

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 24
2018
№ 12

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Битков Ю. В. Разработка методов автоматизации ресурсоориентированной функциональной коррекции логических схем 755

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Барский А. Б., Мельник Д. И. Сетевая технология проектирования асинхронной вычислительной системы на общем решающем поле с самоуправляемой параллельной обработкой данных 763
- Богатырев В. А., Паршутин С. А. Анализ влияния кратности резервирования многопутевых передач на вероятность их своевременного обслуживания 772
- Инютин С. А. Комплексование систем счисления для многоуровневых вычислительных процессов 782

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Фаткиева Р. Р., Рыжков С. Р. Оценка нарушения периметра информационной безопасности в облачной среде 791

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Корепанов В. Д., Кулагин В. П., Халабия Р. Ф. Использование нейронных сетей при модификации генетического алгоритма 799

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Злобина А. Г., Журбин И. В. Сравнительный анализ алгоритмов сегментации данных электропрофилирования при восстановлении границы объекта . . 805
- Указатель статей, опубликованных в журнале "Информационные технологии" в 2018 г. 813

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.,
акад. 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.,
акад. 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.,
акад. 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.
ВАСИЛЬЕВ В. И., д. т. н., проф.
ВИШНЕКОВ А. В., 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.
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. В., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МАРТЫНОВ В. В., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕШУК О. М., д. т. н., проф.
САКСОНОВ Е. А., 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.
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 и базу данных RSCI на платформе Web of Science.

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

INFORMACIONNYYE TEHNOLOGII

Vol. 24
2018
No. 12

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

CAD-SYSTEMS

- Stempkovskiy A. L., Telpukhov D. V., Solovyev R. A., Bitkov Y. V.** Development of Methods for Automating Resource-aware Functional ECO Patch Generation . . . 755

COMPUTING SYSTEMS AND NETWORKS

- Barsky A. B., Melnik D. I.** Network Technology for Designing an Asynchronous Computing System on a Common Decision Field with Self-Controlled Parallel Processing of Data 763

- Bogatyrev V. A., Parshutina S. A.** The Analysis of How the Redundancy Degree of Multipath Transmissions Affects the Probability of their Timely Service 772

- Inyutin S. A.** The Aggregation of Number Systems for Multi-Digit Computational Processes 782

INFORMATION SECURITY

- Fatkieva R. R., Ryzhkov R. R.** Assessment of Violations of Information Security Perimeter in the Cloud 791

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Korepanov V. D., Kulagin V. P., Halabiya R. F.** The Use of Neural Networks in the Modification of the Genetic Algorithm 799

APPLICATION INFORMATION SYSTEMS

- Zlobina A. G., Zhurbin I. V.** Comparative Analysis of Segmentation Algorithms of Electric Profiling Data for Restoration of Boundary of Object 805

Editor-in-Chief:

Stempkovskiy 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., 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., 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., 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.
Vasiliev V. I., Dr. Sci. (Tech.), Prof.
Vishnekov A. V., 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.
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. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya.E., Dr. Sci. (Tech.), Prof.
Martynov V. V., 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.
Saksonov E. A., 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.
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"

А. Л. Стемповский, д-р техн. наук, академик РАН, науч. руководитель,
Д. В. Тельпухов, канд. техн. наук, зав. отд.,
Р. А. Соловьев, канд. техн. наук, вед. науч. сотр., **Ю. В. Битков**, науч. сотр.,
Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН)
e-mail: nofrost@inbox.ru

Разработка методов автоматизации ресурсоориентированной функциональной коррекции логических схем

В современном процессе проектирования СБИС нередко возникают ситуации, когда неправильная работа схемы обнаруживается лишь на этапе заключительной верификации. Хуже, когда ошибки проявляются даже после заключительного этапа проектирования или производства микросхем. Исправление ошибок на таких поздних стадиях требует огромных трудовых и финансовых затрат. Для минимизации этих затрат были разработаны методы внесения функциональных изменений на поздних стадиях проектирования (engeneering change order, ECO) для внесения правок в окончательный проект вместо того, чтобы выполнять полное перепроектирование.

В работе предложены методы автоматизации функциональной коррекции схем на базе структурного анализа, формальных методов и симуляционных подходов. Реализованы программные средства, демонстрирующие высокую эффективность на ряде реальных задач.

Ключевые слова: engeneering change order (ECO), функциональная коррекция, конфликты, ICCAD Contest

Введение

Методы внесения функциональных изменений на поздних стадиях проектирования *ECO* (engeneering change order), описанные в литературе, можно разделить на несколько базовых типов.

Методы на основе использования моделей ошибок [1–3]. Обычно рассматриваются только простые ошибки типа неверных соединений проводников или замены логических элементов. С помощью этих методов ищут ошибки только конкретных типов и применяют заранее известные решения для их исправления. В рамках такого узкого круга случаев задача решается быстро и эффективно. Однако этих моделей ошибок обычно недостаточно для представления всех функциональных ошибок в реальных проектах [4].

Методы, основанные на анализе структуры схем [5–7]. Эффективность данной группы методов базируется на том допущении, что на практике ошибки на уровне регистровых передач обычно сравнительно небольшие и локализованы в одной небольшой области схемы.

Для исправления схемы в этих методах используется сравнение структуры схемы с эталоном. Структурный анализ схем работает только в том случае, если схемы имеют большое число общих частей.

Методы, основанные на синтезе [8–12]. Данные методы не опираются на информацию о структуре эталонной схемы, вместо этого используются моделирование и методы функциональной верификации. Чаще всего методы этой группы работают в два этапа. На первом этапе находят те узлы, исправление которых способно исправить проект. На втором этапе запускается ресинтез для этих узлов. В то время как основным недостатком данной группы методов является слабая масштабируемость и зачастую ориентация на исправление только единичного сигнала, их преимуществом является полная автоматизация.

В данной работе рассматривается задача *ECO*, основанная на методе учета ресурсов при построении корректирующей схемы (патча), предложенная в работе [13]. В этой работе был предложен метод, основанный на учете некоторой физической информации о запасных

ячейках (*spare cells*) и расчете стоимости соединений после технологического маппинга. Таким образом, исправление ошибок в проекте предлагается проводить с учетом некоторой функции стоимости соединений. Это позволяет впервые автоматизировать процесс нахождения исправлений не только с учетом наименьшего числа логических элементов, но также с учетом полной длины межсоединений.

В маршруте промышленного проектирования функциональный инструмент *ECO*, например *Cadence Encounter Conformal ECO Designer* [14], широко применялся в течение многих лет. Базовой характеристикой корректирующей схемы в рамках данных средств является площадь патча. Несмотря на то что размер патча является важной метрикой его качества, для практического решения задачи *ECO* необходимо рассматривать и другие физические аспекты, включая обеспечение заданных временных и мощностных параметров.

В ряде теоретических исследований [4, 6, 8, 11, 12, 15–17] были предложены различные типы алгоритмов для генерации патчей. Основное внимание в работах уделялось минимизации размера патча. Однако сгенерированные патчи могут оказаться непригодными для решения промышленных задач ввиду слишком больших накладных расходов на их физическую реализацию.

1. Постановка задачи

В работе рассматривается второй этап маршрута *ECO* — этап исправления ошибок. На первом этапе — этапе обнаружения ошибок, части проекта, ответственные за ошибку, уже были идентифицированы. Это означает, что в логической схеме существует несколько всяких узлов, которые называются таргетами (*targets*). Необходимо с помощью некоторого набора узлов-кандидатов сформировать под-схему для этих таргетов таким образом, чтобы основная схема стала эквивалентна эталонной схеме. При этом нужно учитывать обобщенный весовой параметр, который имеет каждый узел-кандидат в схеме, что является своего рода метрикой накладных расходов для его использования в патче.

Формализуем задачу, введя некоторые обозначения. Даны две схемы: F — рассматриваемая схема и G — эталонная схема; набор таргетов t_1, t_2, \dots, t_n ; набор весов w_{g_i} , для каж-

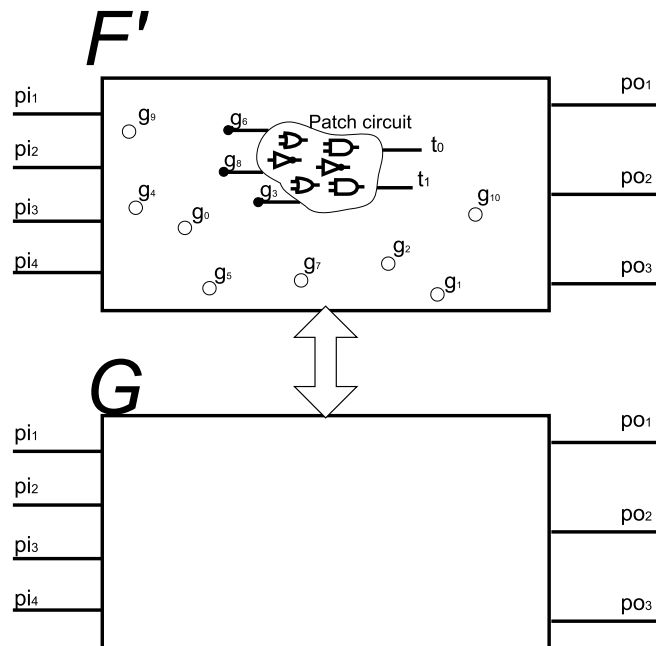


Рис. 1. Постановка задачи автоматизированной функциональной коррекции

дого узла-кандидата g_i , $i = 1..k$ из F . Задача: сгенерировать функции патча с минимальной стоимостью в определенном наборе таргетных точек в F таким образом, чтобы исправленная схема F' и схема G стали эквивалентны (рис. 1). Стоимость патча вычисляется суммированием весов используемых узлов-кандидатов.

Базисом для разрабатываемых методов служит формальный механизм проверки эквивалентности, связанный с представлением схемы в виде направленного ациклического графа, состоящего из инверсий и конъюнкций (And-Inverter Graph (AIG)) с последующим решением задачи выполнимости булевых формул (SAT). Данный механизм реализован в САПР с открытым исходным кодом — *ABC*, разработанном в Калифорнийском университете в Беркли (США) [18]. Все финальные и промежуточные проверки на эквивалентность реализованы с помощью встроенной функции *cec*. Также, не ограничивая общности рассуждений, будем считать, что используются только двух- и одноходовые вентили, поддерживаемые синтаксисом языка *Verilog*.

2. Структурный анализ схемы

Разбиение на подзадачи. Первый этап работы алгоритма заключается в структурном анализе схемы для сокращения вычислительной

нагрузки на последующих этапах. В частности, в рамках такого анализа делается попытка разбиения задачи на подзадачи меньшего размера в случае наличия нескольких таргетов.

Структурный анализ начинается с построения выходных конусов распространения сигнала для выходов каждого таргета и анализа первичных выходов схемы, которые попадают под влияние этих таргетов. Те выходы схемы, которые не попали под влияние таргетов, должны быть заведомо эквивалентны соответствующим выходам схемы G . Этот факт проверяется на начальном этапе, и в случае положительного исхода эти выходы исключаются из дальнейшего рассмотрения. В противном случае делается вывод о том, что никакой патч не сможет исправить эту ошибочную схему, что сигнализирует об ошибке на первом этапе маршрута ECO .

Далее, на основе данных о влиянии таргетов на выходы схемы строится целевая карта чувствительности (*target sensitivity map*, TSM). Эта структура формируется посредством группировки таргетов относительно влияния их на конкретные выходы и представляет собой функцию, определенную на конкретных наборах таргетов и отображающую их в наборы выходов: $TSM: T_i \rightarrow PO_j$, где $T_i = \{t_{i_1}, t_{i_2}, \dots, t_{i_k}\}$ — некоторый набор таргетов; $PO_j = \{po_{j_1}, po_{j_2}, \dots, po_{j_l}\}$ — некоторый набор выходов, po_j — соответствующий выход схемы. Такое отображение означает, что каждый выход из набора PO_j зависит от каждого таргета из набора T_i . Аргументы функции TSM уникальны: отдельные таргеты в них могут повторяться, в то время как po_j в значениях функции повторяться не могут. Ниже представлен пример TSM для одной реальной задачи, описанный в структуре данных языка *Python*. Третья строчка, к примеру, в этом словаре говорит о том, что на выход 'g82' влияют таргеты 't_0', 't_2', 't_7'.

```
{('t_0', 't_1'): ['g65'],
 ('t_0', 't_1', 't_2', 't_3'): ['g66'],
 ('t_0', 't_2', 't_7'): ['g82'],
 ('t_1', 't_2'): ['g67'],
 ('t_1', 't_4'): ['g83'],
 ('t_5', 't_6', 't_7'): ['g89'],
 ('t_7'): ['g85', 'g87', 'g88', 'g86', 'g84']}
```

Используя TSM , можно легко определить замкнутые непересекающиеся классы таргетов, каждый из которых влияет только на не-

которое подмножество выходов схемы, в то время как другие таргеты не влияют на них. Если существует больше одного такого класса, то задача разбивается на подзадачи, которые решаются параллельно и независимо, в том смысле что финальную проверку на эквивалентность для каждой такой подзадачи можно проводить лишь для соответствующих этому классу выходов.

Стратегия выбора таргетов. На основе TSM также реализована стратегия последовательного выбора таргетов для формирования патча. Базовая процедура нахождения патча, которая будет описана в последующих разделах, существенно ограничена обработкой только одного таргета. Поэтому чрезвычайно важно выбрать правильную последовательность обработки таргетов для наиболее эффективного нахождения патчей. Суть стратегии заключается в том, чтобы нахождение патча для каждого следующего таргета позволяло проверить на эквивалентность как можно больше выходов, либо максимально приблизить такую проверку. Алгоритм выбора следующего таргета для оценки представлен ниже.

1. Посчитать новый TSM .

2. Сформировать набор $S = \{T_{k_1}, T_{k_2}, \dots, T_{k_n}\}$, в котором каждый набор T_{k_i} — имеет минимальное число таргетов.

3. Выбрать набор T_{max} из S , имеющий максимальное число зависимых выходов: $|TSM(T_{max})| = \max(|TSM(T_{k_i})|)$; $i = 1..n$.

4. Выбрать случайный таргет из T_{max} .

После оценки очередного таргета и исключения зависящих только от него выходов считается новый TSM относительно оставшихся таргетов и выходов. Для рассмотренного примера последовательность выбора таргетов выглядит следующим образом:

```
't_7' -> 't_0' -> 't_1' -> 't_2' -> 't_3' ->
't_4' -> 't_5' -> 't_6'.
```

Откат таргетов. Стратегия выбора таргетов служит задаче минимизации ситуаций, когда после определенной итерации основного алгоритма оказывается, что патч работает некорректно. Чтобы не откладывать проверку на эквивалентность на самый конец, после каждого найденного таргета проводится формальная проверка тех выходов, которые уже определены всеми таргетами. Тем не менее может случиться, что после какого-то этапа формальная проверка на эквивалентность дает

отрицательный результат. Это может возникнуть за счет того, что базовая процедура поиска патча для отдельного таргета, как правило, опирается только на некоторое подмножество входных стимулов, и в некоторых случаях может генерировать некорректный патч. В этом случае необходимо откатить неправильно найденный таргет. Поиск такого таргета также происходит по *TSM*. Рассмотрим базовую стратегию такого поиска на том же примере. Пусть алгоритм остановился на этапе: 't_7' -> 't_0' -> 't_1' -> 't_2', после чего средство проверки формальной эквивалентности выявило ошибку на первичном выходе 'g82'. Этот выход зависит от трех таргетов: {'t_0', 't_2', 't_7'}. На выходе средства проверки эквивалентности мы получаем некоторый стимул, на котором происходит нарушение эквивалентности (стимул-контрпример). Далее, патчи для каждого таргета поочередно исключаются из "списка подозреваемых" и поиск запускается заново, но уже включая стимул-контрпример. Особенность базовой процедуры поиска патча состоит в том, что, имея стимул-контрпример, уже на первых этапах ее работы становится ясно — возможно ли построить корректный патч. Отрицательный ответ подразумевает, что исключенный таргет был на самом деле корректным. Противоположный случай говорит о том, что исключенный таргет и был источником ошибки. Такой патч исключается из списка найденных, а таргет откатывается.

Определение значащих входов. С помощью анализа структуры схем также можно определить список значащих входов схемы. Это позволит более эффективно использовать симуляци-

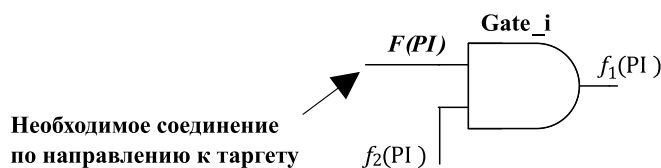


Рис. 2. Поиск функции для входа вентиля при известных функциях на другом входе и выходе

<i>a</i>	<i>b</i>	OR(<i>a,b</i>)
0	0	0
0	1	1
1	0	1
1	1	1

→

<i>a</i>	OR(<i>a,b</i>)	<i>b</i>
0	0	0
0	1	1
1	0	-
1	1	x

Рис. 3. Нахождение обратной функции для вентиля OR2

онное моделирование, обрабатывая те стимулы, на которых с большей вероятностью могут возникнуть конфликты. Получение значащих входов становится возможным за счет наличия эталонной схемы *G*. Для нахождения таких входов достаточно построить входные конусы от всех зависимых выходов в эталонной схеме. Таким образом, мы исключаем из рассмотрения те входы, которые не могут повлиять на эквивалентность тестовой и эталонной схем. Это также снижает размерность задачи и для некоторых сравнительно больших схем возникает возможность использовать полный перебор только по значащим входам, если их число невелико. Это гарантирует нахождение корректного патча.

3. Метод формального поиска патча

Еще один способ гарантированного нахождения корректного патча заключается в использовании формальных методов обратного распространения функции. Однако использование данного метода ограничено определенной структурой схемы. Анализ соответствующих ограничений для формального метода можно также провести на первом этапе работы нашего маршрута *ECO*.

Задача формального метода — найти точную функцию для таргета относительно первичных входов схемы *F*. Базисом для данного метода является процедура обратного распространения функции вентиля. Она формулируется следующим образом: пусть нам известны функции от входов схемы на одном из входов вентиля и функция на его выходе; необходимо найти функцию на другом его входе. Рассмотрим пример. Пусть дан вентиль *OR* и известны функции $f_1(PI)$ и $f_2(PI)$ на выходе и на одном из входов вентиля соответственно, где *PI* — некоторое подмножество основных входов схемы. Задача — найти функцию на втором его входе (рис. 2).

Для этого необходимо вначале определить, какая функция для вентиля *OR* будет обратной. Анализируя таблицу истинности, можно легко определить, что для этой роли подойдет как функция *OR*, так и функция *XOR*, функция второго операнда и инверсия прямой импликации (рис. 3).

Таким образом, функция для первого входа вентиля может быть найдена по формуле $F(PI)n = f_1(PI) + f_2(PI)$, или просто как

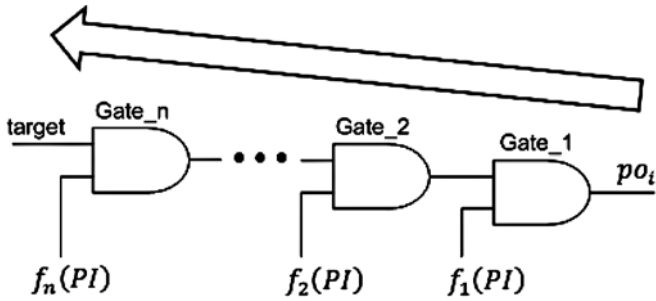


Рис. 4. Принцип обратного распространения для задачи формального поиска патча

$F(PI) = f_2(PI)$. Таким образом можно продвигаться в обратную сторону от выходов к входам схемы. Отправной точкой служит известная функция для основного выхода (*primary output*) эталонной схемы G . Конечной точкой служит вентиль схемы, к которому непосредственно подсоединен таргет.

Возможности данного метода существенно ограничены структурой схемы, показанной на рис. 4. В подсхеме от таргета до выхода не должно быть разветвлений, а все входные сигналы f_i должны зависеть только от входов схемы, и не должны зависеть от других таргетов. Несмотря на довольно строгие структурные ограничения, подобные ситуации встречаются довольно часто — примерно в 20 % случаев.

Достоинствами метода можно считать скорость работы (поскольку требуется только простой структурный анализ и однократная процедура обратного распространения), а также гарантию корректности патча. К недостаткам, помимо структурных ограничений, можно отнести довольно большой размер получаемых патчей. Тем не менее второй недостаток может быть нивелирован дополнительным вторым этапом, на котором проводится минимизация размера патча.

4. Метод генерации патчей на базе моделирования схемы

Ядром разработанного маршрута *ЕСО* является метод генерации патчей основанный на моделировании. Как следует из названия, этот метод принципиально базируется на симуляции схемы. Вследствие бит-параллельного моделирования схемы и эталона получаем набор сигнатур для входов, выходов и узлов-кандидатов схемы F . Сигнатура S — это n -битный вектор на входах, выходах или внутренних уз-

лах схемы, определяющий значения на этом узле, полученные в результате моделирования схемы, где n — это число моделирований схемы. Обращение к n -му биту сигнатуры, как и к любому битовому вектору, будем обозначать $S[k]$. Поскольку моделирование на вычислительных машинах удобно выполнять в бит-параллельном формате, то каждая сигнатура получается параллельно и параметр n характеризует длину входной последовательности при симуляции схемы. После первоначального моделирования получаем некоторый набор сигнатур. Пример представлен в табл. 1.

Следует отметить, что из списка узлов-кандидатов на первом этапе удаляются те узлы, которые зависят от таргетов. Это позволяет обрабатывать таргеты отдельно и независимо, а также уберегает от случайной генерации цепей обратной связи.

Следующий этап связан с построением некоторого массива, определяющего какие комбинации значений на таргетах должны быть для каждой симуляции, для того чтобы первичные выходы схем F и G совпадали. Назовем такой массив целевым (*Target array* (Ta)). Для получения этого массива требуется провести 2^k бит-параллельных симуляций схемы F , где k — число таргетов, построчно сравнивая получившиеся значения выходов F со значениями выходов G (табл. 2).

Этот массив дает исчерпывающую информацию о том, какие сигналы должны формироваться на выходах патча для того чтобы схема работала корректно на заданных стимулах. Дальнейшая работа алгоритма связана с по-

Таблица 1

Таблица сигнатур, полученная в результате бит-параллельного моделирования

№	Схема F						Схема G		
	S_{pi_1}	...	S_{pi_i}	S_{g_1}	...	S_{g_m}	S_{po_1}	...	S_{po_k}
0	0	...	0	0	...	1	0	...	1
1	0	...	0	1	...	0	0	...	1
2	0	...	0	0	...	0	0	...	0
3	0	...	0	1	...	0	0	...	1
...	...								
$n-1$	1	...	1	0	...	0	0	...	1

Число симуляций

Сигнатуры на первичных входах Сигнатуры на вентилях схемы F Сигнатуры на первичных выходах схемы G

следовательной обработкой таргетов в порядке, описанном в предыдущих разделах.

Имея Ta и нужный таргет для обработки, получаем целевой вектор (*Target vector* (Tv)), битовый вектор длиной n , определяющий, какой именно сигнал должен быть подан на этот таргет. Этот вектор похож на сигнатуру, за тем исключением, что кроме 0 и 1 в его разрядах могут быть безразличные состояния (X). При выделении Tv из Ta используется жадный алгоритм, иными словами максимизируется число X в Tv . Далее задача сводится к нахождению базиса из узлов-кандидатов для формирования патча.

Эта задача решается с помощью метода поиска базиса на основе конфликтов. Метод подробно описан в нашей предыдущей работе [19].

5. Результаты экспериментов

В экспериментальной части мы использовали тестовые схемы, которые представляют собой реальные практические случаи с различными сценариями *ECO*. Наборы эталонных тестов были созданы из схем ISCAS, ITC99 в IWLS 2005, OpenCore, LGSynth'93 и некоторых комбинационных участков сложных промышленных проектов.

Были рассмотрены различные сценарии *ECO* с различным числом точек, разным расстоянием между точками до первичных входов/выходов, а также разными размерами схем. Стратегии распределения весов также различались. Эти примеры были заимствованы из ежегодного конкурса ICCAD Contest, проведенного в 2017 г. Все контрольные примеры доступны в открытом доступе на сайте [20]. Результаты экспериментов представлены в табл. 3.

Построение целевого массива в рамках задачи автоматизированной функциональной коррекции схем

Sim. №	Схема F						Схема G			Ta	
	(t_0, t_1, \dots, t_k)			...	(t_0, t_1, \dots, t_k)			S_{po_1}	...		S_{po_k}
	$(0, 0, \dots, 0)$				$(1, 1, \dots, 1)$						
	S_{po_1}	...	S_{po_k}	...	S_{po_1}	...	S_{po_k}	S_{po_1}	...	S_{po_k}	
0	0	...	0	...	0	...	1	0	...	1	$\{(0, 0, \dots, 1), (0, 1, \dots, 1), \dots\}$
1	0	...	0	...	1	...	0	0	...	1	$\{(1, 1, \dots, 1), (1, 0, \dots, 1), \dots\}$
2	0	...	0	...	0	...	0	0	...	0	$\{(1, 0, \dots, 0), (1, 0, \dots, 0), \dots\}$
3	0	...	0	...	1	...	0	0	...	1	$\{(0, 0, \dots, 0), (1, 0, \dots, 1), \dots\}$
...	
$n - 1$	1	...	1	...	0	...	0	0	...	1	$\{(0, 0, \dots, 0), (1, 0, \dots, 1), \dots\}$

Таблица 2

Исследование эффективности маршрута автоматизированной ресурсоориентированной функциональной коррекции схем

Схема	Число таргетов	Число входов	Число выходов	Число элементов	Число элементов в эталоне	Среднее время работы, с	Среднее число элементов в патче
1	1	3	2	5	6	0	3
2	1	157	64	1122	1219	30	9
3	1	411	128	2074	1929	14	8
4	1	11	6	74	77	5	4
5	2	450	282	24 355	21 056	302	123
6	2	99	128	13 832	11 818	451	9
7	1	207	24	2944	1721	264	10
8	1	179	64	2512	3337	41	18
9	4	256	245	5845	4657	104	46
10	2	32	129	1673	2052	322	251
11	8	48	50	2065	2177	400	850
12	1	46	27	13 803	821	76	3
13	1	25	39	377	435	30	47
14	12	17	15	1969	1006	140	274

Заключение

В работе представлен маршрут автоматизированной ресурсо-ориентированной функциональной коррекции схем. Предложены различные структурные подходы, которые позволяют уменьшить вычислительную нагрузку на симуляционный метод поиска базиса.

На языке Python 3.4 был реализован программный модуль для автоматизированной ресурсоориентированной функциональной коррекции схем. Экспериментальные результаты показывают высокую эффективность метода даже на очень сложных задачах с большим числом таргетов, элементов и входов.

Исходный код вместе с тестовыми примерами, а также подробное описание задачи опубликованы в открытом доступе на сайте [20]. Это открывает путь для разработчиков САПР в области ECO для сравнения их алгоритмов и использования разработок для собственного программного обеспечения.

Список литературы

1. Veneris A. G., Hajj I. N. A fast algorithm for locating and correcting simple design errors in VLSI digital circuits // Proc. of Great Lake Symposium on VLSI Design. 1997. P. 45–50.
2. Abadir M., Ferguson J., Kirkland T. Logic design verification via test generation // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 1988. Vol. 7, N. 1. P. 138–148.
3. Chung P.-Y., Hajj I. N. Accord: Automatic catching and correction of logic design errors in combinational circuits // Proc. of Int. ITC, Sep. 1992. P. 742–751.
4. Shao-Lun Huang, Wei-Hsun Lin, Po-Kai Huang, Chung-Yang Huang. Match and Replace: A Functional ECO Engine for Multierror Circuit Rectification // Computer-Aided Design of Integrated Circuits and Systems IEEE Transactions. 2013. Vol. 32. P. 467–478.
5. Brand D., Drumm A. D., Kundu S., Narain P. Incremental synthesis // International Conference on Computer Aided Design. 1994. P. 14–18.
6. Krishnaswamy S., Ren H., Modi N., Puri R. DeltaSyn: An efficient logic difference optimizer for ECO synthesis // International Conference on Computer Aided Design. 2009. P. 789–796.
7. Huang S.-Y., Chen K.-C., Cheng K.-T. AutoFix: A hybrid tool for automatic logic rectification // Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1999. Vol. 18, N. 9. P. 1376–1384.
8. Lin C.-C., Chen K.-C., Marek-Sadowska M. Logic synthesis for engineering change // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 1999. Vol. 18, N. 3. P. 282–292.
9. Yang Y.-S., Sinha S., Veneris A., Brayton R. Automating logic rectification by approximate SPFDs // Proc. ASP-DAC, Jan. 2007. P. 402–407.
10. Ling A., Brown S., Safarpour S., Zhu J. Toward automated ECOs in FPGAs // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., Jan. 2011. Vol. 30, N. 1. P. 18–30.
11. Wu B.-H., Yang C.-J., Huang C.-Y., Jiang J.-H. A robust functional ECO engine by SAT proof minimization and interpolation techniques // Proc. IEEE/ACM Int. Conf. ICCAD. Nov. 2010. P. 729–734.
12. Chang K. H., Markov I., Bertacco V. Fixing design errors with counterexamples and resynthesis // IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., Jan. 2008. Vol. 27, N. 1. P. 184–188.
13. Cheng A.-C., Jiang I. H.-R., Jou J.-Y. Resource-aware functional ECO patch generation // Design, Automation & Test in Europe Conference & Exhibition (DATE). 2016. P. 1036–1041.
14. Cadence Encounter Conformal ECO Designer. URL: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/digital-design-and-signoff/functional-eco/conformal-eco-designer.html
15. Tang K.-F., Wu C.-A., Huang P.-K., Huang C.-Y. Interpolation-based incremental ECO synthesis for multi-error logic rectification // Design Automation Conference (DAC), 2011. P. 146–151.
16. Tang K.-F., Huang P.-K., Chou C.-N., Huang C.-Y. Multi-patch generation for multi-error logic rectification by interpolation with cofactor reduction // Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012. P. 1567–1572.
17. Lin C.-H., Huang Y.-C., Chang S.-C., Jone W.-B. Design and design automation of rectification logic for engineering change // Asia and South Pacific Design Automation Conference (ASP-DAC), 2005. P. 1006–1009.
18. Berkeley Logic Synthesis and Verification Group. ABC: A System for Sequential Synthesis and Verification. URL: <http://www.eecs.berkeley.edu/~alanmi/abc/>
19. Stempkovskiy A. L., Telpukhov D. V., Soloviev R. A. Fast and accurate resource-aware functional ECO patch generation tool // Proc. of Moscow Workshop on Electronic and Networking Technologies (MWENT), 2018. P. 1–5.
20. Исходный код средств автоматизации ресурсоориентированной функциональной коррекции логических схем. URL: https://github.com/IDMIPPM/Functional_ECO

A. L. Stempkovskiy, D. Sci., Scientific Director,

D. V. Telpukhov, Ph. D., Head of Department,

R. A. Solovyev, Ph. D., Chief Researcher, Y. V. Bitkov, Researcher,

Institute for Design Problems in Microelectronics (IPPM RAS), Moscow, 124365, Russian Federation,
nofrost@inbox.ru,

Development of Methods for Automating Resource-aware Functional ECO Patch Generation

In the modern design process of VLSIs, situations often arise when an incorrect operation of the circuit is detected only at the stage of final verification. Worse, when errors occur even after the final stage of design or production of microcircuits. Correction of errors at such late stages requires enormous labor and financial costs. In order to minimize these costs, methods have been developed for making the changes in the advanced stages of design process (engineering change order, ECO) for making corrections to the final design, instead of performing a complete redesign. At present, such corrections are mostly made manually, despite the fact that the main CAD vendors already have some automated tools for ECO.

In this paper, we propose methods for automating the functional correction of circuits based on conflicts, as well as methods for analyzing the structure of the circuit. Implemented software demonstrates high efficiency on a number of real-world tasks.

Keywords: engineering change order, ECO, functional correction, conflicts, ICCAD Contest.

DOI: 10.17587/it.24.755-762

References

1. Veneris A. G., Hajj I. N. A fast algorithm for locating and correcting simple design errors in VLSI digital circuits, *Proc. of Great Lake Symposium on VLSI Design*, 1997, pp. 45–50.
2. Abadir M., Ferguson J., Kirkland T. Logic design verification via test generation, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 1988, vol. 7, no. 1, pp. 138–148.
3. Chung P.-Y., Hajj I. N. Accord: Automatic catching and correction of logic design errors in combinational circuits, *Proc. Int. ITC, Sep. 1992*, pp. 742–751.
4. Huang S.-L., Lin W.-H., Huang P.-K., Huang C.-Y. Match and Replace: A Functional ECO Engine for Multierror Circuit Rectification, *Computer-Aided Design of Integrated Circuits and Systems IEEE Transactions*, 2013, vol. 32, pp. 467–478.
5. Brand D., Drumm A. D., Kundu S., Narain P. Incremental synthesis, *International Conference on Computer Aided Design*, 1994, pp. 14–18.
6. Krishnaswamy S., Ren H., Modi N., Puri R. DeltaSyn: An efficient logic difference optimizer for ECO synthesis, *International Conference on Computer Aided Design*, 2009, pp. 789–796.
7. Huang S.-Y., Chen K.-C., Cheng K.-T. AutoFix: A hybrid tool for automatic logic rectification, *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1999, vol. 18, no. 9, pp. 1376–1384.
8. Lin C.-C., Chen K.-C., Marek-Sadowska M. Logic synthesis for engineering change, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 1999, vol. 18, no. 3, pp. 282–292.
9. Yang Y.-S., Sinha S., Veneris A., Brayton R. Automating logic rectification by approximate SPFDs, *Proc. ASP-DAC, Jan. 2007*, pp. 402–407.
10. Ling A., Brown S., Safarpour S., Zhu J. Toward automated ECOs in FPGAs, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, Jan. 2011, vol. 30, no. 1, pp. 18–30.
11. Wu B.-H., Yang C.-J., Huang C.-Y., Jiang J.-H. A robust functional ECO engine by SAT proof minimization and interpolation techniques, *Proc. IEEE/ACM Int. Conf. ICCAD, Nov. 2010*, pp. 729–734.
12. Chang K. H., Markov I., Bertacco V. Fixing design errors with counterexamples and resynthesis, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, Jan. 2008, vol. 27, no. 1, pp. 184–188.
13. Cheng A.-C., Jiang I. H.-R., Jou J.-Y. Resource-aware functional ECO patch generation, *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 1036–1041.
14. Cadence Encounter Conformal ECO Designer, available at: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/digital-design-and-signoff/functional-eco/conformal-eco-designer.html
15. Tang K.-F., Wu C.-A., Huang P.-K., Huang C.-Y. Interpolation-based incremental ECO synthesis for multi-error logic rectification, *Design Automation Conference (DAC)*, 2011, pp. 146–151.
16. Tang K.-F., Huang P.-K., Chou C.-N., Huang C.-Y. Multi-patch generation for multi-error logic rectification by interpolation with cofactor reduction, *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2012, pp. 1567–1572.
17. Lin C.-H., Huang Y.-C., Chang S.-C., Jone W.-B. Design and design automation of rectification logic for engineering change, *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2005, pp. 1006–1009.
18. Berkeley Logic Synthesis and Verification Group. ABC: A System for Sequential Synthesis and Verification, available at: <http://www.eecs.berkeley.edu/~alanmi/abc/>.
19. Stempkovskiy A. L., Telpukhov D. V., Soloviev R. A. Fast and accurate resource-aware functional ECO patch generation tool, *Proc. Moscow Workshop on Electronic and Networking Technologies (Mwent)*, 2018, pp. 1–5.
20. IDMIPP / Functional-ECO, available at: https://github.com/IDMIPP/Functional_ECO

ГЛАВНОЕ СОБЫТИЕ В ОБЛАСТИ ПРИБОРОСТРОЕНИЯ, ТОЧНЫХ ИЗМЕРЕНИЙ, МЕТРОЛОГИИ И ИСПЫТАНИЙ

МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ ИННОВАЦИОННЫЙ ФОРУМ

ТОЧНЫЕ ИЗМЕРЕНИЯ – ОСНОВА КАЧЕСТВА И БЕЗОПАСНОСТИ

Москва, 15-17 мая 2019 года
ВДНХ, Павильон №75

ОРГАНИЗАТОРЫ:



ВЫСТАВОЧНЫЕ РАЗДЕЛЫ



METROLEXPO
МЕТРОЛОГИЯ, ИЗМЕРЕНИЯ И ИСПЫТАНИЯ



CONTROL&DIAGNOSTIC
КОНТРОЛЬ И ДИАГНОСТИКА



LABTEST
ЛАБОРАТОРНЫЕ ИСПЫТАНИЯ



PROMAUTOMATIC
ПРОМЫШЛЕННАЯ АВТОМАТИЗАЦИЯ



RESMETERING
УЧЁТ ЭНЕРГОРЕСУРСОВ



WEIGHT SALON
ВЕСОВОЙ САЛОН



ЦИФРЫ И ФАКТЫ 2018 года:

Участники – 296 компаний из 24 стран мира

Посетители – 5046 специалистов из 63 регионов России

Площадь экспозиции – 6870 м²

ДИРЕКЦИЯ ФОРУМА

Тел./Факс: +7 (495) 937-40-23

E-mail: metrol@expoprom.ru

Спешите забронировать стенд www.metrol.expoprom.ru

А. Б. Барский, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru,
Д. И. Мельник, канд. техн. наук, ст. науч. сотр., e-mail: mdi_dim@mail.ru,
НИИЦ (г. Москва) ЦНИИ ВВКО Минобороны РФ

Сетевая технология проектирования асинхронной вычислительной системы на общем решающем поле с самоуправляемой параллельной обработкой данных

Исследуется возможность построения макета управляющей двухпроцессорной вычислительной системы, обладающей общим решающим полем, с асинхронной обработкой данных по принципу data flow. Макет выполняется на базе мультимикропроцессорной системы "Сивуч-1", объединяющей 36 микропроцессоров "Эльбрус". Цель макетирования — создание инструментального средства для экспериментальных исследований в области достижения высокой производительности вычислительных средств и специального программного обеспечения, а также высокоустойчивых и безопасных вычислений.

Ключевые слова: сетевая технология, макетирование, data flow, процессор, вычислитель, процессор памяти

Введение

Реализация проектов гонки вооружений конца 70-х — начала 80-х гг. XX века стимулировала интенсивный поиск принципиально новых решений на пути создания высокопроизводительных вычислительных средств¹. В то время общее признание получил так называемый "японский вызов" (опубликован на русском языке в 1980 г.). В рамках этого вызова были сформулированы два пути решения проблемы:

1. Объединение в одной установке десятков тысяч микропроцессоров, где бы совместное решение задач регулировалось по принципу data flow.

2. Применение нейросетевых технологий для решения задач высокой сложности, трудноформализуемых задач и задач управления в реальном времени методом ассоциативных вычислений.

Первые научно-исследовательские разработки [1] в области "поточковых машин", высоко-

кая эффективность которых подкреплена результатами моделирования [2, 3], не вышли на экспериментальный уровень.

В настоящее время появились вычислительные средства, представляющие собой комплексы микропроцессоров "Эльбрус", объединенные на базе сетевой технологии Ethernet. К ним относится вычислительный комплекс (ВК) "Сивуч-1", который можно рассматривать как мини-сеть. Такая мини-сеть является не только превосходным средством моделирования распределенных систем и процессов, но позволяет на новом уровне рассмотреть возможность макетирования перспективных вычислительных систем (ВС) с учетом их функционирования в сложных управляющих системах.

1. Обоснование разработки макета

В предложенном ранее проекте [1–3] была реализована концепция мультимикропроцессорной системы на общем решающем поле [4]. Самоуправляемая параллельная обработка осуществлялась по "классической" схеме data flow. По этой схеме каждая команда не использует адреса операндов, а содержит места для размещения этих операндов, куда они поступают в результате вычислений или из памяти.

¹ Производительность вычислительного средства устанавливается на основе выполнения "машинных" или эквивалентных алгоритмических операций в секунду, а также по временным характеристикам контрольных (функциональных) или тестовых задач.

Наличие всех операндов в команде позволяет приступить к ее выполнению. В динамике вычислений готовыми к выполнению одновременно могут быть несколько команд. Результат их выполнения поступает либо в память (в текст заявки), либо в текст других, заранее распределенных команд. Вычислительный процесс оказывается связанным так, что естественным образом реализуются как параллельные, так и последовательные действия алгоритма.

Однако на этом применение "классической" идеи было закончено, так как ни одна известная попытка ее эффективной реализации не достигла успеха. Помехой тому служило резкое повышение требований к системе оперативного обмена информацией большим числом узлов внутри ВК, исключившее возможность практического использования такой привлекательной (с точки зрения программистов-математиков) идеи.

Поэтому авторами проекта было принято компромиссное решение (рис. 1), суть которого состоит в следующем.

Все микропроцессоры (вычислители), образующие решающее поле ВС (общее арифметическо-логическое устройство, АЛУ), обладают буферами, в которых динамически, в процессе выполнения программы процессором (процессорами — в многопроцессорной ВС), практически выполняющим функции устройства управления (УУ), формируются множества команд вида, свойственного *data flow*. На рис. 1 показан общий вид команды, где θ — код опе-

рации, r_1, \dots, r_4 — операнды, занявшие свои места, R — временно хранящийся результат, A_R — адрес, по которому результат должен быть отправлен.

Одним из вычислителей является *процессор памяти* (ПП). Как и вычислитель, он выполняет заявки асинхронно по мере поступления необходимой информации, но соблюдая частичную упорядоченность операций над памятью, обусловленную программой. В ходе анализа каждой команды ПП формирует заявки к оперативной *памяти данных* (ОПД), располагая адресом регистра буфера вычислителя, куда он должен послать результат считывания. Для выполнения необходимой записи в память вычислитель располагает адресом заявки на запись в буфере ПП (в *очереди заявок* (ОЗП)), куда он пошлет нужный результат. Затем эта заявка может быть рассмотрена для выполнения.

Таким образом, команды оказываются, во-первых, заранее, до установления их готовности к выполнению, распределенными между вычислителями, а во-вторых, динамически связанными между собой промежуточными результатами счета и с памятью данных в соответствии с реализуемым алгоритмом. Процесс коммутации вычислителей и процесс выполнения команд являются асинхронными. Процессоры наполняют буферы вычислителей и связывают их между собой, а также с памятью данных, а вычислители выполняют готовые команды в своем буфере и посылают результаты по сформированным адресам.

Программа такой ВС (программа коммутации) имеет традиционный вид и не предъявляет каких-либо особых требований к языкам программирования. Следует лишь отметить наличие в оттранслированной программе данных специального типа "*вычислитель*". Виртуальным адресом вычислителя удобно именовать результат промежуточных вычислений, подобно условным адресам "рабочих ячеек" при программировании ранних ЭВМ. Присутствующие в программе виртуальные адреса вычислителей образуют условную память, динамически распределяемую адресным генератором в процессе вычислений. Данные в этой памяти обладают короткой жизнью: от объявления до первого использования. После этого виртуальное имя вычислителя может быть использовано вновь.

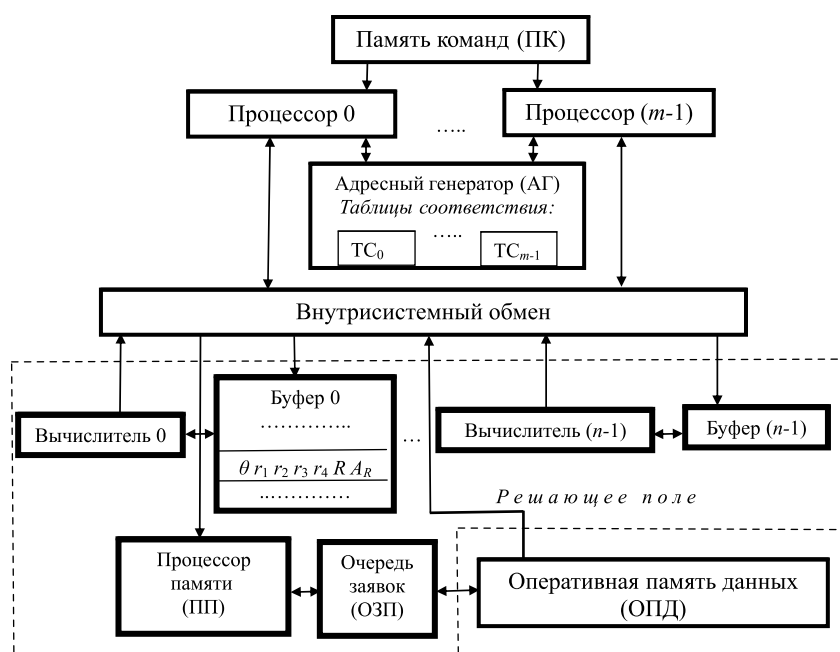


Рис. 1. Схема макетируемой вычислительной системы

Говоря об эффективности трансляции, следует отметить важность промежуточного представления алгоритма в бескобочной (инверсной) записи ПОЛИЗ (предложил польский математик Ян Лукашевич в 1929 г.), что характерно для распространенной технологии трансляции. Тогда при переходе к физическим ресурсам следующие подряд "параллельные" команды оказываются распределенными на "параллельные" вычислители, если их достаточно.

Для понимания принципиальной схемы вычислений кратко представим рассмотренный в работе [1] упрощенный пример программирования и даже трансляции для однопроцессорной ВС рассмотренного типа.

Пусть команда программы коммутации имеет традиционную трехадресную структуру $\theta A_1 A_2 A_3$, где A_1 и A_2 – адреса операндов (ОПД или регистра буфера вычислителя), A_3 – адрес регистра буфера того вычислителя, который будет выполнять инструкцию, сформированную на основе этой команды. Не рассматривая виртуализацию ресурса, адреса вычислителей будем задавать парой {номер вычислителя, номер регистра его буфера}.

Пусть при $n = 4$ следует написать программу счета значения выражения

$$A := ((a + b)(c + d)(e + f + g) - hij):(k(l - \sqrt{m})).$$

Как известно, в результате преобразования в ПОЛИЗ, на основе этой записи легко получается программа в безадресной системе команд, соответствующая выполнению операций на стеке:

$$ab + cd + \times ef + g + \times hi \times j \times - klm \sqrt{-} : \underline{zan} A,$$

где $\underline{zan} A$ – команда "запись по адресу A ".

При последовательном, послыном использовании регистров буферов вычислителей (в примере нумерация начинается с единицы) программа (коммутации) примет вид, показанный на рис. 2.

На рис. 3 представлена схема динамического формирования и коммутации инструкций в регистрах I_{ij} ($i, j = 1, \dots, 4$) буферов вычислителей и в очереди заявок к памяти. Для упрощения примера рассмотрены укороченные регистры, вмещающие только два операнда. В клетках попарно объединены заявки, формируемые при анализе команд программы. Знак $Сч$ означает считывание, знак \underline{zan} означает запись по адресу памяти. Динамика выполнения команд и заявок к памяти не отслеживается.

N_k	КОП	A_1	A_2	A_3
1	+	a	b	(1, 1)
2	+	c	d	(2, 1)
3	+	e	f	(3, 1)
4	\times	h	i	(4, 1)
5	$\sqrt{\quad}$	m		(1, 2)
6	\times	(1, 1)	(2, 1)	(2, 2)
7	+	(3, 1)	g	(3, 2)
8	\times	(4, 1)	j	(4, 2)
9	–	l	(1, 2)	(1, 3)
10	\times	(2, 2)	(3, 2)	(2, 3)
11	\times	k	(1, 3)	(3, 3)
12	–	(2, 3)	(4, 2)	(4, 3)
13	:	(4, 3)	(3, 3)	A

Рис. 2. Программа коммутации

Запрограммируем счет выражения

$$A := \text{if } (a + b) > c \text{ then if } (c - d) > > e \text{ then } X \text{ else } (B - D) \text{ else } Q.$$

Пусть команда УСЛ $A_1 A_2 A_3 A_4 A_5$ записывается в двух словах и интерпретируется как $A_3 := \text{if } (A_1) > (A_2) \text{ then } (A_4) \text{ else } (A_5)$.

Сформируем три команды для коммутации счета арифметических операторов, составляющих это выражение (рис. 4). Затем последовательно используем две команды УСЛ для коммутации внутренней и внешней конструкций типа "if – then – else".

Как видно, ни одной команды условного перехода, задерживающей конвейер выполнения команд, применить не пришлось.

Для составления программы в математических (виртуальных) адресах вычислителей предположим, что мы располагаем единственным условным вычислителем с буфером, объем которого определен специальной областью адресного пространства. Средствами виртуализации вычислительного ресурса в ВС, содержащей m процессоров коммутации, являются адресный генератор (АГ) и таблицы соответствия $ТС_k, k = 0, \dots, m - 1$.

Если любой адрес $A_\mu, \mu = 1, 2, 3$, команды программы, выполняемой на k -м процессоре, является математическим адресом вычислителя, выполняется обращение к $ТС_k$, состоящей

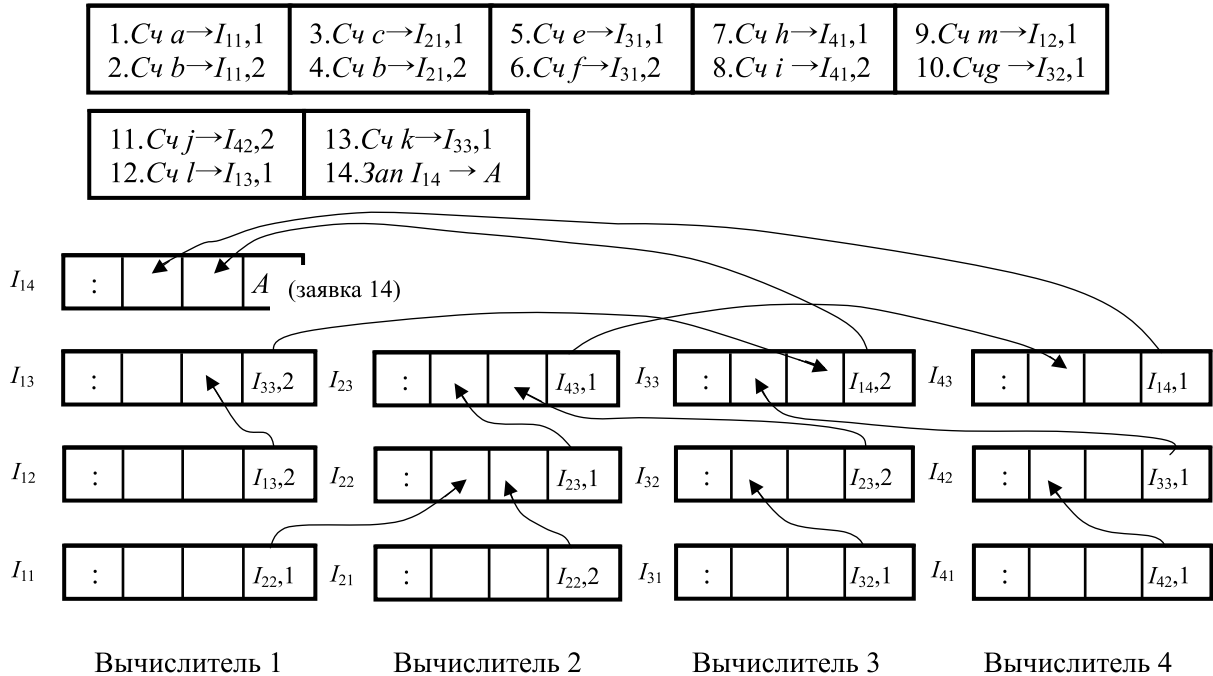


Рис. 3. Схема коммутации вычислителей

1.	+	a	b	(1, 1)
2.	-	c	d	(2, 1)
3.	-	B	D	(3, 1)
4.	УСЛ	(2, 1)	e	(4, 1)
		X	(3, 1)	
5.	УСЛ	(1, 1)	c	A
		(4, 1)	Q	

Рис. 4. Ветвление без условных переходов

из строк соответствия вида $v_i \rightarrow (i_j, s_j)$, где v_i – математический адрес некоторого вычислителя; (i_j, s_j) – соответствующий ему физический адрес. В случае совпадения $A_\mu = v_i$ фиксируется найденный физический адрес, а использованная строка исключается из $ТС_k$. Необходимость исключения строки следует из того, что каждый результат вычислений используется лишь один раз. При безуспешном обращении к $ТС_k$ АГ выдает очередной физический адрес вычислителя в порядке последовательной загрузки вычислителей решающего поля и при наличии свободных регистров в их буферах. Одновременно для данного математического адреса и найденного физического в первом

свободном регистре $ТС_k$ формируется новая строка соответствия.

Если по третьему адресу команды не указан математический адрес вычислителя (например, указан адрес ОПД), то АГ в порядке последовательной загрузки вычислителей формирует физический адрес вычислителя-исполнителя данной команды. По этому адресу записывается инструкция для выполнения операции, указанной в команде.

В рассмотренной в работах [1–3] модели многопроцессорной ВС (см. рис. 1) предполагалось, что АГ является автономным быстродействующим устройством, которое в соответствии с последовательной загрузкой вычислителей выдает всем процессорам одновременно адреса вычислителей для их использования при обработке адресов команд. Для ускорения его работы предлагалось использовать метод ситуационного управления, реализованный на ассоциативной памяти. Однако в макете предложен другой, значительно более простой способ виртуализации адресов вычислителей.

Упорядочение во времени выполнения заявок к ОПД, в которых указан один и тот же адрес, обеспечивается тем, что при выборе для исполнения адрес ячейки ОПД, указанный в каждой заявке, сравнивается с адресами ячеек ОПД, указанными в еще не исполненных, т. е. находящихся в ОПД, заявках, пришедших

ранее. Заявка может быть выполнена лишь в том случае, если выполнены все заявки по указанному в ней адресу, поступившие ранее.

2. Требования к макету

В основу разработки макета положены следующие требования:

1. Вычислительная система управления в реальном времени должна набираться из стандартных модулей в соответствии с функциональным назначением. Программы ВС должны быть инвариантны относительно размерности данных и относительно числа вычислителей.

2. Накладные расходы на управление параллельной работой большого числа модулей ВС должны быть минимальными.

3. Формируемые вычислительные установки должны соответствовать требованиям управления надежностью (избыточной комплектацией, ремонтпригодностью, составом ЗИП).

4. Разработка управляющей ВС диктует возврат к жесткому планированию использования оперативной памяти (ОП). Это связано с современными технологическими возможностями создания больших объемов прямоадресуемой, достаточно быстродействующей, структурированной ОП. Так, если в МК "Эльбрус-2" объем разделяемой ОП составлял 144 Мбайт, то в МК "Эльбрус 401-PC" оперативная память составляет 24 Гбайт. Отказ от трудоемких механизмов ведения (с помощью операционной системы (ОС)) виртуальной памяти обеспечивает весьма ощутимый резерв производительности для режима реального времени.

5. ОС не должна участвовать в вычислениях.

6. Система прерывания должна обеспечивать только временный режим и обслуживать только чрезвычайные ситуации.

7. Стекло процедур, в прошлом вызывающий нарекание разработчиков сложных систем, должен быть реализован только в том случае, если он не поддерживается ОС в режиме управления в реальном времени. (Замечательная команда "Передача управления с возвратом", например в БЭСМ-6, хорошо поддерживала высокую культуру программиста при структуризации программы для выделения "стандартных подпрограмм" — неоднократно используемых участков, которые можно интерпретировать процедурами.)

8. Необходимо снизить влияние условных переходов, задерживающих вычислительный процесс, вводом команды "if-then-else", допу-

скающей любую структуру вложений условного арифметического оператора присваивания.

9. Передача управления по адресу, являющемуся переменной с индексом, позволяет быстро переключать каналы обслуживания, если индекс канала совпадает с параметром цикла.

3. Инструментальное средство макетирования и структура макета

Возврат к идее построения ВС на указанных принципах сегодня обусловлен наличием инструментальных средств, позволяющих с уровня моделирования [3] перейти на уровень макетирования, т. е. создания "живой" архитектуры ВС, позволяющей реализовать рабочий режим, поиск оптимальных архитектурных и схемотехнических решений, внедрять предложения по организации вычислительного процесса, построения операционной системы, по обеспечению надежности, внедрять результаты других исследований.

Очевидно, на макете нельзя воспроизвести ожидаемые достоинства целевой ВС в части производительности, однако коэффициент полезной загрузки оборудования может быть определен с высокой достоверностью.

Предполагается в качестве инструментального средства макетирования взять многопроцессорный МК "Сивуч-1" с распределенной памятью [5].

Данный МК предназначен для решения задач многоканальной цифровой обработки больших объемов информации в режиме реального времени, для функционального и технического управления в области радиолокации, управления оптико-электронными и радиотехническими средствами, для реализации универсальных алгоритмов обработки информации на пунктах управления сложными системами.

Тридцать шесть процессоров МП1С1/V, объединенных в три устройства вычислителя УВ/С в его составе, связаны между собой на базе сетевой технологии *Ethernet* с помощью коммутатора *GbEthernet KGE*. Они условно образуют фрагмент локальной вычислительной сети. Таким образом, МК "Сивуч-1" является хорошим инструментальным средством нового направления — сетевого моделирования.

На этапе макетирования двухпроцессорной асинхронной системы представляется достаточным использование одного устройства УВ/С, содержащего 12 процессорных модулей МП1С1/V, объединенных, в свою очередь, в три ВС МП4С1/V *SYSTEM 0*, *SYSTEM 1*,

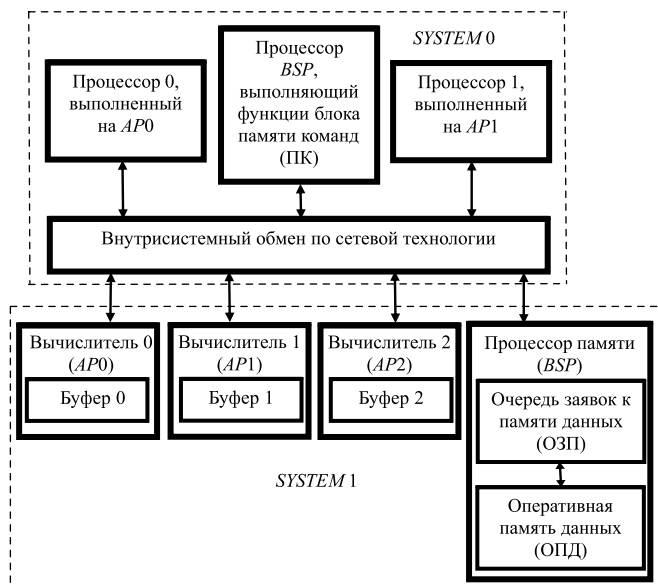


Рис. 5. Распределение функций устройств макетируемой ВС между процессорами ВК "Сивуч-1"

SYSTEM 2. Один модуль МП1С1/V ВС является главным (*Bootstrap, BSP*). Он выполняет начальную загрузку. Каждый из трех дополнительных модулей называется *Application (AP0, AP1, AP2)*.

На рис. 5 представлена схема распределения функций макета среди вычислительных модулей одного УВ/С ВК "Сивуч-1". Для первой очереди макетирования достаточно двух ВС — SYSTEM 0 и SYSTEM 1.

Как сказано выше, найдено решение, исключающее необходимость построения АГ в качестве отдельного устройства.

4. Структура и система команд

Команды макетируемой асинхронной ВС бывают нуль-, одно-, двух-, трех- и пятиадресными. Трехадресная команда образует командное слово основного формата

$$\theta J_1 A_1 J_2 A_2 J_3 A_3,$$

где θ — код операции, $J_j, j = 1, 2, 3$, — адреса регистров-модификаторов, A_j — смещения.

Для каждой команды формируются математические адреса

$$A'_j = (J_j) + A_j, j = 1, 2, 3.$$

Пятиадресная команда занимает два командных слова

$$\begin{aligned} &\theta J_1 A_1 J_2 A_2 J_3 A_3 \\ &0 J_4 A_4 J_5 A_5 0 0. \end{aligned}$$

Адреса A'_1, A'_2, A'_4, A'_5 определяют операнды. В арифметических, логических и операциях отношения эти адреса могут быть математическими (виртуальными) адресами вычислителей или ОПД, адреса A'_2 и A'_5 , кроме того, могут быть кодами целых чисел, непосредственно участвующих в операции, а также адресами модификаторов. Адрес A'_3 определяет результат и может быть адресом ПК, модификатора, ОПД, вычислителя, на котором выполняется данная операция и по адресу которого, следовательно, присваивается значение результата операции.

При макетировании предполагается, что указанный выше тип адресной информации определяется по тегами. Например, два старших дополнительных разряда могут определять теги: 00 — число или адрес ПК, 01 — адрес модификатора, 10 — адрес ОПД, 11 — адрес вычислителя.

В работах [2, 3] подробно обсуждается весьма развитая система команд арифметических и логических операций, операций отношения, условные арифметические операции, операции пересылки и преобразования кодов, операции индексации, групповые, векторные операции и операции цикла.

Там же подробно обсуждаются команды передачи управления. Хотя механизм выполнения вложенных процедур на стеке подвергается критике в применении к управляющим вычислительным средствам, целесообразно для универсальных применений воспользоваться возможностями, уже заложенными в микропроцессорах "Эльбрус" ВК "Сивуч-1". Поэтому предлагаются такие команды, как переход с возвратом внутри процедуры и переход на следующий лексикографический уровень с возвратом. Команда условного перехода ставится в зависимость от сигнала о выработке условия. До появления условия перехода могут выполняться другие команды.

Впервые в соответствии с концепцией системы управления как многоканальной системы массового обслуживания предлагается использовать команду переключения каналов. По этой команде запоминается текущее значение регистра команд и счетчика команд для программы обрабатываемого канала и вводятся или восстанавливаются соответствующие значения для программы обработки канала следующего номера (по $\text{mod} \langle \text{число каналов процессора} \rangle$). С каналами связаны базовые регистры, указывающие на их данные. Выполнение этой команды в цикле позволяет внутри одного

такта управления обработать все объекты, закрепленные за каналами в соответствии со стадией их обслуживания. Один канал головного процессора П0 может закрепляться за супервизором, отслеживающим обстановку и принимающим решения по дальнейшему ходу управления системой: о переходе на следующие стадии обслуживания объектов, о включении новых объектов, о реконфигурации системы и т. д. По всем каналам процессора П1 может проводиться только обслуживание.

С учетом возможностей микропрограммирования следует сделать вывод о целесообразности такого развития системы команд, чтобы обусловленные ими работы были достаточно объемны, существенно загружали вычислители, снижали относительные затраты на управление распараллеливанием, по смыслу приближались к реальным процедурам [3]. Ряд команд должен соответствовать выполнению многих элементарных функций. Специальные применения ВС диктуют и специализацию развиваемой системы команд.

5. Алгоритмы работы устройств макета

1. Обработка команд процессором. Три уровня обработки команд процессором подробно рассмотрены в работе [1]. Однако в связи с отсутствием АГ в макете предлагается другая схема формирования физических адресов вычислителей, реализуемая совместно процессором и выделенным им вычислителем. Вычислитель сообщает процессору адрес регистра, в котором будет формироваться инструкция вычислителю по анализируемой процессором команде. По этой информации в таблице соответствия процессора $ТС_k$, $k = 0, 1$, формируется строка вида " $v \rightarrow$ (номер вычислителя, адрес его буфера)", где v — встретившийся в программе виртуальный адрес вычислителя, выполняющего данную команду.

Рассмотрим эту схему подробнее.

При анализе очередной команды k -м процессором первый раз выбирается вычислитель $j = k$. Затем проводится переадресация для выбора вычислителя при анализе следующей команды: $j = (k + 2)_{\text{mod}2}$. Таким образом осуществляется распределение вычислите-

лей, подобно тому, как распределяются данные между исполнителями по *SPMD*-технологии.

Вычислитель воспринимает свое назначение как заявку на предоставление свободного регистра своего буфера для организации в нем инструкции вычислителю, соответствующей данной команде процессора. Для этого все адреса регистров буфера динамически образуют два списка: *список используемых адресов* и *список освободившихся адресов*. Когда список используемых адресов иссякает, список освободившихся адресов становится списком используемых адресов, а опустевший список используемых адресов становится списком освободившихся адресов.

На рис. 6, а для буфера, состоящего из шести регистров, показано состояние указанных списков в момент начала вычислений. На рис. 6, б показано возможное состояние списков в некоторый момент вычислений с учетом разного времени выполнения инструкций.

2. Обработка инструкций вычислителями.

Вычислитель занимает выделенный регистр пришедшим от процессора кодом операции и передает сообщение этому процессору для формирования строки соответствия в его ТС, в каком регистре располагается новая инструкция. Поскольку каждая инструкция готова к выполнению, то как только в ее текст поступили все операнды, при каждой записи в регистр буфера проверяется и устанавливается признак готовности. Адрес готовой инструкции дополняет очередь "к выполнению". После выполнения инструкции вычислитель направляет результат по информации третьего адреса. Однако в связи с асинхронностью процесса в данный момент адрес отправки результата может еще быть неизвестен. Тогда результат временно сохраняется в отведенном месте текста инструкции, и инструкция остается в очереди "к выполнению", пока в ее текст

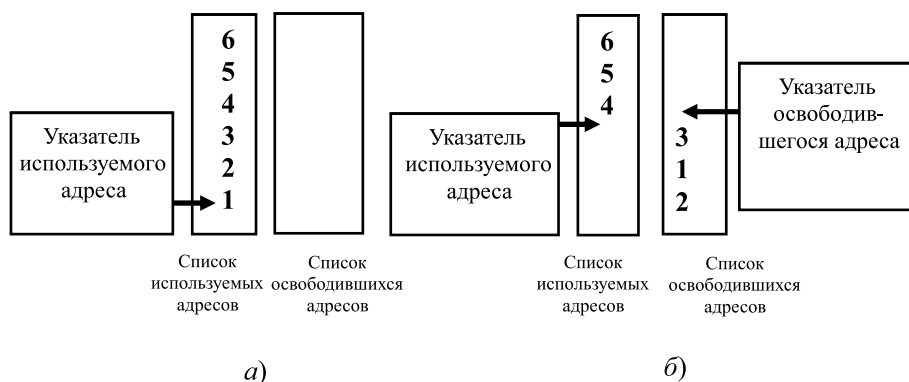


Рис. 6. Возможное состояние списков

не поступит адрес отправления. Выполненные инструкции освобождают занимаемые ими регистры. Адреса этих регистров дополняют список свободных регистров буфера данного вычислителя.

3. Процессор памяти. Принцип работы этого процессора был кратко освещен выше. Добавим, что если ОПД имеет модульную структуру и к одному модулю памяти не может быть одновременно выполнено более одного обращения, то ОЗП формируется к каждому модулю отдельно. Таким образом, заявки к каждому модулю памяти выполняются последовательно в порядке их поступления, а модули памяти работают параллельно.

Однако, если в некоторую заявку на запись по адресу не поступил код, то этот адрес блокирует выполнение последующих заявок к данному модулю с использованием этого же адреса до следующего обзора очереди к модулю. Так исключается возможное нарушение алгоритмического порядка записи и считывания при использовании адреса одной и той же ячейки памяти.

6. Рекомендации по оптимизации обмена в макетируемой ВС

Ранние попытки комплексирования многих ЭВМ для совместного решения задач привели к выводу об исключительно высоких расходах времени на обмен. Этот вывод стимулировал создание многопроцессорных вычислительных систем на общей оперативной памяти, обоснованных В. М. Бахаревым. Хотя первую такую систему CDC-6600 создал С. Крей, справедливо считать, что разработчики МВК "Эльбрус", где главным конструктором был В. С. Бурцев, были пионерами действительно практического, результативного решения, позволившего закрыть ряд важнейших государственных программ 1980-х годов.

Этот опыт в применении к макету говорит, что число запросов на обмен внутри системы следует минимизировать, и развести обмен во времени и пространстве таким образом, чтобы не создавать пиковых нагрузок на главных направлениях.

Ориентируясь на реальную разработку в будущем, в работах [1, 2] предлагается применить шесть основных независимых магистралей обмена данными, связывающих устройства вычислительной системы. Функции ряда второстепенных магистралей могут быть реализованы ими. Магистрали могут быть распределены

за направлениями обмена или быть связанными коммутатором для динамического выбора менее загруженной магистрали для каждого обмена. При наличии коммутатора направления обмена между конкретными модулями ВС могут обладать плавающим приоритетом для установления "равноправия" модулей.

Для сравнения напомним, что, по-видимому, в последней модели С. Крея *Cray Superserver 6400 System (CS6400)* на базе 64 процессоров *SuperSPARC* используются четыре разделяемых магистрали.

Заключение

В основе предлагаемого макета ВС лежит утверждение: познание методов и программирование решения конкретных задач диктуют не только требование к архитектуре и системе команд, но и предлагают простейшие приемы организации взаимодействия модулей при выполнении их функций. В этом смысле вычислительная установка должна быть максимально простой [6, 7]. Разработчик, не сведущий в математических вопросах программирования, не должен "грузить" свою модель всем тем, что, как ему представляется, может понадобиться программисту (когда-нибудь) и на чем можно продемонстрировать кажущуюся эффективность своей разработки. В этом случае изделие неоправданно разрастается, требуя технологического прорыва, стоимость неоправданно растет. Проверка многих параметров при испытаниях становится недоступной.

В работе [3] методом моделирования исследована эффективность предлагаемой архитектуры при решении конкретных задач. Под эффективностью понимается значение коэффициента загрузки вычислителей. Оказывается, что этот коэффициент для единственной решаемой задачи не столь высок, что известно и для многих других параллельных архитектур. Другое дело, когда организована многоканальная обработка информации при управлении многими объектами. В этом случае много задач решаются по многим каналам одновременно. Тогда можно считать, что в сбалансированной системе с достаточно высоким коэффициентом загрузки оборудования число вычислителей должно в три...четыре раза превышать число каналов обслуживания. Поиск оптимальной архитектуры вычислительных средств приводит к особой актуальности исследования их *реальной производительности* в составе сложных управляющих систем.

Включение в состав макета процессора, выполняющего функции памяти команд, наталкивает на вывод о превращении его в *HOST*-процессор, выполняющий основные функции операционной системы и супервизора, а также представляющий данную вычислительную установку во внешних связях.

Список литературы

1. Барский А. Б., Русаков А. Н., Хвоин Б. И. Возможности достижения высокой скорости коммутации и внутрисистемного обмена в вычислительной системе, управляемой потоком данных // Вопросы кибернетики. Разработка и использование суперЭВМ. М., НС по комплексной проблеме "Кибернетика", 1987. С. 130–144.

2. Барский А. Б., Шилов В. В. Вычислительная система, управляемая потоком данных // Информационные технологии. Приложение. 2000. № 8. 24 с.

3. Барский А. Б., Шилов В. В. Потокковая вычислительная система: программирование и оценка эффективности // Информационные технологии. Приложение. 2003. № 7. 24 с.

4. Игнатущенко В. В. Организация структур управляющих многопроцессорных вычислительных систем. М.: Энергоатомиздат, 1984. 184 с.

5. Вычислительный комплекс "Сивуч-1". Руководство по эксплуатации. М.: АО "МЦСТ", 2014. 124 с.

6. Бурцев В. С. Система массового параллелизма с автоматическим распределением аппаратных средств суперЭВМ в процессе решения задачи // Юбилейный сборник трудов институтов ОИВТА РАН. Т. П. М.: ОИВТА РАН, 1993. С. 5–27.

7. Змеев Д. Н., Климов А. В., Левченко Н. Н., Окунев А. С., Стемповский А. Л. Потокковая модель вычислений как парадигма программирования будущего // Информатика и ее применение. 2015. Т. 9, Вып. 4. С. 29–36.

A. B. Barsky, D. Sc., Prof., e-mail: arkbarsk@mail.ru,

D. I. Melnik, Cand. Tech. Sci., Senior Researcher, e-mail: mdi_dim@mail.ru,

NIIC (Moscow) Central Research Institute of the Air Defense Ministry of the Russian Federation

Network Technology for Designing an Asynchronous Computing System on a Common Decision Field with Self-Controlled Parallel Processing of Data

The possibility of constructing a model of a controlling two-processor computing system with a common deciding field with asynchronous data processing based on the data flow principle is investigated. The architecture of the layout is based on a compromise: The switching program executed on the processor has a traditional form, implements the traditional command system and is the subject of user development with the help of a translator. However, its commands are not executed immediately. On their basis, the computation of calculators for joint calculations is performed. The mock-up system implements the management mode of the multi-channel queuing system. On the basis of microprogramming, it is possible to develop a command system for the specialization of the control system.

The model is executed on the basis of the multi-microprocessor system "Sivuch-1", which unites 36 microprocessors "Elbrus". The purpose of prototyping is to create a tool for experimental research in the field of achieving high performance of computing facilities and special software, as well as highly stable and secure computing. Based on the results of prototyping, design proposals can be issued.

Keywords: network technology, prototyping, data flow, processor, calculator, memory processor

DOI: 10.17587/it.24.763-771

References

1. Barskij A. B., Rusakov A. N., Hvojn B. I. Vozmozhnosti dostizheniya vysokoj ckorociti kommutacii i vnutrisistemnogo obmena v vychislitel'noj sisteme, upravlyaemoj potokom dannyh (Opportunities for achieving a high commutation and communication speed in a computing system managed by the flow of data), *Voprosy kibernetiki. Razrabotka i ispol'zovanie super-EVM*, Moscow, NS po kompleksnoj probleme "Kibernetika", 1987, pp. 130–144 (in Russian).

2. Barskij A. B., Shilov V. V. Vychislitel'naya sistema, upravlyaemaya potokom dannyh (Computing system managed by the data flow), *Informacionnye Tehnologii, Prilozhenie*, 2000, no. 8, 24 p. (in Russian).

3. Barskij A. B., Shilov V. V. Potokovaya vychislitel'naya sistema: programmirovaniye i ocenka effektivnosti (The dataflow computing system: programming and efficiency estimation), *Informacionnye Tehnologii, Prilozhenie*, 2003, no. 7, 24 p. (in Russian).

4. Ignatushhenko V. V. Organizaciya struktur upravlyajushhih mnogoprocessornyh vychislitel'nyh sistem (Organization of management structures of multiprocessor computing systems), Moscow, Energoatomizdat, 1984, 184 p. (in Russian).

5. *Vychislitel'nyj kompleks "Sivuch-1". Rukovodstvo po ekspluatcii* ("Sivuch-1" computer complex. Manual), Moscow, AO "MCST", 2014, 124 p. (in Russian).

6. Burcev V. S. Cistema massovogo parallelizma s avtomaticheskim raspredeleniem apparatnyh sredstv super-EVM v processe resheniya zadachi (The system of mass parallelism with automatic distribution of supercomputer hardware in the process of solving the problem), *Jubilejnyj sbornik tpudov institutov OIVTA RAN*, T. P. M., OIVTA RAN, 1993, pp. 5–27 (in Russian).

7. Zmeev D. N., Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovskij A. L. Potokovaya model' vychislenij kak paradigma programmirovaniya budusshego (Flow model of computing as a programming paradigm for the future), *Informatika i eyo Primenenie*, 2015, vol. 9, iss. 4, pp. 29–36 (in Russian).

В. А. Богатырев, д-р техн. наук, проф., e-mail: vladimir.bogatyrev@gmail.com,
С. А. Паршутина, аспирант, e-mail: svetlana.parshutina@gmail.com,
Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Анализ влияния кратности резервирования многопутевых передач на вероятность их своевременного обслуживания

Исследованы возможности повышения вероятности своевременного безошибочного обслуживания в системах многопутевой передачи данных в результате резервированного выполнения критичных к задержкам запросов. На основе результатов имитационного моделирования проанализирована эффективность резервированных многопутевых передач в распределенных компьютерных системах, в которых реализуется доставка копий пересылаемых по многим путям пакетов в один адресуемый узел и/или многопутевое распределение запросов к одному или нескольким узлам обработки, готовым к их обслуживанию.

Ключевые слова: резервирование, многопутевая передача, распределение запросов, критичность к времени пребывания

Введение

Ключевой проблемой проектирования распределенных компьютерных систем и сетей является обеспечение надежности [1–3], безопасности [4–6] и производительности обработки, передачи и хранения данных при взаимодействии узлов через сеть [7–9]. Для реализации надежного взаимодействия узлов распределенной компьютерной системы необходимо обеспечить не только ее структурную надежность и отказоустойчивость за счет резервирования путей, обеспечивающих связанность ее узлов, но и функциональную надежность (устойчивость функционирования) процессов межмашинного обмена через сеть в условиях сбоев, отказов и внешних деструктивных воздействий. Особенно остро проблема обеспечения надежности взаимодействия стоит для систем реального времени, в которых решаются задачи с высокими требованиями к своевременности выполнения и нередко отсутствует возможность повторных передач. В таких системах формируются жесткие требования к задержкам обслуживания запросов; несоблюдение этих требований приводит к нарушениям условий своевременности выполнения запросов при межмашинном обмене.

Повышение качества обслуживания потоков при взаимодействии компьютерных узлов через сеть может достигаться на основе многопутевых передач (в том числе на основе многопутевой маршрутизации) [10–15].

Под маршрутом (путем) понимается последовательность узлов, задействованных при передаче пакетов от узла-источника к узлу-приемнику (адресату). При многопутевой передаче предусматривается наличие двух и более маршрутов между конечными узлами, при этом маршруты могут быть непересекающимися (dis-joint) или могут иметь общие узлы и линии связи. Многопутевая маршрутизация эффективна при высоких нагрузках в сети [16], и значительная часть разрабатываемых моделей нацелена на решение задачи балансировки нагрузки [17–19] для предотвращения и устранения перегрузок и оптимального использования сетевых ресурсов. Превалирует подход, когда распределение потоков данных по разным маршрутам происходит в зависимости от пропускной способности и загрузки линий связи, при этом часто решают задачу минимизации максимальной загрузки каналов сети [20], в программно-конфигурируемых сетях — минимизации очередей в маршрутизаторах [16]. Кроме того, получили распространение модели с выравниванием нагрузки в сети, учитывающие надежность путей на основе критериев: коэффициента готовности каналов [21]; вероятностей их отказов, известных априори [22].

Многопутевая маршрутизация также широко используется для повышения надежности и отказоустойчивости передач в условиях отказов узлов и линий связи, когда один маршрут выступает в роли основного, а остальные — в роли резервных (горячего резерва) [23]. При

отказе основного пути происходит перенаправление потоков по альтернативным маршрутам без потери времени на поиск нового пути.

Распределение пакетов одного потока или потоков от одного узла-источника по разным маршрутам способствует рациональному использованию сетевых ресурсов, предупреждению недоставки пакетов вследствие возникновения перегрузок в сети и потери времени на поиск новых маршрутов в случае отказа основного и, в результате, увеличению вероятности и уменьшению времени передачи пакетов адресату (адресатам).

Известные подходы к повышению надежности систем с многопутевой маршрутизацией связаны со структурным резервированием путей, в то время как функциональная надежность взаимодействия компьютерных систем может быть повышена за счет резервированных передач, когда при использовании существующей совокупности маршрутов реализуется резервирование пересылаемых пакетов (создание их копий), передаваемых по разным физическим путям.

В рамках использования механизмов многопутевых передач в целях повышения надежности взаимосвязи компьютерных систем через сеть в данной статье исследуются возможности повышения вероятности своевременного обслуживания критичных к задержкам запросов на основе их резервированного обслуживания.

Как показано в работах [24–26], снизить среднее время ожидания и повысить вероятность своевременности выполнения критичных к задержкам запросов удастся при их резервированном обслуживании. Исследование многоканальных систем массового обслуживания (СМО) с общей очередью, в которых копии запросов, выполняемых в разных каналах, создаются только при наличии свободных каналов, проведено в работах [27, 28]. Такая дисциплина обслуживания не ориентирована на принципиальное для систем реального времени повышение вероятности своевременного выполнения всех критичных к задержкам запросов. Резервированное обслуживание запросов с заданием числа создаваемых копий в зависимости от их критичности к задержкам ожидания в очередях для одноуровневых кластерных систем, представляемых группой одноканальных СМО с локальными очередями, предложено и исследовано в работах [24–26]. Резервированное обслуживание критичных к ожиданию запросов считается успешным,

если своевременно выполнена хотя бы одна копия запроса. Применение резервированного обслуживания к системам передачи данных с резервированием каналов исследовано в работе [7]. В работах [29, 30] предложено развитие концепции резервирования процессов обработки и передачи данных применительно к многоуровневым системам, предполагающим многоэтапное (последовательное) выполнение копий запросов в узлах разных уровней. Резервированное многоэтапное обслуживание критичных к выполнению запросов считается успешным, если хотя бы для одной копии суммарное (накопленное) время ожидания в последовательно обслуживающих ее узлах (представляемых одноканальными СМО) не превышает заданную предельно допустимую задержку. Для одно- и многоуровневых кластеров эффективность резервированных процессов обслуживания удастся повысить при приоритезации копий запросов в зависимости от их критичности к ожиданию в очередях [30] и при уничтожении неактуальных копий [31, 32] ("просроченных" копий, находящихся в очередях, когда одна из них уже обслужена).

При многопутевой резервированной передаче предлагается создание копий пакетов при их отправке по разным маршрутам с кратностью резервирования (копирования), зависящей от критичности пакетов к допустимым задержкам [24–26, 30–32].

В системах распределенной обработки данных возможна организация взаимодействия, включающего передачу данных (сообщений, пакетов) и распределение (перераспределение) запросов [12, 15, 33, 34]. Таким образом, возможны два направления использования резервированных многопутевых передач:

- S_1 — для резервированной передачи пакетов данных по нескольким путям в один узел-приемник, при этом пересылаемый пакет считается переданным успешно, если за заданное время хотя бы одна его копия доставлена адресату;
- S_2 — для резервированного распределения запросов по нескольким путям к одному или более (согласно заданному числу) узлам-приемникам из некоторого множества возможных адресатов, при этом запрос считается успешно выполненным, если за заданное время хотя бы одна его копия доставлена в один или в заданное число узлов, готовых к ее выполнению.

Цель работы — исследование возможностей повышения надежности и своевременности взаимодействия компьютерных систем при многопутевой резервированной передаче в зависимости от кратности резервирования формируемых копий пакетов при их отправке в узел-приемник (узлы-приемники) по разным маршрутам.

Следует подчеркнуть, что резервированное обслуживание запросов (в том числе на передачу данных через сеть) связано с разрешением технического противоречия, так как, с одной стороны, обслуживание части или всех копий запросов приводит к увеличению загрузки, а значит, к росту времени ожидания копий запросов в очередях, но, с другой стороны (в силу стохастичности задержек передачи по разным путям), существует вероятность того, что некоторая копия будет обслужена значительно быстрее остальных. Если при межмашинном обмене достаточно своевременности обслуживания хотя бы одной копии запроса (пакета), выполнение этого условия приводит к повышению вероятности своевременного обслуживания запросов и уменьшению среднего времени пребывания запроса в сети. Резервированная пересылка пакетов через сеть в условиях возможных ошибок передач усиливает эффект повышения вероятности безошибочности и, следовательно, своевременности передач через сеть.

Разрешение указанного технического противоречия связано с поиском оптимальной кратности резервирования передач в зависимости от интенсивности потоков запросов и их критичности к задержкам обслуживания (к предельно допустимому времени пребывания запросов в сети).

Для достижения поставленной цели впервые решаются задачи анализа эффективности резервированных многопутевых передач в распределенных компьютерных системах, в которых реализуется доставка копий пересылаемых по многим путям пакетов в один адресуемый узел и/или многопутевое распределение запросов к одному или нескольким узлам обработки, готовым к их обслуживанию.

Метод решения задач исследования основан на применении имитационного моделирования.

Имитационные модели многопутевых резервированных передач

Предлагаемые модели разработаны в среде имитационного моделирования AnyLogic 7

Professional в рамках дискретно-событийного подхода, при котором моделируемые процессы представляют в виде последовательности операций, производимых над некими сущностями (например, заявками, требованиями, запросами) с использованием специальных компонентов модели из библиотеки "Process Modeling Library" — "блоков" ("blocks").

Процессы многопутевых резервированных передач данных по одному адресу и распределения запросов по адресам входящих в группу серверов в виде потоковых диаграмм (flowcharts) показаны соответственно на рис. 1, *а* и *б*, при этом выделены этапы:

- создания запросов и их копий в источнике ("блоки" 'source', 'req_clone') с подготовкой к отправке, включающей упаковку формируемых пакетов в кадр используемого протокола и их передачу в канал ('buffer', 'delay');
- выбора пути следования запроса к узлу-получателю из пяти доступных маршрутов, организованных в виде виртуальных каналов, в "блоке" 'path1_5';
- обслуживания запроса в коммуникационных узлах (маршрутизаторах) 'router1', 'router2', ... с возможностью его удаления из системы в "блоках" 'loss1', 'loss2', ..., 'loss5_N';
- обслуживания запроса узлом-получателем ('server', 'server1'—'server5') с возможностью его удаления из системы в "блоках" 'P0', 'P0_1'—'P0_5', 'timeout', 'timeout1'—'timeout5', 'relevant', 'relevant1'—'relevant5';
- выхода запроса из системы ('serviced', 'serviced1'—'serviced5').

Модели на рис. 1 представляют ситуации, в которых имеется пять вариантов (маршрутов) следования запросов от источника до адресуемого сервера (рис. 1, *а*) или пяти серверов кластера (рис. 1, *б*) с четырьмя промежуточными узлами на каждом пути. Маршруты не пересекаются, т. е. не имеют общих узлов.

Задаются следующие параметры модели: интенсивность потока запросов Λ , c^{-1} ; кратность (число создаваемых копий) резервирования передач k ; число пересылаемых запросов N , шт.; максимальное (предельно допустимое) время ожидания t_0 , c ; пропускные способности линий связи BW , бит/с; протяженности сегментов линий связи L , м; задержки в коммуникационных узлах (маршрутизаторах) RD , c ; задержки в узлах назначения (серверах) SD , c ; вероятности (или условия) потерь запросов при передаче LR ; вероятности (интенсивности)

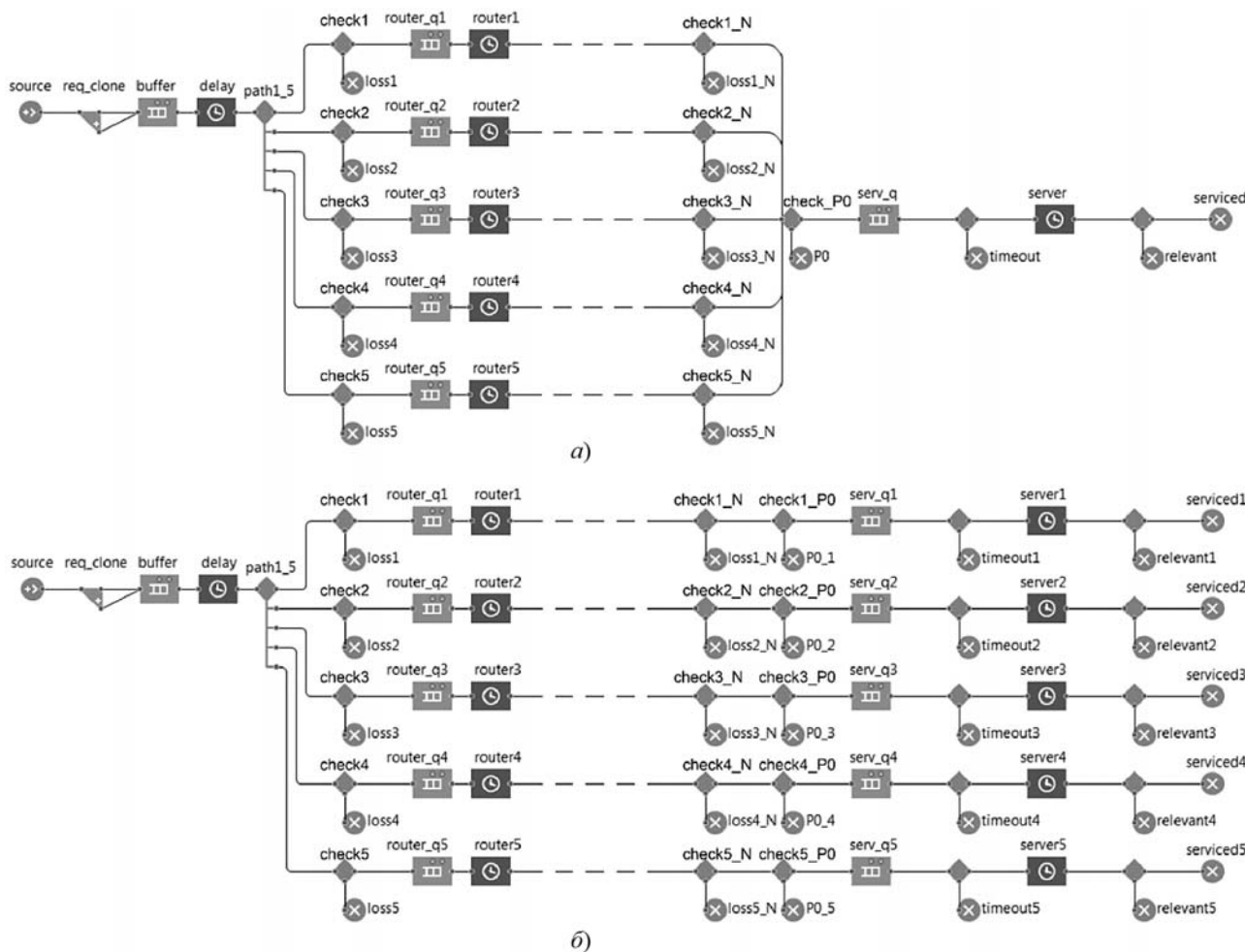


Рис. 1. Фрагмент модели сети с многопутевой передачей данных по одному адресу (а) и многопутевым распределением запросов (б)

битовых ошибок BER ; вероятности готовности узлов назначения (серверов) к обслуживанию запросов P_0 ; вероятность (или алгоритм) обнаружения битовых ошибок в запросах коммуникационными узлами P_1 ; размеры запросов (пакетов данных) RS , бит. В модели также задается: максимальное число и размер буферов выходного порта источника, входных портов промежуточных узлов и узлов назначения. Максимальное число буферов определяется соответственно емкостью (максимальной длиной) очередей — "блока" 'buffer', "блоков" 'router_q1', 'router_q2', ... — RQ (шт.) и "блоков" 'serv_q', 'serv_q1'—'serv_q5' — SQ (шт.). Для простоты будем считать, что размеры буферов совпадают с максимальным размером пересылаемых пакетов (кадров).

Перемещение запросов от одного "блока" к другому с точки зрения модельного времени происходит мгновенно, за исключением "блоков" типа *Delay*, которые запросы покидают по истечении заданного времени или при выпол-

нении некоторого условия. Примем, что задержка запросов при передаче по линиям связи либо отсутствует, либо учитывается в очередях входных интерфейсов следующих узлов. В "блоке" 'source' скорость создания запросов определяется через "интенсивность прибытия" (rate), или экспоненциально распределенное число запросов в единицу времени; "время между прибытиями" (interarrival time), заданное константой или функцией, например, вероятностного распределения, по умолчанию — экспоненциального; расписание (schedule), в том числе загружаемое из базы данных.

Каждому запросу присваивается порядковый номер $reqNumber$ от 1 до N — максимального числа поступающих в систему запросов. Начало его пребывания в системе $startTime$ — текущее время, размер в битах $reqSize$ (например, размер кадра формата Ethernet DIX (II) в компьютерных сетях, построенных на основе технологий семейства Ethernet, от 512 до 12 144 бит). При кратности резервирования

передач $k > 1$ в "блоке" 'req_clone' создается $(k - 1)$ идентичных копий запроса. Запросы и их копии помещаются в "очередь" 'buffer', после чего по одному попадают в "блок" 'delay', в котором задерживаются на время $reqSize/BW$. Далее запросы распределяются по маршрутам ('path1_5') на основе алгоритма кругового обслуживания (Round Robin) и либо удаляются из системы с некоторой вероятностью или по срабатыванию некоторого условия ('loss1', 'loss2', ..., 'loss5_N'), либо попадают в очереди промежуточных узлов 'router_q1', 'router_q2', ... и ожидают обслуживания.

Поток запросов — однородный; дисциплина обслуживания — беспriorитетная; выбор запросов осуществляется в порядке их поступления в очереди 'router_q1', 'router_q2', ..., 'serv_q', 'serv_q1'—'serv_q5' (FIFO).

Задержка обслуживания в "блоках" 'router1', 'router2', ..., 'server', 'server1'—'server5': $RD = SD = t_{prg} + buffTime + procTime$, где t_{prg} — время распространения сигнала; $buffTime$ — время поступления запроса в принимающий буфер; $procTime$ — время обработки запроса (задержка коммутации).

Время распространения сигнала $t_{prg} = L/S$, где L , м — протяженность сегментов линий связи; S , м/с — скорость распространения электромагнитных волн, которая зависит от физической среды и колеблется от 0,6 до 0,9 скорости света в вакууме [35], для витой пары $\approx 0,7$.

Время буферизации $buffTime = (reqSize + 64)/BW + 0,96 \cdot 10^{-6}$, включая преамбулу кадра в 64 бит и время межкадрового интервала $0,96 \cdot 10^{-6}$ с, BW — пропускная способность сегмента линии связи, бит/с.

Достигшие узла назначения запросы либо обслуживаются ('serviced', 'serviced1'—'serviced5'), либо досрочно покидают систему в случае их неактуальности или неготовности серверов ('P0', 'P0_1'—'P0_5'). Запрос теряет актуальность, если время его пребывания в системе (разница между текущим модельным временем и $startTime$) превысило максимальное время ожидания ("блоки" 'timeout', 'timeout1'—'timeout5') или уже была обслужена ранее другая его копия ("блоки" 'relevant', 'relevant1'—'relevant5').

Максимальное время ожидания t_0 задается исходя из требований вычислительного процесса (конкретного приложения/программы).

После выхода каждого запроса из системы осуществляется проверка условия окончания эксперимента: сумма доставленных в узлы на-

значения и досрочно покинувших систему запросов достигает значения kN .

Результаты эксперимента — статистика по числу успешно пересланных запросов, досрочно покинувших систему запросов, минимальному, среднему и максимальному времени пребывания запросов в системе.

Исследование влияния кратности резервирования передач на вероятность своевременного обслуживания запросов при учете интенсивности потоков запросов (трафика) и их критичности к времени пребывания в сети (ограничений на предельно допустимое время пребывания t_0) предполагает анализ результатов многочисленных имитационных экспериментов. Инструментальные средства поддержки проведения серии имитационных экспериментов с варьирующимися значениями параметров модели были предложены в работе [36].

Исследование многопутевых резервированных передач, критичных к времени ожидания

Проанализируем эффективность резервированного взаимодействия в инфокоммуникационной системе в случае резервированной передачи данных по нескольким путям в один узел-приемник (S_1) и в случае резервированного распределения запросов по нескольким путям к одному из нескольких возможных приемников (S_2) [12—15, 34].

При проведении имитационных экспериментов предположим, что используются одинаковые маршрутизаторы, серверы и сегменты линий связи. Линии связи имеют пропускную способность $BW = 100$ Мбит/с. Время распространения сигнала $t_{prg} = 100/0,7 \cdot 2,99 \cdot 10^8 \approx 5 \cdot 10^{-7}$ с, где $2,99 \cdot 10^8$ м/с — скорость света в вакууме; $N = 5 \cdot 10^4$ шт.; $RS = 4096$ бит; $LR = 0$; $BER = 0$; $P_0 = 1$; $P_1 = 1$. Задержки обслуживания в маршрутизаторах RD и серверах SD заданы с помощью встроенной функции экспоненциального распределения 'exponential' в AnyLogic с параметром $\lambda = 6,5$ и минимальным значением аргумента соответственно $RD_{min} = 2,5 \cdot 10^{-6}$ с и $SD_{min} = 10^{-6}$ с.

На рис. 2 показана установленная экспериментально зависимость среднего времени пребывания запросов в системе T , с от интенсивности потока запросов Λ , с⁻¹ при $t_0 = 15$ с для кратностей резервирования передач $k = 1, 2, \dots, 5$ (кривые 1—5 соответственно). Рис. 2, а пред-

ставляет вариант резервированного взаимодействия S_1 , а рис. 2, б — вариант S_2 .

Как видно из рис. 2, существует граница интенсивности потока запросов, ниже которой резервирование процессов передачи (рис. 2, а) и распределения запросов (рис. 2, б) позволяет снизить среднее время пребывания пакетов (запросов) в системе.

Рассмотрим случай многопутевой резервированной передачи данных (S_1) и многопутевого резервированного распределения запросов (S_2) с учетом возможных потерь при обслуживании из-за превышения максимального времени ожидания t_0 .

На рис. 3 представлены результаты моделирования при $t_0 = 2,1 \cdot 10^{-3}$ с и кратности резервирования передач $k = 1, 2, \dots, 5$ (кривые 1—5 соответственно).

Как видно из рис. 2 и рис. 3, изменение максимального времени ожидания может оказывать влияние на диапазон значений интенсив-

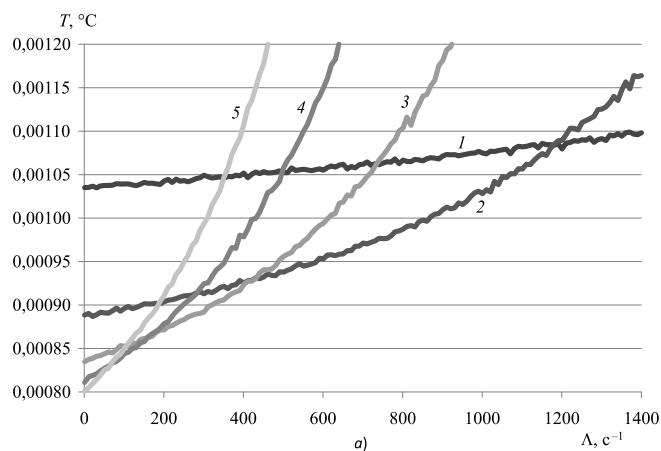


Рис. 2. Среднее время пребывания в системе в случае многопутевой резервированной передачи пакетов (а) и многопутевого резервированного распределения запросов (б) при $t_0 = 15$ с

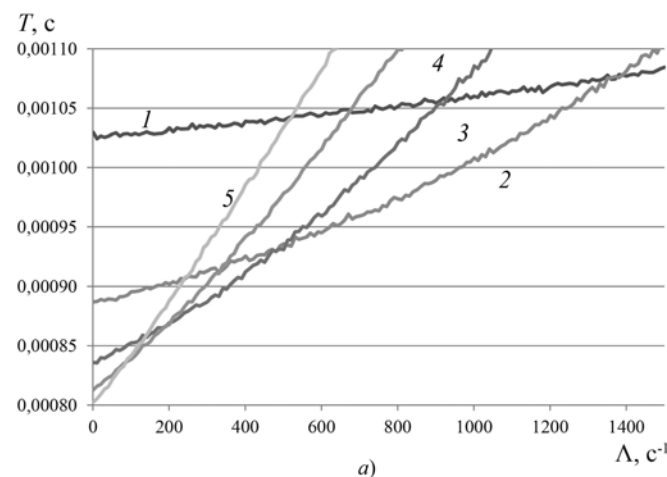
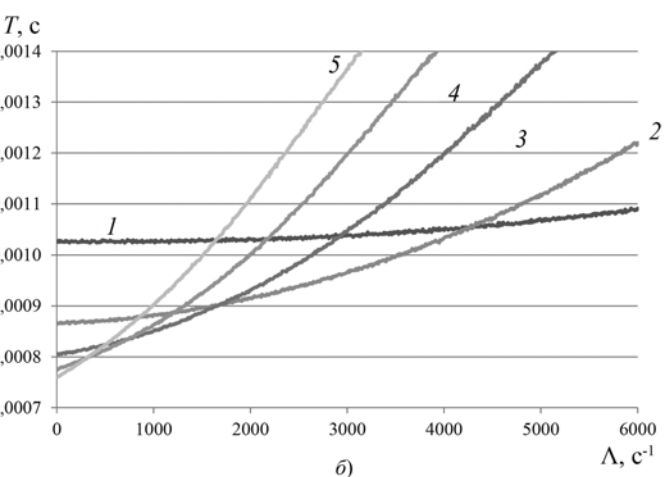
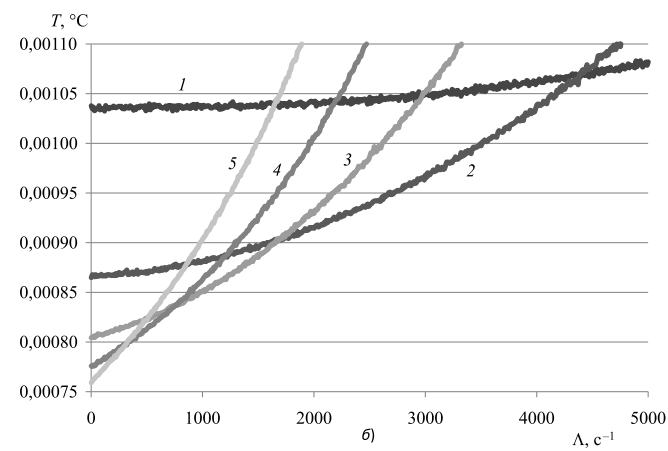


Рис. 3. Среднее время пребывания в системе в случае многопутевой резервированной передачи пакетов (а) и многопутевого резервированного распределения запросов (б) при $t_0 = 2,1 \cdot 10^{-3}$ с

ности потока запросов, при котором резервирование передач целесообразно.

На рис. 4 (см. вторую сторону обложки) показано существование границы эффективности (целесообразности) многопутевой резервированной передачи пакетов (а) и многопутевого резервированного распределения запросов через сеть (б), когда резервирование передач приводит к увеличению вероятности своевременности доставки запросов в узлы назначения P по сравнению с вариантом без резервирования. При увеличении t_0 повышается доля успешно переданных пакетов. По мере возрастания интенсивности потока запросов увеличивается нагрузка в системе и снижается вероятность успешных передач. Установлено, что при уменьшении вероятности доставки P до значений 0,05 и ниже существенно возрастает вариативность среднего времени пребывания запросов в системе. Таким образом, область целесообразности многопутевого резервиро-



ванного обслуживания запросов существует и для передач пакетов по одному адресу (S_1) и для распределения запросов через сеть (S_2).

На рис. 5 (см. вторую сторону обложки) показана зависимость среднего времени пребывания запросов в системе и вероятности их своевременного распределения (доставки к серверам) от их размера: 512 бит (кривые 7–9), 12 144 бит (кривые 1–3) и переменного размера (512, 1024, 2048, 4096, 8192 или 12 144 бит; кривые 4–6). В последнем случае размер пакета (кадра Ethernet) задается согласно равномерному распределению. При этом $t_0 = 2,1 \cdot 10^{-3}$ с, кратность резервирования передач $k = 1$ (кривые 1, 4, 7), $k = 2$ (кривые 2, 5, 8) и $k = 3$ (кривые 3, 6, 9). Здесь $k = 1$ — случай без резервирования.

Анализ резервированного распределения запросов через сеть с учетом ошибок передач и неготовности серверов

Рассмотрим многопутевое резервированное распределение запросов через сеть с учетом их возможных потерь из-за ошибок передач (BER), сбояв и иных ошибок, например, программного и аппаратного обеспечения (LR), неготовности узлов назначения (P_0). Будем предполагать, что возникающие битовые ошибки обнаруживаются промежуточными узлами с вероятностью P_1 , при этом искаженные пакеты всегда обнаруживаются и отбрасываются узлами назначения.

Размер запроса RS варьируется.

На рис. 6 (см. вторую сторону обложки) представлена вероятность доставки запросов к серверам кластера P при $LR = 10^{-4}$ (кривые 1–3 — рис. 6, а и 1–6 — рис. 6, б), $LR = 10^{-3}$ (кривые 4–6 — рис. 6, а), $BER = 10^{-7}$ (кривые 1–3 — рис. 6, а и 1–6 — рис. 6, б), $BER = 10^{-5}$ (кривые 4–6 — рис. 6, а), $P_0 = 0,99$ (кривые 1–3 — рис. 6, а и 1–6 — рис. 6, б), $P_0 = 0,95$ (кривые 4–6 — рис. 6, а), кратности резервирования $k = 1$ соответствуют кривые 1, 4, $k = 2$ — кривые 2, 5, $k = 3$ — кривые 3, 6.

Зависимости на рис. 6, а (см. вторую сторону обложки) построены в предположении идеальности контроля по обнаружению битовых ошибок ($P_1 = 1$). Результаты имитационных экспериментов, приведенные на рис. 6, б, учитывают возможность необнаружения ошибки средствами контроля, при этом вероятности обнаружения ошибки $P_1 = 0,9$ соответствуют кривые

1–3, а случаю отсутствия контроля, когда $P_1 = 0$, соответствуют кривые 4–6. Как видно из рис. 6, а, резервированное многопутевое распределение запросов через сеть в условиях ненадежной доставки позволяет повысить вероятность своевременного распределения запросов. При возрастании интенсивности потока запросов вероятность своевременности резервированного распределения снижается. В то же самое время из-за потерь вследствие ошибок передач и неготовности серверов (кривые 4–6) уменьшается длительность ожидания в очередях узлов и, таким образом, время пребывания запросов в системе, что приводит к повышению вероятности успешных передач в определенном диапазоне интенсивности потока запросов.

Необнаружение битовых ошибок (кривые 4–6), по сравнению со случаем с высокой вероятностью нахождения искажений в пересылаемых пакетах в промежуточных узлах (кривые 1–3), ведет к повышению нагрузки, времени пребывания запросов в системе, уменьшению вероятности своевременного распределения запросов через сеть (рис. 6, б, см. вторую сторону обложки).

На рис. 7 показана вероятность своевременного распределения запросов P при $t_0 = 2,1 \cdot 10^{-3}$ с, $LR = 10^{-3}$, $BER = 10^{-5}$, $P_0 = 0,95$, $P_1 = 1$ и интенсивности потока запросов $\Lambda = 3951, 3776, 2951, 2126, 326$ с $^{-1}$ в зависимости от кратности резервирования передач k .

При решении задачи максимизации доли своевременно и безошибочно распределенных через сеть критичных к задержкам запросов и/или при ограничениях повторных передач представленная зависимость указывает на су-

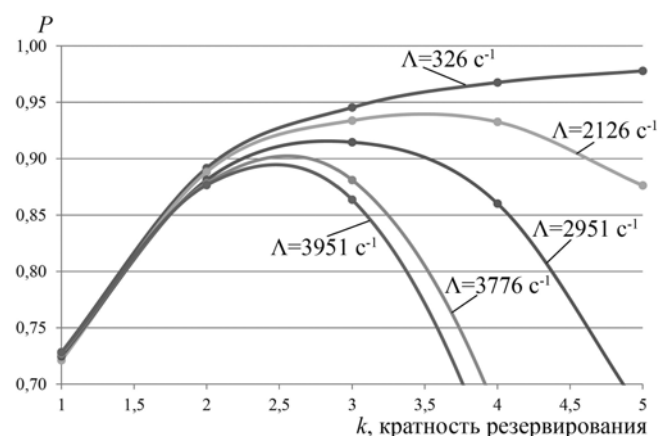


Рис. 7. Вероятность доставки запросов к серверам кластера в зависимости от кратности резервирования передач k

существование оптимальной кратности резервирования запросов в зависимости от интенсивности потока запросов.

Заключение

Исследованы возможности повышения вероятности своевременного безошибочного обслуживания и снижения средних задержек в системах многопутевой передачи данных в результате резервированного выполнения критичных к задержкам запросов.

На основе результатов имитационного моделирования проанализирована эффективность резервированных многопутевых передач в распределенных компьютерных системах, в которых реализуется доставка копий пересылаемых по многим путям пакетов в один адресуемый узел и/или многопутевое распределение запросов к одному или нескольким узлам обработки, готовым к их обслуживанию. Модели разработаны в рамках дискретно-событийного подхода в среде имитационного моделирования AnyLogic 7 Professional.

Показано существование области эффективности резервированной многопутевой передачи данных (пакетов) и резервированного многопутевого распределения запросов в зависимости от их критичности к задержкам и загруженности системы.

Показано существование оптимальной кратности резервирования при многопутевом распределении запросов, обеспечивающей максимизацию вероятности своевременно и безошибочно распределенных через сеть критичных к задержкам запросов или запросов, распределяемых в условиях, исключающих возможность повторных передач.

Установлена зависимость оптимальной кратности резервирования при многопутевом распределении запросов через сеть от интенсивности потока запросов и их критичности к задержкам.

Список литературы

1. **Kopetz H.** Real-Time Systems: Design Principles for Distributed Embedded Applications. Springer, 2011. P. 396.
2. **Sorin D.** Fault Tolerant Computer Architecture. Morgan & Claypool, 2009. 103 p.
3. **Шубинский И. Б.** Надежные отказоустойчивые информационные системы. Методы синтеза. Ульяновск: Областная типография "Печатный двор", 2016. 544 с.
4. **Советов Б. Я., Колбанев М. О., Татарникова Т. М.** Технологии инфокоммуникации и их роль в обеспечении

информационной безопасности // Геополитика и безопасность. 2014. № 1(25). С. 69–77.

5. **Верзун Н. А., Колбанев М. О., Татарникова Т. М.** Технологическая платформа четвертой промышленной революции // Геополитика и безопасность. 2016. № 2 (34). С. 73–78.

6. **Gatchin Y. A., Zharinov I. O., Korobeynikov A. G., Zharinov O. O.** Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems // Modern Applied Science. 2015. Vol. 9, N. 5. P. 197–210.

7. **Богатырев В. А., Богатырев С. В.** Резервированная передача данных через агрегированные каналы в сети реального времени // Известия высших учебных заведений. Приборостроение. 2016. Т. 59. № 9. С. 735–740.

8. **Богатырев В. А.** Надежность и эффективность резервированных компьютерных сетей // Информационные технологии. 2006. № 9. С. 25–30.

9. **Богатырев В. А.** Комбинаторно-вероятностная оценка надежности и отказоустойчивости кластерных систем // Приборы и системы. Управление, контроль, диагностика. 2006. № 6. С. 21–26.

10. **Новиков С. Н.** Классификация методов маршрутизации в мультисервисных сетях связи // Вестник СибГУТИ. 2013. № 1. С. 57–67.

11. **Шувалов В. П., Варакина И. Ю.** Классификация методов многопутевой маршрутизации // Т-Comm #1. 2014. С. 29–32.

12. **Богатырев В. А., Паршутина С. А.** Модели многопутевой отказоустойчивой маршрутизации при распределении запросов через сеть // Вестник компьютерных и информационных технологий. 2015. № 12. С. 23–28.

13. **Богатырев В. А., Паршутина С. А.** Многопутевое резервированное распределение через сеть критичных к задержкам запросов // Вестник компьютерных и информационных технологий. 2016. № 10 (148). С. 41–46.

14. **Bogatyrev V. A., Parshutina S. A.** Efficiency of Redundant Multipath Transmission of Requests Through the Network to Destination Servers // Communications in Computer and Information Science, IET. 2016. Vol. 678. P. 290–301.

15. **Bogatyrev V. A., Parshutina S. A.** Redundant Distribution of Requests Through the Network by Transferring Them Over Multiple Paths // Communications in Computer and Information Science, IET. 2016. Vol. 601. P. 199–207.

16. **Лемешко А. В., Вавенко Т. В.** Усовершенствование потоковой модели многопутевой маршрутизации на основе балансировки нагрузки // Проблемы телекоммуникаций. 2012. № 1 (6). С. 12–29.

17. **Gunnar A., Johansson M.** Robust load balancing under traffic uncertainty — tractable models and efficient algorithms // Telecommun Syst. 2011. 48:93–107.

18. **Banner R., Orda A.** Multipath Routing Algorithms for Congestion Minimization, CCIT Report No. 429, Department of Electrical Engineering, Technion, Haifa, Israel, 2004. URL: <http://www.ee.technion.ac.il/people/ron/Congestion.pdf>.

19. **Жуков И. А., Кулаков Ю. А., Шпак И. Ю.** Адаптивная многопутевая маршрутизация // Проблемы информатизации та управління. 2009. № 3 (27). С. 73–78.

20. **Merindol P.** Improving Load Balancing with Multipath Routing / P. Merindol, J. Pansiot, S. Cateloin // Proc. of the 17-th International Conference on Computer Communications and Networks, IEEE ICCCN 2008. 2008. P. 54–61.

21. **Шувалов В. П., Селянина И. Ю.** Методика обеспечения отказоустойчивости в мультисервисных сетях связи // Проблемы информатики. 2012. № 2 (14). С. 55–62.

22. **Rajeev V., Muthukrishnan C. R.** Reliable backup routing in fault tolerant real-time networks // Proceedings. Ninth IEEE International Conference on Networks, ICON, 2001.

23. **Будылдина Н. В., Трибунский Д. С., Шувалов В. П.** Оптимизация сетей с многопротокольной коммутацией по меткам. М.: Горячая линия-Телеком, 2010. 144 с.

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

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

26. Богатырев В. А., Богатырев А. В. Надежность функционирования кластерных систем реального времени с фрагментацией и резервированным обслуживанием запросов // Информационные технологии. 2016. Т. 22, № 6. С. 409–416.

27. Дудин А. Н., Сунь Б. Многолинейная система MAP/PN с управляемым широкополосным обслуживанием ненадежными приборами // Автоматика и вычислит. техника. 2009. Т. 43, № 5. С. 32–43.

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

29. Богатырев В. А., Богатырев С. В. Своевременность обслуживания в многоуровневых кластерных системах с поэтапным уничтожением просроченных запросов // Вестник компьютерных и информационных технологий. 2018. № 2 (164). С. 28–35.

30. Богатырев В. А., Богатырев С. В. Резервированное обслуживание запросов, критичных к задержкам ожидания, в двухуровневых системах // Известия высших учебных заведений. Приборостроение. 2017. Т. 60, № 10. С. 945–950.

31. Богатырев В. А., Богатырев С. В. Резервированное обслуживание в группе одноканальных систем с назначением различных приоритетов копиям запроса // Известия высших учебных заведений. Приборостроение. 2017. Т. 60, № 11. С. 1033–1039.

32. Богатырев В. А., Богатырев С. В. Резервированное обслуживание в кластерах с уничтожением неактуальных запросов // Вестник компьютерных и информационных технологий. 2017. № 1 (151). С. 21–28.

33. Богатырев В. А., Богатырев А. В. Модель резервированного обслуживания запросов реального времени в компьютерном кластере // Информационные технологии. 2016. Т. 22, № 5. С. 348–355.

34. Богатырев В. А., Богатырев С. В. Надежность мультикластерных систем с перераспределением потоков запросов // Известия высших учебных заведений. Приборостроение. 2017. Т. 60, № 2. С. 171–177.

35. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы. СПб.: Питер, 2016. 992 с.

36. Паршутина С. А. Организация имитационных экспериментов при проектировании распределенных компьютерных систем с резервированной передачей данных // Научно-технический вестник информационных технологий, механики и оптики. 2017, Т. 17, № 5. С. 841–849.

V. A. Bogatyrev, D. Sc., Professor, e-mail: vladimir.bogatyrev@gmail.com,

S. A. Parshutina, Postgraduate, e-mail: svetlana.parshutina@gmail.com,

St. Petersburg National Research University of Information Technologies, Mechanics and Optics

The Analysis of How the Redundancy Degree of Multipath Transmissions Affects the Probability of their Timely Service

The possibility of using redundant processing of delay-sensitive requests to increase the probability of timely faultless service and reduce delays in multipath data transmission systems is investigated. Simulation is used to evaluate the effectiveness of redundant multipath transmissions: the process of sending multiple copies of packets along different paths in a given network, in order to deliver them to one destination node or to one of several service nodes which are ready to execute incoming requests. The models are developed based on the discrete-event approach in AnyLogic 7 Professional simulation environment.

The degree of redundancy defines the number of copies of packages which are distributed among different routes in the network and depends on the sensitivity of these packets to time delays.

In distributed computer systems, the redundant multipath data transmission and the redundant multipath distribution of requests are considered the ways of increasing the efficiency of communication between computational processes which exchange messages that are critical to service delays over the network.

The redundant data transmission means that data packets are cloned and copies of those packets travel along several paths to one destination node. Data are regarded to be transmitted successfully if at least one copy of each packet has been delivered to that node in time, i.e. within a given period of time. The redundant distribution of requests over the network implies that copies of requests are forwarded along several routes to one or more destination nodes, which have been selected out of all possible ones in some way. A request is regarded to be distributed successfully if at least one copy of it has been delivered to all selected nodes within the specified time and those nodes have been ready for serving incoming requests. The effectiveness of redundant multipath transmissions — when the time of delivery of packets is limited — is determined by the probability of timely faultless service.

The existence of the scope of efficiency for the redundant multipath data transmissions and the redundant multipath distribution of requests, with considering their sensitivity to time delays, is shown.

It is found that there exists the optimal degree of redundancy that ensures the maximization of the probability of timely and faultless distribution of the multiplied requests over a given network when those requests are critical to time delays and/or can be transmitted only once.

It was also discovered that the optimal redundancy degree of multipath transmissions depends on the intensity of the flow of requests and their sensitivity to time delays.

Keywords: redundancy, multipath transmission, distribution of requests, sensitivity to time delays

DOI: 10.17587/it.24.772-781

References

1. **Kopetz H.** Real-Time Systems: Design Principles for Distributed Embedded Applications, Springer, 2011, 396 p.
2. **Sorin D.** Fault Tolerant Computer Architecture, Morgan & Claypool 2009, 103 p.
3. **Shubinskij I. B.** *Nadezhnye otkazoustojchivye informacionnye sistemy. Metody sinteza* (Reliable fault-tolerant information systems. Synthesis method), Ul'yanovsk, Oblastnaya tipografiya "Pechatnyj dvor", 2016, 544 p. (in Russian).
4. **Sovetov B. Ya., Kolbanyov M. O., Tatarnikova T. M.** Tekhnologii infokommunikacii i ih rol' v obespechenii informacionnoj bezopasnosti (Infocommunication technologies and their role in information security), *Geopolitika i bezopasnost'*, 2014, no. 1(25), pp. 69–77 (in Russian).
5. **Verzun N. A., Kolbanev M. O., Tatarnikova T. M.** Tekhnologicheskaya platforma chetvertoj promyshlennoj revolyucii (Technology platform of the fourth industrial revolution), *Geopolitika i Bezopasnost'*, 2016, no. 2 (34), pp. 73–78 (in Russian).
6. **Gatchin Y. A., Zharinov I. O., Korobeynikov A. G., Zharinov O. O.** Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems, *Modern Applied Science*, 2015, vol. 9, no. 5, pp. 197–210.
7. **Bogatyrev V. A., Bogatyrev S. V.** Rezervirovannaya poredacha dannyh cherez agregirovannye kanaly v seti real'nogo vremeni (Redundant data transmission through aggregated channels in real-time network), *Izvestiya vuzov. Priborostroenie*, 2016, vol. 59, no. 9, pp. 735–740 (in Russian).
8. **Bogatyrev V. A.** Nadezhnost' i ehffektivnost' rezervirovannyh komp'yuternyh setej (Reliability and efficiency of redundant computer networks), *Informacionnye Tekhnologii*, 2006, no. 9, pp. 25–30 (in Russian).
9. **Bogatyrev V. A.** Kombinatorno-veroyatnostnaya ocenka nadezhnosti i otkazoustojchivosti klasternyh sistem (Combinatorial probabilistic evaluation of reliability and fault tolerance of cluster systems), *Pribory i sistemy. Upravlenie, kontrol', diagnostika*, 2006, no. 6, pp. 21–26 (in Russian).
10. **Novikov S. N.** Klassifikaciya metodov marshrutizacii v mul'tiservisnyh setyah svyazi (Classification of routing methods in multiservice communication networks), *Vestnik SibGUTI*, no. 1, 2013, pp. 57–67 (in Russian).
11. **Shuvalov V. P., Varaksina I. Yu.** Klassifikaciya metodov mnogoputevoj marshrutizacii (Classification of multipath routing methods), *T-Comm*, 2014, no. 1, pp. 29–32 (in Russian).
12. **Bogatyrev V. A., Parshutina S. A.** Modeli mnogoputevoj otkazoustojchivoj marshrutizacii pri raspredelenii zaprosov cherez set' (Multipath fault-tolerant routing models for distributing queries through the network), *Vestnik Komp'yuternyh i Informacionnyh Tekhnologij*, 2015, no. 12, pp. 23–28 (in Russian).
13. **Bogatyrev V. A., Parshutina S. A.** Mnogoputevoe rezervirovannoe raspredelenie cherez set' kritichnyh k zaderzhkam zaprosov (Multipath redundant distribution of delay-sensitive queries through the network), *Vestnik Komp'yuternyh i Informacionnyh Tekhnologij*, 2016, no. 10 (148), pp. 41–46 (in Russian).
14. **Bogatyrev V. A., Parshutina S. A.** Efficiency of Redundant Multipath Transmission of Requests Through the Network to Destination Servers, *Communications in Computer and Information Science, IET* – 2016, vol. 678, pp. 290–301.
15. **Bogatyrev V. A., Parshutina S. A.** Redundant Distribution of Requests Through the Network by Transferring Them Over Multiple Paths, *Communications in Computer and Information Science, IET* – 2016, vol. 601, pp. 199–207.
16. **Lemeshko A. V., Vavenko T. V.** Usovshenstvovanie potokovoj modeli mnogoputevoj marshrutizacii na osnove balansirovki nagruzki (Improving the threading model of multipath routing based on load balancing), *Problemy Telekomunikacij*, 2012, no. 1 (6), pp. 12–29 (in Russian).
17. **Gunnar A., Johansson M.** Robust load balancing under traffic uncertainty – tractable models and efficient algorithms, *Telecommun Syst.*, 2011, vol. 48, pp. 93–107.
18. **Banner R., Orda A.** Multipath Routing Algorithms for Congestion Minimization, CCIT Report No. 429, Department of Electrical Engineering, Technion, Haifa, Israel, 2004, available at: <http://www.ee.technion.ac.il/people/ron/Congestion.pdf>.
19. **Zhukov I. A., Kulakov Yu. A., Shpak I. Yu.** Adaptivnaya mnogoputevaya marshrutizaciya (Adaptive multipath routing), *Problemy Informatizacii ta Upravlinnya*, 2009, no. 3 (27), pp. 73–78 (in Russian).
20. **Merindol P., Pansiot J., Catelein S.** Improving Load Balancing with Multipath Routing, *Proc. of the 17-th International Conference on Computer Communications and Networks, IEEE ICCCN 2008*, 2008, pp. 54–61.
21. **Shuvalov V. P., Selyanina I. Yu.** Metodika obespecheniya otkazoustojchivosti v mul'tiservisnyh setyah svyazi (A method of providing fault tolerance in multi-service networks), *Problemy Informatiki*, 2012, no. 2(14), pp. 55–62 (in Russian).
22. **Rajeev V., Muthukrishnan C. R.** Reliable backup routing in fault tolerant real-time networks, *Proceedings. Ninth IEEE International Conference on Networks, ICON*, 2001.
23. **Budyldina N. V., Tribunskij D. S., Shuvalov V. P.** Optimizaciya setej s mnogoprotokol'noj kommutaciej po metkam. (Optimization of networks with Multiprotocol label switching), Moscow, Goryachaya liniya-Telekom, 2010, 144 p. (in Russian).
24. **Bogatyrev V. A., Bogatyrev A. V.** Optimizaciya rezervirovannogo raspredeleniya zaprosov v klasternyh sistemah real'nogo vremeni (Optimization of redundant query distribution in real-time cluster systems), *Informacionnye Tekhnologii*, 2015, vol. 21, no. 7, pp. 495–502 (in Russian).
25. **Bogatyrev V. A., Bogatyrev A. V.** Functional Reliability of a Real-Time Redundant Computational Process in Cluster Architecture Systems, *Automatic Control and Computer Sciences, IET*, 2015, vol. 49, no. 1, pp. 46–56.
26. **Bogatyrev V. A., Bogatyrev A. V.** Nadezhnost' funkcionirovaniya klasternyh sistem real'nogo vremeni s fragmentaciej i rezervirovannyh obsluzhivaniem zaprosov (Reliable operation of real-time cluster systems with fragmentation and redundant service requests), *Informacionnye Tekhnologii*, 2016, vol. 22, no. 6, pp. 409–416 (in Russian).
27. **Dudin A. N., Sun' B.** Mnogolinejnaya sistema MAP/PH/N s upravlyaemym shirokoveshchatel'nym obsluzhivaniem nenadezhnymi priborami (Multi-line MAP/PH/N system with managed broadcast service by unreliable devices), *Avtomatika i Vychislit. Tekhnika*, 2009, vol. 43, no. 5, pp. 32–43.
28. **Dudin A. N., Sun' B.** Mnogolinejnaya nenadezhnaya sistema s upravlyaemym shirokoveshchatel'nym obsluzhivaniem (Multi-line unreliable system with managed broadcast service), *Avtomatika i Telemekhanika*, 2009, vol. 70, no. 12, pp. 147–160 (in Russian).
29. **Bogatyrev V. A., Bogatyrev S. V.** Svoevremennost' obsluzhivaniya v mnogourovnevnyh klasternyh sistemah s poehtapnym unichtozheniem prosrochennyh zaprosov (Timeliness of service in multi-level cluster systems with phased elimination of overdue requests), *Vestnik Komp'yuternyh i Informacionnyh Tekhnologij*, 2018, no. 2(164), pp. 28–35 (in Russian).
30. **Bogatyrev V. A., Bogatyrev S. V.** Rezervirovannoe obsluzhivanie zaprosov, kritichnyh k zaderzhkam ozhidaniya, v dvuhurovnevnyh sistemah (Redundant service for latency-critical requests in two-tier systems), *Izvestiya vuzov. Priborostroenie*, 2017, vol. 60, no. 10, pp. 945–950 (in Russian).
31. **Bogatyrev V. A., Bogatyrev S. V.** Rezervirovannoe obsluzhivanie v gruppe odnokanal'nyh sistem s naznacheniem razlichnyh prioritetov kopiyam zaprosa (Redundant service in a group of single-channel systems with different priorities assigned to query copies), *Izvestiya vuzov. Priborostroenie*, 2017, vol. 60, no. 11, pp. 1033–1039 (in Russian).
32. **Bogatyrev V. A., Bogatyrev S. V.** Rezervirovannoe obsluzhivanie v klasterah s unichtozheniem neaktual'nyh zaprosov (Redundant service in clusters with the destruction of irrelevant requests), *Vestnik komp'yuternyh i informacionnyh tekhnologij*, 2017, no. 1(151), pp. 21–28 (in Russian).
33. **Bogatyrev V. A., Bogatyrev A. V.** Model' rezervirovannogo obsluzhivaniya zaprosov real'nogo vremeni v komp'yuternom klasterne (Real-time redundant query service model in a computer cluster), *Informacionnye Tekhnologii*, 2016, vol. 22, no. 5, pp. 348–355 (in Russian).
34. **Bogatyrev V. A., Bogatyrev S. V.** Nadezhnost' mul'tiklasternyh sistem s pereraspredeleniem potokov zaprosov (The reliability of multi-cluster systems with redistribution query), *Izvestiya vuzov. Priborostroenie*, 2017, vol. 60, no. 2, pp. 171–177 (in Russian).
35. **Olifer V., Olifer N.** *Komp'yuternye seti. Principy, tekhnologii, protokoly* (Computer networks. Principles, technologies, protocols), SPb., Piter, 2016. 992 p.
36. **Parshutina S. A.** Organizaciya imitacionnyh eksperimentov pri proektirovanii raspredelyonnyh komp'yuternyh sistem s rezervirovannoj peredachej dannyh (Organization of simulation experiments in the design of distributed computer systems with redundant data transmission), *Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki*, 2017, vol. 17, no. 5, pp. 841–849 (in Russian).

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

Комплексирование систем счисления для многоразрядных вычислительных процессов

Введены конструкции и обоснованы варианты сочетаний внешней и внутренней систем счисления — математические основы методов вычислений в больших компьютерных диапазонах (многоразрядных вычислений). Систематизированы и обоснованы форматы данных для многоразрядных вычислений. Детально исследована конструкция, содержащая внутреннюю позиционную и внешнюю степенную модулярную системы. На основе кодонезависимых алгоритмов Ререп и Lucas-Lehmer введены кодозависимые модулярные модификации алгоритмов, имеющих меньшую временную сложность за счет больших затрат памяти. Исследованы линейные пространства модулярных векторов. Выявлены особенности и обоснованы преимущества модулярной системы с квадратичным диапазоном и соответствующих параллельных компьютерных форматов данных для модулярных реконфигурируемых вычислительных систем. Показанные преимущества обосновывают использование внешней квадратичной модулярной и внутренней позиционной системы счисления в пределах машинного диапазона для разрабатываемого программного инструментария организации вычислительных процессов в больших компьютерных диапазонах на компьютерах SIMD-архитектуры.

Ключевые слова: многопроцессорные реконфигурируемые системы SIMD-архитектуры, многоразрядный вычислительный процесс, кодозависимые алгоритмы, линейная и квадратичная сложность алгоритмов, параллельные компьютерные форматы квадратичной модулярной системы

Введение

При численных расчетах в быстро развивающейся прикладной теории чисел, при математическом моделировании в космологии, а также при проектировании точных и сложных технологических процессов в ряде отраслей техники и производства появляются отдельные вычислительные задачи, требующие целочисленных многоразрядных вычислений или вычислений в больших компьютерных диапазонах. На современном уровне развития вычислительной техники под этой проблематикой понимаются вычисления с числовыми величинами, содержащими более миллиона десятичных разрядов.

Важной научной и практической задачей является разработка программно-алгоритмических основ вычислительного инструментария для многоразрядных вычислений, относящихся к задачам с большой алгоритмической сложностью. Для организации этих вычислений на суперкомпьютерах и серийных вычислительных системах требуется специальная организация вычислительных процессов. Справиться с возникающими проблемами и выполнить числовые расчеты невозможно без специальной арифметики и производных типов данных на базе соответствующей компью-

терной алгебры [1, 2]. В этих процессах для ряда вычислительных задач значения операндов операций (переменных функций), в частности целочисленных, значительно, на много (сто, тысячу и более) порядков превышают максимум типового компьютерного диапазона серийной вычислительной техники.

С длиной машинного слова связан типовой компьютерный диапазон представления целых и рациональных числовых величин. В машинных словах или их совокупностях отображаются числовые данные в различных внутренних форматах. Данные в этих форматах обрабатываются в операционных регистрах процессора. Обработка форматов данных в регистрах поддерживается аппаратными и микропрограммными способами. Машинная арифметика для обработки этих форматов числовых данных, не превышающих максимума типового компьютерного диапазона, имеет максимальную скорость выполнения и является основной в компьютере [1].

1. Базовая терминология вычислений в больших диапазонах

Для корректного анализа проблем организации вычислительных процессов в диапазонах, значительно превышающих типовой компью-

терный диапазон серийной вычислительной техники, введем специальную терминологию.

Диапазонные числа — числовые величины для машинных расчетов, отображаемые в разрядной сетке процессора компьютера. Например, арифметический целочисленный компьютерный диапазон для 32-разрядного процессора есть отрезок $[-2^{31}, 2^{31} - 1]$.

Большие числа — конечные вектора с компонентами, являющимися диапазонными числами. *Сверхбольшие числа* — конечные или бесконечные последовательности больших чисел, полученные на этапах проблемных алгоритмов, как правило, в виде вычетов от больших чисел по модулю большого числа. Программная поддержка чисел двойной разрядности (или точности) не меняет приведенной классификации.

Для многоразрядных вычислений или расчетов в больших компьютерных диапазонах выделим внешнюю и внутреннюю системы счисления, учитывая, что в вычислительной технике нашли применение позиционные и модулярные системы счисления [1, 2].

Внутреннюю систему счисления можно выбрать позиционную или модулярную и использовать для представления диапазонных чисел, отображаемых в 32 разрядах (4 байтах) или в 64 разрядах (8 байтах).

Внешнюю систему счисления также можно выбрать позиционную или модулярную и использовать для представления больших чисел.

Представляет интерес поиск и обоснование оптимального сочетания внешней и внутренней систем счисления. Затем для них строят соответствующие арифметики и производные форматы числовых данных. Базовые операции в различных арифметиках имеют разную сложность.

Рассмотрим кратко варианты комплексирования внешней и внутренней систем счисления, приведенные в таблице.

Вариант	Внешняя система	Внутренняя система
1	Позиционная	Позиционная
2	Позиционная	Модулярная
3	Модулярная	Позиционная
4	Модулярная	Модулярная

1. ПозПоз — типовой вариант, исследованный в работах [1, 2].

2. ПозМод — проблематичный вариант, имеющий большую алгоритмическую сложность при реализации [1].

3. МодПоз — вариант представляет наибольший интерес, в нем для внутреннего представления используется одинарная или квадратичная модулярная система счисления, что позволяет организовать оптимальную эмуляцию на серийных машинах с аппаратно-поддерживаемой быстрой позиционной целочисленной 16-, 32-, 64-разрядной арифметикой на компьютерах соответствующей разрядности.

4. МодМод — вариант представляет интерес для модулярных систем со сжатием (перераспределением избыточности) внутреннего представления. Для этого варианта, требующего отдельного описания, ориентируясь на стандартный двоичный размер байта, можно предложить взаимно простые модули — основания модулярной арифметики: $\{p_1, p_2, p_3, p_4, p_5\} = \{253, 255, 256, 257, 259\}$.

2. Область приложения многоразрядных вычислений

Детально проанализируем вариант 3, наиболее перспективный на наш взгляд. Целесообразно рассмотреть особенности комплексирования систем счисления, опираясь на один из классов алгоритмов, предназначенных для операций с операндами из сверхбольших диапазонов. К этому классу относятся алгоритмы тестирования на простоту чисел специального вида — числа Ферма и Мерсенна, мультипликативное каноническое разложение которых неизвестно. Задачи тестирования чисел на простоту имеют важные приложения, но большую алгоритмическую и временную сложность [2, 3]. Соответствующие критерии простоты дают теоремы.

Теорема 1. Pepin. Необходимое и достаточное условие простоты числа Ферма $F_n = 2^{2^n} + 1$ есть выполнение сравнения:

$$3^{\frac{F_n-1}{2}} \equiv -1 \pmod{F_n}.$$

Теорема 2. Lucas-Lehmer. Число Мерсенна $M_q = 2^q - 1$, $q \in N$, — простое тогда и только тогда, когда M_q делит нацело W_{q-2} член последовательности $\{W_n\}$, построенной следующим образом:

$$W_0 = 4, W_{n+1} = W_n^2 - 2, n \geq 0.$$

Приведем тексты алгоритмов *Pepen* и *Lucas-Lehmer* в кодонезависимой постановке с указанием сложности на алгоритмическом псевдоязыке (в нотации П. Нодена [3]). Алгоритмы можно считать типовыми для вычислительных процессов, использующих большие компьютерные диапазоны.

Algorithm A-1. Pepen.

1. $m \leftarrow 0, A \leftarrow 3$
 2. $A \leftarrow A^2 \pmod{F_n}, m \leftarrow m + 1$
 3. if $2^n - m = 0$ then end else goto 2
 4. end: if $A = 2^{2^n}$ then F_n — prime else F_n — not prime
- Complexity: $time = 2^n \cdot (t_m + t_{mod}) = 2^n \cdot (n \cdot 2^n + 2 \cdot 2^n)$

Algorithm A-2. Lucas-Lehmer.

1. $m \leftarrow 0, A \leftarrow 4$
 2. $A \leftarrow A^2 - 2 \pmod{M_n}, m \leftarrow m + 1$
 3. if $n - 2 - m = 0$ then end else goto 2
 4. end: if $u \equiv 0 \pmod{M_n}$ then M_n — prime else M_n — not prime
- Complexity: $time = (n - 2) \cdot (t_m + t_{mod}) = (n - 2)(n \cdot \log_2 n + 3 \cdot n)$

Характерной особенностью этих алгоритмов является однотипная последовательность операций с применением арифметики больших чисел на каждом шаге алгоритма:

- возведение в квадрат большого числа, полученного на предыдущем шаге алгоритма;
- выполнение при необходимости арифметических операций с квадратом большого числа;
- вычисление вычета от результата предыдущего шага алгоритма, т. е. квадрата большого числа по большому модулю с неизвестным каноническим разложением.

Критерий останова этих алгоритмов заключается в сравнении больших чисел — результатов, полученных на каждом шаге алгоритма, с константой или повторение однотипных шагов алгоритма фиксированное число раз и затем сравнение большого числа с константой.

Перспективным путем решения проблем, возникающих при вычислениях в больших компьютерных диапазонах, является выбор варианта 3, т. е. использование для внешней системы модулярного представления данных и соответствующей модулярной арифметики. Внутренняя система является аппаратно поддерживаемой позиционной. При этом возникает проблема конвертации кодонезависимых проблемных вычислительных алгоритмов в форму модулярных проблемных алгоритмов

[4, 5]. Введем модификации алгоритмов *Pepen* и *Lucas-Lehmer* тестирования на простоту чисел специального вида Ферма и Мерсенна, ориентированные на использование модулярной компьютерной арифметики [2]. Из оценок следует, что это позволяет уменьшить их временную сложность, хотя организация вычислительного процесса усложняется и затраты специальным образом организованной памяти увеличиваются [6].

Algorithm A-3. Pepen for residue codes.

1. $|P - F_n| \rightarrow \min, P = \prod_{i=1}^k p_i > F_n, p_i$ — prime
 2. $F_n \pmod{P^2} = 2^{2^n} + 1 \pmod{P^2}$
 3. $m \leftarrow 0, A \pmod{P^2} \leftarrow 3 \pmod{P^2}$
 4. $A \leftarrow |A^2|_{F_n} \pmod{P^2}, m \leftarrow m + 1$
 5. if $2^n - m = 0$ then end else goto 4
 6. end: if $A = F_n - 1 \pmod{P^2}$ then F_n — prime else F_n — not prime
- Complexity (one processor): $time_1 = 2^n \cdot (t_m + t_{mod}) = (2^n)(O_1(k) + O_2(k)) + O_3(k)$
- Complexity (k — modular processor): $time_k = O(2^n)$

Algorithm A-4. Lucas-Lehmer for residue codes.

1. $|P - M_n| \rightarrow \min, P = \prod_{i=1}^k p_i > M_n, p_i$ — prime
 2. $M_n = 2^n - 1 \pmod{P^2}$
 3. $m \leftarrow 0, A \pmod{P^2} \leftarrow 4 \pmod{P^2}$
 4. $A \pmod{P^2} \leftarrow |A^2 \pmod{P^2} - 2 \pmod{P^2}|_{M_n} \pmod{P^2}, m \leftarrow m + 1$
 5. if $n - 2 - m = 0$ then end else goto 4
 6. end: if $A = 0 \pmod{P^2}$ then M_n — prime else M_n — not prime
- Complexity (one processor): $time_1 = (n - 2) \cdot (t_m + t_{mod}) = (n - 2)(O_1(k) + O_2(k)) + O_3(k)$
- Complexity (k — modular processor): $time_k = O(n - 2)$

3. Свойства инструментальной вычислительной алгебры

Переход к использованию модулярной арифметики требует уточнения терминологии применительно к математическим и алгоритмическим основам программного инструментария на базе модулярной арифметики.

Вычислительная проблема — формулировка математической расчетной задачи или комплекса взаимосвязанных задач в кодонезависимой постановке, т. е. независимой от системы счисления, необходимой для представления операндов и арифметики для оперирования с ними.

Вычислительная задача — формулировка математической расчетной задачи или комплекса взаимосвязанных задач в кодозависимой постановке, в частности, модулярная вычислительная задача ориентирована на вычисления с модулярными величинами и на операции модулярной арифметики.

В общем случае под *модулярной степенной системой* (k -й степени) будем понимать множество типов модулярных данных, способ ввода арифметического аддитивного и мультипликативного диапазонов, алгоритмы выполнения модульных операций, выбор количественной характеристики отношения порядка и связанных с ней алгоритмов выполнения немодульных операций. *Степень модулярной системы* при указанных условиях определяет число четырехбайтных машинных слов, необходимых для хранения одного степенного вычета.

Конкретизируем некоторые используемые понятия, полагая, что модулярные системы одинарного (первой степени) диапазона исследованы в достаточной степени [1, 2, 5].

Модулярная система, ориентированная на эмуляцию на серийном компьютере, должна удовлетворять некоторым требованиям [8]:

- иметь максимально возможные значения модулей для минимизации присущей модулярной арифметике информационной избыточности, неизбежной при отображении на двоичные конечные разрядные сетки компьютера;
- значения модулей системы должны лежать в пределах, "удобных" для вычислений с модулярными компонентами на 32-разрядных компьютерах;
- использовать знаковый арифметический целочисленный диапазон стандартной компьютерной позиционной арифметики 32-разрядных компьютеров, являющейся "быстрой" благодаря аппаратной или микропрограммной поддержке.

Под термином "удобства" вычислений будем понимать минимизацию затрат операционного оборудования или машинного времени для рассматриваемой задачи на выполнение машинных операций, а также последующую минимизацию затрат оперативной и долговременной памяти. Для уменьшения информационной избыточности модулярной системы, отображаемой в двоичных разрядных сетках 32-разрядных компьютеров, целесообразно применение простых модулей, лежащих в диа-

пазоне $(2^{30}, 2^{31} - 1]$. Оценка Чебышева дает более 50 млн простых чисел в этом диапазоне [2].

Целесообразно назвать числовой величиной элемент множества целых чисел. Для рассматриваемых задач значение числовой величины на много порядков превышает максимум целочисленного машинного диапазона современных компьютеров с ограниченной разрядной сеткой.

Целочисленная числовая величина отображается (представлена) в любой системе счисления вектором с конечным числом компонент:

$$A \leftrightarrow (a_1, \dots, a_i, \dots, a_n),$$

где $A \in \mathbf{Z}/\mathbf{Z}_P$ принадлежит подмножеству целых чисел или кольцу по модулю P — произведению оснований системы счисления; компоненты $a_i \in p|p_i$ принадлежат кольцу по модулю p или p_i — одиночным основаниям системы счисления.

Для биективности отображения необходимо ограничение значений числовой величины некоторым конечным диапазоном и фиксацией системы счисления, так как из представления значение числовой величины однозначно можно вычислить только в соответствующей системе счисления.

Обратное отображение компонент вектора представления в значение числовой величины однозначно определяет разложение числовой величины в сумму компонент представления с некоторыми числовыми коэффициентами, детерминированными выбранной системой счисления.

В модулярной системе числовые величины представлены модулярными векторами с компонентами-вычетами по простым (взаимно простым) модулям для системы одинарного диапазона или по степеням простых модулей для систем степенных диапазонов:

$$A \pmod{P^m} \leftrightarrow (a_1 \pmod{p_1^m}, \dots, a_i \pmod{p_i^m}, \dots, a_n \pmod{p_n^m}),$$

где $a_i \equiv |A|_{p_i^m} \pmod{p_i^m}$ — степенной вычет числовой величины по одному из модулей (оснований) степенной модулярной системы.

Рассмотрим метрическое пространство векторов с модулярными компонентами. Для удобства рассмотрения ограничимся случаем $m = 2$, т. е. квадратичной модулярной системой, которую будем обозначать $MC(P^2)$.

Для векторов с модулярными компонентами известно остаточное расстояние (аналог метрики Хэмминга), введенное для задач помехозащитного кодирования и учитывающее характер модульной ошибки [1].

Определение. Остаточное расстояние $d(A, B)$ между двумя модулярными векторами $A(\text{mod } P^2)$, $B(\text{mod } P^2)$ есть остаточный вес модульной разности двух модулярных векторов:

$$d(A, B) = \sum_{i=1}^n \delta(|a_i - b_i|_{p_i^2}),$$

где символ Кронекера

$$\delta(|a_i - b_i|_{p_i^2}) = \begin{cases} 1 & \text{при } a_i \neq b_i, \\ 0 & \text{при } a_i = b_i. \end{cases}$$

Остаточное расстояние является метрикой, для него выполняются соответствующие аксиомы.

Введем аналоги евклидового расстояния на множестве n -мерных векторов через их скалярное произведение, применимые для анализа модулярных вычислительных процессов.

Определение. Первое скалярное произведение двух модулярных величин — векторов $A, B(\text{mod } P^2)$ из $\text{MC}(P^2)$ есть число

$$s_1 = \sum_{i=1}^n a_i \cdot b_i = (A, B).$$

Замечание. $s_1 \leq n(p_i^2 - 1)^2$.

Определение. Второе скалярное произведение двух модулярных величин — векторов $A, B(\text{mod } P^2)$ из $\text{MC}(P^2)$ есть число

$$s_2 = \sum_{i=1}^n |a_i|_{p_i^2} |b_i|_{p_i^2}.$$

Замечание 1. $s_2 \leq n(p_i^2 - 1)^2$.

Замечание 2. $s_2 = \sum_{i=1}^n |a_i|_{p_i^2}^2$ при $A = B(\text{mod } P^2)$.

Введем два типа расстояний на множестве модулярных векторов, связанных со скалярными произведениями.

Определение. Евклидово расстояние между двумя модулярными величинами — векторами $A, B(\text{mod } P^2)$ из $\text{MC}(P^2)$ есть число

$$\varepsilon(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} = \sqrt{(A - B, A - B)}.$$

Евклидово расстояние есть метрика, так как не отличается от соответствующей математиче-

ской конструкции в евклидовом пространстве векторов с целочисленными компонентами.

Определение. Модулярное расстояние между двумя модулярными величинами — векторами $A, B(\text{mod } P^2)$ из $\text{MC}(P^2)$ есть число

$$l(A, B) = \sqrt{\sum_{i=1}^n |a_i - b_i|_{p_i^2}^2} = \sqrt{(|A - B|_{p_i^2}, |A - B|_{p_i^2})}.$$

Для модулярного расстояния докажем следующую теорему.

Теорема Т-3. Модулярное расстояние между двумя модулярными векторами является метрикой.

Доказательство. Используя арифметическое значение квадратного корня, по определению получим:

1. $l(A, B) \geq 0$.
2. $l(A, B) = 0$, если $A \equiv B(\text{mod } P^2)$.
3. $l(A, B) = l(B, A)$.
4. Неравенство треугольника рассмотрим подробнее.

Действительно, выполняется следующее равенство:

$$\begin{aligned} |x_i - y_i|_{p_i^2} &= ||x_i - z_i|_{p_i^2} + |z_i - y_i|_{p_i^2}|_{p_i^2} = \\ &= |x_i - z_i|_{p_i^2} + |z_i - y_i|_{p_i^2} - 0|_{p_i^2}. \end{aligned}$$

Следовательно

$$|x_i - z_i|_{p_i^2} + |z_i - y_i|_{p_i^2} \geq |x_i - y_i|_{p_i^2}.$$

Возведем обе части неравенства в квадрат и просуммируем:

$$\begin{aligned} \sum_{i=1}^n (|x_i - z_i|_{p_i^2}^2 + 2|x_i - z_i|_{p_i^2} |z_i - y_i|_{p_i^2} + |x_i - y_i|_{p_i^2}^2) &\geq \\ &\geq \sum_{i=1}^n |x_i - y_i|_{p_i^2}^2. \end{aligned}$$

Применим неравенство Коши—Буняковского, которое для соотношений по модулям имеет следующий вид:

$$\begin{aligned} \sum_{i=1}^n (|x_i - z_i|_{p_i^2} |z_i - y_i|_{p_i^2}) &\leq \\ &\leq \sqrt{\sum_{i=1}^n |x_i - z_i|_{p_i^2}^2} \sqrt{\sum_{i=1}^n |z_i - y_i|_{p_i^2}^2}. \end{aligned}$$

Получим:

$$\begin{aligned} \sqrt{\sum_{i=1}^n |x_i - z_i|_{p_i^2}^2} + \sqrt{\sum_{i=1}^n |z_i - y_i|_{p_i^2}^2} &\geq \\ &\geq \sqrt{\sum_{i=1}^n |x_i - y_i|_{p_i^2}^2}. \end{aligned}$$

Следствие. Первое и второе скалярные произведения определяют метрики, а не нормы [7].

Для кодозависимых алгоритмов, ориентированных на модулярную арифметику, важно соотношение между модульными и немодульными операциями в линейных и циклических частях алгоритмов. Степень сложности модульных и немодульных операций может меняться в различных модулярных системах, что ставит задачу выбора оптимальной степени модулярной системы. Причем эта сложность зависит как от свойств самой модулярной системы, так и от компьютерной базы реализации программного инструментария [2, 9].

Покажем целесообразность выбора квадратичной модулярной системы, опираясь на удобство программно-алгоритмической реализации на 32-разрядных IBM-совместимых серийных компьютерах.

Рассмотрим проблему эффективности выполнения немодульных операций, опираясь на внутренние свойства самой квадратичной модулярной системы.

Числовая величина в квадратичной модулярной системе имеет следующее P -разложение:

$$A(\bmod P^2) = |A|_P + K \cdot P(\bmod P^2),$$

где $0 < |A|_P, K < P$.

Пусть для числового значения модулярной величины выполняется неравенство $0 < A < P^2$, тогда для модулярной величины

$$A(\bmod P^2) \leftrightarrow (\dots, a_i(\bmod p_i^2), \dots).$$

В модулярной системе квадратичного диапазона введем три представления и разложение модулярной величины, а также рассмотрим связи между ними.

Определение. Первое (естественное) представление модулярной величины есть выражение следующего вида:

$$A(\bmod P^2) \leftrightarrow (\dots, a_i(\bmod p_i^2), \dots),$$

или $A(\bmod P^2) \leftrightarrow a_i(\bmod p_i^2),$

где $a_i \equiv A(\bmod p_i^2)$.

Будем при необходимости использовать краткую запись, отображая знаком \leftrightarrow биективность отображения числовой величины в пространство образов — модулярных представлений числовой величины.

Определение. Второе (каноническое) представление модулярной величины есть выражение следующего вида:

$$A(\bmod P^m) \leftrightarrow (\dots, a_{i0} + a_{i1} \cdot p_i(\bmod p_i^2), \dots)$$

или $A(\bmod P^2) \leftrightarrow a_{i0} + a_{i1} \cdot p_i(\bmod p_i^2),$

где

$$a_{i0} \equiv A(\bmod p_i),$$

$$a_{i0} \equiv a_i(\bmod p_i) \equiv |a_i|_{p_i}(\bmod p_i^2),$$

$$a_{i1} = [a_i/p_i] = (a_i - |a_i|_{p_i})/p_i.$$

Каноническое представление применяется для выполнения арифметических операций, в частности деления в $MC(P^2)$. Для канонического представления двухкомпонентный вектор (a_{i0}, a_{i1}) хранится в двух машинных словах серийного компьютера при выборе оснований модулярной системы, меньших максимального значения машинного диапазона.

Каноническому представлению соответствует разложение модулярной величины $A(\bmod P^2) = A_0 + A_1 \cdot P(\bmod P^2)$, причем неравенство $A_0 < P$ в общем случае не выполняется.

Определение. Третье представление (согласованное с P -разложением) модулярной величины есть

$$A(\bmod P^2) \leftrightarrow (\dots, s_i + k_i \cdot p_i(\bmod p_i^2), \dots)$$

или $A(\bmod P^2) \leftrightarrow s_i + k_i \cdot p_i(\bmod p_i^2),$

где

$$s_i \equiv |A|_P(\bmod p_i^2) \equiv s_{i0} + s_{i1} \cdot p_i(\bmod p_i^2);$$

$$s_i \neq |a_i|_{p_i}(\bmod p_i^2), s_{i0} = |a_i|_{p_i}(\bmod p_i^2);$$

$$P(\bmod p_i^2) \leftrightarrow g_i \cdot p_i(\bmod p_i^2) = \bar{p}_i(\bmod p_i^2);$$

$$A_1(\bmod P) = K(\bmod P) = (A - |A|_P)/P(\bmod P);$$

$$K(\bmod P) \leftrightarrow k_i(\bmod p_i) \neq g_i \bar{k}_i + s_{i1}(\bmod p_i) \equiv a_{i1}(\bmod p_i);$$

$$g_i = |P/p_i|_{p_i}(\bmod p_i), h_i = |P/p_i|_{p_i}^{-1}(\bmod p_i);$$

$$\bar{p}_i = g_i \cdot p_i(\bmod p_i^2), \bar{k}_i = h_i \cdot k_i(\bmod p_i^2).$$

Согласованное представление применяется в алгоритмах вычисления количественных характеристик отношения порядка для модулярных величин, в частности, с использованием алгоритма исчерпывания [2].

При рассмотрении мы ограничились случаем $m = 2$, но все предыдущее обобщается

на случай произвольной модулярной системы m -степени.

Сказанное выше является основой методов преобразования канонического в согласованное представление модулярной величины. Такое преобразование уменьшает алгоритмическую сложность в алгоритмах вычисления количественных характеристик отношения порядка, а это, в свою очередь, уменьшает сложность процедур выполнения немодульных операций в проблемных вычислительных алгоритмах, ориентированных на применение модулярной арифметики.

4. Преимущества модулярной квадратичной арифметики

Покажем ряд преимуществ модулярной системы квадратичного диапазона при выполнении операций модулярной арифметики.

При выполнении эмулируемых арифметических операций аддитивного алгебраического сложения и мультипликативной операции умножения двух модулярных операндов нормальным этапом является переполнение типового машинного диапазона. Переполнение вызывает аппаратное прерывание. Временные затраты на обработку прерывания минимальные, если факт переполнения однозначно формирует числовой результат без анализа кратности переполнения. Такой анализ требует дополнительных операций и временных затрат на выполнение алгоритма для обработки прерывания.

Пусть квадратичный диапазон $P^2 = \prod_{i=1}^n p_i^2$ задан совокупностью простых оснований $\{\dots, p_i, \dots\}$, а $A(\bmod P^2), B(\bmod P^2)$ — модулярные величины в квадратичном диапазоне, представленные векторами с соответствующими компонентами, которые являются наименьшими неотрицательными вычетами от числовой величины по квадратам модулей-оснований модулярной системы.

Рассмотрим аддитивную операцию алгебраического сложения двух модулярных величин, имеющих разложения в канонические представления:

$$\begin{aligned} A(\bmod P^2) &= A_0 + A_1 \cdot P(\bmod P^2) \leftrightarrow \\ &\leftrightarrow a_i^0 + a_i^1 \cdot p_i(\bmod p_i^2); \\ B(\bmod P^2) &= B_0 + B_1 \cdot P(\bmod P^2) \leftrightarrow \\ &\leftrightarrow b_i^0 + b_i^1 \cdot p_i(\bmod p_i^2). \end{aligned}$$

Модулярные величины в одинарном модулярном диапазоне, являющиеся коэффициентами позиционного аддитивного разложения, имеют следующий вид:

$$\begin{aligned} A_0(\bmod P) &\leftrightarrow a_i^0(\bmod p_i), A_1(\bmod P) \leftrightarrow \\ &\leftrightarrow a_i^1(\bmod p_i). \end{aligned}$$

Результат алгебраического сложения двух модулярных величин должен иметь вид:

$$\begin{aligned} A + B(\bmod P^2) &\equiv C(\bmod P^2) \equiv \\ &\equiv C_0 + C_1 \cdot P(\bmod P^2) \leftrightarrow c_i^0 + c_i^1 \cdot p_i(\bmod p_i^2). \end{aligned}$$

Операции целочисленной знаковой арифметики в процессоре 32-разрядного компьютера, учитывая ограниченность разрядной сетки, выполняются по модулю $m = 2^{31}$. Оценим число прерываний по переполнению при суммировании максимальных значений компонент векторов:

$$\begin{aligned} c_i^0 + c_i^1 \cdot p_i &\leq p_i - 2 + 1 \cdot p_i + (p_i - 2)p_i + 1 \cdot p_i^2, \\ \left| a_i^0 + b_i^0 \right|_m + r_0 m, \left| a_i^1 + b_i^1 \right|_m + r_1 m, \end{aligned}$$

где $r_0, r_1 = 0|1$.

Эти покомпонентные соотношения в модулярной системе примут вид:

$$\begin{aligned} \left| a_i^0 + b_i^0 \right|_m \Big|_{p_i} + r_0 \Big|_m \Big|_{p_i} + e_1 p_i + e_2 p_i; \\ \left| a_i^1 + b_i^1 \right|_m + r_1 m \Big|_{p_i} = \left| a_i^1 + b_i^1 \right|_m \Big|_{p_i} + r_1 \Big|_m \Big|_{p_i}, \end{aligned}$$

где $e_0, e_1 = 0|1$.

Переполнения, вызывающие прерывания, отражены в коэффициентах $r_0, r_1 = 0|1$. Кратности переполнений имеют единичные значения, следовательно, соответствующие числовые значения определяются фактом переполнения.

В модулярных системах больших степеней сложность операций растет, например, в $MC(P^3)$ число процедур обработки переполнений удваивается при выполнении операций над соответствующими представлениями:

$$\begin{aligned} A(\bmod P^3) &= A_0 + A_1 \cdot P + A_2 \cdot P^2(\bmod P^3) \leftrightarrow \\ &\leftrightarrow a_i^0 + a_i^1 \cdot p_i + a_i^2 \cdot p_i^2(\bmod p_i^3); \\ B(\bmod P^3) &= B_0 + B_1 \cdot P + B_2 \cdot P^2(\bmod P^3) \leftrightarrow \\ &\leftrightarrow b_i^0 + b_i^1 \cdot p_i + b_i^2 \cdot p_i^2(\bmod p_i^3). \end{aligned}$$

Рассмотрим операцию умножения двух модулярных величин, заданных каноническими представлениями в модулярной системе квадратичного диапазона. Результат должен быть получен в виде:

$$A \times B \pmod{P^2} \equiv C \pmod{P^2} \equiv \\ \equiv C_0 + C_1 \cdot P \pmod{P^2} \leftrightarrow c_i^0 + c_i^1 \cdot p_i \pmod{p_i^2}.$$

Рассмотрим процесс формирования результата:

$$(a_i^0 + a_i^1 \cdot p_i \pmod{p_i^2}) \times (b_i^0 + b_i^1 \cdot p_i \pmod{p_i^2}) \leftrightarrow \\ \leftrightarrow c_i^0 + c_i^1 \cdot p_i + c_i^2 \cdot p_i^2 + c_i^3 \cdot p_i^3,$$

где $c_i^0, c_i^1, c_i^2, c_i^3$ — коэффициенты разложения модулярных компонент произведения с учетом переполнений.

Результат умножения в стандартной компьютерной арифметике операндов, имеющих машинный формат целых чисел, заносится в два последовательных операционных регистра или соседних машинных слова, что позволяет последующим делением получить целую часть и наименьший неотрицательный вычет от результата произведения. В этом случае аппаратно формируемый сигнал переполнения сигнализирует о выходе за пределы удвоенного машинного диапазона без информации о кратности переполнения.

Оценим влияние отдельных компонент операндов на число переполнений при формировании коэффициентов $c_i^0, c_i^1, c_i^2, c_i^3$, используя при умножении максимальные значения компонент векторов-операндов:

$$a_i^0 \cdot b_i^0 \pmod{p_i^2} \equiv c_i^0 + c_i^{01} \cdot p_i \pmod{p_i^2} \leq \\ \leq 1 + (p_i - 2)p_i \leq m^2; \\ a_i^0 \cdot b_i^1 + a_i^1 \cdot b_i^0 \pmod{p_i^2} \equiv c_i^{10} + c_i^{11} \cdot p_i + \\ + c_i^{12} \cdot p_i^2 \pmod{p_i^2} \leq 2 + (p_i - 4)p_i + 1 \cdot p_i^2 \geq m^2; \\ a_i^1 \cdot b_i^1 \pmod{p_i^2} \equiv c_i^{22} + c_i^3 \cdot p_i \pmod{p_i^2} \leq \\ \leq 1 + (p_i - 2)p_i \leq m^2,$$

где $c_i^0, c_i^{01}, c_i^{10}, c_i^{11}, c_i^{12}, c_i^{22}, c_i^3$ — коэффициенты разложения слагаемых модулярных компонент произведения с учетом переполнений.

Для квадратичного модулярного диапазона отдельные из соотношений, имеющих, соответственно, множители 1, p , p^2 , лишь косвенно

отражаются на результате операции умножения. Переполнение, подлежащее обработке, появляется во втором соотношении рассматриваемой совокупности, причем коэффициент кратности переполнения единичный.

Заключение

Целесообразность применения для вычислений в больших компьютерных диапазонах в качестве внешней модулярной системы квадратичного диапазона обусловлена рядом причин, из которых основными являются:

— минимальные временные затраты на обработку прерываний при выполнении основных арифметических операций целочисленной арифметики при эмуляции в процессоре серийного компьютера с использованием знаковой целочисленной арифметики;

— возможность получения для многозарядных вычислительных процессов соотношений, имеющих меньшую временную сложность для вычисления позиционных характеристик модулярного представления и некоторых важных для проблемных алгоритмов операций, например, вычисления вычета по большому модулю без операции деления в явном виде и ряда других.

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. **Ноден П., Китте К.** Алгебраическая алгоритмика. М.: Мир, 1999. 720 с.
4. **Инютин С. А.** Анализ сложности многозарядных вычислительных процессов // Научные труды МАТИ. 2014. Вып. 22 (94). С. 154—159.
5. **Амербаев В. М., Стемпковский А. Л., Соловьев Р. А.** Принципы рекурсивных модулярных вычислений // Информационные технологии. 2013. № 2. С. 22—27.
6. **Шилов В. В., Столярский Е. З.** Организация и работа кэш-памяти // Информационные технологии. 2000. № 7. С. 2—8.
7. **Inyutin S. A.** Parallel Square Modular Computer Algebra // Transaction of Parallel Processing and Applied Mathematics PPAAM-2003. Poland-Denmark: Springer, 2003. P. 117—123.
8. **Инютин С. А.** Метод вычисления количественной характеристики модулярной величины // Информационные технологии. 2016. Т. 22, № 5. С. 343—347.
9. **Инютин С. А.** Метод вычисления характеристики отношения порядка для параллельных форматов данных // Информационные технологии. 2017. Т. 23, № 8. С. 569—574.

The Aggregation of Number Systems for Multi-Digit Computational Processes

Designs and variants of combinations of external and internal number systems — mathematical bases of calculation methods in large computer ranges (multi-digit calculations) are introduced. Data formats for multi-digit calculations are systematized and substantiated. Studied design, containing internal and external positional power of the modular system. On the basis of code-independent Pepen and Lucas-Lehmer algorithms, code-dependent modular modifications of algorithms with less time complexity due to high memory costs are introduced. Linear spaces of modular vectors are investigated. The features and advantages of a modular system with a quadratic range and the corresponding parallel computer data formats for modular reconfigurable computing systems are revealed. The shown advantages justify the use of an external quadratic modular and internal positional number system within the machine range for the developed software tools for the organization of computational processes in large computer ranges on SIMD architecture computers.

Keywords: multiprocessor reconfigurable systems SIMD architecture, multi-row computational process, code-dependent algorithms, linear and quadratic complexity of algorithms, parallel computer formats of quadratic modular system

DOI: 10.17587/it.24.782-790

References

1. Amerbaev V. M. *Teoreticheskie osnovy mashinnoy arifmetiki* (Theoretic base computer arithmetic), Alma-Ata: Nauka, 1976. 320 p. (in Russian).
2. Inyutin S. A. *Osnovy modulyarnoy algoritmiki* (Base at modular algorithmic), Hantyi-Mansiysk, Poligrafist, 2009. 237 p. (in Russian).
3. Noden P., Kittle K. *Algebraicheskaya algoritmika* (Algebra algorithmic). Moscow, Mir, 1999, 720 p. (in Russian).
4. Inyutin S. A. Analiz slojnosti mnogorazryadnyh vyichislitelnyh protsessov (Analysis many digital calculation process), *Nauchnyye trudy MATI*, 2014, vol. 22 (94), pp. 154—159 (in Russian).
5. Amerbaev V. M., Stempkovsky A. L., Solovjev R. A. Printsipy rekursivnyh modulyarnykh vyichisleniy (Principles of recursive modular computing), *Informatcionnye tehnologii*, 2013, no. 2, pp. 22—27 (in Russian).
6. Shilov V. V., Stolyarskiy E. Z. Organizatsiya i rabota kesh-pamyati (Planning work Kech-memory), *Informatcionnye tehnologii*, 2000, no. 7, pp. 2—8 (in Russian).
7. Inyutin S. A. Parallel Square Modular Computer Algebra, *Transaction of Parallel Processing and Applied Mathematics PPAM-2003*, Poland-Denmark: Springer, 2003. pp. 117—123.
8. Inyutin S. A. Metod vyichisleniy kolichestvenoy harakteristic modulyarnoy velichina (Method calculation quantitative characteristic for computer modular value), *Informatcionnye tehnologii*, 2016, vol. 22, no. 5, pp. 343—347 (in Russian).
9. Inyutin S. A. Metod vyichisleniy harakteristic otnosheniy porjadka dliy parallel formatov dannih (Method calculation characteristic the order for parallel computer format date), *Informatcionnye tehnologii*, 2017, vol. 23, no. 8, pp. 569—574 (in Russian).



17—19 июня 2019 года в Москве состоится

XIII Всероссийское совещание по проблемам управления,

посвященное 80-летию Института проблем управления
имени В. А. Трапезникова РАН

Направления работы Совещания:

- ❖ Теория систем управления
- ❖ Управление подвижными объектами и навигация
- ❖ Интеллектуальные системы в управлении
- ❖ Управление в промышленности и логистике
- ❖ Управление системами междисциплинарной природы
- ❖ Средства измерения, вычислений и контроля в управлении
- ❖ Системный анализ и принятие решений в задачах управления
- ❖ Информационные технологии в управлении
- ❖ Проблемы образования в области управления: современное содержание и технологии обучения

Подробную информацию о Совещании смотрите на сайте
<http://vspu2019.ipu.ru>

Р. Р. Фаткиева, канд. техн. наук, доц., e-mail: rikki2@yandex.ru,

С. Р. Рыжков, программист, e-mail: Ryzhkov@iias.spb.su,

Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук

Оценка нарушения периметра информационной безопасности в облачной среде¹

Рассмотрена задача оценивания безопасности виртуальной среды клиента, начиная с момента установления соединения и заканчивая получением результатов при использовании сервиса. Предложено использовать марковскую модель для выявления наиболее значимых угроз и прогнозирования состояния системы. Показано, что за счет применения комплекса мер по обеспечению информационной безопасности повышается не только вероятность успешной штатной работы, но и вероятность сбора дополнительных данных злоумышленником в связи с повышением интереса к системе.

Ключевые слова: информационная безопасность, динамический периметр, облачная среда, информационные атаки

Введение

Постепенная эволюция от замкнутых информационных сред с централизованной системой управления к формированию облачной среды с динамическим периметром и децентрализованным процессом управления приводит к тому, что для обеспечения информационной безопасности (ИБ) цифровых платформ недостаточно установки какого-либо одного средства защиты информации, поскольку имеет место их совместное использование. Важным условием для обеспечения ИБ цифровых платформ является реализация механизмов управления доступом. В работе [1] отмечается, что виртуальность облачных технологий привела к исчезновению традиционного физического периметра на основе контрольных точек, обеспечивавшего конфиденциальность информации. Периметр в цифровых платформах виртуализован, так как он определяется параметрами виртуальной среды и выполняемых приложений. На сегодняшний день существует актуальная проблема районирования и защиты гео-

графического ландшафта электронно-вычислительной структуры облачных вычислений при использовании цифровых платформ. Проблема обусловлена необходимостью:

- гарантированного доступа к отечественным и зарубежным информационным ресурсам [2];
- обеспечения безопасности и конфиденциальности миграции данных и приложений при использовании облачных вычислений в географически распределенных облачных хранилищах [3, 4];
- использования разнотипных вычислительных средств и систем, составляющих основу облачных вычислений, нередко разнородных и от разных производителей, что может повлечь за собой не только сбои в работе виртуальной среды, но и множество различных атак на нее [5, 6];
- понижения вероятности преодоления систем защиты при получении несанкционированного доступа к управлению виртуальной средой [7, 8];
- ограничения доступа физическим и юридическим лицам к сайтам в сети Интернет, содержащим информацию, распространение которой в Российской Федерации запрещено;

¹ Работа выполнена при частичной поддержке гранта РФФИ 16-29-09482.

- атрибуции и отслеживания перемещения информации в целях создания матрицы доступа и контроля над уровнем предоставляемых прав [9—11].

Для определения общего числа возможных уязвимых мест (поверхностей атак) целесообразно классифицировать угрозы, систематизировать их в виде таксономии.

Таксономия как иерархическая система соподчиненных рангов безопасности облачных вычислений рассматривается в свете количественного анализа актуальных проблем безопасности для облачных вычислений, которые зависят от архитектуры. Базовая архитектурная классификация включает сети, хосты, приложения, данные (информацию об их безопасности и хранении), управление безопасностью, идентификацией и доступом — все эти элементы напрямую связаны с инфраструктурой и архитектурой реализации облачных решений.

Однако при использовании облачных вычислений периметр динамически изменяется, размывается за счет многоуровневой туннелированной вложенности и зависит от прохождения пакета по стеку протоколов. Каждый уровень вложенности позволяет создать снимок своего периметра, что позволяет, используя привязку к физическому периметру, затем разместить их на карте.

На рис. 1 отображены векторы атаки злоумышленника вне периметра провайдера облачных вычислений и внутри, в случае, когда злоумышленник получил доступ в качестве легитимного пользователя облачной инфраструктуры. Облачные вычисления позволяют отказаться от классической архитектуры, где в слу-

чае нарушения периметра вся защищаемая сеть находится под угрозой. Периметр безопасности, поставляемый как услуга, физически и логически отделен и изолирован от защищаемой сети. Физический периметр в системах контроля и управления доступом (СКУД) также оперирует понятием "граница" в контексте границы контролируемой зоны. В настоящее время известны модели, которые основываются на противоречиях между уровнем защищенности и доступности информационных ресурсов в пределах периметра [12—16]. Однако в описываемых моделях не учтена вероятность повышения интереса злоумышленника при обнаружении уровня защищенности, не соответствующего предполагаемой им ценности защищаемого информационного ресурса. Также на практике реализована методика, включающая в себя обоснование мероприятий обеспечения информационной безопасности в информационной системе [17, 18], в которой не охвачен вопрос использования облачных технологий, предоставляемых гарантированным поставщиком.

В работе [14] представлено комплексное решение для повышения уровня доверия в цифровых платформах, в основе этого решения — мандатный контроль доступа и надежные вычислительные технологии (измеряемая загрузка, аттестации и запечатывания). Такой подход создает гарантированную среду и явно связывает зашифрованные виртуальные машины с ранее аттестованными узлами, однако не представляет для оценки количественные значения.

Таким образом, важным условием для обеспечения безопасности информационной системы, построенной с использованием технологий облачных вычислений, является определение уязвимого для атаки места на всей поверхности защищаемого периметра исполняемых приложений в целях ее предотвращения. В статье рассматривается задача определения риска нарушения безопасности как внутри периметра облачных вычислений, так и "снаружи" миграции данных за счет перехвата трафика, а также подбора мероприятий для их снижения.

1. Постановка задачи

Рассмотрим постановку задачи на практическом примере доступа к облачному ресурсу. Ключевым вопросом при использовании облачной среды является безопасность виртуальной среды клиента, начиная с момента уста-



Рис. 1. Атаки на физический и облачный периметр

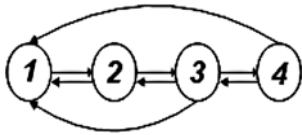


Рис. 2. Штатный процесс установления соединения

новления соединения и заканчивая получением результатов при использовании сервиса. В обобщенном виде процесс доступа к вычислительному ресурсу может быть представлен в виде графа состояний (рис. 2).

В частности, он соответствует процессу установления соединения по протоколу TCP/IP. Согласно графу (рис. 2) в процессе доступа могут быть выделены следующие состояния: 1 — клиент запускает исполняемый файл; 2 — осуществляется открытие сессии; 3 — передача данных по каналам связи; 4 — установление соединения.

Требуется разработать подход, позволяющий оценить риски нарушения безопасности в облачной среде, а также выработать мероприятия по противодействию представленным атакам.

Поскольку события описываемого операционного процесса дискретны, а время выполнения непрерывно, то сам процесс может быть представлен как процесс с отсутствием последствия, т. е. он обладает марковским свойством. В этом случае процесс перехода из состояния в состояние возможно описать системой дифференциальных уравнений, позволяющей учитывать не только вероятности наступления того или иного события, но влияние мероприятий по защите на процессы, происходящие в облачной среде.

2. Модель облачной среды в условиях информационных угроз

Определим показатель периметра защищенности облачных вычислений как вероятность того, что нарушение безопасности не произойдет. В этом случае защищенность облачных вычислений можно определить через математическое ожидание ущерба от нарушения защищаемых хостов $M_j = \sum_{i=0}^N \gamma_i P_{ij}(t)$, где $P_{ij}(t)$ — вероятность нарушения безопасности i -го хоста в момент времени t при j -м методе защиты; $i = \overline{1, N}$ — число представленных в системе виртуальных хостов; $j = \overline{1, K}$ — число возможных вариантов защиты; γ_i — значение ущерба от нарушения безопасности i -го хоста. Выполнение оценки возможно при условии, что ата-

ки являются независимыми, а работа виртуальных хостов не коррелируется.

Для нахождения $P_{ij}(t)$ на каждом i -м виртуальном хосте определим методику, включающую следующие шаги:

Шаг 1. Определение процессов S_i , происходящих в системе, в штатном режиме функционирования.

Шаг 2. Формирование перечня атак $A_i(t)$ и построение модели угроз в облачной среде.

Шаг 3. Определение интенсивностей λ_i перехода процесса S_i из состояния в состояние и нахождение начальных состояний на момент $t = 0$.

Шаг 4. Решение системы дифференциальных уравнений относительно полученных интенсивностей перехода и начальных состояний.

Шаг 5. Получение значений вероятностей $P_{igr}(t)$ перехода из состояния q в состояние r на основе рассмотренной ранее системы дифференциальных уравнений для различных наборов значений интенсивностей, характерных для альтернативных мероприятий защиты.

Шаг 6. Влияние на процесс функционирования системы. Определение наилучшего варианта мероприятий по обеспечению информационной безопасности защиты согласно

$$M_j = \min_{\lambda \in \Omega} \sum_{i=0}^N \gamma_i P_{ij}(t).$$

Для построения простейшей модели расширения процесс доступа к вычислительному ресурсу (рис. 3) с учетом возможных угроз. В этом варианте процесса можно выделить следующие состояния: 1 — запуск программы; 2 — открытие сессии; 3 — передача данных по каналам связи; 4 — установление соединения с сервером; 5 — процесс аутентификации; 6 — про-

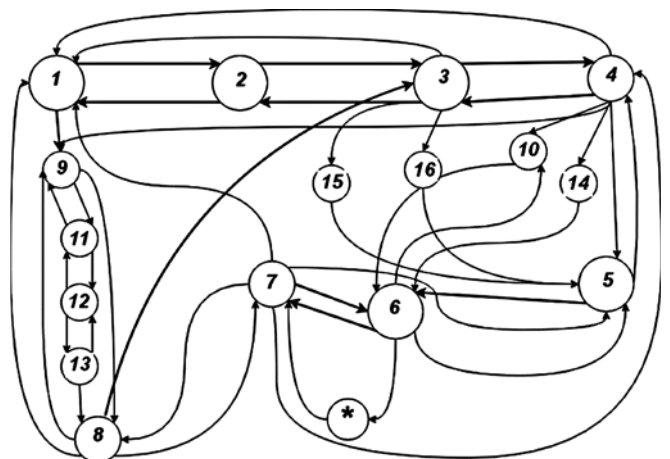


Рис. 3. Процесс доступа к вычислительному ресурсу при нарушениях ИБ

цесс авторизации; 7 — атака, направленная на подбор пароля; 8 — процесс получения доступа к управляемому ресурсу; 9 — атака, направленная на анализ приложения; 10 — атака Ddos; 11 — атака, направленная на сканирование порта; 12 — процесс получения данных о порте; 13 — процесс получения доступа к ОС; 14 — внедрение произвольного кода в SQL-запрос; 15 — процесс перехвата трафика; 16 — атака, направленная на подмену маршрутизатора; * — подделка (компрометация) криптоключа.

Оценка процессов, происходящих при доступе к сервису, позволяет выделить четыре основных этапа, которые необходимо осуществить для получения доступа к ресурсу: установление соединения, аутентификация, авторизация, доступ к ресурсу. Каждый из этапов представляет собой последовательный набор действий, приводящий к результату, но затрудняющий работу пользователя при осуществлении атаки. В связи с этим целесообразно рассмотреть частные модели возможных угроз на каждом из этапов.

3. Результаты моделирования

Моделирование нарушения установления соединения. В отличие от стандартного исполняемого файла, содержащего программу в виде, в котором она может быть исполнена компьютером, среда исполнения, а следовательно, и периметр исполняемого приложения, располагается не только на локальном вычислительном устройстве, но и в облаке. Перед исполнением программа загружается в память, и выполняются некоторые подготовительные операции (настройка окружения, загрузка библиотек), необходимые для связи с распределенными вычислительными ресурсами. Далее осуществляется открытие сессии и установление соединения. Для построения модели угроз и оценки состояния нарушения осуществим моделирование процесса получения доступа с возможными нарушениями:

- *атака, направленная на подбор пароля:* атака, в основе которой лежит метод перебора по словарю. С точки зрения теории вероятностей выбор пароля детерминирован и закономерен. В качестве пароля могут выступать: дата рождения, имя, предмет, набор цифр, последовательность близко расположенных на клавиатуре букв. В общем случае происходит конкатенация вышеперечисленного, поэтому предопределенность в выборе па-

роля играет ключевую роль в выборе алгоритмов, на которых основан метод перебора по словарю. Алфавитный пароль, сгенерированный человеком, неравномерно распределен в пространстве алфавитных последовательностей. Данное условие может быть учтено с большой точностью в марковских фильтрах нулевого и первого порядка (*нулевой порядок модели Маркова*: каждый символ генерируется в соответствии со своим вероятностным распределением и независимо от предыдущих символов; *первый порядок модели Маркова*: каждой диаграмме (упорядоченной паре) символов присваивается вероятность и каждый символ порождается в условной зависимости от предыдущего);

- *атака, направленная на перехват сетевых пакетов:* поскольку при передаче данных создается непрерывный динамический канал, проходящий через отдельные сегменты сети, создается возможность прослушивания и перехвата сетевых пакетов;
- *атака, направленная на навязывание ложного маршрутизатора:* атака канального или сетевого уровня, приводящая к перенаправлению сетевых пакетов жертвы или всего трафика сегмента на ненадлежащий адрес или к отказу в обслуживании (no route to host). Для защиты от ложного маршрута и подмены сервера применим технологию DNS поверх TLS. Целью данного метода является повышение конфиденциальности и безопасности пользователей путем предотвращения перехвата и манипулирования данными DNS.

Для представленных атак граф перехода из состояния в состояние и соответствующая система дифференциальных уравнений представлены на рис. 4, а, б.

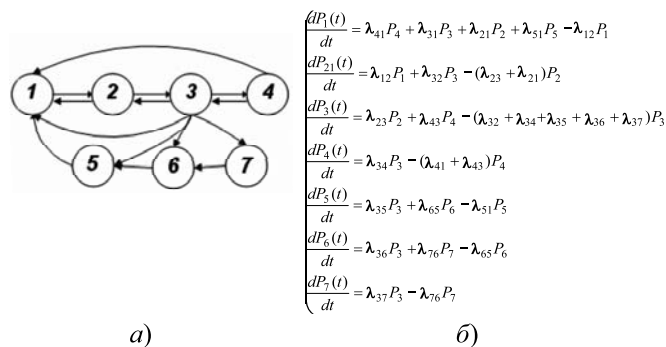


Рис. 4. Граф перехода из состояния в состояние и соответствующая система дифференциальных уравнений:

а — штатный процесс установления соединения; б — система дифференциальных уравнений

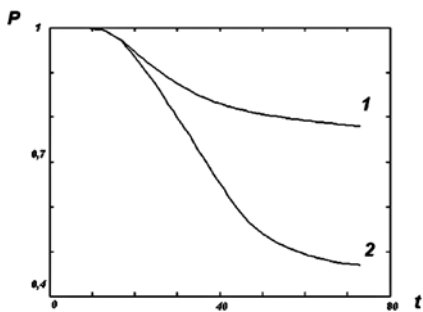


Рис. 5. Вероятность запуска программы

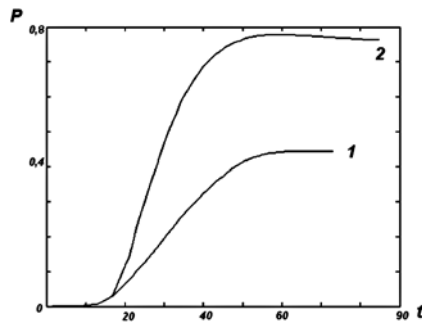


Рис. 6. Вероятность открытия сессии

В момент запуска программы (рис. 5, кривая 1) вероятность приближена к единице и с течением времени снижается (согласно проблеме останова) до состояния, когда можно утверждать, что программа зависла. Критическая точка процесса определена 20-й секундой. На рис. 5 представлен график кривой 2, полученный с использованием моделирования мероприятий по повышению вероятности успешного запуска (использование "песочницы", т. е. изолированной среды исполнения, антивирусной защиты, контроля целостности приложений и разделение прав доступа пользователей). Это позволяет оценить влияние введения тех или иных мер по защите на протекание процесса.

Зависимость процесса открытия сессии от процесса запуска программы. Вероятность открытия сессии обеспечивается запуском программы (рис. 6, кривая 1), созданием нового соединения, инициируемого исполняемым файлом, с передачей пакета. Будем считать, что одновременно с созданием сессии открывается порт TCP/IP для последующей передачи данных и установления соединения с распределенными вычислительными ресурсами. Такая последовательность действий обеспечивает рост вероятности до определенного момента, на которую могут повлиять атаки: MITM, DDOS. Для обеспечения успешного открытия сессии, а также в целях противодействия нарушениям безопасности и в качестве мер повышения вероятности было предложено шифрование сетевого трафика на канальном уровне (аппаратный VPN "точка-точка" с криптографическим модулем). Анализ результатов моделирования показал эффективность проведенных мероприятий по защите (рис. 6, кривая 2).

Вероятность успешной передачи данных по каналам связи. Как и на предыдущем графике, прослеживается зависимость процесса передачи данных по каналам связи от процессов, его образующих (рис. 7, кривая 1). Критической точкой жизнеспособности процесса является тот же момент, при котором в процессе запуска приложения появляются те

или иные проблемы с загрузкой. На рис. 7 представлен график кривой 2, полученный после моделирования мер по повышению вероятности успешной передачи данных с использованием туннелирования в каналах передачи данных (VPN-канал). Моделирование показало, что влияние туннелирования на процесс передачи не существенно, однако вероятность передачи повышается.

Результаты оценки вероятности того, что в данный момент происходит установление соединения с сервером, аналогичны предыдущему процессу (рис. 8, кривая 1). На рис. 8 (кривая 2) изображена вероятность успешного установления соединения с сервером при противодействии нарушению безопасности. Были реализованы следующие меры: ограничение доступа по геолокации (GeoIP Block), динамическая блокировка IP-адресов "разрешенных стран" согласно правилам фильтрации пакета, установлен VPN-канал.

Вероятность атаки, направленной на подбор пароля, моделируется в момент соединения программы с сервером (рис. 9, кривая 1). С точки зрения теории вероятности выбор пароля детерминирован, поэтому применение генераторов паролей с аппаратным датчиком случайных чисел снижает вероятность успешной атаки (рис. 9, кривая 2).

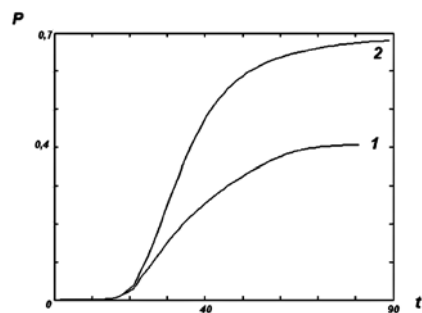


Рис. 7. Вероятность успешной передачи данных по каналам связи

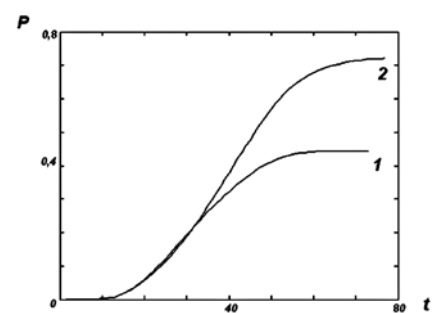


Рис. 8. Вероятность установления соединения с сервером

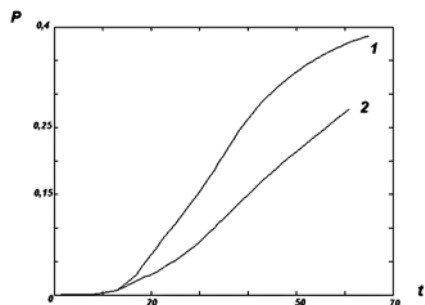


Рис. 9 Вероятность успешной атаки подбора пароля

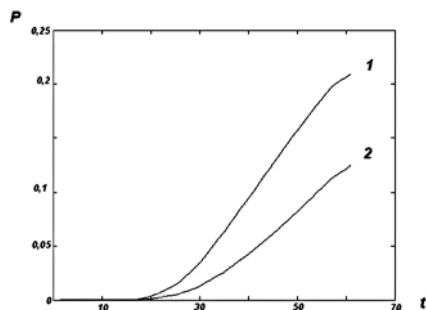


Рис. 10. Вероятность успешного перехвата сетевых пакетов

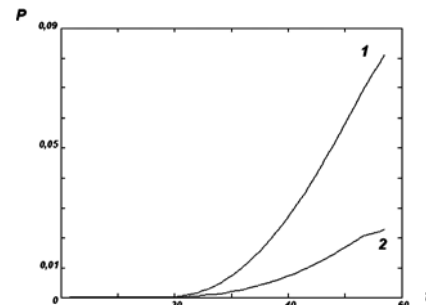


Рис. 11. Вероятность успешного навязывания ложной маршрутизации пакетов

Вероятность успешного перехвата сетевых пакетов в целях их дальнейшего анализа рассчитывается на временном интервале (рис. 10, кривая 1) и связана с установлением соединения между приложением и сервером. Рис. 10 (кривая 2) демонстрирует, что процесс обнаружения атак (процесс оценки подозрительных действий узлов сети) понижает вероятность перехвата.

Атака канального или сетевого уровня приводит (рис. 11, кривая 1) к перенаправлению сетевых пакетов жертвы или всего трафика сегмента на ненадлежащий адрес или к отказу в обслуживании (no route to host). Вероятность атаки зависит от успешности соединения с сервером, повышается по мере его установления. Результаты моделирования противодействия навязывания ложной маршрутизации пакетов с применением фильтрации проходящих ICMP-сообщений Redirect понижает вероятность атаки почти вдвое (рис. 11, кривая 2).

Полученные результаты моделирования с учетом всей совокупности введенных мероприятий по защите (в таблице и на рис. 12) позволяют определить влияние мер по обеспечению безопасности на уязвимые элементы в анализируемых процессах. Полученные решения применимы при планировании мероприятий по защите критически важных объектов от рассматриваемых угроз с учетом количественной оценки вероятности нарушения. Использование данной методики позволяет оценить целесообразность мероприятий обеспечения безопасности в анализируемом объекте.

Целесообразность таких мероприятий определяется достижением максимальной защиты при минимальных затратах и характеризуется следующими факторами: масштабом и ресурсоемкостью сети, необходимостью применения однотипных технологий и средств защиты информации при проведении мероприятий по обеспечению системы защиты информации.

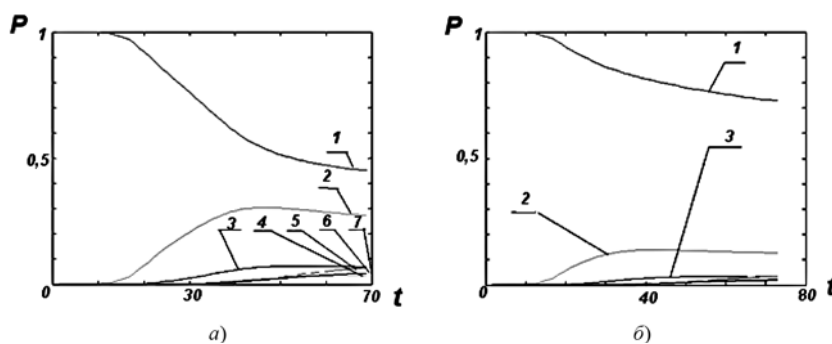


Рис. 12. Оценка вероятности нарушения безопасности:

а — без мер защиты; б — с учетом всей совокупности введенных мероприятий по защите

Оценка вероятности нарушения безопасности

Процесс	Вероятность без мероприятий по обеспечению безопасности	Вероятность при мероприятиях по обеспечению безопасности
Запуск программы (кривая 1 на рис. 12)	0,47	0,77
Открытие сессии (кривая 2 на рис. 12)	0,3	0,7
Передача данных (кривая 3 на рис. 12)	0,07	0,5
Соединение с сервером (кривая 4 на рис. 12)	0,07	0,6
Атака, направленная на подбор пароля (кривая 5 на рис. 12)	0,03	0,18
Атака, направленная на перехват сетевых пакетов (кривая 6 на рис. 12)	0,03	0,09
Атака, направленная на навязывание ложного маршрутизатора (кривая 7 на рис. 12)	0,03	0,04

Заключение

Анализ полученных в ходе исследования результатов моделирования процессов безопасности виртуальной среды клиента и применения комплексного подхода к обеспечению информационной безопасности показал следующее. Современные стереотипы по защите процессов клиентов виртуальной среды не в полной мере готовы противостоять текущим угрозам и новым вызовам безопасности. В связи с этим возникает необходимость совершенствования научно-методического аппарата. На примере конкретной задачи с типовыми процессами использована марковская модель для выявления наиболее значимых угроз, целесообразных мероприятий защиты и прогнозирования состояния системы. Аналогичная методика может быть использована для других моделей безопасности, для нахождения целесообразных мероприятий защиты и прогнозирования состояния системы. Применение полученных результатов в процессе построения, эксплуатации и модернизации системы информационной безопасности, предоставляющей облачные услуги, ведет к повышению эффективности и качества обслуживания и позволяет эффективно управлять процессом информационной безопасности.

Список литературы

1. **Analysis** of Cloud related Security and risks mitigation IRACST // International Journal of Advances Computing, Engineering and Applications (IJACEA). 2012. Vol. 1, N. 2. P. 40–49, 2012.
2. **Гюнтер Е. С., Нарутта Н. Н., Шахов В. Г.** "Облачные" вычисления и проблемы их безопасности // Омский научный вестник. 2013. № 2 (120). С. 279–282.
3. **Саклаков В. М.** Облачные вычисления: современные модели предоставления услуг и возможные риски // Молодежь и современные информационные технологии: сб. тр. XIII Междунар. науч.-практ. Конф. студентов, аспирантов и молодых ученых, г. Томск, 9–13 ноября 2015 г. Томск: Изд-во ТПУ, 2016. Т. 2. С. 104–106.
4. **Фролов Д. Б., Грунюшкина С. А., Старостин А. В.** Информационная геополитика и сеть Интернет. М.: РФК-Имидж Лаб, 2008. 404 с.
5. **Юсупов Р. М., Шишкин В. М.** О некоторых противоречиях в решении проблем информационной безопасности // Труды СПИИРАН. 2008. № 6. С. 11–23.

6. **Петров Д. Л.** Оптимальный алгоритм миграции данных в масштабируемых облачных хранилищах // Управление большими системами: сб. тр. 2010. № 30. С. 180–197.
7. **Воробьев В. И., Рыжков С. Р., Фаткиева Р. Р.** Защита периметра облачных вычислений // Программные системы: теория и приложения. 2015. Т. 6, № 1 (24). С. 61–71.
8. **Коваленко О. С.** Обзор состояний, проблем и перспектив хранения и анализа данных в "облаке" // Информатика, вычислительная техника и инженерное образование. 2011. № 5. С. 39–49.
9. **Каменщиков А. А.** Облачные технологии и интероперабельность информационных систем в здравоохранении // Журнал радиоэлектроники: электронный журнал. 2013. № 2. URL: <http://jre.cplire.ru/jre/feb13/11/text.pdf>. (дата обращения: 28.06.18).
10. **Чемеркин Ю. С.** Облачные вычисления как инструмент обработки конфиденциальной информации // Вестник РГГУ. Сер.: Документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. 2012. № 14. С. 53–65.
11. **Mauw S., Verschuren J. H. S., de Vink E. P.** A formalization of anonymity and onion routing // Computer Security—ESORICS 2004. Springer Berlin Heidelberg, 2004. С. 109–124.
12. **Gupta S., Kumar P.** Taxonomy of cloud security // International Journal of Computer Science, Engineering and Applications (IJCEA). October 2013. Vol. 3, N. 5, P. 47–67.
13. **Lonea A. M., Popescu D. E., Tianfield H.** Detecting DDoS attacks in cloud computing environment // International Journal of Computers Communications & Control. 2013. Vol. 8. P. 70–78.
14. **Simma A.** Trusting Your Cloud Provider: Protecting Private Virtual Machines // Magdeburger Journal zur Sicherheitsforschung, Vol. 1. P. 530–539. Retrieved June 17, 2015, URL: <http://www.sicherheitsforschung-magdeburg.de/publikationen.html> (Version 2015/06/16 20: 01).
15. **Pivoting in Amazon Clouds** URL: <http://andresriancho.github.io/nimbostratus/pivoting-in-amazon-clouds.pdf> html (Version 2013/10/27 20: 48).
16. **Adrian D. et al.** Imperfect forward secrecy: How Diffie-Hellman fails in practice // Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015. P. 5–17.
17. **Осипов В. Ю., Носаль И. А.** Обоснование мероприятий обеспечения информационной безопасности // Информационно-управляющие системы. 2013. № 2 (63). С. 48–53.
18. **Осипов В. Ю., Носаль И. А.** Обоснование периода пересмотра мероприятий по защите информации // Информационно-управляющие системы. 2014. № 1. С. 63–69.
19. **Потапов В. И.** Постановка и решение игровой задачи противоборства аппаратно-избыточной динамической системы с атакующим противником, действующим в условиях неполной информации в процессе конфликта // Мехатроника, автоматизация, управление. 2017. Т. 18, № 8. С. 525–531.

Assessment of Violations of Information Security Perimeter in the Cloud

The processes influencing the safety of the virtual environment of the client, from the moment of establishing the connection till obtaining the results, were investigated. It is proposed to use the Markov model to identify the most significant threats and to predict the state of the system. It is shown that due to the information security, not only the probability of successful regular workload is increased, but also the probability of collecting additional data by an intruder due to increased interest in the system. The analysis of the results of modeling the processes of security of the virtual environment of the client and the application of an integrated approach to ensuring information security, showed the following. Modern stereotypes to protect the processes of virtual environment are not fully ready to withstand current threats and new security challenges. In this regard, there is a strong need to improve the scientific and methodological apparatus. Using the example of a specific task with typical processes, the Markov model was used to identify the most significant threats, expedient measures for protecting and predicting the state of the system. A similar methodology can be used for other security models, for finding appropriate protection measures and predicting the state of the system. Application of the results obtained in the process of building, operating and upgrading the information security for cloud services leads to increased efficiency and quality of service and allows to effectively manage the information security.

Keywords: information security, dynamic perimeter, cloud environment, attacks on data

DOI: 10.17587/it.24.791-798

References

1. **Analysis** of Cloud related Security and risks mitigation IRACST, *International Journal of Advances Computing, Engineering and Applications (IJACEA)*, 2012, vol. 1, no. 2, pp. 40–49.
2. **Giunter E. S., Narutta N. N., Shahov V. G.** "Oblachnyye" vychisleniia i problemy ikh bezopasnosti (Cloud Computing and Security Problems), *Omskii Nauchnyi Vestnyk*, 2013, no. 2 (120), pp. 279–282.
3. **Saclakov V. M.** Oblachnyye vychisleniia: sovremennye modeli predostavleniia uslug i vozmozhnye riski (Cloud computing: modern service delivery models and possible risks), *Molodezh i sovremennye informatsionnye tekhnologii: sbornik trudov XIII Mezh-dunarodnoi nauchno-prakticheskoi konferentsii studentov, aspirantov i molodykh uchennykh, g. Tomsk, 9–13 noiabria 2015 g.* Tomsk, Publishing house of TPU, 2016, vol. 2, pp. 104–106 (in Russian).
4. **Frolov D. B., Gruniushkina S. A., Starostin A. V.** *Informatsionnaia geo-politika i set Internet* (Information geopolitics and the Internet), Moscow, RFK-Imidzh Lab, 2008, 404 p. (in Russian).
5. **Iusupov R. M., Shishkin V. M.** O nekotorykh protivorechiakh v reshenii problem informatsionnoi bezopasnosti (On some contradictions in solving information security problems), *Trudy SPIIRAN*, 2008, no. 6, pp. 11–23 (in Russian).
6. **Petrov D. L.** Optimalnyi algoritm migratsii dannykh v masshtabiruemykh oblachnykh khranilishchakh (The optimal algorithm for data migration in scalable cloud storage), *Upravlenie bolshimi sistemami: sbornik trudov*, 2010, no. 30 (in Russian).
7. **Vorobev V. I., Ryzhkov S. R., Fatkueva R. R.** Zashchita perimetra oblachnykh vychislenii (Protecting the perimeter of cloud computing), *Programmnye sistemy: teoriia i prilozheniia*, 2015, vol. 6, no. 1 (24), pp. 61–71 (in Russian).
8. **Kovalenko O. S.** Obzor sostoiianii, problem i perspektiv khraneniia i analiza dannykh v "oblake" (Overview of the states, problems and perspectives of data storage and analysis in the "cloud"), *Informatika, Vychislitelnaia Tekhnika i Inzhenernoe Obrazovanie*, 2011, no. 5, pp. 39–49 (in Russian).
9. **Kamenshchikov A. A.** Oblachnye tekhnologii i interoperabelnost informatsionnykh sistem v zdravookhraneni (Cloud technologies and interoperability of information systems in healthcare), *Zhurnal Radioelektroniki*, 2013, no. 2, available at: http://j.re.cplire.ru/j_re/feb/13/11/text.pdf. Version 2018/06/28 15: 01 (in Russian).
10. **Chemerkhin Iu. S.** Oblachnyye vychisleniia kak instrument obrabotki konfidentsialnoi informatsii (Cloud computing as a tool for processing confidential information), *Vestnyk RGGU. Seriya: Dokumentovedenie i arkhivovedenie. Informatika. Zashchita informatsii i informatsionnaia bezopasnost*, 2012, no. 14, pp. 53–65 (in Russian).
11. **Mauw S., Verschuren J. H. S., de Vink E. P.** A formalization of anonymity and onion routing, *Computer Security—ESORICS 2004*, Springer Berlin Heidelberg, 2004, pp. 109–124.
12. **Gupta S., Kumar P.** Taxonomy of cloud security, *International Journal of Computer Science, Engineering and Applications (IJCSA)*, October 2013, vol. 3, no. 5, pp. 47–67.
13. **Lonea A. M., Popescu D. E., Tianfield H.** Detecting DDoS attacks in cloud computing environment, *International Journal of Computers Communications & Control*, 2013, vol. 8, pp. 70–78.
14. **Simma A.** Trusting Your Cloud Provider: Protecting Private Virtual Machines, *Magdeburger Journal zur Sicherheitsforschung*, Vol. 1, pp. 530–539. Retrieved June 17, 2015, URL: <http://www.sicherheitsforschung-magdeburg.de/publikationen.html> Version 2015/06/16 20: 01. — 2015.
15. **Pivoting in Amazon Clouds** available at: <http://andresrianchogithub.io/nimbostratus/pivoting-in-amazon-clouds.pdf> html Version 2013/10/27 20: 48. — 2013.
16. **Adrian D.** et al. Imperfect forward secrecy: How Diffie-Hellman fails in practice, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM*, 2015, pp. 5–17.
17. **Osipov V. Iu., Nosal I. A.** Obosnovanie meropriiati obeshpecheniia informatsionnoi bezopasnosti (Rationale for measures to ensure information security), *Informatsionno-Upravliaiushchie Sistemy*, 2013, no. 2 (63), pp. 48–53 (in Russian).
18. **Osipov V. Iu., Nosal I. A.** Obosnovanie perioda peresmotra meropriiati po zashchite informatsionnoi bezopasnosti (Justification of the revision period for information protection measures), *Informatsionno-Upravliaiushchie Sistemy*, 2014, no. 1, pp. 63–69 (in Russian).
19. **Potapov V. I.** Postanovka i reshenie igrovoi zadachi protivoborstva apparatno-izbytochnoi dinamicheskoi sistemy s atakuiushchimi protivnikom, deistvuiushchimi v usloviakh nepolnoi informatsii v protsesse konflikta (Statement and solution of the game problem of confrontation of the hardware-redundant dynamic system with the attacking enemy operating under incomplete information in the process of conflict), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18, no. 8, pp. 525–531 (in Russian).

В. Д. Корепанов, студент, e-mail: korepanov.viacheslav@yandex.ru,
В. П. Кулагин, д-р техн. наук, проф., e-mail: Kulagin_VP@mgupi.ru,
Р. Ф. Халабия, канд. техн. наук, доц., e-mail: rustam-capitan@mail.ru,
Российский технологический университет

Использование нейронных сетей при модификации генетического алгоритма

Генетические алгоритмы находят все большее применение при разработке современного программного обеспечения в системах искусственного интеллекта, при решении оптимизационных задач, при построении искусственных нейронных сетей, а также в других отраслях знаний. Актуальным и перспективным направлением является совместное использование генетических алгоритмов и нейронных сетей, в результате чего образуются гибридные системы, предназначенные для решения конкретных практических задач. В статье рассматривается подход к модификации генетического алгоритма элементами нейронной сети. Целью данной модификации является построение нового, более эффективного алгоритма, превосходящего на определенном классе задач классический генетический алгоритм. Анализ эффективности предложенного алгоритма проведен на основе результатов, полученных при исследовании устройств распознавания запаха широкого класса химических веществ "электронный нос".

Ключевые слова: генетический алгоритм, особь, хромосома, ген, размножение, мутация, приспособленность (пригодность), нейронная сеть

Введение

В настоящее время генетические алгоритмы находят все большее применение при разработке программного обеспечения, в системах искусственного интеллекта, при решении оптимизационных задач, при построении искусственных нейронных сетей, а также в других отраслях знаний. Следует отметить, что сейчас с помощью генетических алгоритмов решаются задачи, для которых ранее использовались только нейронные сети. Генетические алгоритмы часто применяются совместно с нейронными сетями, образуя гибридные системы, предназначенные для решения конкретных задач.

В настоящей статье рассматриваются вопросы модификации генетического алгоритма элементами нейронных сетей. Целью данной модификации является построение нового, более эффективного алгоритма, превосходящего на определенном классе задач классический генетический алгоритм. В качестве данных для анализа эффективности предложенного алгоритма взяты результаты, полученные при

исследовании устройств распознавания широкого класса химических веществ "электронный нос" [8–13].

Классический генетический алгоритм

Генетический алгоритм представляет собой математическую модель эволюции популяции особей [4]. На примере описания алгоритма введем основные термины, которые будут использованы в данной работе. Каждая особь популяции характеризуется определенным числом хромосом h . Хромосома представляется последовательностью генов (цепочкой элементов) $S_k^m = (S_{k1}^m, S_{k2}^m, \dots, S_{kn}^m)$, где m — номер хромосомы k -й особи. Будем считать, что каждая хромосома имеет одинаковое число генов n . Хромосома определяет приспособленность (пригодность) особи к селекции и репродукции. Функцию пригодности обозначим $f_k = f(S_k)$, $k = \overline{1, N}$, где N — численность популяции. Цель генетического алгоритма состоит в нахождении особи с максимальным

значением функции приспособленности $f(S_k)$. Эволюция популяции представляется последовательностью поколений. На каждом следующем шаге состав популяции меняется. Каждое следующее поколение репродуцирует особи с относительно большими значениями функции $f(S_k)$. Соответствующие хромосомы приспособленных особей ("родителей") "скрещиваются" и подвергаются "мутациям". Процесс размножения продолжается до тех пор, пока не достигнуто условие окончания эволюционного поиска (например, достижение заданного предела числа поколений, определяемого эмпирически).

Модификация генетического алгоритма

Модификация классического генетического алгоритма посредством нейронной сети выполняется в целях уменьшения вычислительной трудоемкости. Следует отметить, что при этом происходит некоторая потеря точности. Однако для определенного класса задач (например, для распознавания образа газовой смеси [8–13] или при анализе и сравнении химических составов жидких и твердых объектов [21]) указанная потеря точности не наносит непоправимого ущерба, но время, выигранное при этом, в некоторых случаях имеет очень важное (первостепенное) значение.

Опишем модель размножения особей на примере функционирования персептрона [2, 3]. Хромосомы родителей поэлементно подаются на входы B_1 и B_2 (рис. 1). На выходе C поэлементно формируются хромосомы новой особи.

Пусть имеется множество значений, которые получены при анализе некоторого объекта и которые можно использовать в качестве эталонных при обучении персептрона. При обучении персептрона воспользуемся дельта-правилом, которое укрупненно можно представить в виде следующих шагов [7]:



Рис. 1. Схема работы персептрона для размножения (S_i^x , S_i^y — i -й ген хромосом с номерами x и y соответственно, подаваемый на входы B_1 и B_2 соответственно; w_1 , w_2 — веса входов)

- 1) задание весов входов персептрона (малые случайные величины);
- 2) подача на входы персептрона данных для обучения;
- 3) определение выходных значений персептрона;
- 4) нахождение разности между эталонным значением и значением, полученным на выходе персептрона;
- 5) корректировка весов в целях уменьшения ошибки;
- 6) повторение шагов 2–5, пока ошибка не достигнет одного из минимумов (локального либо глобального).

В процессе корректировки новое значение веса определяется в соответствии с выражением

$$w_j(t+1) = w_j(t) + E\eta z_j, \quad (1)$$

где j — номер входа персептрона; $w_j(t+1)$ — новое значение веса j -го входа; $w_j(t)$ — старое значение веса j -го входа; E — ошибка (разность между эталонным значением и текущим значением на выходе персептрона), при $t = 0$ в качестве текущего значения берется заданная малая величина; $0 < \eta < 1$ — коэффициент, задающий скорость (точность) работы алгоритма; z_j — данные, пришедшие на j -й вход персептрона [5].

Следует отметить, что при подстановке вместо персептрона некоторой нейронной сети все приведенные выше рассуждения остаются в силе [1, 6].

Распознавание образа вещества на основе модифицированного генетического алгоритма

Для демонстрации возможностей модифицированного алгоритма воспользуемся экспериментальными данными, полученными на установке "электронный нос" [20], функциональная схема которой представлена на рис. 2 (см. третью сторону обложки).

Аналоговый образ газовой смеси, снимаемый с датчиков установки, приведен на рис. 3 (см. третью сторону обложки).

Цифровая форма аналогового образа представлена на рис. 4. Данный одномерный массив будем называть хромосомой. В нашем случае образ газоздушной смеси (особь) формируется восемью хромосомами, что в данном случае определяется числом датчиков газочувствительной матрицы.

Рис. 4. Цифровая форма образа изопропила

Гены хромосомы соответствуют точкам графика. Число точек (генов) определяется шагом оцифровывания и выбирается произвольно исходя из точности представления аналогового сигнала.

В соответствии с данными, поступающими со входа установки, создаем N одинаковых особей, каждая из которых мутирует в соответствии с некоторым правилом (рассмотрено ниже). Таким образом, N особей образуют популяцию.

Каждая особь поочередно выбирается в качестве первого родителя. Затем определяется факт размножения особи на данном этапе эволюции в соответствии со следующим правилом:

$$f(r_0) = \begin{cases} 0, r_0 \geq q_0; \\ 1, r_0 < q_0, \end{cases} \quad (2)$$

где r_0 — псевдослучайное вещественное число из отрезка $[0; 1]$; q_0 — постоянная величина, равная некоторому вещественному числу из отрезка $[0; 1]$, подбираемая эмпирически.

Если на данном этапе эволюции $f(r_0) = 1$, то происходит размножение особи. В противном случае происходит переход к анализу следующей особи. Номер второго родителя определяется случайно.

Новая хромосома очередной особи, в случае использования классического алгоритма, создается из двух соответствующих хромосом-родителей, каждый ген новой хромосомы вычисляется по формуле

$$\begin{cases} S_i^m = S_i^x + c(S_i^y - S_i^x), r \bmod 2 = 0; \\ S_i^m = S_i^y - c(S_i^y - S_i^x), r \bmod 2 \neq 0, \end{cases} \quad (3)$$

где S_i^m — i -й ген новой хромосомы с номером m ; r — псевдослучайное целое число; c — постоянная величина, равная некоторому вещественному числу из отрезка $[0; 1]$, подбираемая эмпирически.

В случае использования модифицированного алгоритма размножение происходит с учетом работы персептрона (см. рис. 1).

Новая хромосома с вероятностью q мутирует, т. е. псевдослучайным образом выбирается один из ее генов и изменяется по формуле

$$S_i^m = pS_i^m, r_0 < q, \quad (4)$$

где S_i^m — новое значение выбранного i -го гена m -й хромосомы; p — параметр мутации (псевдослучайное вещественное число, выбираемое на отрезке $[a, b]$); q — постоянная величина, равная некоторому вещественному числу на отрезке $[0; 1]$, подбираемая эмпирически; a, b — постоянные величины, равные некоторым вещественным числам, подбираемым эмпирически: $a, b \in (-\infty; +\infty)$.

Процесс размножения, например, может быть закончен в момент достижения наперед заданного числа поколений.

Для каждой хромосомы с номером m вычисляем значение отклонения (погрешности):

$$\delta = \sum_{i=1}^N |S_i^m - Y_i^m|, \quad (5)$$

где S_i^m — i -й ген m -й хромосомы особи, полученной в результате работы алгоритма с данными со входа установки; Y_i^m — i -й ген m -й хромосомы особи, представленной образом из базы данных.

Итоговая особь образуется из хромосом с наименьшей погрешностью для взятого из базы данных образа. Погрешность итоговой особи Δ есть сумма погрешностей всех ее хромосом.

Каждому образу соответствует единственная итоговая особь. Итоговую особь, имеющую наименьшую погрешность Δ_{\min} , предлагается рассматривать в качестве результата идентификации исследуемого образа. Будем использовать обозначения Δ и Δ_{\min} для модифицированного алгоритма, Δ_0 и $\Delta_{0\min}$ — для классического.

Сравнительный анализ алгоритмов

Сравнительный анализ классического и модифицированного алгоритмов проведем на примере данных, полученных с помощью установки "электронный нос", представленной на рис. 2. С этой целью введем критерии оценки: величину μ , определяемую отношением времени работы модифицированного генетического алгоритма t_1 ко времени работы классического генетического алгоритма t_0 при анализе одного и того же подмножества образов, и величину ε , характеризующую относительную точность работы алгоритмов:

$$\mu = \frac{t_1}{t_0}; \quad (6)$$

$$\varepsilon = \frac{\Delta_{0\min}}{\Delta_{\min}}. \quad (7)$$

Анализируя выражение (3), можно отметить, что в случае применения классического алгоритма для вычисления одного гена хромосомы необходимо выполнить следующие шаги:

- 1) генерация числа r ;
- 2) вычисление выражения $r \bmod 2$;
- 3) сравнение результата, полученного на шаге 2, с нулем;
- 4) получение разности $(S_i^y - S_i^x)$;
- 5) умножение результата, полученного на шаге 4, на значение c , либо на $(-c)$;
- 6) сложение результата, полученного на шаге 5, с S_i^x , либо с S_i^y .

Будем считать, что шаги 1 и 5 являются более трудоемкими и описываются длинными операциями.

В случае использования модифицированного алгоритма вычисление одного гена хромосомы определяется следующими шагами:

- 1) умножение S_i^x на вес w_1 ;
- 2) умножение S_i^y на вес w_2 ;
- 3) сложение результатов, полученных на первых двух шагах.

В этом случае шаги 1 и 2 представляются длинными операциями, а 3 шаг — короткой.

Пусть в базе данных содержится P образов. На распознавание подается последовательность образцов, включающая R элементов. Среднее арифметическое числа длинных и числа коротких операций классического алгоритма в случае распознавания одного образа обозначим M_0 и M'_0 соответственно. Аналогично M_1

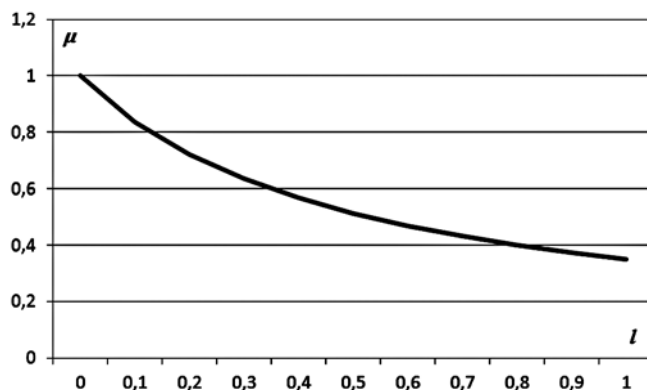


Рис. 5. График функции $\mu(l)$

и M'_1 — для модифицированного алгоритма. Введем величину l , определяемую отношением времени выполнения короткого шага dt ко времени выполнения длинного шага $d\tau$:

$$l = \frac{dt}{d\tau}. \quad (8)$$

Отсюда выражение (6) может быть преобразовано к следующему виду:

$$\mu = \frac{t_1}{t_0} = \frac{M'_1 dt + M_1 d\tau}{M'_0 dt + M_0 d\tau} = \frac{M'_1 l + M_1}{M'_0 l + M_0}. \quad (9)$$

Для нашего случая при $P = 33$ и $R = 100$ имеем:

$$\mu = \frac{13\ 204\ 308l + 264\ 086\ 160}{527\ 939\ 400l + 263\ 969\ 700}. \quad (10)$$

Построим график зависимости μ от l (рис. 5).

Из графика видно, что модифицированный алгоритм является более предпочтительным при больших значениях l .

Для демонстрации работы модифицированного алгоритма распознаем один из образов базы данных.

Пусть необходимо распознать образ изопронила (рис. 6, см. третью сторону обложки).

На основе полученных вычислений имеем:

$$\mu = \frac{87,4227}{123,2537} \approx 0,7093; \quad \varepsilon = \frac{138,6752}{142,3551} \approx 0,9741.$$

Модифицированный алгоритм тем предпочтительнее, чем сложнее способ размножения (чем большее число длинных операций требуется для вычисления одного гена хромосомы).

Заключение

Следует отметить, что модифицированный генетический алгоритм является более предпочтительным в тех случаях, когда процедура размножения хромосом включает достаточно большое число трудоемких (длинных) операций. Возможная потеря точности при этом не имеет решающего значения для рассматриваемого класса задач, что подтверждается экспериментом.

Описанный алгоритм гибридной системы с успехом может быть применен и для анализа других информационных потоков, например космического излучения, представляющего сегодня актуальнейшее направление современных исследований [14–19].

Список литературы

1. Аксенов С. В., Новосельцев Б. В. Организация и использование нейронных сетей (методы и технологии) // Томск: Издательство научно-технической литературы, 2006. 128 с.
2. Заенцев И. В. Нейронные сети: основные модели // Воронеж: Воронежский государственный университет, 1999. 74 с.
3. Каллан Р. Основные концепции нейронных сетей // М.: Издательский дом "Вильямс", 2001, 287 с.
4. Подлазова А. В. Генетические алгоритмы на примерах решения задач раскроя // Пробл. Управл., 2008, вып. 2. С. 57–63.
5. Хайкин С. Нейронные сети, полный курс. Издание второе, исправленное // М.: Издательский дом Вильямс, 2006. 1104 с.
6. Norbert Wiener. Cybernetics or Control and Communication in the Animal and the Machine. (Hermann & Cie Editeurs, Paris, The Technology Press, Cambridge, Mass., John Wiley & Sons Inc., New York, 1948)
7. Rumelhart D. E., Hinton G. E., Williams R. J. Learning Internal Representations by Error Propagation // Parallel Distributed Processing. Vol. 1. — Cambridge, MA: MIT Press, 1986.
8. Ivanov A. I., Kulagin V. P., Kapferko A. F., Kuznetsov Y. M., Obolyaeva N. M., Chulkova G. M., Yurin A. N., Shustrov A. V. Instrumental means for the control and analysis of gas mixtures based on the use of artificial neural networks (Conference Paper), in: Proceedings of the 2017 International Conference "Quality Management, Transport and Information Security, Information Technologies", St. Petersburg; Russian Federation; 23–30 September 2017. P. 734–738.
9. Kulagin V., Kuznetsov Y. M., Chulkova G. M., Львов С., Шустров А., Иванов А. Intelligent multi-sensor measurement device for recognition of gas-air mixture samples with the use of artificial neural networks, in: Proceedings of the XX IEEE International Conference on Soft Computing and Measurements (SCM 2017), 24–26.05.2017, Saint Petersburg. Издательство СПбГЭТУ "ЛЭТИ", 2017. doi P. 574–576.
10. Ivanov A. I., Kapferko A., Kuznetsov Y. M., Kulagin V., Chulkova G. M., Shustrov A. Information Engineering for Rapid

Recognition of Odors with the Help of "Electronic Nose", in: 11th IEEE International Conference on Application of Information and Communication Technologies (AICT) Vol. 1: 11th IEEE International Conference on Application of Information and Communication Technologies Conference Proceedings. Session 2 Data Mining and Data Engineering. M., 2017. Ch. 2. P. 210–212.

11. Kulagin V., Ivanov A. I., Kuznetsov Y. M., Chulkova G. M. Multidimensional mutual ordering of patterns using a set of pre-trained artificial neural networks // Journal of Physics: Conference Series. 2017. Vol. 803. N. 1.

12. Kulagin V., Kuznetsov Y. M., Chulkova G. M., Ivanov A., Ivannikov A. D. HighDimensional NeuralNetwork Artificial Intelligence Capable of Quick Learning to Recognize a New Smell, and Gradually Expanding the Database, in: 2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC). M.: IEEE, 2016. P. 332–335.

13. Кулагин В. П., Кузнецов Ю. М., Чулкова Г. М., Иванов А. Алгоритмы цифровой обработки информации на основе нейронных сетей для распознавания широкого класса химических веществ // Сб. докладов Всероссийской конференции "Радиоэлектронные средства получения, обработки и визуализации информации" (РСПОВИ-2016). М.: РНТОРЭС им. А. С. Попова, 2016. С. 204–209.

14. Истратов А. Ю., Захарченко К. В., Каперко А. Ф., Колюбин В. А., Кулагин В. П., Курочкин Р. И. Применение нейросетевого подхода к измерениям потоков космического излучения // Измерительная техника. 2016. № 3. С. 49–54.

15. Afanasiev V., Kapferko A., Kulagin V., Kolyubin V. Method of Adaptive Filtering in the Problem of Restoring Parameters of Cosmic Radiation Automation and Remote Control, 2017, Vol. 78, No. 3, pp. 397–412 / Пер. с рус. // Automation and Remote Control. 2017. No. 3. P. 397–412. doi

16. Каперко А. Ф., Кулагин В. П. Интеллектуальные методы и информационные технологии в процессах контроля и управления потоками ионизирующего излучения // Информационные технологии моделирования и управления. 2017. № 3. С. 45–50.

17. Афанасьев В. Н., Каперко А. Ф., Колюбин В. А., Кулагин В. П. Метод адаптивной фильтрации в задаче восстановления параметров космического излучения // Автоматика и телемеханика. 2017. № 3. С. 15–33.

18. Zakharchenko K. V., Kolyubin V., Kapferko A., Kulagin V., Nedosekin P., L'vov S. A. Modeling the operation of a sensor unit in a monitoring device for the parameters of cosmic ray fluxes / Пер. с рус. // Measurement Techniques. 2016. Vol. 59. No. 8. P. 884–891.

19. Zakharchenko K. V., Kapferko A., Kolyubin V., Kulagin V., L'vov S. A., Nedosekin P. Modeling the Operation of a Sensor Unit in a Monitoring Device for the Parameters of Cosmic Ray Fluxes // Measurement Techniques. 2016. Vol. 59. No. 8. P. 884–891.

20. Разработка нового поколения быстрообучаемых средств нейросетевого распознавания широкого класса химических веществ (высокоинтеллектуального искусственного носа) на основе твердотельных газочувствительных матриц / Номер гос. учета — АААА-Б 18-218030690084-1.

21. Lapshenkov E. M., Volkov V. Y., Kulagin V. P. Synthesis of compact patterns for NMR relaxation decay in intelligent "electronic tongue" for analyzing heavy oil composition // Journal of Physics: Conference Series, Volume 1015, Mathematical simulation and data processing, 2018. doi: 10.1088/1742-596/1015/3/032083

V. D. Korepanov, Bachelor's Degree student, e-mail: korepanov.viacheslav@yandex.ru,
V. P. Kulagin, Professor, e-mail: Kulagin_VP@mgupi.ru,
R. F. Halabiya, Associate Professor, e-mail: rustam-capitan@mail.ru,
MIREA — Russian Technological University

The Use of Neural Networks in the Modification of the Genetic Algorithm

Genetic algorithms are increasingly used in the development of modern software, in artificial intelligence systems, in the solution of optimization problems, in the construction of artificial neural networks, as well as in other branches of knowledge. An actual and promising direction is the joint use of genetic algorithms and neural networks. As a result hybrid systems are formed to solve specific practical problems. In the article the approach to modification of the genetic algorithm by elements of neural networks is considered. The purpose of this modification is to construct a new more efficient algorithm that surpasses the classical genetic algorithm in a certain class of problems. Efficiency analysis of the proposed algorithm is based on the results obtained in the "electronic nose" investigation (devices for odors recognition of a wide chemicals class).

Keywords: genetic algorithm, individual, chromosome, gene, reproduction, mutation, fitness, neural network

DOI: 10.17587/it.24.799-804

References

1. **Aksenov S. V., Novoselcev B. V.** Organizaciya i ispol'zovanie nejronnyh setej (metody i tekhnologii), Tomsk, Izdatel'stvo nauchno-tekhnicheskoy literatury, 2006. 128 p. (in Russian).
2. **Zaencev I. V.** Nejronnye seti: osnovnye modeli, Voronezh, Voronezhskij Gosudarstvennyj universitet, 1999, 74 p. (in Russian).
3. **Kallan R.** Osnovnye koncepcii nejronnyh setej, Moscow, Izdatel'skij dom "Vil'yams", 2001, 287 p. (in Russian).
4. **Podlazova A. V.** Geneticheskie algoritmy na primerah resheniya zadach raskroya, *Problemy Upravleniya*, 2008, iss. 2, pp. 57–63 (in Russian).
5. **Hajkin S.** Nejronnye seti, polnyj kurs. Izdanie vtoroje, ispravlennoe, Moscow, Izdatel'skij dom Vil'yams, 2006, 1104 p.
6. **Wiener N.** Cybernetics or Control and Communication in the Animal and the Machine, Hermann & Cie Editeurs, Paris, The Technology Press, Cambridge, Mass., John Wiley & Sons Inc., New York, 1948.
7. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning Internal Representations by Error Propagation, *Parallel Distributed Processing*, vol. 1, Cambridge, MA, MIT Press, 1986.
8. **Ivanov, A. I., Kulagin, V. P., Kaperko, A. F., Kuznetsov, Y. M., Obolyaeva, N. M., Chulkova, G. M., Yurin, A. N., Shustrov, A. V.** Instrumental means for the control and analysis of gas mixtures based on the use of artificial neural networks (Conference Paper), *Proceedings of the 2017 International Conference "Quality Management, Transport and Information Security, Information Technologies"*, St. Petersburg; Russian Federation; 23–30 September 2017, pp. 734–738.
9. **Kulagin V., Kuznetsov Y. M., Chulkova G. M., L'vov S., Shustrov A., Ivanov A.** Intelligent multi-sensor measurement device for recognition of gas-air mixture samples with the use of artificial neural networks, *Proceedings of the XX IEEE International Conference on Soft Computing and Measurements (SCM 2017)*, 24–26.05.2017, Saint Petersburg. Izdatel'stvo SPbGEHTU "LEHTI", 2017, pp. 574–576.
10. **Ivanov A. I., Kaperko A., Kuznetsov Y. M., Kulagin V., Chulkova G. M., Shustrov A.** Information Engineering for Rapid Recognition of Odors with the Help of "Electronic Nose", *11th IEEE International Conference on Application of Information and Communication Technologies (AICT)*, vol. 1: 11th IEEE International Conference on Application of Information and Communication Technologies Conference Proceedings. Session 2 Data Mining and Data Engineering. Moscow, 2017, part 2, pp. 210–212.
11. **Kulagin V., Ivanov A. I., Kuznetsov Y. M., Chulkova G. M.** Multidimensional mutual ordering of patterns using a set of pre-trained artificial neural networks, *Journal of Physics: Conference Series*, 2017, vol. 803, no. 1.
12. **Kulagin V., Kuznetsov Yu. M., Chulkova G. M., Ivanov A., Ivannikov A. D.** High Dimensional Neural Network Artificial Intelligence Capable of Quick Learning to Recognize a New Smell, and Gradually Expanding the Database, *2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, Moscow, IEEE, 2016, pp. 332–335.
13. **Kulagin V. P., Kuznecov Yu. M., Cpulkova G. M., Ivanov A.** Algoritmy cifrovoy obrabotki informacii na osnove nejronnyh setej dlya raspoznavaniya shirokogo klassa himicheskikh veshchestv, *Sbornik dokladov Vserossijskoj konferencii "Radioelektronnye sredstva polucheniya, obrabotki i vizualizacii informacii" (RSPOVI–2016)*, Moscow, RNTOREHS im. A. S. Popova, 2016, pp. 204–209 (in Russian).
14. **Istratov A. Yu., Zaharchenko K. V., Kaperko A. F., Kolyubin V. A., Kulagin V. P., Kurochkin R. I.** Primenenie nejrosetevogo podhoda k izmereniyam potokov kosmicheskogo izlucheniya, *Izmeritel'naya Tekhnika*, 2016, no. 3, pp. 49–54 (in Russian).
15. **Afanasiev V., Kaperko A., Kulagin V., Kolyubin V.** Method of Adaptive Filtering in the Problem of Restoring Parameters of Cosmic Radiation, *Automation and Remote Control*, 2017, vol. 78, no. 3, pp. 397–412.
16. **Kaperko A. F., Kulagin V. P.** Intellektual'nye metody i informacionnye tekhnologii v processah kontrolya i upravleniya potokam ioniziruyushchego izlucheniya, *Informacionnye Tekhnologii Modelirovaniya i Upravleniya*, 2017, no. 3, pp. 45–50 (in Russian).
17. **Afanas'ev V. N., Kaperko A. F., Kolyubin V. A., Kulagin V. P.** Metod adaptivnoj fil'tracii v zadache vosstanovleniya parametrov kosmicheskogo izlucheniya, *Avtomatika i Telemekhanika*, 2017, no. 3, pp. 15–33 (in Russian).
18. **Zakharchenko K. V., Kolyubin V., Kaperko A., Kulagin V., Nedosekin P., L'vov S. A.** Modeling the operation of a sensor unit in a monitoring device for the parameters of cosmic ray fluxes, *Measurement Techniques*, 2016, vol. 59, no. 8, pp. 884–891.
19. **Zakharchenko K. V., Kaperko A., Kolyubin V., Kulagin V., L'vov S. A., Nedosekin P.** Modeling the Operation of a Sensor Unit in a Monitoring Device for the Parameters of Cosmic Ray Fluxes, *Measurement Techniques*, 2016, vol. 59, no. 8, pp. 884–891.
20. **Razrabotka** novogo pokoleniya bystro obuchaemyh sredstv nejrosetevogo raspoznavaniya shirokogo klassa himicheskikh veshchestv (vysokointellektual'nogo iskusstvennogo nosa) na osnove tverdotel'nyh gazochuvstvitel'nyh matric, Nomer gos. ucheta — AAAA-B 18-218030690084-1 (in Russian).
21. **Lapshenkov E. M., Volkov V. Y., Kulagin V. P.** Synthesis of compact patterns for NMR relaxation decay in intelligent "electronic tongue" for analyzing heavy oil composition, *Journal of Physics: Conference Series*, vol. 1015, Mathematical simulation and data processing, 2018, doi: 10.1088/1742-596/1015/3/0320.

А. Г. Злобина, канд. техн. наук, науч. сотр., e-mail: ELF54@yandex.ru,
И. В. Журбин, д-р ист. наук, канд. техн. наук, гл. науч. сотр., e-mail: zhurbin@udm.ru,
Удмуртский федеральный исследовательский центр
Уральского отделения Российской академии наук, г. Ижевск

Сравнительный анализ алгоритмов сегментации данных электропрофилеирования при восстановлении границы объекта

Эффективным способом построения линии границы объекта искусственного происхождения, расположенного в приповерхностном слое грунта, является сегментация данных электропрофилеирования. Сравнительный анализ результатов обработки данных компьютерного моделирования и натурального эксперимента показал, что использование алгоритмов нечеткой кластеризации — fuzzy c-means и Густафсона—Кесселя — обеспечивает наибольшую степень соответствия восстановленной и истинной границы объекта поиска.

Ключевые слова: электропрофилеирование, объекты искусственного происхождения, локальная аномалия, сегментация, линия границы объекта, алгоритм fuzzy c-means, алгоритм Густафсона—Кесселя, критерий Hausdorff's distance, критерий pixel distance error, моделирование, натуральный эксперимент

Введение

Малоглубинная геофизика ориентирована на изучение верхней части геологического разреза, структура которого изменена в результате антропогенного или техногенного воздействия и содержит *объекты искусственного происхождения* — подземные инженерные сооружения и трубопроводы, места захоронения промышленных и бытовых отходов, объекты историко-культурного наследия и пр. Обычно такие объекты не выражены в рельефе, поэтому актуальной задачей является определение их расположения и геометрических параметров (форма, размеры, глубина залегания).

Для решения этой задачи часто применяют модификацию метода электроразведки — площадное электропрофилеирование [1, с. 173—185]. Методика исследований предполагает последовательное перемещение измерительной установки постоянного размера и конфигурации по выбранной сети наблюдений. В результате формируется *массив данных* — массив значений кажущегося удельного сопротивления грунта ρ_k (*кажущееся сопротивление*). Визуализация массива данных в виде карты распределения сопротивления (рис. 1) позволяет оце-

нить изменение электрических свойств грунта на участке исследований.

Анализ геофизических данных основан на аддитивной модели поля, в соответствии с которой результаты измерений фиксируют суммарное влияние региональной и локальной ано-

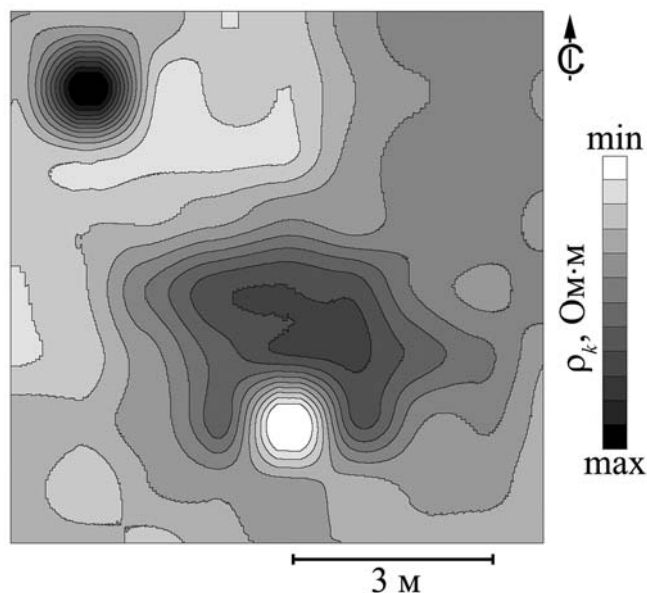


Рис. 1. Исследования Кушманского городища Учцакар IX—XIII вв. н.э. (фрагмент)

малый и некоррелированной помехи [1, с. 26]. *Локальной аномалией* принято считать такую составляющую геофизического поля, которая по линейным или площадным размерам меньше изучаемых профилей или площадей наблюдений в пять и более раз. Обычно эти аномалии вызваны *объектами поиска* — объектами искусственного происхождения (см. аномалию в центральной части участка на рис. 1).

Региональные аномалии соответствуют крупным геологическим структурам или явлениям, которые по своим размерам соизмеримы или превышают площадь наблюдений. Их влияние проявляется в виде плавного изменения сопротивления (северо-восточная часть участка на рис. 1). На территории, подвергавшейся антропогенному или техногенному воздействию, такие аномалии могут быть вызваны изменением мощности культурного слоя (результат эрозионных процессов на участках распашки, зоны накопления отходов, перепланировки территории при строительстве и пр.) или природными факторами (вариация уровня грунтовых вод, локальные изменения рельефа верхней части материковых пород и т. п.).

Под *помехами* (северо-западный угол участка, южная граница локальной аномалии, рис. 1) понимается всякое возмущение поля, обусловленное методическими и аппаратными погрешностями, препятствующее выделению полезного сигнала. Во многих случаях применение методов фильтрации помех [2—5] неприемлемо: корректировка предполагает пересчет всего массива данных, что приводит к искажению конфигурации границ локальных аномалий и изменению их контрастности относительно фона.

В рамках аддитивной модели интерпретация геофизических данных предполагает выделение вклада локальных аномалий на фоне региональной аномалии и помех и на этой основе определение местоположения объектов искусственного происхождения. Необходимо отметить, что объекты поиска расположены в приповерхностном слое грунта, их минимальный линейный размер в плане значительно превышает глубину залегания верхней кромки объекта и размеры измерительной установки. В этом случае форма и простираение локальных аномалий соответствуют положению объекта. Следовательно, по выявленной локальной аномалии возможна оценка геометрических параметров объектов поиска — построение *линии границы* объекта

в плане. Одним из способов решения этой задачи является применение методов классификации к массивам данных кажущегося сопротивления с последующей сегментацией карты распределения сопротивления. Реконструкция формы границы позволяет соотнести выявленные локальные аномалии с объектами поиска различных типов, что существенно уточняет интерпретацию геофизических данных.

Методы сегментации

Обычно сегментацию проводят в целях декомпозиции изображения на части, области или сегменты, более удобные для дальнейшего анализа [6, с. 362]. Выделяют несколько подходов к сегментации, отличающихся по способу анализа исходных данных.

Методы пороговой сегментации основаны на сравнении амплитуды измеренного сигнала с одной или несколькими пороговыми величинами [7, с. 850—874]. При многоуровневой сегментации одним из способов определения границ интервалов является формула Стерджесса. На участке, структура грунта которого изменена в результате антропогенного и техногенного воздействия, обычно фиксируют региональные аномалии, которые связаны с различной мощностью культурного слоя, и помехи, вызванные неоднородным поверхностным слоем (строительный и бытовой мусор). Региональная аномалия будет играть определяющую роль при классификации пороговым методом, а помехи будут проявляться в виде локальных аномалий малой площади, которые искажают границы локальных аномалий объектов поиска.

Методы наращивания областей учитывают не только амплитуду сигнала, но и взаимное положение элементов изображения. Алгоритмы основаны на оценке связности смежных сегментов изображения и последующем объединении их в однородную область [7, с. 874—881]. Одна из возможных реализаций данного подхода — метод разрастания региона. В соответствии с алгоритмом к первому элементу последовательно присоединяются "похожие" на него смежные элементы, иначе — создается новый регион. В качестве метрики "похожести" выступает критерий однородности: отличие элемента от среднего в регионе не должно превышать заданного порога. Ограничения этой группы методов при сегментации данных электропро-

филирования аналогичны описанным ограничениям для методов пороговой сегментации.

Градиентными методами осуществляют поиск границ объектов по резкому изменению амплитуды сигнала между смежными элементами изображения [7, с. 819—836]. Наиболее эффективным методом выделения границ считается алгоритм Саппу [6, с. 386—387]. Алгоритм обнаруживает границы, выполняя поиск локальных максимумов градиента функции. Двойная пороговая фильтрация обеспечивает нахождение сильных и слабых границ. Итоговые границы определяются путем подавления слабых, не связанных с сильными. С точки зрения аддитивной модели поля градиентными методами наиболее контрастно будут выделены границы помех. Границы локальных и региональных аномалий будут либо исключены из рассмотрения, либо выделены фрагментарно.

Методы на основе кластеризации предполагают разбиение элементов массива данных на кластеры на основе их "схожести" [6, с. 364—365]. В метрическом пространстве "схожесть" определяется через расстояние в пространстве признаков. При жесткой кластеризации каждый элемент однозначно приписывается одному и только одному кластеру. При нечеткой кластеризации элемент может принадлежать ко всем выделяемым кластерам с различной степенью принадлежности. Примером жесткой кластеризации является алгоритм *k-means* [8, с. 460—462], разделяющий все данные на заданное число кластеров. Основная идея алгоритма заключается в итеративном уточнении центров кластеров для минимизации среднего квадратичного отклонения значений элементов данных, входящих в кластеры, от их центров. Аналогичным образом реализуются алгоритмы нечеткой кластеризации *fuzzy c-means* (FCM) и Густафсона—Кесселя (GK) [9], позволяющие представить выделенные кластеры в виде нечетких множеств. Отличие алгоритмов FCM и GK друг от друга состоит в выделении кластеров разной формы. В FCM-алгоритме используется евклидова метрика. Это определяет возможность обнаружения кластеров строго сферической формы. Использование метрики Махаланобиса в алгоритме GK обеспечивает формирование кластеров эллипсоидальной формы. Существенным ограничением представленных алгоритмов кластеризации при анализе данных электрофилирования является необходимость априорного определе-

ния числа классов и чувствительность к помехам в массиве данных.

Байесовские методы классификации базируются на критериях принятия статистических решений [6, с. 152—156]. Примером данной группы методов служит EM-алгоритм (*expectation-maximization*), основанный на методике итеративного вычисления оценок максимального правдоподобия [8, с. 272—279]. Предполагается, что элементы в каждом классе подчиняются нормальному закону распределения. Задача классификации заключается в "подгонке" распределений выделенных классов, а затем в определении вероятностей принадлежности элемента к каждому классу. Эффективность их применения зависит от амплитуды зафиксированного сигнала — абсолютного значения измеренного кажущегося сопротивления. Следовательно, локальные аномалии, расположенные на участке региональной аномалии и вне его, будут отнесены к различным классам, что в значительной степени усложняет истолкование полученных результатов. Кроме того, существенное влияние на результаты сегментации будут оказывать случайные ошибки измерений.

Методы на основе графов позволяют представить данные в виде иерархической структуры [8, с. 520—528]. Один из подходов — построение дерева решений — предполагает последовательное объединение исходных элементов массива данных и получение заданного числа классов. Основанием для объединения нижележащих узлов дерева в один вышележащий является отношение расстояний между парами классов. Эффективность применения этих методов зависит от полноты априорной информации о предполагаемых объектах поиска или связанных с ними аномалиях. При этом практически исключается возможность использования существующего набора априорных сведений при изучении других участков грунта, отличающихся по своим физическим свойствам (плотность, влажность, пористость и пр.), что ограничивает применимость данного подхода.

Предварительный анализ различных принципов классификации показал, что каждый из рассмотренных алгоритмов имеет ограничения при сегментации массивов кажущегося сопротивления. Поэтому дальнейшая оценка их эффективности осуществлялась на основе компьютерного моделирования и натурных экспериментов.

В общем случае выявление аномалий на геофизической карте можно рассматривать как задачу классификации данных, поскольку каждый элемент массива данных (каждая точка на изображении геофизической карты) будет отнесена в один из классов: "Фон", "Объект 1", ... "Объект N". Восстановленная линия границы объекта проходит по внешнему контуру выявленной локальной аномалии. Для оценки степени соответствия восстановленной и истинной границ объекта использовали два критерия.

1. Критерий $H(I_t, I_s)$ — Hausdorff's distance [10]:

$$H(I_t, I_s) = \max(h(I_t, I_s), h(I_s, I_t)),$$

где $I_t = \{t_1, t_2, \dots, t_l\}$ и $I_s = \{s_1, s_2, \dots, s_k\}$ — множество точек на сегментированной карте, принадлежащих истинной и восстановленной границам соответственно; $h(I_t, I_s)$ — максимальное расстояние от точки из множества I_t до ближайшей к ней точки из множества I_s . Аналогичным образом определяется величина $h(I_s, I_t)$.

Критерий Hausdorff's distance характеризует максимальное абсолютное отклонение восстановленной границы от истинной.

2. Критерий ε — pixel distance error [11]:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{l},$$

где N — число ошибочно классифицированных точек на сегментированной карте, отнесенных к восстановленной границе; l — общее число точек, принадлежащих истинной границе; d_i — расстояние между i -й ошибочно классифицированной точкой, отнесенной к восстановленной границе, и ближайшей к ней, принадлежащей истинной.

Критерий pixel distance error определяет среднее отклонение восстановленной границы объекта от истинной.

Таким образом, чем меньше численные значения критериев, тем больше степень соответствия восстановленной и истинной границ (лучше качество сегментации). При $H(I_t, I_s) = 0$ и $\varepsilon = 0$ восстановленная и истинная границы совпадают.

Оценку качества сегментации при использовании перечисленных алгоритмов к данным электропрофилирования проводили на основе решения прямой задачи электроразведки для слоистой среды, содержащей псевдотрехмерные объекты конечной высоты. Компьютерное моделирование (программа Res3Dmod, Geotomo Software, Малайзия) выполнено для электропрофилирования дипольной экваториальной установкой с добавлением 5 % белого шума. Рассмотрены две принципиально отличающиеся модели, параметры которых приближены к реальной структуре культурного слоя средневековых археологических памятников Приуралья.

Модель 1 — двухслойная среда, содержащая локальный трехмерный объект поиска (рис. 2). Верхний слой мощностью 0,25 м имитирует пахотный горизонт — гумусированный суглинок с удельным сопротивлением 40 Ом·м. Нижний слой (0,25...2,50 м) имитирует материковую породу — глину с удельным сопротивлением 10 Ом·м. В этом слое (диапазон глубин 0,25...0,90 м) расположена прямая призма с квадратным основанием (удельное сопротивление 80 Ом·м). Данная трехмерная неоднородность имитирует хозяйственную яму (объект поиска). Обычное заполнение таких объектов — сильногумусированный суглинок высокого удельного сопротивления с включениями угля, золы, камней и артефактов.

Модель 2 — трехслойная среда с локальным трехмерным объектом поиска (рис. 3). Верхний

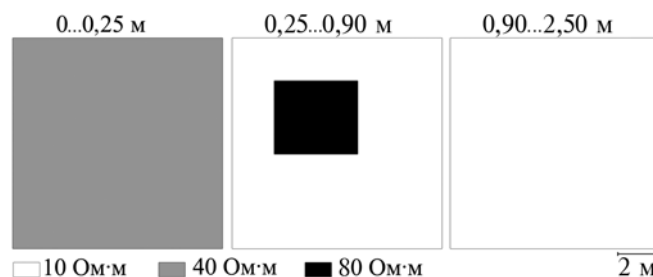


Рис. 2. Геометрические и физические параметры модели 1

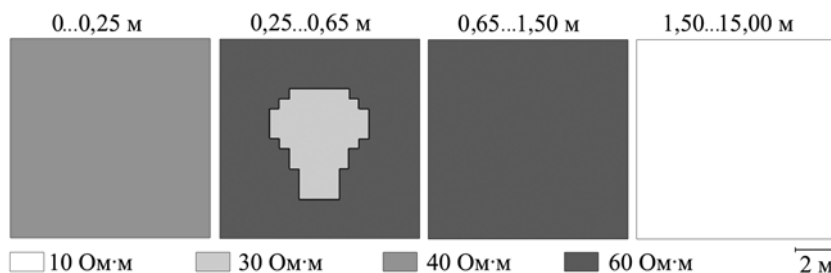


Рис. 3. Геометрические и физические параметры модели 2

слой мощностью до 0,25 м имитирует пахотный горизонт — гумусированный суглинок с удельным сопротивлением 40 Ом·м. Средний слой (0,25...1,50 м) соответствует культурному слою с удельным сопротивлением 60 Ом·м (гумусированный суглинок с включениями тонких прослоек глины, супеси и древесного тлена). В диапазоне 0,25...0,65 м введена прослойка относительно малого удельного сопротивления (объект поиска), которая имитирует фрагмент фундамента древнего сооружения. Обычно такие объекты сформированы из уплотненной или прокаленной глины с удельным сопротивлением около 30 Ом·м. Нижний слой (1,50...15,00 м) соответствует материковой породе.

Несмотря на то что рассмотренные модели отличаются по числу слоев и соотношению удельного сопротивления объекта поиска и фона, результаты сегментации демонстрируют сходные тенденции. Применение методов пороговой сегментации (рис. 4, а, 5, а) и разрастания региона (рис. 4, б, 5, б) приводит к чрезмерной сегментации внутренней области объекта. Применение EM-алгоритма (рис. 4, ж, 5, ж) позволяет локализовать объект. При этом фиксируется существенное смещение восстановленной границы объекта в область фона (табл. 1). Эта особенность выявлена при различном соотношении удельного сопротивления объекта поиска и фона (см. рис. 2, 3). В отличие от рассмотренных, алгоритмы Canny (рис. 4, в, 5, в), *k-means* (рис. 4, з, 5, з), FCM (рис. 4, д, 5, д), GK (рис. 4, е, 5, е) и дерево решений (рис. 4, з, 5, з) обеспечивают хорошее согласие формы восстановленной и истинной границ объекта поиска. Наилучшее качество сегментации для обеих моделей обеспечивают алгоритмы *k-means*, FCM, GK (табл. 1).

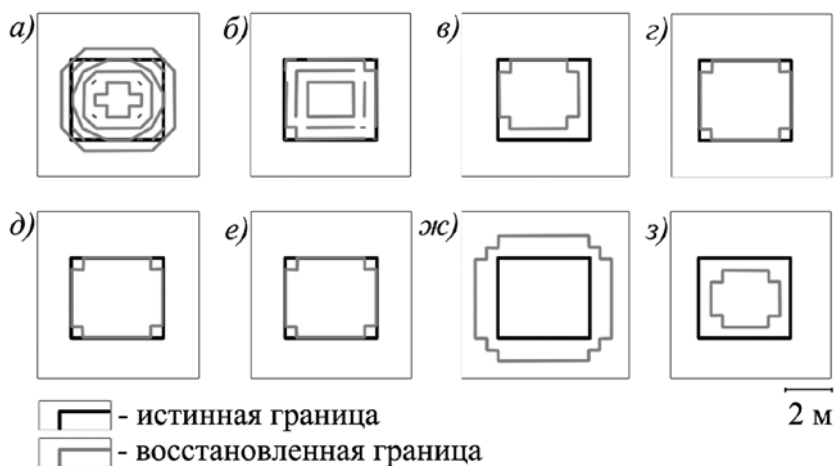


Рис. 4. Результат сегментации и построения линии границы объекта для модели 1:

а — пороговая сегментация; б — разрастание региона; в — алгоритм Canny; з — алгоритм *k-means*; д — алгоритм FCM; е — алгоритм GK; ж — алгоритм EM; з — построение дерева решений

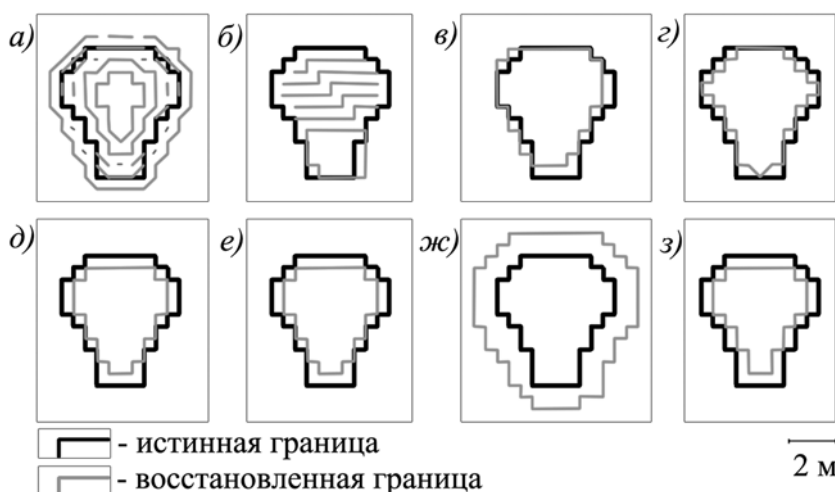


Рис. 5. Результат сегментации и построения линии границы объекта для модели 2:

а — пороговая сегментация; б — разрастание региона; в — алгоритм Canny; з — алгоритм *k-means*; д — алгоритм FCM; е — алгоритм GK; ж — алгоритм EM; з — построение дерева решений

Таблица 1

Оценка качества сегментации для моделей 1 и 2

Мо-дель	Критерий	Метод поро-говой сегмен-тации	Метод разрас-тания регио-на	Canny	k-means	FCM	GK	EM	Де-рево реше-ний
1	$H(I_p, I_s)$, м	—	—	1,12	0,50	0,50	0,50	1,00	1,12
	ε , м	—	—	0,07	0,03	0,03	0,03	0,21	0,10
2	$H(I_p, I_s)$, м	—	—	0,71	0,50	0,71	0,71	1,41	0,71
	ε , м	—	—	0,05	0,05	0,05	0,05	0,18	0,06

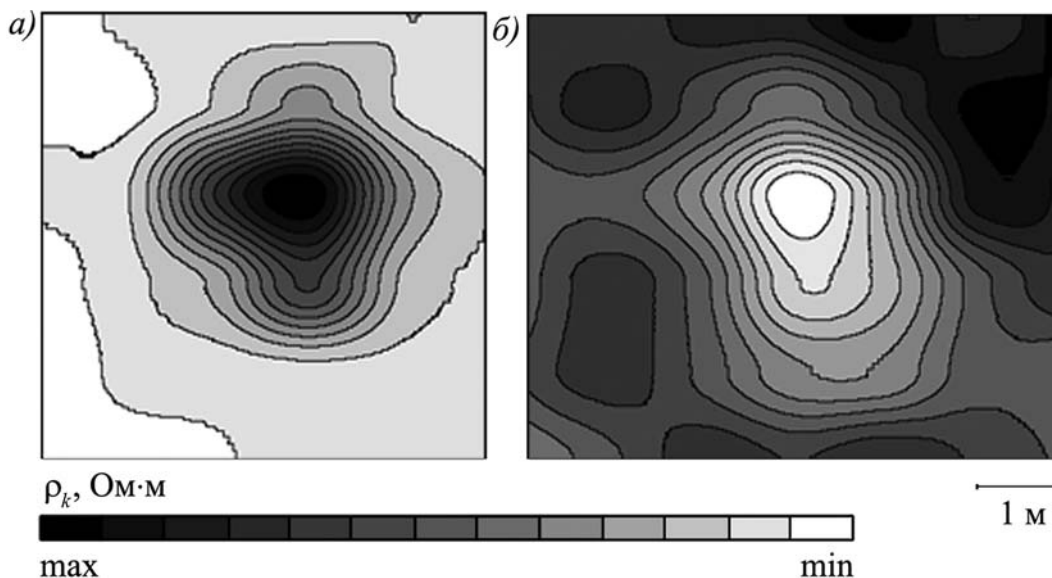


Рис. 6. Данные электропрофилеирования:
 а — аномалия 1; б — аномалия 2

Натурный эксперимент

На территории средневекового города Уч-какар IX—XIII вв. н.э. (Удмуртская Республика) проведена геофизическая съемка методом электропрофилеирования [12]. В частности, зафиксированы две локальные аномалии (рис. 6) разного сопротивления относительно вмещающего грунта.

Аномалия 1 — аномалия повышенного сопротивления (рис. 6, а) обусловлена хозяйственной ямой. По результатам археологических раскопок установлено, что яма заполнена сильно-гумусированным слоем и заглублена в материковую породу (плотная красная глина). Яма расположена в диапазоне глубин 0,20...0,95 м от поверхности. Перекрывающий слой — пахотный горизонт. Таким образом, исследуемый участок соответствует модели 1 (см. рис. 2).

Аномалия 2 — аномалия пониженного сопротивления (рис. 6, б), источником которой является фундамент средневекового сооружения. Данный объект представляет собой глиняную площадку, расположенную в гумусированном слое выше уровня материка. По своим геометрическим и физическим параметрам исследуемый участок грунта близок к модели 2 (см. рис. 3).

Методы пороговой сегментации и разрастания региона для сегментации этих данных не применяли, так как при компьютерном моделировании была показана недостаточная эффективность этих методов.

Результаты сегментации данных натурального эксперимента существенно уточняют результаты компьютерного моделирования. Алгоритмы Canny (рис. 7, а, 8, а), *k-means* (рис. 7, б, 8, б) и построения дерева решений (рис. 7, в, 8, в) фиксируют лишь фрагменты границ объектов поиска. Вероятно, это связано с отсутствием резкого контраста между объектом и вмещающим грунтом (в отличие от моделей 1 и 2 без ярко выраженных точек перегиба). Вместе с тем при использовании алгоритмов FCM (рис. 7, г, 8, г), GK (рис. 7, з, 8, з) и EM

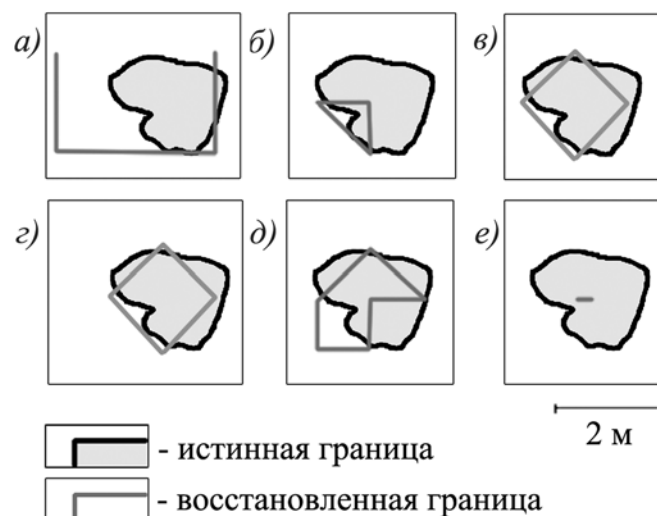


Рис. 7. Результат сегментации и построения линии границы объекта для аномалии 1:

а — алгоритм Canny; б — алгоритм *k-means*; в — алгоритм FCM; г — алгоритм GK; д — алгоритм EM; е — построение дерева решений

Оценка качества сегментации для аномалий 1 и 2

Аномалия	Критерий	Canny	<i>k-means</i>	FCM	GK	EM	Дерево решений
1	$H(I_x, I_y)$, м	—	1,41	1,00	1,00	1,00	—
	ε , м	—	0,13	0,00	0,00	0,13	—
2	$H(I_x, I_y)$, м	—	—	1,00	1,00	2,00	—
	ε , м	—	—	0,15	0,15	0,15	—

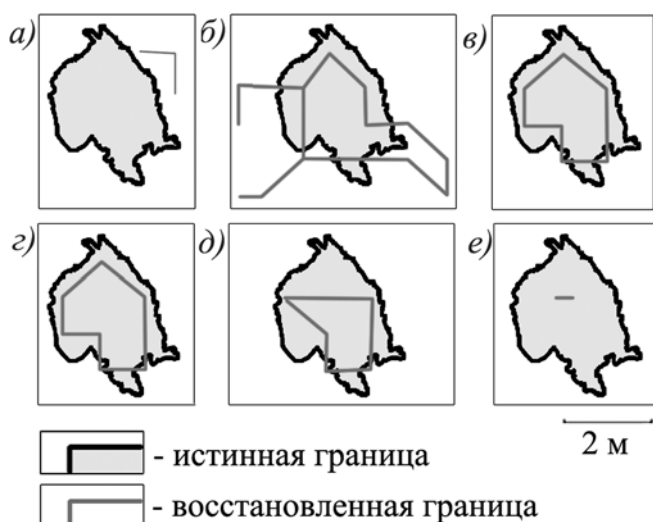


Рис. 8. Результат сегментации и построения линии границы объекта для аномалии 2:

a — алгоритм Canny; *б* — алгоритм *k-means*; *в* — алгоритм FCM; *г* — алгоритм GK; *д* — алгоритм EM; *е* — построение дерева решений

(рис. 7, *д*, 8, *д*) конфигурация восстановленной границы локальной аномалии согласуется с истинной границей объекта. При этом алгоритмы FCM и GK обеспечивают наименьшее отклонение восстановленной границы объекта от истинной (табл. 2). Поскольку результаты применения этих алгоритмов совпадают, выбор метрики при нечеткой кластеризации данных электропрофилирования не играет существенной роли. Отмеченные тенденции сохраняются при различном отношении сопротивления объекта и фона (см. рис. 6, *а* и *б*).

Заключение

Сравнительный анализ результатов классификации массивов данных, полученных при компьютерном моделировании и натурном эксперименте, показал эффективность алгоритмов FCM и GK для сегментации карт распределения

сопротивления. Алгоритмы нечеткой кластеризации обеспечивают корректное восстановление линии границы объекта искусственного происхождения в грунте в условиях влияния регионального фона и помех. Это обеспечивает решение актуальной задачи — оценки формы и геометрических параметров объектов поиска по данным геофизики.

Список литературы

1. **Электроразведка:** Справочник геофизика. В двух книгах/ Под ред. В. К. Хмелевского и В. М. Бондаренко. Книга первая. — 2-е изд., перераб. и доп. М.: Недра, 1989. 438 с.
2. **Haigh J. G. B.** Automatic grid balancing in geophysical survey // Computer Applications and Quantitative Methods in Archaeology — 1991. Oxford, 1992. P. 191—196.
3. **Paasche H., Eberle D.** Automated integration of large geophysical data sets using three partitioning cluster algorithms // 11th SAGA Biennial Technical Meeting and Exhibition Swaziland. 2009. P. 286—291.
4. **Петров А. В., Юдин Д. Б., Хоу С.** Обработка и интерпретация геофизических данных методами вероятностно-статистического подхода с использованием компьютерной технологии "КОСКАД 3D" // Вестник КРАУНЦ. Серия: Науки о Земле. 2010. № 2, вып. № 16. С. 126—132.
5. **Давыденко А. Ю.** Интерпретация геофизических данных с использованием технологии GELIOSMI // Вопросы естествознания. 2016. № 1(9). С. 120—124.
6. **Шапиро Л.** Компьютерное зрение. М.: БИНОМ. Лаборатория знаний, 2006. 752 с.
7. **Гонсалес Р., Вудс Р.** Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
8. **Hastie T., Tibshirani R., Friedman J.** The elements of statistical learning: data mining, inference, and prediction. 2nd ed. Springer-Verlag, 2009. 746 p.
9. **Bezdek J. C.** Pattern recognition with fuzzy objective function algorithms. N.Y.: Plenum Press, 1981. 272 p.
10. **Zhang Y. J.** Advances in image and video segmentation. IBM Press, 2006. 473 p.
11. **Yasnoff W., Miu J., Bacus J.** Error measures for scene segmentation // Pattern Recognition. 1977. P. 217—231.
12. **Журбин И. В., Иванова М. Г.** Геофизические исследования Кушманского городища Уччакар в Прикамье // Археология, этнография и антропология Евразии. 2018. Т. 46. № 1. С. 76—85.

A. G. Zlobina, PhD, Researcher, e-mail: ELF54@yandex.ru,
I. V. Zhurbin, D. Sc., Chief Researcher, e-mail: zhurbin@udm.ru,
Udmurt Federal Research Center of Ural Branch of the Russian Academy of Sciences,
Ijevsk, Russian Federation

Comparative Analysis of Segmentation Algorithms of Electric Profiling Data for Restoration of Boundary of Object

In this article assessment of applicability of methods of segmentation for the restoration of the boundary line for the artifacts (underground engineering structures and pipelines, industrial and domestic disposal sites, objects of historical and cultural heritage, etc.) based on geophysical data — the data of areal electric profiling — is carried out. Reconstruction of the shape and geometric parameters allows to correlate the detected local anomalies with search objects of different types. This significantly clarifies the interpretation of the results of the geophysical survey of the territories that have been subjected to human and man-made impact. Methods of classification are applied to the arrays of the apparent resistivity data with the subsequent segmentation of the resistivity distribution map.

The assessment of efficiency of methods of segmentation was carried out on the basis of computer modeling and a natural experiment (geophysical and archaeological studies at the Kushmanskoye (Uchkakar) fortified settlement dating to the 9th–13th centuries). The assessment of quality of segmentation was carried out by criteria of Hausdorff's distance and pixel distance error. Comparative analysis of the results of applying different approaches to the segmentation of soil resistivity distribution maps showed that the algorithms fuzzy c-means and Gustafson-Kessel of fuzzy clustering are the most effective from the point of view of the restoration of the boundary line of object on local anomaly.

Keywords: electric profiling, artifacts, local anomaly, segmentation, boundary line of object, fuzzy c-means algorithm, Gustafson–Kessel algorithm, Hausdorff's distance, pixel distance error, modeling, natural experiment.

DOI: 10.17587/it.24.805-812

References

1. **Elektrorazvedka: Spravochnik geofizika.** V dvuh knigah (Electrical survey: manual for geophysicist. In two books), Ed. V. K. Khmelevsky i V. M. Bondarenko. Kniga pervaya. 2-e izd., pererab. i dop. Moscow, Nedra, 1989. 438 p. (in Russian).
2. **Haigh J. G. B.** Automatic grid balancing in geophysical survey, *Computer Applications and Quantitative Methods in Archaeology — 1991*, Oxford, 1992, pp. 191–196.
3. **Paasche H., Eberle D.** Automated integration of large geophysical data sets using three partitioning cluster algorithms, *11th SAGA Biennial Technical Meeting and Exhibition Swaziland*, 2009, pp. 286–291.
4. **Petrov A. V., Judin D. B., Hou S.** Obrabotka i interpretatsiya geofizicheskikh dannykh metodami veroyatnostno-statisticheskogo podkhoda s ispol'zovaniem komp'yuternoy tekhnologii "KOSKAD 3D" (Processing and interpretation of geophysical data by methods of probabilistic and statistical approach with use of the computer technology "KOSKAD 3D"), *Vestnik KRAUNTS. Nauki o Zemle*, 2010, no. 2, iss. 16, pp. 126–132 (in Russian).
5. **Davydenko A. J.** Interpretatsiya geofizicheskikh dannykh s ispol'zovaniem tekhnologii GELIOSMI (Geophysical data interpretation based on GelioSMI technology), *Voprosy estestvoznaniya*, 2016, no. 1, pp. 120–124 (in Russian).
6. **Shapiro L.** *Kompyuternoe zrenie* (Computer Vision), Moscow, BINOM. Laboratoriya znaniy, 2006. 752 p. (in Russian).
7. **Gonsales R., Vuds R.** *Tsifrovaya obrabotka izoyurazheniy* (Digital image processing). Moscow: Tehnosfera, 2005. 1072 p. (in Russian).
8. **Hastie T., Tibshirani R., Friedman J.** *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer-Verlag, 2009. 746 p.
9. **Bezdek J. C.** *Pattern Recognition with Fuzzy Objective Function Algorithms*. N.Y., Plenum Press, 1981. 272 p.
10. **Zhang Y. J.** *Advances in image and video segmentation*. IBM Press, 2006. 473 p.
11. **Yasnoff W., Miu J., Bacus J.** Error measures for scene segmentation, *Pattern Recognition*, 1977, pp. 217–231.
12. **Zhurbin I. V., Ivanova M. G.** Geophysical Studies at the Kushmanskoye (Uchkakar) Fortified Settlement, Kama Basin, *Archaeology, Ethnology and Anthropology of Eurasia*, 2018, vol. 46, no. 1, pp. 76–85 (in Russian).

**Указатель статей,
опубликованных в журнале
"Информационные технологии"
в 2018 г.**

ОБЩИЕ ВОПРОСЫ

Юрченко А. В. О сервисном подходе к формированию и оценке востребованности киберинфраструктуры науки № 4

**ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ
И ТЕХНОЛОГИИ**

Акопов А. С., Бекларян А. Л., Хачатрян Н. К., Фомин А. В. Разработка адаптивного генетического алгоритма с использованием методов агентного моделирования № 5

Сметанин Ю. Г., Ульянов М. В., Пестова А. С. О числе возможных реконструкций слов по подсловам при окне переменного сдвига № 4

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

Агасиев Т. А., Карпенко А. П. Современные техники глобальной оптимизации. Обзор № 6

Асадов Х. Г., Джахидзаде Ш. Н., Керимова М. И. Метод дискретно-вариационной оптимизации систем дистанционного зондирования № 3

Бронштейн Е. М., Гиндуллин Р. В. Решение задачи о заборе и доставке грузов одним транспортным средством с использованием пакета CPLEX . № 6

Буравцев А. В., Цветков В. Я. Анализ структуры сложной организационно-технической системы № 1

Гермашев И. В., Дербишер В. Е., Дербишер Е. В., Маркушевская Е. А. Анализ нечетких данных для оптимизации химико-технологических систем № 3

Гурарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л. Редуцирование модели электрической схемы с учетом характеристик внешних цепей № 9

Гурарий М. М., Русаков С. Г., Ульянов С. Л., Жаров М. М. Итерационный метод решения параметризованной системы линейных алгебраических уравнений с многими правыми частями № 6

Денисова И. П., Костиков Ю. А., Лебедева М. В., Пасисниченко М. А. Математическое, алгоритмическое и программное обеспечение системы лазерной локации космических аппаратов с использованием подвижного телескопа № 9

Дубанов А. А. Задача преследования. Решение в системе вычислительной математики MathCAD. № 4

Зак Ю. А. Принципы построения систем имитационного моделирования производственных систем № 11

Иванников А. Д., Северцев В. Н. Математическая модель множества входных воздействий цифровых систем при их моделировании на уровне логических сигналов. № 10

Кобзаренко Д. Н. Технологии геоинформационного моделирования пространственного распределения геотермальных ресурсов № 4

Ковалев С. П., Шаймарданов М. Ю. Метод учета технологических ограничений при оптимизации топологии изделий машиностроения № 2

Курейчик В. В., Жиленков М. А., Курейчик Вл. Вл. Генетический алгоритм решения задачи компоновки с учетом электромагнитной совместимости. № 4

Курейчик В. М., Логунова Ю. А. Анализ перспективности генетического алгоритма при решении задачи коммивояжера № 11

Левин В. И. Интервальные уравнения в задачах исследования неопределенных систем № 3

Попков С. И. Метод внешней оптимизации для идентификации марковских процессов . . № 10

Резчиков А. Ф., Яндыбаева Н. В., Кушников В. А., Иващенко В. А., Богомолов А. С., Филимонюк Л. Ю. Информационно-советующая система для моделирования и прогнозирования динамики показателей национальной безопасности на основе причинно-следственных комплексов и уравнений Форрестера № 1

Сирота А. А., Гончаров Н. И. Моделирование конфликтного взаимодействия систем с использованием формализма гибридных автоматов . № 1

Стенников В. А., Барахтенко Е. А., Соколов Д. В. Разработка принципов построения интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики № 5

Тарасов В. Н., Бахарева Н. Ф., Липилина Л. В. Анализ немарковских сетей массового обслуживания на основе уравнений баланса потоков № 5

Ульянов М. В., Фомичев М. И. Подходы к организации поискового дерева решений в методе ветвей и границ для асимметричной задачи коммивояжера № 11

Черняев А. В., Горбачев С. И., Михаленко М. А., Метелкин Е. В. Комплексное моделирование последствий аварийных разливов нефти в малых водотоках для информационной системы их прогнозирования № 11

Чертовской В. Д. Методология математического описания и моделирования адаптивной автоматизированной системы управления производством. № 2

**СИСТЕМЫ АВТОМАТИЗИРОВАННОГО
ПРОЕКТИРОВАНИЯ**

Божко А. Н. Формализация размерных ограничений на проектные решения сборочного периода № 7

Божко А. Н. Методы структурного анализа сложных изделий в интегрированных CAD/CAM-системах № 8

- Гурарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л.** Минимаксная оптимизация в задачах схемотехнического проектирования № 7
- Романов П. С., Романова И. П.** Методика принятия решений по выбору наилучшей технической системы по набору ее характеристик . № 7
- Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Битков Ю. В.** Разработка методов автоматизации ресурсоориентированной функциональной коррекции логических схем № 12
- Хабаров В. В., Шуталева Н. С., Костин В. И.** Информационная система проектирования предприятий № 4

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Боровский А. С., Шумилина Н. А.** Метод интеллектуальной поддержки при принятии управленческих решений в проектах промышленных предприятий № 9
- Ефимов Н. О., Жиялков Е. Г.** Распознавание фрагментов изображений рукописного текста. Характерные частотные интервалы № 7
- Исаева О. С.** Технология построения комплексных моделей в инфраструктуре имитационного моделирования № 7
- Корепанов В. Д., Кулагин В. П., Халабия Р. Ф.** Использование нейронных сетей при модификации генетического алгоритма № 12
- Лиманова Н. И., Седов М. Н.** Алгоритм нечеткого поиска реквизитов физических лиц в базах данных на основе метрики Левенштейна № 8
- Лядова Л. Н., Сухов А. О., Медведева Е. Ю.** Алгоритмы синтаксического разбора для текстовых динамически настраиваемых предметно-ориентированных языков № 10
- Макаров С. Л.** Информационные технологии поиска аналогов исследования по каталогам диссертаций № 6
- Наумов А. С.** Разработка программного обеспечения для реконструкции фрагментированных изображений № 7
- Сорокин А. Б., Кушнарев А. П.** Морфологический анализатор текста для выявления полноты информации № 11
- Сорокин А. Б., Лобанов Д. А.** Концептуальное проектирование интеллектуальных систем . . . № 1
- Сухов А. О., Медведева Е. Ю.** Подход к разработке языкового инструментария для создания текстовых предметно-ориентированных языков . . № 1
- Харахинов В. А.** Генетический алгоритм как альтернатива обучения слоя Кохонена № 10
- Цветков В. Я.** Алгоритмы как средство познания № 8
- Шмелева А. Г., Ладынин А. И., Бахметьев А. В.** Построение взвешенных решений управления сложными производственными системами с применением теории массового обслуживания № 6
- Шучалова Ю. С., Ланин В. В.** Исследовательский портал для анализа и оценки стиля научных публикаций № 8

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Абдуллаев Н. Т., Гасанкулиева М. М., Джабиева А. Д.** Применение нейросетевой технологии при фликкер-шумовой спектроскопии электрокардиограммы № 6
- Горбатков С. А., Полупанов Д. В.** Оптимальный отбор и агрегирование экзогенных переменных в нейросетевых моделях банкротств на основе функций Харрингтона № 2
- Коляда А. А., Кучинский П. В., Червяков Н. И.** Редукционный метод позиционно-модулярного преобразования больших чисел для нейронных сетей на конечных кольцах № 5
- Куликов С. В., Захаров О. С., Андреев Д. Ю.** Исследование возможности совместного применения нейросетевого преобразователя биометрия—код и глубокой сверточной нейронной сети в распознавании лиц № 2
- Лютикова Л. А., Шматова Е. В.** Логический подход к коррекции результатов работы СП-нейронных сетей № 2
- Пантюхин Д. В., Карелова Е.** Повышение качества классификации компьютерных атак сверточной нейронной сетью посредством балансировки обучающей выборки № 6

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Асратян Р. Э.** Коммуникационно-ориентированная архитектура распределенных информационных систем на основе службы обработки защищенных сообщений № 6
- Бурькова Е. В.** Программная реализация выбора средств физической защиты объекта информатизации № 11
- Васильев В. И., Вульфин А. М., Гузаиров М. Б.** Оценка рисков информационной безопасности с использованием нечетких продукционных когнитивных карт № 4
- Васильев В. И., Вульфин А. М., Гузаиров М. Б., Кириллова А. Д.** Интервальное оценивание информационных рисков с помощью нечетких серых когнитивных карт № 10
- Осадчая Т. С., Щеглов А. Ю.** Комплексное решение задачи защиты от атак с правами привилегированного пользователя № 2
- Стюгин М. А., Золотарев В. В., Паротькин Н. Ю.** Подход к защите информационных систем от уязвимостей, не выявленных на стадии проектирования системы № 9
- Фаткиева Р. Р., Рыжков С. Р.** Оценка нарушения периметра информационной безопасности в облачной среде № 12
- Щеглов К. А., Щеглов А. Ю.** Моделирование угроз целевых атак № 5

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Барский А. Б., Желенков Б. В., Шаменков Н. А.** Высоконадежные управляющие компьютерные

- сети с нейросетевой адаптивной маршрутизацией на базе перспективных моделей серии "Эльбрус" № 5
- Барский А. Б., Мельник Д. И.** Сетевая технология проектирования асинхронной вычислительной системы на общем решающем поле с самоуправляемой параллельной обработкой данных. № 12
- Богатырев В. А., Паршуткина С. А.** Анализ влияния кратности резервирования многопутевых передач на вероятность их своевременного обслуживания. № 12
- Гаевой С. В., Ахмед В. М. А., Фоменков С. А.** Использование имитационного моделирования для определения показателей обслуживания кластера № 7
- Инютин С. А.** Комплексование систем счисления для многозарядных вычислительных процессов № 12
- Санталов Г. Д., Артемьев Б. В.** Беспроводные сенсорные сети с централизованной обработкой данных № 5

БАЗЫ ДАННЫХ

- Зыкин С. В., Зыкин В. С.** Коммутативные преобразования в базе данных при редактировании многотабличных запросов № 5
- Игнатъев Н. А.** Выбор собственного пространства объекта с использованием нелинейных преобразований признаков № 10
- Чернышов А. В.** Исследование свойств долговременных электронных архивных хранилищ информации на оптических дисках, организованных в структуры RAID-5 № 9

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Дам В. Н.** OFDM-модуляция в задаче автоматического распознавания вида цифровой модуляции. № 5
- Назаров Л. Е., Шишкин П. В.** Характеристики помехоустойчивых блоковых турбо-кодов на основе низкоплотностных кодов № 6

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Авдошин С. М., Песоцкая Е. Ю.** Интернет вещей: Транспорт № 2
- Александров А. Е.** Оценка достоверности результатов контроля металла на основе альтернативного алгоритма № 8
- Доронина Ю. В.** Метод структурно-функционального синтеза информационной системы с учетом нечеткой привязки функций № 9
- Злобина А. Г., Журбин И. В.** Сравнительный анализ алгоритмов сегментации данных электропрофилирования при восстановлении границы объекта № 12

- Карпова И. П.** Хранение и обработка распределенных данных в группе мобильных роботов . . . № 3
- Кобзаренко Д. Н., Камилова А. М., Газанова Н. Ш., Дадашев А. М.** Применение непрерывного вейвлет-преобразования в изучении временных рядов ветромониторинга на примере Дагестана. . . . № 3
- Кузнецова Е. В., Усадова Е. В.** Управление проектными рисками в организациях сферы ИТ-услуг. № 11
- Ле Ба Чунг, Холопов Ю. А.** Асимметричный межмодульный интерфейс. № 2
- Ле Тхань Тунг Нгуен, Кравец А. Г., Нгок Зыонг Буй.** Анализ средств и моделей взаимодействия между компонентами в системе управления корпоративной мобильностью № 1
- Савин С. И., Ворочаева Л. Ю., Мальчиков А. В., Ворочаев А. В.** Метод генерации последовательности шагов для шестиногого внутритрубного робота, перемещающегося по пространственно-изогнутым трубам, с использованием развертки трубы. № 3
- Чунг Ле Ба, Холопов Ю. А.** Информационная среда цифровых систем управления, использующая для управления периферией концепцию "вынесенной руки". № 8

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

- Родионов А. Н.** Метод и алгоритм структурного анализа достижимости узлов в сетях потоков работ. № 4

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Абрамов А. Г.** Панъевропейский научно-образовательный сетевой консорциум GÉANT: особенности инфраструктуры, ключевые проекты и сервисы. № 8
- Загидуллин Р. Р., Занин Д. С.** Определение интегральной оценки профессиональной квалификации студента вуза на различных этапах обучения. № 1
- Лебедев Г. Н., Канушкин С. В., Кузнецова Т. И., Царегородцева М. Г.** Автоматизированная компьютерная система индивидуального обучения при освоении простых и сложных навыков. . . № 11
- Печников Д. А.** Модели обеспечения оперативности процесса критериально-ориентированного тестирования № 3
- Романов А. А., Волчек Д. Г., Муромцев Д. И.** Семантические технологии в массовых открытых онлайн-курсах: экстракция, обогащение и связывание данных электронных учебных курсов для образовательной платформы Open edX. . . . № 8
- Тимофеева Т. В., Нестеренко М. А.** Параметризация кривых в AutoCAD на примере овалов Кассини № 3

Чирцов А. С., Козуненко Д. А. Использование технологий создания стереоскопических изображений при численном моделировании сложных физических систем для лекционных демонстраций и учебных исследований № 1

КОМПЬЮТЕРНАЯ ГРАФИКА И ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Филинских А. Д., Корсаков К. С. Интерактивная трехмерная модель Нижегородского Кремля № 9

КОДИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ

Гриняк В. М., Девятисильный А. С., Шурыгин А. В. Оценка возможностей использования bluetooth-устройств для навигации внутри помещений № 9

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

Май В. П. Система объемной визуализации синоптических объектов № 10

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИМЕДИЦИНСКИХ СИСТЕМАХ

Новик В. П., Матвеев И. А., Леонов В. Ю. Сравнение информативных признаков изображений радужки глаза методом оптимального пути № 10

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Мыльников Л. А., Селедкова А. Б. Способ выбора метода прогнозирования и горизонта планирования параметров с использованием оценки риска № 2

Разумовский А. И. Информационная избыточность в разработке программных систем № 1

WEB-ТЕХНОЛОГИИ

Антоненко В. А., Смелянский Р. Л., Ермилов А. А., Романов А. Р., Пинаева Н. М., Плакунов А. В. Платформа управления виртуальными сетевыми функциями С2 № 3

Лобов И. В., Готман В. Г. Использование контейнера Ogg для организации потоковой трансляции в реальном времени над протоколом HTTP методом опережающей загрузки. № 2

Мальцева С. В., Думский Д. В., Комаров М. М. Исследование рынка новых доменных имен. . . № 3

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова.*

Корректор *Е. В. Комиссарова.*

Сдано в набор 10.10.2018. Подписано в печать 23.11.2018. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT1218. 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.
