глазу были просверлены тонкие отверстия, через которые продевалась золотая проволока, с помощью которой глаз закреплялся в глазнице. Микроскопические исследования обнаружили следы золотой проволоки, а это говорит о том, что искусственный глаз постоянно использовался. Скелет датируется периодом около 2900— 2800 гг. до н. э. По всей видимости, это самый старый из когда-либо найденных глазных протезов.

Считалось, что подобные протезы создавались начиная с V века до н. э. Большинство данных о древней офтальмологии хранилось в Александрийской библиотеке и, к сожалению, они были утрачены во время пожара. Протезы делали из крашеной глины, прикрепляли к полотну и закрепляли его над глазницей. В более позднее время для создания глазных протезов использовали золото, драгоценные камни, серебро, медь и стекло. Первые стеклянные протезы появились в Венеции. Секрет их производства тщательно хранился вплоть до конца XVIII века.

Еще одна уникальная находка из Египта — протез большого пальца ноги, обнаруженный пристегнутым к правому пальцу мумии, опознанной как Табакетенмут, дочери священника, жившей между 950 и 710 гг. до н. э. (рис. 2).

Предполагается, что Табакетенмут потеряла палец из-за гангрены, усугубленной сахарным диабетом.

Протез Табакетенмут сделан из дерева и кожи и даже снабжен специальным шарниром, придающим пальцу подвижность. Археологам известна масса примеров, когда мумиям восполняли недостающие конечности — глаза, носы, чтобы человек



Рис. 2. Большой палец Табакетенмут

мог достойно перейти в мир мертвых. Однако в случае с пальцами, уверяют ученые, можно говорить о том, что эти приспособления были предназначены для использования в этой жизни. Компьютерная модель показала, что эти протезы были в состоянии выдержать до 40 % веса человеческого тела, и сконструированы таким образом, что помогали человеку двигаться вперед. Кроме того, двум добровольцам, у которых ампутированы пальцы ног, предложили примерить точные копии древних протезов, и эти протезы прекрасно себя показали.

Существует большой палец стопы, также египетского происхождения, датируемый 300 г. до н. э. и экспонирующийся в Британском музее Лондона. Он известен как *"Большой палец Гренвилла Грестера"* (по имени человека, который приобрел его в 1881 г. для музея) и сделан из папье-маше, в составе которого был лен, животный клей и гипс. Он тоже имеет следы ношения, однако, как полагают, был косметическим.

Современное протезирование

По мере развития механики, ближе к современности, стали появляться более совершенные типы протезов, хорошо имитирующие потерянную часть тела или даже способные двигаться за счет встроенных механизмов.

В первых трех фильмах эпопеи "Звездные войны" есть множество кадров и целых сцен, сразу же вошедших в историю мирового кинематографа. Одна из таких — момент, когда Люка Скайуокера оснащают высокотехнологическим протезом руки, потерянной в ходе поединка с Дартом Вейдером (рис. 3, см. третью сторону обложки).

Протез выглядит и функционирует неотличимо от натуральной руки и даже, похоже, передает тактильные ощущения, т. е. является почти совершенным бионическим протезом. К сожалению, земной науке до тех технологий, которые применялись "давным-давно, в далекой Галактике", еще идти и

идти, хотя цель поставлена, а идеи компоновки протеза просматриваются вполне реально.

Решающим прорывом в сфере протезирования было появление направления, получившего название "биомехатроника". Принципиальное отличие протезов нового поколения, или "бионических" ("биоэлектрических"), от обычных заключается в их способности регистрировать электрические сигналы, вырабатываемые при сокращении мышц конечностей, к которым они крепятся, и совершать необходимые человеку движения. Функциональность таких протезов может даже превосходить возможности живых рук и ног. Некоторые из них можно дистанционно программировать на выполнение определенных манипуляций. Более того, подбираемый по желанию пациента дизайн искусственной руки или ноги может обеспечивать практически полную маскировку увечья либо привлекать всеобщее внимание к "человеку-киборгу".

Биомехатроника: практика

Посмотрите на известного бегуна (рис. 4), чемпиона параолимпийских игр, бегающего на двух ножных протезах. Он может быть феноменально быстрым. Но в конце гонки он скидывает свои протезы прочь, так как для обычной жизни они непригодны и не обеспечивают всю свободу движений в быту. Ведь человек хочет и бегать, и лазать по скалам, и плавать, и ходить... Возможно ли это делать на протезах?

Вот известный спортсмен использует биопротез, полностью заменяющий конечность от бедра до стопы (рис. 5). Он первый британец, обладатель такого протеза, и до недавнего времени только сумасшедший мог бы сказать, что он сможет свободно кататься на велосипеде, как и все обычные люди.

Однако современный, моторизированный протез не только подарил ему езду на велосипеде. Спортсмен теперь может так же успешно кататься на лыжах, ходить задом наперед и с легкостью подниматься и опускаться по лестнице — роскошь, недоступная обладателям обычных ножных протезов. Большинство протезов используют для области колена шарнирное соединение, а тут протез полностью сделан из углеродного волокна, алюминиевых сплавов и титанов, и в коленной чашечке у него спрятаны микропроцессор и четыре датчика. Датчики посылают в процессор сведения о движении тела, распределении веса и угле наклона. Это значит, что конечность может предвидеть следующее движение владельца и среагировать на него. Аналогично, если человек не двигается, а стоит в одном положении, то датчики блокируют и фиксируют позу.

Существует бионический ручной протез *i-LIMB*, особенностью которого является облегченный алюминиевый корпус и технология, позволяющая совершать более точные движения пальцами (рис. 6). Протез позволяет выполнять огромное число за-



Рис. 4. Чемпион параолимпийских игр



Рис. 5. На велосипеде



Рис. 6. Протез *i-LIMB*

даний, вплоть до завязывания шнурков или застегивания ремня.

Похожий протез в 2011 г. установили 5-летней Тилли Локи. Девочка в состоянии контролировать каждый пальчик своей новой руки. Управление кистью в бионическом протезе осуществляется с помощью специальных датчиков, которые фиксируют электрические сигналы от оставшихся мышц плеча. Бионическая рука позволяет намного лучше контролировать сжатие кисти, что позволило ребенку учиться писать и рисовать.



Рис. 7. Бионический глаз



Рис. 8. Кот Оскар



Рис. 9. Рука SmartHand

Одним из самых важных достижений бионического протезирования, скорее всего, является протез Argus II, известный еще под названием Бионический глаз. В 2007 г. такой протез установили 75-летнему пациенту (рис. 7).

В ходе 4-часовой операции мужчине имплантировали в глаз 60 мельчайших электродов и микрочип. Протез работает через обнаружение света, далее импульсы через тончайшие электроды поступают на искусственную сетчатку глаза. Получаемая картинка еще недостаточно детально, чтобы отказаться от собаки-поводыря. Но больной уже может самостоятельно отсортировать, к примеру, белье для стирки по цвету. На молодых пациентах испытания показывают более успешные результаты. Дети могут видеть и различать буквы и цифры.

Самым последним достижением в области протезирования является разработка технологии остеоинтеграции — вживления протезов в кость. Основными преимуществами данной технологии является отсутствие риска натирания и травмирования культи, а также практически полный контроль над искусственной конечностью. В месте выхода наружу имплантируемые в кости фрагменты таких протезов покрывают специальным пористым материалом, не только имитирующим ткань, обеспечивающую соединение кости и мягких тканей, но и защищающим организм от проникновения инфекций. Пожалуй, самым знаменитым пациентом, быстро бегающим на остеоинтегрированных протезах, является кот Оскар (рис. 8), который, попав под комбайн, потерял обе задние лапы. Единственное удовольствие, которого Оскар лишился навсегда, — это способность лазать по деревьям.

Чудом биомехатроники является искусственная рука SmartHand (рис. 9).

Особенность данного протеза заключается в том, что благодаря работе четырех электродвигателей и 40 датчиков он не только имитирует движения руки человека, но и воспроизводит ощущения от прикосновения к объектам.

Одним из наиболее ярких персонажей, олицетворяющих современный этап протезирования, является родившаяся в 1976 г. американка Айми Маллинс, которой из-за врожденного заболевания в годовалом возрасте пришлось ампутировать обе ноги ниже колена. В студенческие годы девушка добилась выдающихся результатов в соревнованиях по легкой атлетике, где она принимала участие наравне со здоровыми спортсменами. Она также выходила на подиум в качестве модели и снялась в нескольких фильмах. Однако ее известность обусловлена преимущественно тем, что всех своих достижений она добивалась на красивых ножках-протезах (рис. 10).

Культовой личностью для пользователей протезов является американский альпинист, инженер-биофизик, доцент Массачусетского технологиче-



Рис. 10. Протезы Айми Маллинс

ского института Хью Герр, который является владельцем целого арсенала "ног" собственной разработки (рис. 11, см. третью сторону обложки).

В обычной жизни он пользуется протезами со спрятанными в туфлях пружинами из углеродного волокна, которые для утренних пробежек заменяет длинными карбоновыми дугами. У него есть длинные алюминиевые протезы с небольшой резиновой стопой, превращающие его в гиганта ростом 2,1 м, протезы со стопой в виде алюминиевых когтей и клиновидные протезы-ледорубы из полиэтилена.

Во всех рассмотренных выше случаях работает совершенное биоуправление: берется сигнал от мышц тела и превращается в команды для исполнительного устройства.

Новой задачей изобретателей является создание биопротеза, реагирующего на сигналы из мозга инвалидов. Это особенно важно для тех пациентов, которые полностью парализованы и не могут двигаться. Так, в 2009 г. успешно прошли испытания в Японии, во время которых ученые заставили инвалидную коляску двигаться под воздействием сигналов из мозга.

Необходимость и возможность интеллектуализации протезирования конечностей

Дальнейшее развитие интеллектуального протезирования возможно при широком внедрении информационных технологий. Поэтому следующие исследования базируются на важном *предположении о возможности считывания сигналов непосредственно с усеченных нервных окончаний культи*. Образующиеся на них сигналы, формируемые мозгом, действительно теоретически могут быть связаны с управляемым протезом. Однако на деле следует предположить нарушение управляющих связей, искажение той картины управляющих воздействий,

которая способна привести к адекватному сценарию действий протеза. Ведь не следует забывать о так называемых фантомных болях — о проблеме, возникающей, в основном, в результате травмы.

В головном мозге предусмотрена достаточная избыточность: повреждение целых областей мозга после длительных тренировок приводит к нахождению обходных путей распространения возбуждений и даже к новому формированию логических связей с помощью здоровых нейронов. В конечностях такое резервирование отсутствует. Ведь нервы, ведущие к органу тела, являются аксонами некоторых нейронов, находящихся в головном мозге. Конечности не располагают средствами автономного управления, т. е. не располагают фрагментами нейронной сети.

Таким образом, следует исходить из того, что усеченные нервные окончания обеспечивают *нечеткие* исходные *данные*, которые требуют расшифровки для адекватного реагирования протеза. Необходима интеллектуализация протеза, его следует обучить адекватному восприятию искаженных управляющих сигналов, т. е. протез должен обладать интеллектуальными средствами автономного управления.

Вместе с тем, человек в результате длительных тренировок и самообучения также должен приспособиться к управлению протезом.

Возможность применения логических нейронных сетей

В качестве средств искусственного интеллекта (ИИ) при протезировании конечностей предлагается использовать однослойные логические нейронные сети (ЛНС) [6—11].

Напомним, что нейрон, или нейроподобный элемент, независимо от технологии воплощения, является *прибором*, выполняющим пороговую функцию активации вида

$$V = \begin{cases} \frac{\sum_i^K \omega_i V_i}{\sum_i^K \omega_i}, & \text{если это значение не ниже } h; \\ 0 & \text{— в противном случае.} \end{cases} \quad (1)$$

Здесь V — значение возбуждения нейрона; V_i — значение возбуждения рецептора, связанного с данным нейроном, и пришедшее на его i -й вход; ω_i — вес i -й связи (i -го входа); K — число связей (входов) нейрона; h — порог (выбирается экспериментально для устранения излишней "нервозности").

Нейроны, как и вся ЛНС, могут быть реализованы как аппаратно, так и программно. Второй вариант предпочтителен, так как может базироваться на применении встроенного нейрокомпьютера, роль которого может выполнять любой достаточно миниатюрный сигнальный микропроцессор, используемый в бытовой технике. Кроме того, программная реализация нейросети позволяет без ог-

раничений осуществлять переобучение, развитие, оперативный экспериментальный и индивидуальный подбор связей и других параметров.

При программном исполнении нейрону соответствует стандартная процедура, выполняющая функцию (1), а связи нейронов с рецепторами отражаются в рассмотренной ниже матрице следования, что способствует малой сложности и высокой скорости расчетов значений возбуждения нейронов.

На рис. 12 представлен примерный вид ЛНС для протеза кисти руки.

На рецепторный слой ЛНС должны подаваться сигналы с сохранившихся "живых" нервных окончаний культи. При необходимости эти сигналы должны быть усилены (вряд ли ослаблены). Веса $\{\omega\}$, соответствующие степени участия каждого нерва в совершении конкретного движения, определяются экспериментально. При этом нулевое значение связи указывает на разрыв испытываемой связи рецептора с нейроном. (На рисунке веса не распределены по стрелкам, а выделены в одно, не индексированное, множество.)

Связи рецепторов с нейронами на рисунке показаны условно. Они выбираются во многом индивидуально в соответствии с влиянием возбуждения рецептора на требуемую реакцию, например, как частично показано на рисунке. Так строится обученная ЛНС.

Одновременное использование одним нейроном значений возбуждения нескольких рецепторов позволяет осуществлять результирующий комплекс движений подобно "живому" управлению. В то же время должна быть предусмотрена возможность "комплексного" приказа: "захват карандаша", "захват молотка" и др.

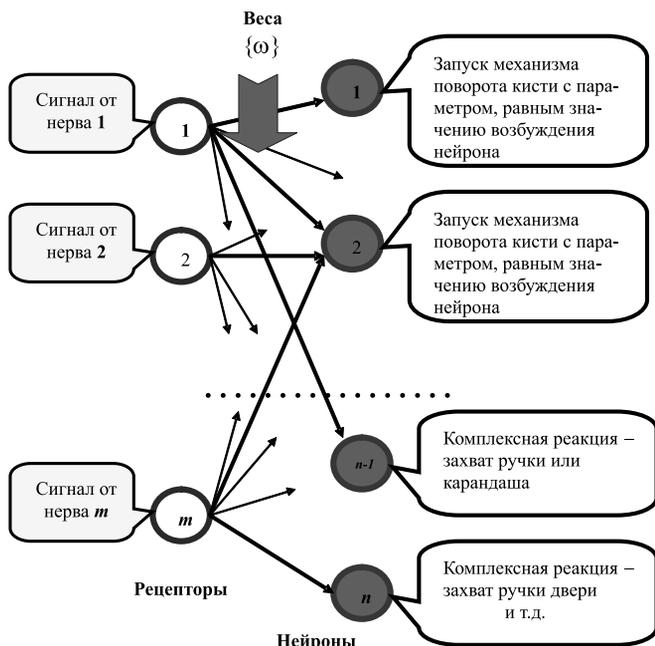


Рис. 12. Примерный вид логической нейронной сети для протеза кисти руки

Напомним, что в "живой" нейронной сети — мозге — функция активации, подобная математической модели (1), реализуется биохимически на основе примерно 240 химических реакций. Веса связей реализуются с помощью *синапсов*, по сути представляющих собой переменные сопротивления. Синапсы на стыках разделяют ветви аксонов (здесь — выходы нейронов-рецепторов, возбуждаемых извне) и дендриты (входы) нейронов, принимающих возбуждение соответствующих рецепторов. Их значение устанавливается в результате обучения для направленного прохождения сигналов в сети. Поэтому связи между нейронами называются *синаптическими*. В результате обучения, вследствие направленного распространения сигналов на основе попутной "подкрутки" синапсов, образуются связи-отношения вида "посылка — следствие", "если — то". Эти отношения являются основой мышления человека, реализуя жизненные функции ассоциативной памяти, логического вывода, распознавания, принятия решений и управления движением органов.

Необходимо помнить, что функционирование человека и его мозга, как и всякой системы автоматического управления, тактируется. Импульсы-сигналы, например, для сокращения мышцы вырабатываются в каждом такте (у уставшая мышца дрожит). Это и позволяет динамически, во времени, менять поведение.

Для микропрограммного выполнения достаточно несложных расчетов матрица следования ЛНС исчерпывающим образом представляется таблицей. В приведенной таблице отражены связи на рис. 1.

Возбуждение каждого нейрона (значение функции активации (1)) находится в результате скалярного умножения строки возбуждения рецепторов на строку, определяющую соответствующее (этому нейрону) движение. Результат делится на сумму весов в данной строке и корректируется при сравнении с порогом.

Значение возбуждения нейрона является значением параметра, определяющего силу или интенсивность выполнения соответствующего движения в совокупности или одновременно с другими движениями протеза, т. е. "физический смысл" каждого параметра заключается в коэффициенте усилия, с которым действует данное движение. Ведь каждое сложное, составное движение определяется сово-

Матрица следования однослойной логической нейронной сети

Рецепторы	1	2	...	m
Значения возбуждения рецепторов нервами	V_1	V_2	...	V_m
Движение 1	$\omega_{(1 \rightarrow 1)}$	$\omega_{(2 \rightarrow 1)}$		
Движение 2	$\omega_{(1 \rightarrow 2)}$	$\omega_{(2 \rightarrow 2)}$		$\omega_{(m \rightarrow 2)}$
...				
Движение n				$\omega_{(m \rightarrow n)}$

купностью сигналов на сокращение определенных мышц с разной силой. В случае протеза — на усилия определенных тяг, пропорциональные уровню сигнала. По множеству усилий отдельных решений складывается составное движение как результат композиции принимаемых решений.

Таким образом, осуществляется целевое, сложное и составное движения на основе множества сигналов. Адекватность движений регулируется экспериментальным подбором связей ω . Здесь кроется искусство алгоритмиста-программиста, призванного заниматься столь благородным делом.

Возможная компоновка интеллектуального протеза

Возможная компоновка интеллектуального протеза кисти руки показана на рис. 13 (сравните с рис. 3). На этом этапе не рассматривается возможность обратной связи, т. е. передачи в мозг тактильных сигналов. Поэтому основными элементами управления являются: датчик сигналов нервных окончаний, нейрокомпьютер, блок питания и дисплей, используемый для настройки и контроля работы протеза. Предполагается, что кристалл микропроцессора обладает достаточным объемом сверхоперативной памяти, необходимой для микропрограммирования. Работа с дисплеем может осуществляться с помощью "мышки". Механическая часть протеза комплексно отработывает команды нейронной сети, поступившие в такте управления в соответствии со значением их параметров.

Существуют сигнальные микропроцессорные вычислительные системы, где на одном кристалле выполнены несколько параллельно работающих процессоров. Там же расположена и сверхоперативная память достаточного объема. Тогда обработка логической нейронной сети может распараллеливаться подобно работе мозга, что резко увеличивает ее быстродействие.

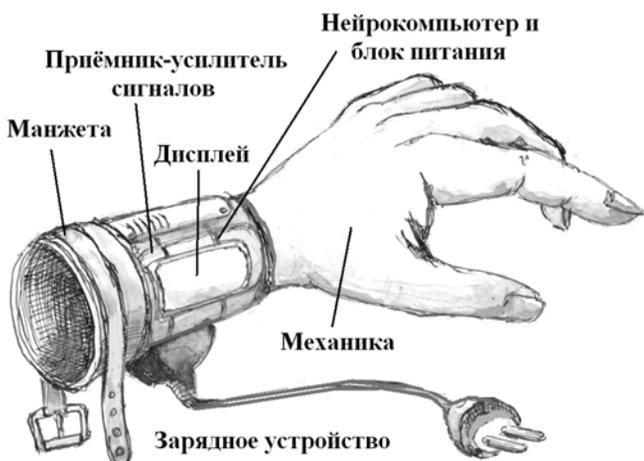


Рис. 13. Возможная компоновка интеллектуального протеза кисти руки

Заключение

Протезы, которые раньше лишь скрывали внешние дефекты, постепенно становятся биомеханическими, реально помогают людям с ограниченными возможностями не только вести обычный образ жизни, но и заниматься спортом. И это в полной мере доказали спортсмены всего мира, участвовавшие в Паралимпийских играх 2014 г.

Успешно решается и задача подключения искусственной конечности непосредственно к мозгу. Ведь команда рождается в мозге, подключить протез к этой команде — это быстрее и естественнее.

Однако не только травмы и болезни приводят к необходимости протезирования. Наш организм стареет, выходят из строя те или иные органы. Многие люди носят протезы органов движения, однако на очереди протезы внутренних органов — печени, сердца. Необходимо научиться непосредственно подключать мозг к управлению протезами разных органов тела.

"Мы уже давно вступили в эпоху антропотехнологических преобразований человека, — считает доктор философских наук Давид Дубровский, главный научный сотрудник Института философии РАН. — Сегодня мы можем протезировать почти все — конечности, внутренние органы. Но где граница? Теоретически можно заменить всего человека. Но человек — это самоорганизующаяся система. Главная задача — это создание самоорганизующейся системы на небиологическом субстрате". Движение "Россия 2045" заявила о таком суперпроекте, где речь идет о создании искусственного человеческого тела и о переносе сознания и психики человека, т. е. собственного "Я", на небиологический субстрат.

Список литературы

1. Агеева У. О. Бионические протезы // Межрегиональные VI Мартыновские чтения. Секция "В мире физико-математических наук". Саратов, 2014. URL: <http://www.openclass.ru/node/4677> 64.
2. Бернштейн Н. А. О построении движений. М.: Медгиз, 1947. 254 с.
3. Донской Е. Д. Н. А. Бернштейн и история отечественной биомеханики // Теория и практика физической культуры. 1996. № 11. С. 4—9. URL: <http://lib.sportedu.ru/press/tpfk>
4. Клиническая биомеханика. Под ред. В. И. Филатова. Л.: Медицина, 1980. 200 с.
5. Александр Р. Биомеханика. М.: Мир, 1970, 341 с.
6. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений. М.: Финансы и статистика. 2006. 175 с.
7. Барский А. Б. Нейронные сети логического вывода. Курс лекций. Saarbücken, Germany: LAP LAMBERT Academic Publishing. 2011. 400 с.
8. Барский А. Б., Дмитриев А. А., Барская О. А. Медицинские информационно-справочные системы на логических нейронных сетях // Информационные технологии. Приложение. 2010. № 1. 32 с.
9. Барский А. Б., Милютин Л. Б., Тимофеев А. Е. Реагирующие объекты для систем интеллектуального отображения // Информационные технологии. 2007. № 2. С. 2—11.
10. Вахромцев Е. М. Применение методов "живого" моделирования для производственного мониторинга // Информационные технологии. 2010. № 7. С. 10—15.
11. Барский А. Б. "Живое" моделирование и объемная экранизация // Информационные технологии. 2015. № 11. С. 863—870.

U. O. Ageeva, Pupil, V. G. Ageeya, Teacher, School № 1 Saratov Region,
A. B. Barskiy, Professor, e-mail: arkbarsk@mail.ru,
Moscow State University of Railway Engineering (MIIT)

Bionic Intelligence Limbs Prosthetic and Logical Neural Networks

Analyzes the history of prosthetics and advances in prosthetic of limbs based on the bio-mechatronics. It emphasizes the importance of solving the problem of the direct connection of the prosthesis to the brain. For this proposed use signals formed on the truncated nerve endings. Recognition and conversion of these signals into movement of the prosthesis is offered to make with the help of an artificial single-layer logical network. We consider the approximate form of a network and the activation function of the neuron. A neural network is described by the following matrix, which provides a low complexity of processing — calculation of the value of the excitation of neurons, as well as the ability to use built-in signal processors as neurocomputers. These micro-processors serve as signal amplification, and control settings. The value of the excitation of the neuron is a parameter that determines the amount of force with which the movement is performed, at which the neuron indicates. The composition of movements triggered neurons determines the resulting effect. The possible layout of the prosthesis considering reading signals from the stump, its power supply and configuration of the neural network.

Keywords: prosthetics, bio-mechatronics, logical neural network, activation function, neurocomputer, intelligent prosthesis

References

1. Ageeva U. O. Bionicheskie protezy, *Mezhregional'nye VI Martynovskie chteniya. Sekcija "V mire fiziko-matematicheskikh nauk"*. Saratov, 2014. URL: <http://www.openclass.ru/node/467764>
2. Bernshtejn N. A. *O postroenii dvizhenij*. Moscow: Medgiz, 1947. 254 p.
3. Donskoj E. D. N. A. Bernshtejn i istorija otechestvennoj biomehaniki, *Teorija i praktika fizicheskoj kul'tury*, 1996, no. 11, pp. 4—9. URL: <http://lib.sportedu.ru/press/tpfk>
4. *Klinicheskaja biomehanika*. Pod red. V. I. Filatova. Leningrad: Medicina, 1980. 200 p.
5. Aleksander R. *Biomehanika*. Moscow: MTR 1970. 341 p.
6. Barskiy A. B. *Nejronnye seti: Paspoznavanie, upravlenie, priyatye reshenij*. Moscow: Finansy i statistika. 2006. 175 p.
7. Barskiy A. B. *Nejronnye seti logicheskogo vyvoda*. Kurs lekcij. Saarbrücken, Germany: LAP LAMBERT Academic Publishing. 2011. 400 p.
8. Barskiy A. B., Dmitriev A. A., Barskaya O. A. Medicinskie informacionno-spravochnye sistemy na logicheskikh nejronnyh setyah, *Informacionnye tehnologii. Prilozhenie*, 2010, no. 1, 32 p.
9. Barskiy A. B., Miljutin L. B., Timofeev A. E. Reagirujushhie ob'ekty dlja sistem intellektual'nogo otobrazhenija, *Informacionnye tehnologii*, 2007, no. 2, pp. 2—11.
10. Vahromcev E. M. Primenenie metodov "zhivogo" modelirovanija dlja proizvodstvennogo monitoringa, *Informacionnye tehnologii*, 2010, no. 7, pp. 10—15.
11. Barskiy A. B. "Zhivoe" modelirovanie i ob'yomnaja ekranizacija, *Informacionnye tehnologii*. 2015, no. 11, pp. 863—870.

УДК 004.891.3

А. Е. Прасолова, канд. техн. наук, зав. каф., e-mail: prasolova@bsu.edu.ru
Старооскольский филиал Белгородского государственного национального
исследовательского университета, г. Старый Оскол, Белгородская область, Россия

Коллективный нейросетевой алгоритм диагностики инфаркта миокарда

Приведено описание нейросетевой системы диагностики инфаркта миокарда. Описано применение нейронной сети Хемминга, РБФ-сети, самоорганизующихся карт Кохонена, а также коллективного нейросетевого алгоритма в диагностике инфаркта миокарда при использовании лишь данных ЭКГ. Представлены результаты вычислительного эксперимента.

Ключевые слова: нейронная сеть, диагностика инфаркта миокарда, коллективный нейросетевой алгоритм, сеть Хемминга, сеть Кохонена, РБФ-сеть

Введение

Заболевания сердечно-сосудистой системы, в том числе инфаркт миокарда различной локализации, в настоящее время являются самыми распространенными в мире. По данным статистических исследований, инфаркт миокарда возникает у каждой

двадцать пятой женщины и у каждого десятого мужчины. Заболеваемость инфарктом миокарда составляет 36,3 случаев на каждую 1000 человек. Тем не менее до сих пор не существует точных методов диагностики. На электрокардиограмме не всегда содержатся явные признаки инфаркта мио-

карда. Согласно статистике, врач правильно диагностирует данное заболевание у 88 % больных и ставит этот диагноз ошибочно в 29 % случаев. Необходимо пояснить, в каких параметрах оценивается качество диагноза в общем случае. Предположим, что из десяти человек, у которых инфаркт действительно есть, диагностический метод позволяет обнаружить заболевание у восьми. Тогда чувствительность метода составит 80 %. Если же мы возьмем десять человек, у которых инфаркта нет, а метод диагностики заподозрит его у трех человек, то доля ложных тревог составит 30 %, при этом дополнительная к нему характеристика — специфичность метода — будет равна 70 %.

Как видно из приведенных данных, специфичность диагностики врачом слишком низкая, т. е. чересчур много гипердиагностики. Различные методы обработки данных в целях повышения качества диагностики данного заболевания применяют не одно десятилетие, однако лучшие из них позволяют сократить число случаев гипердиагностики не более чем на 3 % [1].

Нейросетевые алгоритмы хорошо зарекомендовали себя в задачах обработки многомерных данных [2—4]. В работе [2] описано применение в решении данной задачи искусственной нейронной сети Хемминга.

Постановка задачи

В данной работе описано применение нейронных сетей разной конфигурации в диагностике инфаркта миокарда различной локализации при использовании лишь данных ЭКГ. Были рассмотрены следующие шесть параметров: размер зубца Q , наличие зубца R , значение интервала QRS , положение и наклон интервала ST относительно изолинии, размер и положение зубца T относительно изолинии (рис. 1).

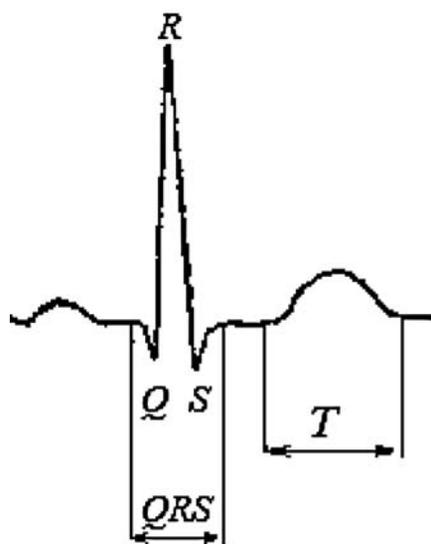


Рис. 1. Зубцы и интервалы ЭКГ

Цель работы — оценить чувствительность и специфичность коллективного нейросетевого алгоритма в задаче диагностики инфаркта миокарда при малом числе заданных параметров, сравнить экспериментальные данные, характеризующие коллективный нейросетевой алгоритм, с результатами, полученными для каждого нейросетевого алгоритма, являющегося структурной частью данного алгоритма, а также с известными результатами подобных исследований.

В исследовании были использованы данные ретроспективного обследования больных кардиологического отделения больницы скорой медицинской помощи г. Курска и Курской областной клинической больницы с клинически подтвержденным инфарктом миокарда различной глубины и локализации (40 чел.) и больных, у которых по данным обследования инфаркт миокарда был исключен (40 чел.). Таким образом, объем контрольной выборки составил 80 человек.

В ходе исследования использовали электрокардиограф SCHILLER CARDIOVIT AT-101 и данные стандартной 12-канальной электрокардиографии. В стандартную комплектацию данного кардиографа входит программное обеспечение (Software M), которое применяли для цифровой обработки. В результате были получены значения требуемых зубцов и интервалов. Эти значения подавали на вход нейросетевого программного комплекса диагностики. Разработанный программный комплекс ставил диагноз: наличие или отсутствие инфаркта миокарда в соответствующем отделении.

В ходе работы было проведено исследование показателей качества диагностики инфаркта миокарда с помощью нейронной сети Хемминга, сети с радиально-базисной функцией активации (РБФ-сети), самоорганизующихся карт Кохонена и коллективного нейросетевого алгоритма. Были рассмотрены вероятность правильной диагностики, чувствительность и специфичность алгоритмов как качественные параметры диагностики. Проведен анализ зависимости рассмотренных параметров от объема обучающей выборки. Объем контрольной выборки — 80 ЭКГ для всех параметров.

Нейросетевые алгоритмы

На рис. 2 представлена структурная схема сети Хемминга.

Обучение данной нейронной сети сводится к расчету весовых коэффициентов нейронов первого слоя. Весовые коэффициенты полагают равными следующим значениям [5]:

$$w_{ik} = \frac{x_i^k}{2}, \quad i = 0 \dots n - 1, \quad k = 0 \dots m - 1,$$

где x_i^k — i -й элемент k -го образца. В нашем случае x_i^k рассчитывали исходя из данных обучающей выборки, как выборочные средние.

На рис. 3 представлена структурная схема самоорганизующихся карт Кохонена. При обучении данной нейронной сети использован метод последовательных приближений. На вход сети последовательно подаются обучающие примеры. В ходе этого процесса определяется нейрон, у которого минимальны значения скалярного произведения весов и входного вектора. Данный нейрон объявляется победителем и при подстройке весов у соседних нейронов является центром. Расчет проводим по формуле [6]:

$$w_n = w_c + \alpha(x - w_c).$$

Такое обучающее правило реализует "соревновательное" обучение. Для каждого примера из обу-

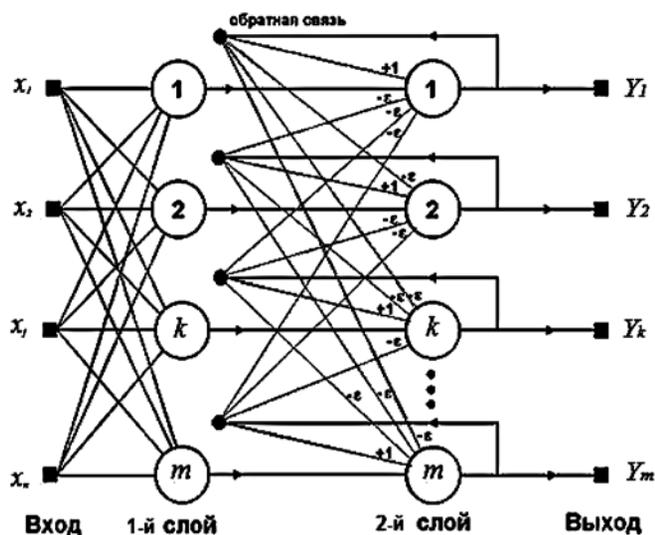


Рис. 2. Структурная схема сети Хемминга

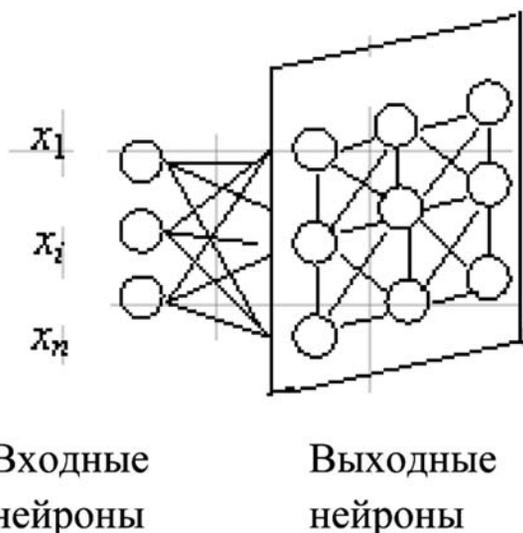


Рис. 3. Структурная схема самоорганизующихся карт Кохонена

чающей выборки основной итерационный алгоритм Кохонена последовательно проходит ряд эпох. Каждый нейрон в процессе обучения наделяется "чувством справедливости". Нейрон временно увеличивает свой порог, если он становится победителем чаще, чем примерно в $1/k$ раз (где k — число нейронов Кохонена), что уменьшает его шансы на победу. Таким образом другие нейроны получают возможность обучаться.

На рис. 4 представлена структурная схема РБФ-сети. Обучение ячеек данной нейронной сети происходит путем подбора центра и отклонения каждой из них. Центры ячеек t_j рассчитывают как выборочное среднее. Параметр разброса для каждой ячейки σ определяется как половина расстояния до ближайшего центра ячейки, соответствующей другому классу. Выходной слой РБФ-сети обучался по методу обратного распространения ошибки.

В ходе обучения нейронной сети Хемминга потребовалось настроить 12 весовых коэффициентов нейронов первого слоя, в ходе обучения РБФ-сети — четыре весовых коэффициента нейронов выходного слоя и центры и отклонения двух РБФ-ячеек. В ходе обучения нейронной сети на базе карт Кохонена потребовалось рассчитать 12 весовых коэффициентов.

Схема окончательной постановки или исключения диагноза инфаркта миокарда по данным электрокардиографии осуществляется коллективным нейросетевым алгоритмом, схема которого представлена на рис. 5.

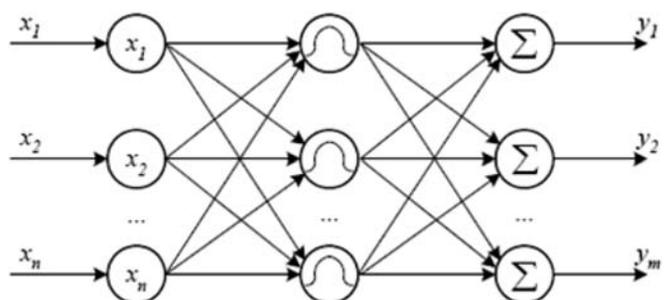


Рис. 4. Структурная схема РБФ-сети

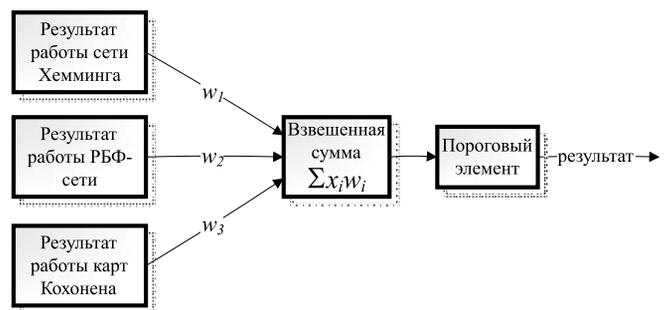


Рис. 5. Схема коллективного нейросетевого алгоритма окончательной постановки или исключения диагноза инфаркта миокарда

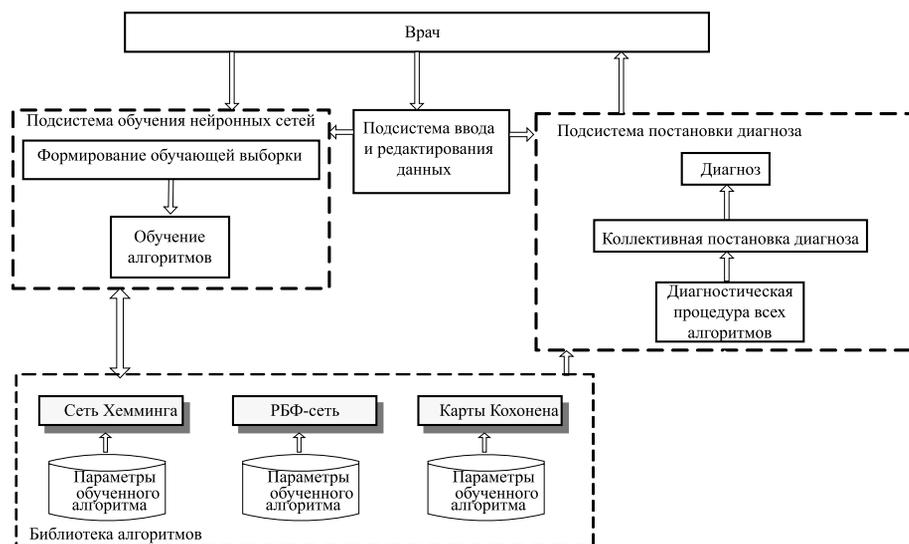


Рис. 6. Функциональная схема программного комплекса диагностики

В ходе исследования был разработан программный комплекс диагностики, функциональная схема которого представлена на рис. 6.

Диагностирующий блок реализует алгоритм на базе рассмотренных нейронных сетей с учетом их коэффициентов достоверности, приведенных в табл. 1. Коэффициенты достоверности рассчитывали как среднюю вероятность правильного диагностирования. Таким образом, решение о постановке или исключении диагноза инфаркта миокарда коллективный нейросетевой алгоритм принимает с учетом диагнозов, поставленных всеми рассмотренными алгоритмами.

Результаты вычислительного эксперимента, в ходе которого определялась зависимость вероятности правильного диагностирования от объема обучающей выборки, представлены в табл. 2.

Полученные результаты

В табл. 3 представлены чувствительность и специфичность различных методов диагностики: нейронная сеть Хемминга, РБФ-сеть, карты Кохонена — наши результаты (объем обучающей выборки — 20, контрольной выборки — 80), врач (данные статистических исследований), многослойный персептрон (результаты, полученные Уильямом Бакстом [1]).

Заключение

Полученные результаты иллюстрируют множество особенностей в поведении качественных параметров рассмотренных нейросетевых алгоритмов при увеличении объема обучающей выборки. В ходе эксперимента искусственная нейронная сеть Хемминга сталкивается с проблемой высокой специфичности метода, как и врач. Однако сеть при увеличении чувствительности не теряет специфичности. В ходе эксперимента РБФ-сеть и самоорганизующиеся карты Кохонена не изменяли показатели качества диагностики при увеличении объема обучающей выборки. Однако их показатели качества все равно оказались выше показателей качества диагностики, демонстрируемой врачом. Самое высокое значение вероятности правильного диагностирования продемонстрировал коллективный нейросетевой алгоритм, однако он показал меньшую чувствительность, чем сеть Хемминга и карты Кохонена, но большую специфичность. Коллективный нейросетевой алгоритм продемонстрировал большую чувствительность, но меньшую специфичность в сравнении с РБФ-сетью.

Однако рассмотренные виды нейронных сетей имеют некоторые существенные недостатки. Так, в алгоритмах обучения всех рассмотренных нейро-

Таблица 1

Коэффициенты достоверности нейросетевых алгоритмов

Алгоритм	Коэффициент достоверности
Сеть Хемминга	0,87
Карты Кохонена	0,83
РБФ-сеть	0,9

Таблица 2

Зависимость вероятности правильного диагностирования от объема обучающей выборки

Алгоритм	Объем обучающей выборки					Средняя
	8	10	12	16	20	
Сеть Хемминга	0,79	0,85	0,861	0,919	0,93	0,87
РБФ-сеть	0,9	0,9	0,9	0,9	0,9	0,9
Карты Кохонена	0,83	0,83	0,83	0,83	0,83	0,83
Коллективный нейросетевой алгоритм	0,93	0,93	0,93	0,93	0,93	0,93

Таблица 3

Чувствительность и специфичность методов

Алгоритм	Чувствительность, %	Специфичность, %
Сеть Хемминга	95	91
РБФ-сеть	80	100
Карты Кохонена	95	71
Коллективный нейросетевой алгоритм	89	97
Персептрон (результаты Бакста)	92	96
Врач (данные статистики)	88	71

сетевых алгоритмов нет правил для изменения числа элементов сети в процессе обучения. Может возникнуть ситуация, когда число нейронов и связей между ними, заданное первоначально, является недостаточным для достижения требуемого качества обучения. Также в процессе обучения сети Хемминга и РБФ-сети изменяются параметры всех элементов. В результате при увеличении числа нейронов вычислительные затраты на обучение также увеличиваются. К недостаткам рассмотренных сетей можно отнести невозможность извлечь алгоритм решения задачи, т. е. извлечь знания из данных. Также в рассмотренных нейросетевых структурах число принимаемых решений закладывается единожды на этапе проектирования сети и не может неограниченно расширяться.

Исходя из сказанного, представляется целесообразным рассмотреть возможность применения в задаче диагностики логических нейронных сетей, лишенных указанных недостатков, и сравнить по-

казатели качества диагностики нейронных сетей данного класса с уже рассмотренными алгоритмами. Построение комплекса диагностики на базе логических нейронных сетей можно считать задачей на будущее.

Список литературы

1. **Ежов А. А., Чечёткин В. Р.** Нейронные сети в медицине // Открытые системы. 1997. № 4. С. 34–37.
2. **Прасолова А. Е., Прасолов А. В.** Искусственная нейронная сеть Хэмминга в приложении к диагностике инфаркта миокарда // Нейрокомпьютеры: разработка, применение. 2005. № 7. С. 41–45.
3. **Прасолова А. Е.** Нейросетевые и статистические алгоритмы в задаче обнаружения сигналов // Телекоммуникации. 2010. № 2. С. 2–6.
4. **Прасолова А. Е.** Коллективный нейросетевой алгоритм распознавания изображений // Телекоммуникации. 2010. № 10. С. 22–28.
5. **Короткий С.** Нейронные сети Хопфилда и Хэмминга // http://lii.newmail.ru/NN/KOROTKY/N4/kor_nn4.htm.
6. **Kohonen T.** *Self-organization and associative memory* // Berlin, Heidelberg; New York, Tokyo: Springer Verlag, 1984. 255 p.

A. E. Prasolova, Chief of the Department of Economics,

Computer Science and Mathematics, e-mail: prasolova@bsu.edu.ru

Stary Oskol Branch of Belgorod National Research University, Stary Oskol, Belgorod region, Russia

Collective Neural Network Algorithm for the Diagnosis of Myocardial Infarction

The article describes a neural network diagnostic system of myocardial infarction. The application of Hamming neural network, RBF network, Kohonen self-organizing maps, and collective neural network algorithm in the diagnosis of myocardial infarction using only ECG data.

Objective — to evaluate the sensitivity and specificity of the collective neural network algorithm in the problem of diagnosing myocardial infarction in a small amount of the specified parameters. Compare the experimental data with the results obtained for each neural network algorithm, which is a structural part of the algorithm, as well as with the known results of such studies. The results of computational experiments are represented. This work describes the use of neural networks of various configurations in the diagnosis of myocardial infarction by using different localization only ECG data. We considered the following 6 parameters: the size of the Q-wave, the presence of the R-wave, the value of QRS interval, the position and the slope of the ST interval relative contour, size and position relative to the T wave contour. In a retrospective study of patients examination cardiology department with clinically confirmed myocardial infarction with different depths and localization (40 pers.) And patients who according to a survey of myocardial infarction was excluded (40 pers.) were used. Thus, the volume of the control sample consisted of 80 people.

Keywords: neural network, diagnosis of myocardial infarction, collective neural network algorithm, Hamming network, Kohonen network, RBF network

References

1. **Ezhov A. A., Chechetkin V. R.** Neironnye seti v medicine (Neural networks in medicine), *Otkrytye sistemy*, 1997, no. 4, pp. 34–37.
2. **Prasolova A. E., Prasolov A. V.** Iskusstvennaja nejronnaja set' Hjemminga v prilozhenii k diagnostike infarkta miokarda (Artificial hamming neural network in application to myocardial infarction diagnostics), *Nejrokomputery: razrabotka, primeneniye*, 2005, no. 1, pp. 41–45.
3. **Prasolova A. E.** Nejrosetevye i statisticheskie algoritmy v zadache obnaruzheniya signalov (Neural network and statistical algorithms in problem of signal detection), *Telekommunikacii*, 2010, no. 2, pp. 2–6.
4. **Prasolova A. E.** Kollektivnyj nejrosetevoj algoritim raspoznavaniya izobrazhenij (Collective neural network images recognition algorithm), *Telekommunikacii*, 2010, no. 10, pp. 22–28.
5. **Korotkij S.** *Neironnye seti Hopfilda i Hjemminga*, URL: http://lii.newmail.ru/NN/KOROTKY/N4/kor_nn4.htm.
6. **Kohonen T.** *Self-organization and associative memory*, Berlin; Heidelberg; New York; Tokyo: Springer Verlag, 1984, 255 p.

УДК 519.68:15:681.5

В. М. Гриняк, канд. техн. наук, доц., e-mail: victor.grinyak@gmail.com,
Дальневосточный федеральный университет,
Ю. С. Иваненко, студент,
Владивостокский государственный университет экономики и сервиса,
А. С. Девятисильный, д-р техн. наук, гл. науч. сотр.,
Институт автоматизи и процессов управления ДВО РАН

Погрешность измерения координат компьютеризированными РЛС, обусловленная скоростью передачи данных в распределенных информационных системах*

Статья посвящена проблеме построения цифровых систем радиолокационного наблюдения. Выдвигается тезис о том, что передача данных между компонентами распределенной системы обработки данных РЛС является важным фактором, ограничивающим точность измерений координат наблюдаемых объектов. Приведены данные натурного эксперимента по определению скорости передачи данных между двумя типовыми ЭВМ общего назначения, объединенных в сеть с помощью стандартного оборудования. На основе этих данных проведены расчеты возможных значений погрешностей измерений, обусловленных ограничениями на частоту и диапазон дискретизации РЛС-сигнала. Сделан вывод о том, что скорость передачи данных в распределенной информационной системе достаточна для измерения координат наблюдаемых объектов с точностью 5...10 м.

Ключевые слова: распределенная информационная система, радиолокационная станция, цифровая обработка сигнала, передача данных, погрешность измерения

Введение

Радиолокационные системы наблюдения являются важнейшим объектом приложения достижений современных информационных технологий и играют значительную роль в обеспечении контроля за движением объектов различного типа [1—3]. Несмотря на развитость современных спутниковых средств навигации (например, автоматическая идентификационная система (АИС) в судовождении [4]), именно радиолокационные инструменты продолжают составлять информационную основу систем наблюдения за морским и воздушным движением [5, 6], так как способны обеспечить бесперебойность и автономность их функционирования. Например, береговые и бортовые системы управления движением судов обязательно включают в себя одну или несколько радиолокационных станций (РЛС) кругового обзора [4].

В настоящее время характерна компьютеризация РЛС, состоящая в их сопряжении с ЭВМ с помощью специальных аналого-цифровых устройств

(радар-процессоров [7]). Радиолокационный отраженный эхосигнал оцифровывается и "вводится" в память ЭВМ (первичная обработка сигнала). Получаемые таким образом первичные цифровые данные передаются другим компонентам информационной системы (как правило, распределенной) для вторичной обработки.

Аналого-цифровое преобразование исходного эхосигнала характеризуется частотой и диапазоном его дискретизации. Высокая частота и большой диапазон дискретизации очень желательны, так как позволяют достигать высокой точности измерения координат наблюдаемых объектов и решать дополнительные задачи (например, выделять объекты на фоне помех, распознавать тип объекта по его радиолокационному образу, оценивать метеоусловия и т. п.) [8]. Вместе с тем, высокие значения частоты и диапазона дискретизации приводят к значительным объемам получаемых данных. Суть проблемы в том, что передача данных от блока первичной обработки к блоку вторичной обработки распределенной информационной системы имеет ограничения по скорости, обусловленные характеристиками используемых компьютеров, программной

* Работа поддержана грантом РФФИ, проект 15-08-00234.

платформы и сетевого оборудования. В условиях этих ограничений выбор тех или иных параметров аналого-цифрового преобразования исходного эхосигнала представляет собой актуальную задачу.

Промышленные системы наблюдения строятся с использованием специализированных ЭВМ и программных платформ, что определяет их высокую стоимость и ограниченную доступность специалистов-разработчиков и эксплуатационников нужной квалификации. Поэтому если решаемые системами наблюдения задачи не предъявляют высоких требований к надежности и защищенности (например, научно-исследовательские задачи), целесообразно обратиться к типовым компьютерам и программному обеспечению общего назначения.

Настоящая работа посвящена исследованию ограничений характеристик систем наблюдения, создаваемых на основе компьютеризированных РЛС кругового обзора с использованием типового компьютерного оборудования и программных платформ и связанных с конечностью скорости передачи данных в распределенных информационных системах.

Основные проблемные аспекты и постановка задачи

Пусть имеется типовая РЛС кругового обзора с периодом обращения антенны T_ϕ и частотой посылок ν_ϕ (частота дискретизации по азимуту). При таких значениях число линеек на один оборот антенны (т. е. число дискретных значений азимута) будет равно $n_\phi = \nu_\phi T_\phi$.

Пусть шкале дальности РЛС соответствует время приема линейки T_r и пусть каждая линейка преобразуется аналого-цифровым устройством с частотой дискретизации ν_r в массив значений. Тогда число элементов в этом массиве значений (т. е. число дискретных значений дальности) у каждой линейки будет равно $n_r = \nu_r T_r$.

Один оборот антенны РЛС формирует радиолокационный образ наблюдаемого пространства, который путем аналого-цифрового преобразования представляется в виде матрицы \mathbf{A} амплитуд отраженного эхосигнала с числом строк, равным n_ϕ , и числом столбцов, равным n_r . Матрица амплитуд обновляется с каждым оборотом антенны, т. е. один раз за время T_ϕ .

Система наблюдения, построенная на базе такой РЛС, имеет следующую архитектуру (рис. 1). Электрические видео- и синхронизирующие сигналы РЛС подаются на вход подключенного к ЭВМ аналого-цифрового преобразователя (АЦП). Полученная в результате такой первичной обработки матрица амплитуд \mathbf{A} передается по сетевому протоколу другой ЭВМ для вторичной обработки, включающей в себя сопровождение траекторий наблюдаемых объектов, определение их параметров движения и визуализацию навигационной обстановки в пользовательском интерфейсе.

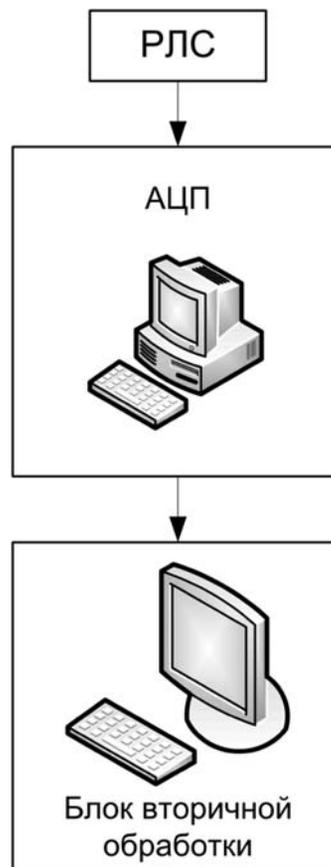


Рис. 1. Архитектура компьютеризированной радиолокационной системы наблюдения

В общем случае производительность современных типовых ЭВМ достаточна для решения базовых задач первичной и вторичной обработки сигнала. Вместе с тем, ввиду большого объема матрицы \mathbf{A} и ее частого обновления передача данных от блока первичной обработки к блоку вторичной обработки по сетевому протоколу является "узким местом" рассматриваемой системы радиолокационного наблюдения.

Архитектура современных программных платформ (операционных систем) позволяет обеспечить построение распределенных информационных систем путем использования многозадачности и передачи данных между процессами. Эти процессы, составляющие программную основу распределенной информационной системы, могут быть запущены параллельно друг другу и обмениваться данными как на одной, так и на различных ЭВМ, соединенных сетью. Имеется целый ряд инструментов для обмена данными между процессами [9].

Интерфейс динамического обмена данными (DDE-интерфейс). DDE-интерфейс предоставляет набор инструментов и сообщений, посылаемых между приложениями, использующими общую память, чтобы известить приложение об изменении данных. Приложения могут использовать протокол DDE как для однократной передачи данных, так и

для непрерывного обмена, в котором приложения посылают сообщения друг к другу, как только новые данные становятся доступны. Как правило, этот способ используется для передачи небольших объемов данных.

Почтовые слоты (mail slots). Mail slots — это специальные области памяти, для доступа к которым используются стандартные API-функции операционной системы для работы с файлами. Любой процесс может выполнять роль mail slot-сервера, создавая слот. Процесс, выполняющий роль mail slot-клиента, подключается к уже существующему слоту. Этот слот может располагаться как на текущем, так и на удаленном компьютере. И клиент, и сервер могут осуществлять как чтение, так и запись данных.

Именованные каналы (pipes). Именованные каналы используются для гарантированной передачи данных между двумя процессами, это своего рода "выделенная линия" между процессами. Одна из программ выполняет роль pipe-сервера, создавая канал и записывая туда данные, вторая программа — роль pipe-клиента, подключаясь к каналу (который может располагаться как на текущем, так и на удаленном компьютере) и читая данные. Достоинство этого способа — возможность обеспечить синхронизацию клиента и сервера.

Сокеты (sockets). Сокеты — абстрактные объекты, предоставляющие интерфейс для обмена данными между процессами. Каждый процесс может создать серверный сокет и привязать его к какому-нибудь порту операционной системы. Другой процесс создает клиентский сокет и подсоединяется к нему, после чего становится возможной передача данных через файловый дескриптор. Каждый сокет характеризуется своим сетевым адресом, что дает возможность передавать данные между процессами, запущенными на различных компьютерах.

Файлы, отображаемые в память (memory mapped files). Это способ работы с файлами, при котором всему файлу или его части ставится в соответствие определенный участок оперативной памяти. Для высокопроизводительного обмена данными между процессами используются непостоянные (non-persisted) файлы, отображаемые в память, которые не связаны с исходным файлом на диске. В этом случае фактически операционной системой создается общая память, доступ к которой процессы получают через файловый дескриптор. Достоинство этого способа — платформонезависимость.

Ввиду особенностей функциональности системы радиолокационного наблюдения для программной реализации обмена данными между блоками первичной и вторичной обработки наиболее подходящим является механизм именованных каналов (pipes). Его характеризует возможность синхронизировать работу блока вторичной обработки (pipe-клиента) с обновлением данных блока первичной обработки (pipe-сервером). При этом следует отме-

тить методологическую общность работы ядра операционной системы при реализации различных механизмов обмена данными между процессами, что приводит к близости их производительности и ресурсоемкости, в том числе — в разрезе различных операционных систем.

В настоящей работе ставится задача об оценке характерной скорости передачи данных от блока первичной к блоку вторичной обработки при использовании механизма именованных каналов (pipes). Это, в свою очередь, позволит оценить возможные предельные значения частот дискретизации радиолокационного образа по азимуту v_φ и дальности v_r , определяющих размерность матрицы амплитуд A .

Набор сценариев для оценки скорости передачи данных

Оценку характерной скорости передачи данных от pipe-сервера к pipe-клиенту будем проводить по результатам натурных испытаний. Имеются две ЭВМ, соединенные локальной сетью. На одной из них запускается программа-сервер, создающая именованный канал и генерирующая и записывающая в него данные. На второй ЭВМ запускается программа-клиент, читающая данные из именованного канала.

Скорость передачи данных определяется не только конфигурацией ЭВМ и сетевых устройств, но также используемой операционной системой и степенью загрузки ресурсов ЭВМ другими системными и прикладными задачами. Для анализа скорости передачи данных в различных условиях реализуем следующие тестовые сценарии.

Сценарий 1. Работа клиента и сервера без дополнительной нагрузки.

Сценарий 2. Параллельно клиенту и серверу запускается проигрывание видеофайлов с жесткого диска.

Сценарий 3. Параллельно клиенту и серверу запускается проигрывание видеофайлов из сети.

Сценарий 4. Параллельно клиенту и серверу запускаются ресурсоемкие приложения.

Первый сценарий условно моделирует идеальный вариант работы системы наблюдения как программного продукта, второй, третий и четвертый — различные аспекты загрузки ресурсов ЭВМ (процессор, память, сеть). В четвертом сценарии роль ресурсоемкого приложения выполняли компьютерные игры с богатой функциональностью.

Результаты натурных измерений скорости передачи данных

При постановке эксперимента использовались ЭВМ со следующими характеристиками: процессор Intel Core i5 с частотой 2,5 ГГц, память 4 Гбайт, сетевой адаптер со скоростью 1 Гбит/с, ОС Win7(64). Такие характеристики вполне соответствуют определению "типовая ЭВМ общего назначения".

На рис. 2 показаны графики, характеризующие значения скорости передачи данных C от *peer*-сервера к *peer*-клиенту. Рис. 2, *a* соответствует первому сценарию, рис. 2, *б, в* и *г* — второму, третьему и четвертому. В момент времени $t = 0$ последовательно запускались процессы сервера и клиента, в момент времени $t = 20$ с, после стабилизации распределения ресурсов ЭВМ, начиналось логирование данных. Из графиков видно, что скорость передачи данных составляет порядка 50 Мбайт/с (что более чем в 2 раза меньше теоретически возможной, обусловленной характеристиками используемого сетевого адаптера 1 Гбит/с). При этом имеют место колебания скорости порядка 5...10 %. Значения скорости передачи данных близки во всех четырех сценариях, т. е. скорость передачи данных не зависит явно от загрузки ресурсов ЭВМ, если нагрузка не экстремально велика.

Значения размерности матрицы амплитуд отраженного экосигнала A — число строк n_ϕ и число столбцов n_r — при известных размере одного элемента матрицы s (определяется диапазоном дискретизации), скорости передачи данных от блока первичной обработки к блоку вторичной обработки C и периоде обращения антенны РЛС T_ϕ связаны соотношением

$$n_r = \frac{CT_\phi}{sn_\phi}.$$

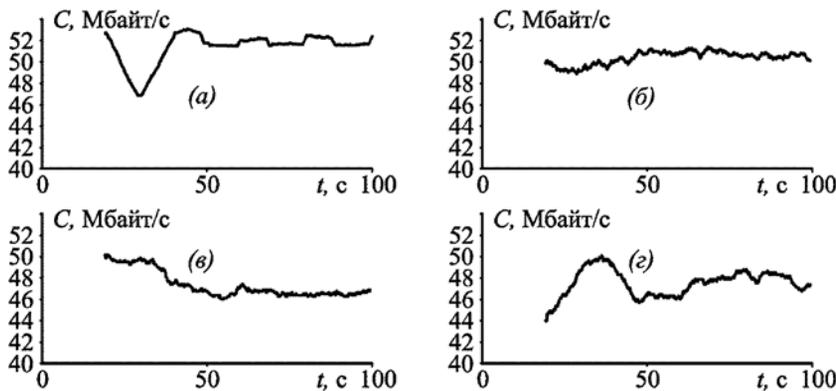


Рис. 2. Скорость передачи данных клиент — сервер

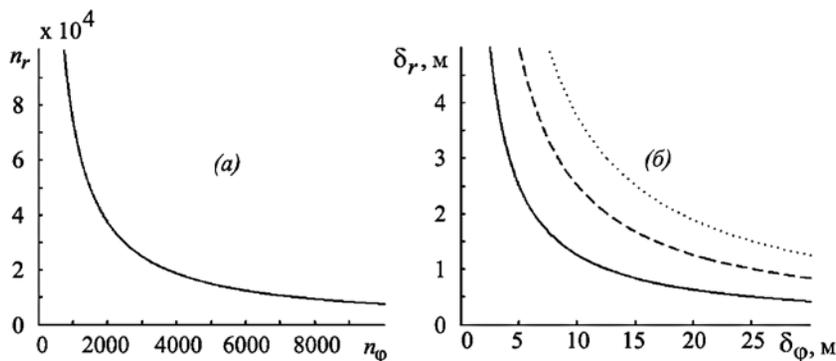


Рис. 3. Максимально возможные размеры матрицы амплитуд и соответствующие погрешности измерения координат

Погрешности измерения угла и дальности, обусловленные дискретизацией и выраженные в метрах, при размерности матрицы амплитуд $n_\phi \times n_r$ равны, соответственно, $\delta_\phi = 2\pi r/n_\phi$ и $\delta_r = R/n_r$, где r — расстояние до наблюдаемого объекта, R — максимальная дальность работы РЛС (определяемая выбранной шкалой дальности).

Рис. 3 иллюстрирует возможные значения погрешностей измерения координат наблюдаемых объектов, если скорость передачи данных от блока первичной обработки к блоку вторичной обработки C составляет 50 Мбайт/с, период обращения антенны РЛС T_ϕ равен 3 с, а размер одного элемента матрицы амплитуд s равен 2 байт. На рис. 3, *a* показана зависимость максимального числа столбцов n_r от числа строк n_ϕ при указанных значениях. Например, если частота посылок РЛС соответствует 4000 вращений на один оборот антенны, то максимальное возможное число дискретных значений дальности приблизительно равно 20 000.

На рис. 3, *б* показана зависимость минимально возможной погрешности измерения дальности δ_r от погрешности измерения угла δ_ϕ при $R = 30$ км и $r = 5$ км (сплошная линия), $r = 10$ км (штриховая линия), $r = 15$ км (пунктирная линия). Видно, что, например, для объекта, находящегося на удалении $r = 5$ км, при погрешности измерения угла δ_ϕ , равной 4 м, погрешность измерения дальности δ_r равна 3 м, что соответствует погрешности измерения местоположения в целом 5 м.

Заключение

Как показывают приведенные в статье эксперименты и расчеты, реально достижимая на типовой ЭВМ со скоростью сетевого адаптера 1 Гбит/с скорость передачи данных составляет порядка 50 Мбайт/с. В этом случае обусловленная дискретизацией РЛС сигнала погрешность измерения координат наблюдаемых объектов на характерных дальностях 10...15 км составляет порядка 10...5 м. Такие погрешности достаточны для решения многих задач наблюдения, связанных, например, с оценкой навигационной обстановки и обеспечением безопасности коллективного движения наблюдаемых объектов [10, 11].

Для задач, требующих на порядок меньшей погрешности измерения координат ~1 м [12], необходима на два порядка большая скорость передачи данных ~5 Гбайт/с, недостижимая для типового сетевого оборудования общего назначения. Решение таких задач требует разработки специальных приемопередающих устройств и алгоритмов обработки данных.

Список литературы

1. Лучков Н. В. Анализ объединения данных РЛС, их временная и пространственная привязки // Автоматизация процессов управления. 2015. № 1. С. 21—26.
2. Дмитриев В. И., Соляков О. В., Турецкий Н. В. Автоматизированное рабочее место судоводителя — настоящее и будущее // Вестник гос. университета морского и речного флота им. адмирала С. О. Макарова. 2014. № 4. С. 42—47.
3. Donderi D. C., Mercer C. R., Hong M. B., Skinner C. D. Simulated Navigation Performance with Marine Electronic Chart and Information Display Systems (ECDIS) // Journal of Navigation. 2004. Vol. 57. P. 189—202.
4. Ростопшин Д. Я., Антонова Д. А. О проблемах использования данных автоматической идентификационной системы в задачах управления движением судов // Мехатроника, автоматизация, управление. 2007. № 9. С. 63—69.
5. Дорожко В. М. Информационная технология идентификации начального этапа разгона морского судна // Информационные технологии. 2009. № 10. С. 65—70.
6. Дегтярёв О. В., Орлов В. С. Алгоритмы обнаружения и децентрализованного разрешения опасных сближений самолетов в воздухе, основанные на методе потенциальных полей // Изв. РАН. Теория и системы управления. 2013. № 5. С. 93—113.
7. Радар Процессор РП5М [Электронный ресурс]. URL: <http://www.ipmce.ru/custom/path3/> (01.09.2015)
8. Десятисильный А. С., Дорожко В. М., Гриняк В. М. Технология компьютерного моделирования радиолокационного эхосигнала // Информационные технологии. 2002. № 3. С. 42—49.
9. Сафонов В. О. Основы современных операционных систем. М.: БИНОМ. Лаборатория знаний, 2011. 584 с.
10. Дорожко В. М. Радиолокационная регистрация начального этапа разгона морского судна // Измерительная техника. 2010. № 6. С. 23—28.
11. Гриняк В. М., Головченко Б. С., Десятисильный А. С. Нейронечеткая обучаемая система распознавания опасного сближения судов // Информационные технологии. 2014. № 9. С. 68—73.
12. Гриняк В. М., Десятисильный А. С. Идентификация воздушных объектов в системах управления движением судов // Транспорт: наука, техника, управление. 2012. № 8. С. 38—40.

V. M. Grinyak, Professor, Far Eastern Federal University, e-mail: victor.grinyak@gmail.com,
Yu. S. Ivanenko, Student,
Vladivostok State University of Economics and Service,
A. S. Devyatisilny, Professor, Institute of Automation and Control Processes, FEBRAS

Data Rate in Distributed Information System and its Effect on Digital Radar Measurement Error

Current paper is about evaluation of measurement errors at distributed digital radar system with limited network bandwidth. Radar system considered under this paper consists of 2 main blocks: ADC convertor that digitalize raw radar signal and block of secondary data processing. Since high sampling frequency generates big digital datasets that should be transmitted from ADC to secondary processing block(s) bandwidth has crucial impact on system accuracy. Authors emphasize their efforts on analysis the systems where secondary processing block is based on common widely available hardware. Typical personal computer as common hardware is considered as suitable for research purposes. This paper is emphasis on research of system limitation based on two-coordinate radar. Problem is states as following. ADC generates amplitude matrix by signals from radar. Rows of the amplitude matrix represent angular sampling and columns represent distance sampling. The matrix is transmitted to secondary processing block that is responsible for tasks like: recognizing objects, objects tracking and rendering. The paper refer to comparison of data transmission protocols between 2 physical hardware (PC based) connected into network. Following protocols are considered as potentially suitable for inter-process communications: DDE, Mail-slots, Named-pipes, Sockets, memory-mapped files. In this paper preference is given to named-pipes. Name-pipes have been used for all computational experiments of data transmitting speed evaluation. Computational experiments for the article have been performed for system of 2 personal computers connected to local network. In order to simulate different external conditions following types of tests have been performed on the system: both hosts running data receiving-transmission without extra load, both hosts running data receiving-transmission with extra CPU utilization, both hosts running data receiving-transmission with extra memory utilization, both hosts running data receiving-transmission with extra network utilization.

The paper presents the results of data transmission experiments on charts. For 1 Gbit/s Ethernet, average transmission speed is about 50 Mbyte/s with 5—10 % fluctuations rate. One of the results of the experiments that data transmission speed does not directly depend on CPU, memory, network utilization. Based on the experimental values of data transmission speed as main result of the paper given charts of dependency of maximum size of amplitude matrix and measurement error for typical radar system.

Keywords: distributed information system, radar, digital signal processing, data transfer, measurement error

References

1. **Luchkov N. V.** Analiz obyedineniya dannikh RLS, ikh vremennaya i prostranstvennaya privyazki, *Avtomatizatsiya processov upravleniya*, 2015, no. 1, pp. 21–26.
2. **Dmitriev V. I., Solyakov O. V., Turecky N. V.** Avtomatizirovannoye rabochee mesto sudovoditelya — nastoyashee i budushee, *Vestnik gos. universiteta morskogo i rechnogo flota im. admirala S. O. Makarova*, 2014, no. 4, P. 42–47.
3. **Donderi D. C., Mercer C. R., Hong M. B., Skinner C. D.** Simulated Navigation Performance with Marine Electronic Chart and Information Display Systems (ECDIS). *Journal of Navigation*. 2004, vol. 57, pp. 189–202.
4. **Rostopshin D. Ya., Antonova D. A.** O problemakh ispolzovaniya dannykh avtomaticheskoy identifikatsionnoy systemy v zadachack upravleniya dvizheniya sudov, *Mekhatronika, avtomatizatsiya, upravleniye*, 2007, no., 9, pp. 63–69.
5. **Dorozhko V. M.** Informatsionnaya tekhnologiya identifikatsii nachalnogo etapa razgona morskogo sudna, *Informatsionnye tekhnologii*, 2009, no. 10, pp. 65–70.
6. **Degtyarev O. V., Orlov V. S.** Algoritmy obnaruzheniya i detsekrizirovannogo razresheniya opasnykh sblizheniy samoletov v vozduhe (Broadcast algorithms for detection and decentralized resolution of unsafe approach of airborne aircraft based on the force field method), *Izvestiya RAN. Teoriya i sistemy upravleniya* (Journal of computer and systems sciences international), 2013, no. 5, pp. 764–782.
7. **Radar Processor RP5M**, URL: <http://www.ipmce.ru/custom/path3/> (01.09.2015)
8. **Dev'yat'silny A. S., Dorozhko V. M., Grinyak V. M.** Tekhnologiya komputernogo modelirovaniya radiolokatsionnogo ekhhsignala, *Informatsionnye tekhnologii*, 2002, no. 3, pp. 42–49.
9. **Safonov V. O.** Osnovy sovremennikh operatsionnykh system. Moscow: BINOM. Laboratoriya znaniy, 2011. 584 p.
10. **Dorozhko V. M.** Radiolokatsionnaya registratsiya nachalnogo etapa razgona morskogo sudna (Radar logging of the initial stage of acceleration of a vessel) // *Izmeritel'naya tekhnika* (Measurement techniques), 2010, no. 6, pp. 629–636.
11. **Grinyak V. M., Golovchenko B. S., Dev'yat'silny A. S.** Neyronechetkaya obuchaemaya sistema raspoznavaniya opasnogo sblizheniya sudov, *Informatsionnye tekhnologii*, 2014, no. 9, pp. 68–73.
12. **Grinyak V. M., Dev'yat'silny A. S.** Identifikatsiya vozdukhnykh objektov v sistemakh upravleniya dvizheniyem sudov, *Transport: nauka, tekhnika, upravleniye*, 2012, no. 8, pp. 38–40.

УДК 004.946

Р. Ю. Скоробогатов, аспирант, ассистент, e-mail: ro-m-a-n@yandex.ru,
Федеральное государственное образовательное бюджетное учреждение
высшего профессионального образования "Сибирский государственный университет
телекоммуникаций и информатики", Новосибирск

Расширение интерактивности компьютерной модели в телевизионной среде

Рассматривается методика внедрения виртуального персонажа в среду виртуальной телевизионной студии с дублированием движений от реального актера. Рассматриваются методы создания видеоролика с трехмерным аватаром. Показано, что Kinect второго поколения имеет значительно более качественные характеристики по сравнению с первым поколением. Приведены параметры оборудования, которые отвечают современным требованиям телеиндустрии.

Ключевые слова: Kinect, 3D, виртуальная телевизионная студия, рендер, виртуальная реальность, дополненная реальность

Введение

В последнее время телевидение переживает бурный рост. Трансляции таких важных событий, как зимняя Олимпиада 2014 г. в Сочи, чемпионаты мира по футболу и хоккею, которые пройдут в России в ближайшем будущем, требуют не только всестороннего обзора от команды работников, осуществляющих непосредственную трансляцию во время проведения матчей и соревнований, но также дополнительную работу в телевизионной студии: освещение состояния игроков, разбора матчей и многое другое, связанное с основным событием. Также для каждого крупного спортивного мероприятия разрабатывается свой персонаж-талисман, кото-

рый создается как в графическом исполнении — двухмерный рисунок, так и в 3D-формате.

Для ускорения и удешевления решения этих задач были разработаны виртуальные телевизионные студии (ВТС) [1] с интерактивной составляющей (смена декораций, появление дополнительных предметов и графиков в соответствии с событиями в студии). Однако при появлении виртуального персонажа в студии телеведущий взаимодействовать с ним не может, что делает показ телевизионной картинки "искусственным" для зрителя.

В качестве решения данной проблемы была разработана методика дополнения виртуальной студии компании СофтЛабНск "Фокус" [2] управляемым виртуальным персонажем.

Постановка задачи

В настоящее время существует два основных метода персонажной анимации.

1. *Keyframe animation* (анимация по ключевым кадрам) [3] — оперирует не только значением анимированного параметра, но и моментом времени. Все ключи анимации одного типа располагаются на отдельном треке анимации. Таким образом, для каждого объекта имеется минимум семь треков — три для положения объекта по координатным осям X , Y , Z , три для поворота и один для изменения масштаба объекта. Для каждого параметра устанавливается трек с соответствующим параметром: длина, ширина, высота, радиус и др.

Представленный тип анимации требует высокой квалификации от сотрудника и, следовательно, имеет высокую стоимость при производстве. Также требуется довольно длительное время для придания объекту плавности и реалистичности в движениях.

2. *Motioncapture* (захват движения) [4] создает анимацию посредством размещения на теле человека датчиков отслеживания движения и считывания их параметров сенсорами [5], установленными по краям площадки, либо с применением технологии "компьютерное зрение" [6].

Такая технология позволяет создавать максимально приближенные к реальности движения. Необходимое специальное оборудование и малое число специалистов в этой области делают производство слишком дорогостоящим и подходит, как правило, только крупным киноконцернам.

В данной работе рассматривается возможность персонажной анимации с использованием Kinect второго поколения, а также рассматривается возможность внедрения полученного результата непосредственно в телевизионную трансляцию.

Основная часть

Kinect первого поколения [7] предоставил возможность отслеживать реального актера и совмещать его с движением виртуального персонажа. Однако устройство имело ограниченные технические параметры, которые не давали высокой точности передачи движения и, следовательно, качественного выходного изображения.

Второе поколение Kinect позволяет [8] уменьшить артефакты при передаче движений, а новая версия SDK (Software Development Kit — комплект разработки программного обеспечения) дает возможность реализации более производительных алгоритмов для обработки сигналов, получаемых с сенсоров.

На рис. 1 представлены изображение Kinect второго поколения с отображением датчиков, имеющихся в представленном устройстве, и их расположение.



Рис. 1. Сенсоры и датчики Kinect второго поколения

Слева располагается Color Camera — RGB-видеокамера с разрешением 1080 p. Далее идет Depth Sensor (датчик глубины), состоящий из инфракрасной камеры (IR Camera) и инфракрасного излучателя.

ИК излучатель сканирует пространство перед собой, а ИК камера принимает искаженный сигнал и на расхождении между идеальным сигналом, заложенным программно, и принимаемым сигналом создает трехмерную картину окружающего пространства. Снизу располагается массив микрофонов, который позволяет записывать качественный звук, совпадающий с движением губ персонажа.

В Kinect второго поколения добавлено несколько новых режимов просмотра:

а) режим "Инфракрасное зрение", который совместно с новыми инструментами моделирования человеческого тела можно использовать для отслеживания движений мышц и взаимной ориентации частей тела;

б) режим "Глубокое изображение", действующий как радар, где каждый из 220 тыс. пикселей сенсора записывает данные независимо, позволяя создавать точное и детализированное отображение комнаты.

В Kinect второго поколения предусмотрены настройки камеры, инвариантные свету (остающиеся неизменными при изменении параметров источника света). При таких настройках Kinect дает одинаковый результат вне зависимости от освещения комнаты. На практике это означает, что можно использовать Kinect в темноте или помещении со световыми шумами. Например, прожектор, направленный прямо на датчик, не повлияет на производительность устройства. Описанная функция была протестирована в ряде тестов и оказалась работоспособной [9].

Ниже представлены технические характеристики Kinect второго поколения в сравнении с первым поколением устройства [10].

Основные характеристики Kinect второго поколения относительно первого поколения

Минимальное необходимое расстояние от Kinect	Снижено до 1,58 м
ИК сенсор	Сканирование помещения активным ИК лучом (в отличие от негативного в первом поколении устройства)
Разрешение матрицы	Повышено до 1920 × 1080 пикселей (предыдущая версия — 640 × 480 пикселей)
Разрешение по глубине	512 × 424 (было 320 × 240)
Диапазон глубины	0,5...4,5 м (было 0,8...4,0 м)
Активное ИК разрешение	512 × 424 (раньше не было реализовано)
Углы обзора	Увеличины на 60 %
Частота кадров	30 кадров/с (ранее 25 кадров/с)
Распознавание людей	6 человек (ранее 2 человека)
Скорость отклика	33 мс

Приложение для Kinect второго поколения может создать сетку из более чем 1000 точек для более точного представления лица, что означает повышение разрешения в 20 раз.

На рис. 2 представлен процесс создания видеоролика с персонажной анимацией в виде схемы.

Первоначально актер перед Kinect совершает движение, после чего включается процесс записи в видеокамере и сканирования помещения датчиком глубины в ИК диапазоне. Когда датчик глубины начинает принимать искаженный сигнал, анализирующий модуль определяет, что перед устройством стоит человек. Далее запускается модуль, отвечающий за создание скелета, соответствующего скелету актера, и передачу его виртуальному персонажу. Как только произошла синхронизация по скелету, данные начинают записываться в программу в виде последовательности ключевых кадров. Данный метод позволяет программно устанавливать промежуток между ключевыми точками движений, следовательно, чем он больше, тем менее реалистично передаются движения, но в то же время снижается нагрузка на вычислительные мощности компьютера. В завершение последовательность движений сохраняется в видеофайл, в который отдельной дорожкой записывается звук с блока микрофонов, находящегося в устройстве.

Данный метод создания анимации может быть изображен в виде многократно повторяющейся циклической схемы (рис. 3).

Данная схема может быть продублирована для большего числа персонажей. Технические ограничения позволяют использовать до шести аватаров одновременно.

Программа начинает работать при выполнении следующего условия:

$$|S_1| = |S_2|, \quad (1)$$

где $|S_1|$ — трехмерная матрица, описывающая узлы (суставы) скелета реального актера; $|S_2|$ — трехмерная матрица, описывающая узлы скелета 3D-персонажа.

Таким образом, создание скелетной анимации происходит по следующей формуле:

$$|A_0| = |S_{10}| = |S_{20}|, \quad (2)$$

где $|A_0|$ — нулевое значение матрицы анимации, соответствующее моменту времени, когда программа синхронизировала значения матриц со значениями виртуального скелета и скелета реального человека; $|S_{10}|$ — трехмерная матрица, описывающая узлы скелета реального актера в начальный момент времени — 0-й кадр; $|S_{20}|$ — трехмерная матрица, описывающая узлы скелета 3D персонажа в начальный момент времени — 0-й кадр.

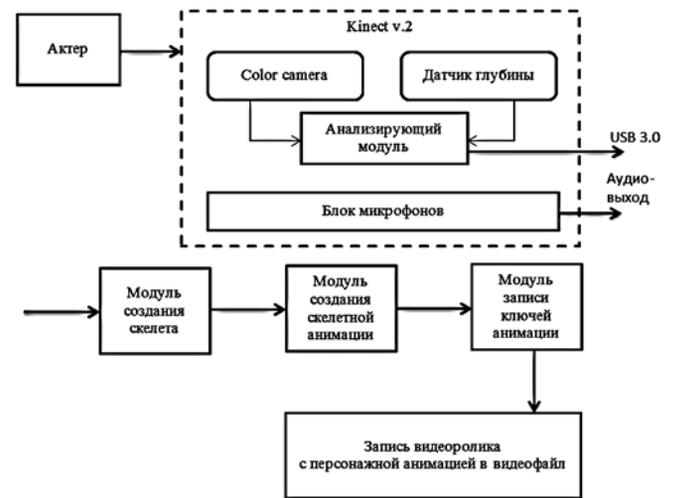


Рис. 2. Блок-схема создания видеоролика с персонажной анимацией



Рис. 3. Цикл создания скелетной анимации

Цикл получения матрицы для первого кадра анимации происходит по формуле

$$|A_1| = |S_{20}| + \Delta|S_{11}|, \quad (3)$$

где $|A_1|$ — матрица анимации для первого кадра анимации; $|S_{11}|$ — трехмерная матрица, описывающая узлы скелета реального актера в первом кадре.

В результате получаем:

$$|A_1| = |S_{21}|, \quad (4)$$

где $|S_{21}|$ — трехмерная матрица, описывающая узлы скелета 3D-персонажа в первом кадре.

Данный цикл может быть повторен для n движений, при условии достаточного количества места на жестком диске компьютера, по следующей формуле:

$$|A_n| = |S_{2n}| + \Delta|S_{1(n+1)}|, \quad (5)$$

где n задает номер кадра.

Таким образом, качество анимации зависит от величины Δ , которую можно программно увеличивать или уменьшать в соответствии с требуемым качеством и возможностями вычислительных мощностей оборудования, используемого для просчета входных данных.

Удешевление производства видеороликов с персонажной анимацией происходит благодаря использованию Kinect второго поколения и разработанного для его применения современного SDK 2.0 (Softwaredevelopmentkit — набор средств для разработки программ) [11]. Стоимость самого устройства составляет 7999 руб. [12]. Это значительно дешевле среднерыночной цены программного обеспечения для создания трехмерной персонажной анимации, например, для такого как 3dsmax 2016 от компании Autodesk, — его стоимость составляет порядка 140 000 руб. за одну лицензию [13]. Стоимость систем Motioncapture варьируется в довольно широком диапазоне, начиная от 50 000 руб. за лицензию на ПО за один год [14]. Таким образом, описанный в данной статье метод значительно дешевле существующих на рынке аналогов, при этом он предоставляет высокое качество итоговой анимации.

Заключение

На основе проведенного структурного анализа процессов создания персонажной анимации с при-

менением Kinect второго поколения получены следующие результаты:

- рассмотрены условия функционирования процессов создания скелетной анимации в виртуальном пространстве;
- для предложенного цикла создания скелетной анимации показаны соответствующие самоопределяющиеся структуры и дана их краткая характеристика.

Дополнительным преимуществом описанного метода является отсутствие длительного обучения персонала для начала работы. Большинство операций, такие как соединение предварительно загруженного фона с виртуальным персонажем, запись и сведение аудио- и видеодорожки, выполняются автоматически средствами вычислительных мощностей ПК. Данный алгоритм позволяет обогащать видеоряд без привлечения дополнительных сотрудников в компанию, так как программа проста в использовании и не требует специализированных навыков работы.

Таким образом, представленный метод соединяет в себе преимущества анимации, создаваемой по ключевым кадрам, и анимации, создаваемой по средствам захвата движения, но имеет низкую стоимость производства видеороликов при возможности работы в режиме реального времени.

Список литературы

1. Shimoda S., Yamanouchi Y., Fukaya T. New Video Production Techniques for Virtual Studios // Broadcast Technology. 2005. N. 21. P. 6—9.
2. Сайт <http://d-graphica.ru/mediawiki/>.
3. Martin J. B. 3D Theory — Keyframe animation [Electronic resource] // 2015. URL: <http://www.euclideanspace.com/threed/animation/keyframing/>.
4. Реут Д. Использование Blender в короткометражном фильме "TheMessage" // Международная конференции по компьютерной графике "CGEvent-2012", М., Россия, 13—14 декабря 2012.
5. Макаров С. Vicon Motion Systems. Профессиональные решения для захвата движения // 625. 2013. № 10.
6. Телековский С. Прорыв в современных технологиях. Техника и технология кино // 625. 2010. № 5.
7. Barbagallo R. S. TUIO based multitouch Kinect (SDK V1.0) for Windows7. Catania, 2012. 2 p.
8. Сайт <https://msdn.microsoft.com/en-us/library/dn785530.aspx>
9. Сайт <http://www.softrew.ru>.
10. Kinect for Windows — Human interface guidelines v2.0. California, Microsoft Corporation, 2014.
11. Сайт <https://www.microsoft.com/en-us/download/details.aspx?id=44561>.
12. Сайт <https://www.microsoft.com/en-us/kinectforwindows/purchase/default.aspx>.
13. Сайт <http://store.softline.ru/autodesk/autodesk-3ds-max/>.
14. Сайт <http://www.syssoft.ru/iPi-Soft/iPi-Motion-Capture-Standard-Edition/>.

R. Yu. Skorobogatov, Postgraduate, Assistant of the Department CAD, e-mail: ro-m-a-n@yandex.ru
Federal State-Financed Educational Institution of Higher Professional Education
"Siberian State University of Telecommunications and Information Sciences", Novosibirsk

Introduction Virtual Characters in the Space of TV Studios

The article describes a technique for introducing a virtual character in a virtual television studio (VTS) with overlapping movements of real actors. Methods of creating a video with a three-dimensional avatar. It is shown that the second generation Kinect has significantly better specifications compared to the first generation. The parameters of the equipment that meet modern requirements of the industry body.

Keywords: *Kinect, 3d, virtual TV studio, rendering, virtual reality, augmented reality*

References

1. **Shimoda S., Yamanouchi Y., Fukaya T.** New Video Production Techniques for Virtual Studios, *Broadcast Technology*, 2005, no. 21, pp. 6–9.
2. **Sajt.** <http://d-graphica.ru/mediawiki/>.
3. **Martin J. B.** *3D Theory — Keyframe animation* [Electronic resource], 2015. URL: <http://www.euclideanspace.com/threed/animation/keyframing/>.
4. **Reut D.** Ispol'zovanie Blender v korotkometrazhnom fil'me "TheMessage", *Mezhdunarodnaja konferencii po komp'yuternoj grafike "CGEvent—2012"*, Moscow, Rossija, 13–14 dekabnja 2012.
5. **Makarov S.** Vicon Motion Systems. Professional'nye reshenija dlja zahvata dvizhenija, *625*, 2013, no. 10.
6. **Telekovskij S.** Proryv v sovremennyh tehnologijah // *Tehnika i tehnologija kino*, *625*, 2010, no. 5.
7. **Barbagallo R. S.** *TUIO based multitouch Kinect (SDK V1.0) for Windows7*. Catania, 2012. 12 p.
8. **Sajt.** <https://msdn.microsoft.com/en-us/library/dn785530.aspx>
9. **Sajt.** <http://www.softrew.ru>.
10. **Kinect** for Windows — Human interface guidelines v2.0 California, Microsoft Coloration, 2014.
11. **Sajt.** <https://www.microsoft.com/en-us/download/details.aspx?id=44561>.
12. **Sajt.** <https://www.microsoft.com/en-us/kinectforwindows/purchase/default.aspx>.
13. **Sajt.** <http://store.softline.ru/autodesk/autodesk-3ds-max/>.
14. **Sajt.** <http://www.syssoft.ru/iPi-Soft/iPi-Motion-Capture-Standard-Edition/>.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *З. В. Наумова*.

Сдано в набор 01.03.2016. Подписано в печать 22.04.2016. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT516. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.