

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ



НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ

САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО
ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Информатика. Телекоммуникации.
Управление

3(198) 2014

Издательство Политехнического университета
Санкт-Петербург
2014

НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ САНКТ-ПЕТЕРБУРГСКОГО
ГОСУДАРСТВЕННОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА
ИНФОРМАТИКА. ТЕЛЕКОММУНИКАЦИИ. УПРАВЛЕНИЕ

РЕДАКЦИОННЫЙ СОВЕТ ЖУРНАЛА

Председатель

Юсупов Р.М., чл.-кор. РАН;

Редакционный совет:

Абрамов С.М., чл.-кор. РАН;

Арсеньев Д.Г., д-р техн. наук, профессор;

Воеводин В.В., чл.-кор. РАН;

Заборовский В.С., д-р техн. наук, профессор;

Козлов В.Н., д-р техн. наук, профессор;

Фотиади А.Э., д-р физ.-мат. наук, профессор;

Черноруцкий И.Г., д-р техн. наук, профессор.

РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА

Главный редактор

Коротков А.С., д-р техн. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия;

Редакционная коллегия:

Бабкин А.В., д-р экон. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия;

Ицыксон В.М., канд. техн. наук, доцент, Санкт-Петербургский государственный политехнический университет, Россия;

Prof. Dr. Philippe Ferrari, Head of the RF and Millimeter-Wave Lab IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble Alpes University, France;

Карпов Ю.Г., д-р техн. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия;

Клавдиев В.Е., канд. техн. наук, доцент, Санкт-Петербургский государственный политехнический университет, Россия;

Prof. Dr. Wolfgang Krautschneider, Head of Nanoelectronics Institute, Hamburg University of Technology, Germany;

Кучерявый Е.А., канд. техн. наук, профессор, Tampere University of Technology, Finland.

Dr. Fa-Long Luo, Chief Scientist, Element CXI, San Jose, USA;

Макаров С.Б., д-р техн. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия;

Prof. Dr. Emil Novakov, IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble, France;

Устинов С.М., д-р техн. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия;

Цикин И.А., д-р техн. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия;

Шкодырев В.П., д-р техн. наук, профессор, Санкт-Петербургский государственный политехнический университет, Россия.

Журнал с 1995 года издается под научно-методическим руководством Российской академии наук. С 2008 года выпускается в составе сериального периодического издания «Научно-технические ведомости СПбГПУ» ISSN 1994-2354.

Журнал с 2002 года входит в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Сведения о публикациях представлены в Реферативном журнале ВИНТИ РАН, в международной справочной системе «Ulrich`s Periodical Directory».

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ПИ № ФС77-51457 от 19.10.2012 г.

Подписной индекс **47517** в объединенном каталоге «Пресса России».

Журнал включен в базу данных «Российский индекс научного цитирования» (РИНЦ), размещенную на платформе Научной электронной библиотеки на сайте <http://www.elibrary.ru>

При перепечатке материалов ссылка на журнал обязательна.

Точка зрения редакции может не совпадать с мнением авторов статей.

Адрес редакции и издательства: Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29.
Тел. редакции (812) 552-62-16.

© Санкт-Петербургский государственный политехнический университет, 2014

THE MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION



ST. PETERSBURG STATE
POLYTECHNICAL UNIVERSITY
JOURNAL

Computer Science.
Telecommunications and Control Systems

3(198) 2014

Polytechnical University Publishing House
Saint Petersburg
2014

ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY JOURNAL COMPUTER SCIENCE. TELECOMMUNICATIONS AND CONTROL SYSTEMS

EDITORIAL COUNCIL

Head of the editorial council

Prof. Dr. *Rafael M. Yusupov* (corresponding member of the Russian Academy of Sciences)

Members:

Prof. Dr. *Sergey M. Abramov* (corresponding member of the Russian Academy of Sciences),

Prof. Dr. *Dmitry G. Arseniev*,

Prof. Dr. *Vladimir V. Voevodin* (corresponding member of the Russian Academy of Sciences),

Prof. Dr. *Vladimir S. Zaborovsky*,

Prof. Dr. *Vladimir N. Kozlov*,

Prof. Dr. *Alexandr E. Fotiadi*,

Prof. Dr. *Igor G. Chernorutsky*.

EDITORIAL BOARD

Editor-in-chief

Prof. Dr. *Alexander S. Korotkov*, St. Petersburg State Polytechnical University, Russia;

Members:

Prof. Dr. *Alexandr V. Babkin*, St. Petersburg State Polytechnical University, Russia;

Assoc. Prof. Dr. *Vladimir M. Itsykson*, St. Petersburg State Polytechnical University, Russia;

Prof. Dr. *Philippe Ferrari*, Head of the RF and Millimeter-Wave Lab IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble Alpes University, France;

Prof. Dr. *Yuri G. Karpov*, St. Petersburg State Polytechnical University, Russia;

Assoc. Prof. Dr. *Vladimir E. Klavdiev*, St. Petersburg State Polytechnical University, Russia;

Prof. Dr. *Yevgeni Koucheryavy*, Tampere University of Technology, Finland.

Prof. Dr. *Wolfgang Krautschneider*, Head of Nanoelectronics Institute, Hamburg University of Technology, Germany;

Dr. *Fa-Long Luo*, Chief Scientist, Element CXI, San Jose, USA;

Prof. Dr. *Sergey B. Makarov*, St. Petersburg State Polytechnical University, Russia;

Prof. Dr. *Emil Novakov*, IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble, France;

Prof. Dr. *Viacheslav P. Shkodyrev*, St. Petersburg State Polytechnical University, Russia;

Prof. Dr. *Igor A. Tsikin*, Professor, St. Petersburg State Polytechnical University, Russia;

Prof. Dr. *Sergey M. Ustinov*, St. Petersburg State Polytechnical University, Russia.

The journal is published under scientific and methodical guidance of the Russian Academy of Sciences since 1995. The journal is published since 2008 as part of the periodical edition «Nauchno-tekhnicheskie vedomosti SPbGPU» (ISSN 1994-2354).

The journal is included in the List of Leading Peer-Reviewed Scientific Journals and other editions to publish major findings of PhD theses for the research degrees of Doctor of Sciences and Candidate of Sciences.

The publications are presented in the VINITI RAS Abstract Journal and Ulrich's Periodical Directory International Database.

The journal is registered with the Federal Service for Supervision in the Sphere of Telecom, Information Technologies and Mass Communications (ROSKOMNADZOR). Certificate ПИ № ФС77-51457 issued Oct. 19, 2012.

Subscription index **47517** in the «Press of Russia» Joint Catalogue.

The journal is on the Russian Science Citation Index (RSCI) database

© Scientific Electronic Library (<http://elibrary.ru/>).

No part of this publication may be reproduced without clear reference to the source.

The views of the authors can contradict the views of the Editorial Board.

The address: 195251 Polytekhnicheskaya Str. 29, St. Petersburg, Russia.

© St. Petersburg State Polytechnical University, 2014

Содержание

Инфокоммуникационные технологии

Котенко И.В., Саенко И.Б., Юсупов Р.М. <i>Новое поколение систем мониторинга и управления инцидентами безопасности</i>	7
---	---

Системный анализ и управление

Бородин А.А. <i>Исследование нагрузочных способностей компьютерных систем</i>	19
Черный С.Г., Жиленков А.А. <i>Идентификация внешних параметров сигналов для экспертных под-систем в составе устройств судовых электроэнергетических систем</i>	28

Вычислительные машины и программное обеспечение

Демидова Л.А., Нестеров Н.И., Тишкин Р.В. <i>Возможностно-нечеткая сегментация изображений земной поверхности с применением генетических алгоритмов и искусственных нейронных сетей</i>	37
--	----

Математическое моделирование: методы, алгоритмы, технологии

Ботнев В.А., Устинов С.М. <i>Методы решения прямой и обратной геодезических задач с высокой точностью</i>	49
--	----

Приборы, информационно-измерительные системы

Четвериков С.Ф. <i>Информационно-измерительная система для изучения статистических свойств и суточно-сезонных вариаций интенсивности естественного ОНЧ радиошума</i>	59
---	----

Высокопроизводительные вычисления

Червяков Н.И., Бабенко М.Г., Ляхов П.А., Лавриненко И.Н., Лягин А.М. <i>Умножение и деление в системе остаточных классов с использованием полей Галуа GF(p)</i>	65
--	----

Конференция «Технологии Microsoft в теории и практике программирования»

Тютин Б.В., Веселов А.О., Котляров В.П. <i>Тестирование на основе ключевых слов с использованием диаграмм последовательности событий</i>	78
Дробинцев П.Д., Никифоров И.В., Воинов Н.В., Котляров В.П. <i>Подход к тестированию параллельных систем на основе UCM-спецификаций</i>	85
Осенняя Д.И., Закирова И.И., Дробинцев П.Д. <i>Разработка программного продукта для подбора конфигураций информационных центров</i>	92

Contents

Infocommunication Technologies

- Kotenko I.V., Saenko I.B., Yusupov R.M.** *New generation of security information and event management systems.* 7

System Analysis and Control

- Borodin A.A.** *Research of computer systems load capability* 19
- Cherny S.G., Zhilenkov A.A.** *Identification of external parameters for the expert subsystems in the composition of the shipboard of electric power systems* 28

Computer Systems and Software

- Demidova L.A., Nesterov N.I., Tishkin R.V.** *Possibilistic-fuzzy segmentation of earth surface images by means of genetic algorithms and artificial neural networks* 37

Mathematical Modelling: Methods, Algorithms, Technologies

- Botnev V.A., Ustinov S.M.** *Methods for direct and inverse geodesic problems solving with high precision*..... 49

Devices, Information-Measuring Systems

- Chetverikov S.F.** *Using the information and measurement system to study statistical characteristics and daily and seasonal variation in the intensity of VLF atmospheric radio noise*..... 59

High-performance Computing

- Chervyakov N.I., Babenko M.Gr., Lyakhov P.A., Lavrinenko I.N., Lyagin A.M.** *Multiplication and division in the residue number system using Galois fields $GF(p)$* 65

Conference «Technologies Microsoft in the Theory and Programming Practice»

- Tiutin B.V., Veselov A.O., Kotlyarov V.P.** *Sequence diagrams application for keyword-driven testing*... 78
- Drobintsev P.D., Nikiforov I.V., Voinov N.V., Kotlyarov V.P.** *Approach to concurrent systems testing based on UCM specification*..... 85
- Osennya D.I., Zakirova I.I., Drobintsev P.D.** *Developing software for data center configuration* 92

УДК 004.056.5

И.В. Котенко, И.Б. Саенко, Р.М. Юсупов

**НОВОЕ ПОКОЛЕНИЕ СИСТЕМ МОНИТОРИНГА
И УПРАВЛЕНИЯ ИНЦИДЕНТАМИ БЕЗОПАСНОСТИ**

I.V. Kotenko, I.B. Saenko, R.M. Yusupov

**NEW GENERATION OF SECURITY INFORMATION
AND EVENT MANAGEMENT SYSTEMS**

Обоснована технологическая необходимость разработки нового поколения систем мониторинга и управления инцидентами безопасности, основанных на технологии управления информацией и событиями безопасности. Приведены типовая архитектура и основные решения по построению отдельных модулей таких систем, осуществляющих устойчивый сбор данных о событиях безопасности, их универсальную трансляцию, масштабируемую обработку, гибридное онтологическое хранение и многофункциональную визуализацию, а также межуровневую корреляцию событий, моделирование атак и прогностический анализ безопасности. Сформулированы предложения по применению таких систем в предметных областях, касающихся обеспечения безопасности в критических инфраструктурах.

МОНИТОРИНГ И УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ; КОМПЬЮТЕРНАЯ СЕТЬ; СОБЫТИЕ БЕЗОПАСНОСТИ; ИНФОРМАЦИОННАЯ ИНФРАСТРУКТУРА.

The given paper justifies the technological necessity to develop a new generation of security monitoring and event management systems based on security information and event management technology. We have focused on the typical architecture and key solutions to design the individual modules of such systems collecting constant security data, their universal translation, scalable processing, hybrid ontological storage and rich visualization, as well as a cross-level correlation of events, attack modelling and predictive security analysis. We have also stated some proposals to use such systems in the domains related to security protection in critical infrastructures.

SECURITY MONITORING AND MANAGEMENT; COMPUTER NETWORK; SECURITY EVENT; INFORMATION INFRASTRUCTURE.

Технология управления событиями и информацией безопасности (Security Information and Events Management – SIEM) является новым, бурно развивающимся направлением в области информационной безопасности, обладающим достаточно большим потенциалом по обнаружению угроз и выработке контрмер по обеспечению требуемого уровня безопасности информационной инфраструктуры.

Функционирование SIEM-систем, или, как принято говорить в русскоязычной литературе, *систем мониторинга и управления инцидентами безопасности*, заключается в оперативном сборе, хранении и аналитической обработке данных о событиях безопасности, которые первоначально формируются и фиксируются в системных журналах различных аппаратных и программных элементов, образующих информационные ин-

фраструктуры: серверы, рабочие станции, маршрутизаторы, межсетевые экраны, системы управления базами данных, системы обнаружения атак, антивирусные средства и т. д. [1, 2]

SIEM-системы в настоящее время имеют большое количество коммерческих реализаций, выполненных ведущими разработчиками и интеграторами средств и систем защиты информации: компаниями IBM (система QRadar) [3], HP (ArcSight) [4], Symantec (Symantec Security Information Manager) [5], Novell (Novell Sentinel) [6] и др. Однако область применения данных систем, которые можно назвать *SIEM-системами первого поколения*, не выходит за рамки информационных процессов, протекающих в компьютерной сети.

В то же время при мониторинге безопасности становится все более актуальной задача выявления атак и прочих злонамеренных воздействий не только на основе анализа событий безопасности, зафиксированных в журналах сетевых инфраструктурных элементов, но и на уровне бизнес-процессов, а также на уровне физических датчиков. Кроме того, известные коммерческие SIEM-системы испытывают значительные затруднения при обеспечении безопасности компьютерных сетей большой размерности.

Эти и ряд других недостатков существующих коммерческих SIEM-решений обусловили необходимость исполнения международного проекта MASSIF [7], предназначенного для построения SIEM-систем, свободных от этих недостатков и определяемых как системы *нового поколения*.

Цель настоящей работы — обобщение основных результатов, полученных в рамках проекта MASSIF, связанных с построением нового поколения систем мониторинга и управления инцидентами безопасности.

Требования к SIEM-системам нового поколения

Появление нового поколения SIEM-систем связывается с рядом вполне конкретных предпосылок, которые определяются, с одной стороны, последними достижениями в области информационно-

коммуникационных технологий (ИКТ), а с другой — новыми требованиями, предъявляемыми к современным информационным технологиям. Рассмотрим их подробнее.

Современные тенденции развития ИКТ тесно связываются с поддержкой распределенных информационных инфраструктур, ориентированных на широкое использование Интернета. К числу таких тенденций можно отнести, например, использование усовершенствованных протоколов передачи защищенных данных, сервис-ориентированных архитектур (Service-Oriented Architecture — SOA) [8], архитектур распределенных приложений, основанных на «передаче репрезентативного состояния» (Representational State Transfer), новых технологий взаимодействия с различными типами устройств.

Кроме того, специфическими технологиями, в настоящее время притягивающими к себе внимание разработчиков ИКТ, являются:

- применение методов виртуализации (Virtualization) для оптимизации информационной инфраструктуры;
- «облачные вычисления» (Cloud Computing) как основа для различных моделей обмена данными;
- контекстно-зависимые приложения (Context-aware Applications);
- семантические и контекстуальные глобальные сети (Semantic and Contextual Web);
- «зеленые» информационные технологии (Green IT);
- обработка сложных событий (Complex Event Processing);
- «социальные вычисления» (Social Computing), в т. ч. концепции Web 2.0 и Enterprise 2.0;
- усовершенствованные графические интерфейсы для человеко-машинного взаимодействия и др.

На вершине современной пирамиды ИКТ находятся парадигмы «Интернета будущего» (Future Internet) и «Интернета вещей» (Internet of Things), которые становятся реальностью по мере разработки упомянутых выше технологий.



Реализация этих парадигм приведет к значительному возрастанию информационных потоков, а также типов и количества сенсоров и устройств, распространяющих данные о событиях безопасности через Интернет. Интернет станет более интеллектуальным, т. к. он будет обрабатывать намного большее количество разнородной информации, необходимой для построения обоснованных решений, поддержки моделирования, прогнозирования, выработки рекомендаций, что в конечном итоге значительно повысит эффективность процесса принятия решений в целом.

Системы мониторинга и управления инцидентами безопасности в ближайшем будущем будут увеличивать свою значимость, поскольку они наиболее близки к этим событийно-ориентированным парадигмам и направлены на решение новых проблем безопасности. Однако в этом случае мониторинг и управление инцидентами безопасности становятся более сложными, т. к. они должны затрагивать распределенные бизнес-процессы в условиях недоступности и/или возможного несанкционированного изменения передаваемых и обрабатываемых данных.

Хотя традиционные SIEM-системы (т. е. системы первого поколения) уже нашли применение в корпоративных инфраструктурах, сценарии их будущего развертывания должны учитывать возможность того, что связи между управляемыми устройствами и поставщиками услуг будут проходить по общедоступным путям в Интернете. Кроме того, SIEM-системы нового поколения могут быть развернуты как «облачные» службы. В этом случае актуальным становится обеспечение доверия и конфиденциальности.

Таким образом, SIEM-системы нового поколения должны быть ориентированы на решение следующих проблем [1, 2, 7, 9]:

- разработку надежных и устойчивых средств обеспечения осведомленности пользователей о безопасности инфраструктуры;
- совершенствование механизмов распределенного управления безопасностью для адаптивного конфигурирования политик безопасности;

- улучшенную масштабируемость, обеспечивающую требуемое увеличение производительности при увеличении количества обрабатываемых данных;

- использование инновационных моделей прогнозирования безопасности, обеспечивающих проактивную обработку инцидентов и событий безопасности;

- децентрализацию сбора и обработки событий безопасности между центральными механизмами и удаленными коллекторами.

Тогда в качестве основных требований, предъявляемых к SIEM-системам нового поколения, следует указать возможность реализации следующего перечня новых функциональных возможностей:

- межуровневой корреляции событий безопасности, поступающих из неоднородных источников;

- адаптивной и высокомасштабируемой обработки событий, обеспечивающей управление большими объемами данных о безопасности в реальном или близком к реальному времени;

- прогностического анализа безопасности, позволяющего осуществлять проактивное обнаружение и предотвращение атак путем принятия соответствующих контрмер;

- высокой доступности и отказоустойчивости сбора данных о событиях безопасности в условиях территориально-распределенного построения информационной инфраструктуры и активного вредоносного и/или непреднамеренного воздействия на элементы инфраструктуры.

Обобщенная архитектура SIEM-системы нового поколения

Обобщенная архитектура SIEM-системы нового поколения представлена на рис. 1. В ней можно выделить четыре уровня: сети, данных, событий и приложений. Уровень сети является внешним и охватывает сетевые элементы, являющиеся источниками данных о событиях безопасности. На уровне данных происходит предварительная обработка поступивших данных, выделение из них событий безопасности и преобразование их в единый внутренний формат.

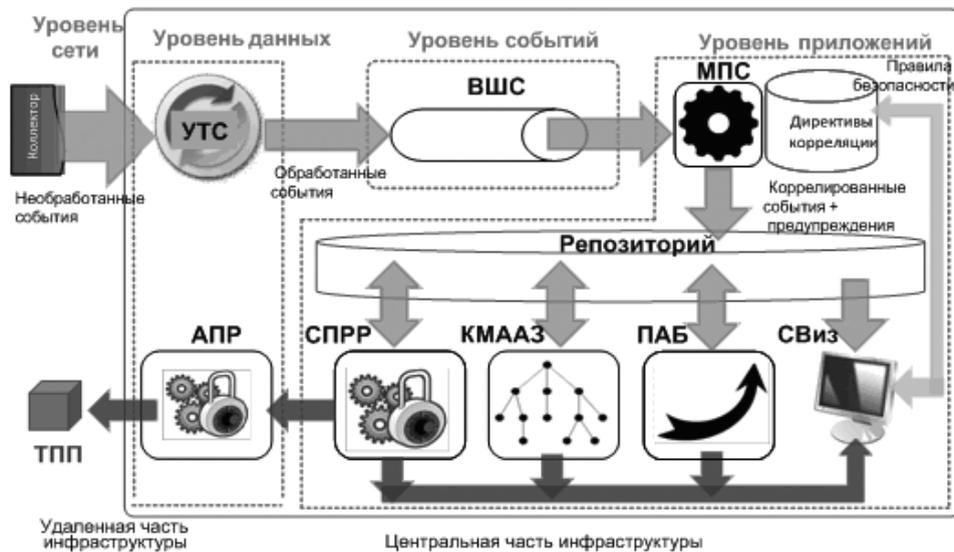


Рис. 1. Архитектура SIEM-системы нового поколения

На уровне событий осуществляется обмен событиями между всеми компонентами системы. На уровне приложений производится хранение и аналитическая обработка событий.

Основными компонентами SIEM-системы нового поколения являются:

коллектор, под которым понимаются источники данных о событиях безопасности, такие как сетевые устройства, серверы, рабочие станции, базы данных, межсетевые экраны, антивирусы, сенсоры и т. д. (уровень сети);

универсальный транслятор событий (УТС), предназначенный для первичной обработки данных о событиях безопасности, поступающих в SIEM-систему (уровень данных);

высоконадежная шина событий (ВШС), предназначенная для распространения данных о событиях безопасности и их гарантированной доставки требуемым компонентам SIEM-системы (уровень событий);

масштабируемый процессор событий (МПС), обеспечивающий адаптивную поддержку всех задач по обработке событий и их решение в реальном времени (уровень приложений);

репозиторий (хранилище) данных о безопасности (уровень приложений);

система принятия решений и реагирования

(СПРР), предназначенная для верификации и управления политиками безопасности, обеспечивающими защиту инфраструктурных элементов (уровень приложений);

компонент моделирования атак и анализа защищенности (КМАЗ), обеспечивающий дополнительные аналитические возможности SIEM-системы за счет реализации функций моделирования атак и анализа защищенности (уровень приложений);

прогностический анализатор безопасности (ПАБ), обеспечивающий расширенные возможности мониторинга безопасности, заключающиеся в моделировании состояния элементов информационной инфраструктуры в ближайшей перспективе и предсказании возможных нарушений безопасности (уровень приложений);

система визуализации (СВиз), предназначенная для представления информации о безопасности в графическом виде, обеспечивающем ее наибольшую степень восприятия и визуального анализа (уровень приложений);

агент принятия решений (АПР), обеспечивающий доведение решений по безопасности до инфраструктурных элементов (уровень данных);

точки применения политик безопасности (ТПП), под которыми понимаются логические сущности или оборудование, реа-

лизирующие по запросам пользователей команды управления доступом и решения по безопасности.

Характеристика компонентов

Приведем характеристику перечисленных компонентов, подробнее останавливаясь на решениях, полученных учеными Санкт-Петербургского института информатики и автоматизации РАН (СПИИ РАН).

Компонент УТС является программной системой, осуществляющей сбор событий и информации безопасности. Он обеспечивает управление неоднородными данными и их конфиденциальность за счет реализации процедур межуровневого сбора данных, расширенной обработки форматов, многоуровневой корреляции, агрегации, шифрования полей событий и анонимизации. Кроме того, он генерирует форматы выходных событий, пригодные для некоторых известных коммерческих SIEM-систем, например, систем OSSIM [10] и Prelude [11], основанных на открытом исходном коде.

Компонент ВШС, по сути, является телекоммуникационной подсистемой, пригодной для распределенных приложений, в которых обмен данными должен обладать высокой устойчивостью. Этот компонент в реальном масштабе времени осуществляет поиск оптимальных путей, учитывая, с одной стороны, избыточную доступность физической сети, а с другой – воздействие компьютерных атак и других негативных факторов. Для восстановления пропущенных пакетов применяются процедуры повышения достоверности, позволяющие минимизировать повторную передачу пакетов и тем самым максимально снизить временные задержки.

Компонент МПС обеспечивает адаптивную вычислительную поддержку в реальном масштабе времени всех задач обработки событий. Он способен обрабатывать несколько сотен тысяч событий в секунду, не требуя каких-либо корректировок правил или управления событиями. Благодаря постоянному хранению выбранных событий в памяти, становится возможным проведение анализа событий. Обработка событий может выполняться в распреде-

ленной среде, в которой высокая масштабируемость МПС обеспечивает достаточно высокую эффективность фильтрации, преобразования, агрегации, абстрагирования и корреляции событий. Вычислительная адаптивность МПС означает, что он может управлять входной нагрузкой. В случае резкого возрастания нагрузки компонент автоматически инициирует выполнение задач на новых узлах, что позволяет убрать пиковые нагрузки и равномерно распределить задания. Аналогично, в случае недоиспользования ресурсов происходит завершение функционирования всех ненужных узлов.

Компонент ПАБ использует в качестве входных данных различные модели обработки событий, политики безопасности, требования безопасности и события, поступающие в систему в реальном масштабе времени. Целью его функционирования является оказание помощи в принятии важнейших решений, касающихся выработки контрмер по противодействию атакам и угрозам безопасности, которые воздействуют на компьютерную сеть в текущий момент времени.

Компонент СПРР ориентирован на модель доступа организационного типа OgBAC [12]. Он осуществляет конфигурирование политик безопасности внешних систем, вызываемых соответствующими средствами. При этом не требуется знания правил конфигурирования других средств, достаточно правил конфигурирования самого компонента.

Репозиторий данных является средством кроссплатформенной интеграции различных компонентов SIEM-системы [13–15]. В качестве основы для его реализации положена архитектура SOA (рис. 2). Как видно из рисунка, архитектура репозитория разделяется на два уровня: уровень хранения и уровень веб-сервисов.

Уровень хранения включает в себя реляционную базу данных (РБД), базу XML-данных и базу данных в формате RDF, называемых «триплетами», т. к. они отражают отношения «субъект – предикат – объект». Тем самым обеспечивается гибридный подход к хранению данных о событиях безопасности, сочетающий достоинства всех

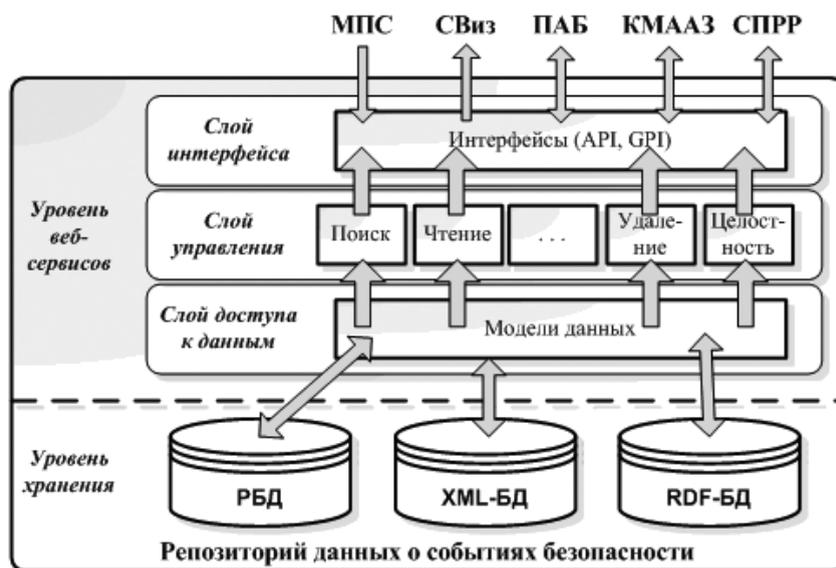


Рис. 2. Архитектура репозитория данных

базовых моделей представления данных и обеспечивающий, с одной стороны, задание моделей предметной области в виде онтологий, а с другой – использование логического вывода для выработки решений.

Уровень реализации веб-сервисов делится на три основных слоя: доступа к данным, управления и интерфейса. Слой доступа обеспечивает обращение к базам данных с помощью моделей данных. Слой управления реализует операции над данными. Слой интерфейса реализует различные виды взаимодействия с компонентами SIEM-системы.

Компонент КМАЗ способен генерировать графы атак, вычислять метрики защищенности, оценивать защищенность сети посредством анализа графов атак, генерировать отчеты с рекомендациями по повышению безопасности и анализировать события безопасности для обнаружения атакующих действий, что позволяет распознавать модели поведения возможного злоумышленника и его последующие шаги [16–22]. Входными данными для КМАЗ являются: конфигурация компьютерной сети (системы); политики безопасности, определяемые множеством полномочий или правил доступа; формируемые предупреждения; внешние базы данных уязвимостей, атак, платформ и т. д.;

профили нарушителей; требуемые значения метрик безопасности. Основными результатами работы компонента КМАЗ являются: обнаруженные уязвимости; возможные маршруты (графы) атак и целей атак; зависимости между сервисами; «узкие места» в безопасности компьютерной сети; скорректированные деревья атак, основанные на изменениях, произошедших в сети; предсказания дальнейших шагов нарушителя; метрики безопасности, которые могут использоваться для оценки общего уровня безопасности; последствия атак и контрмер; предложения по увеличению уровня безопасности; решения, основанные на мерах, политиках и средствах безопасности.

Архитектура компонента КМАЗ показана на рис. 3 [23].

Загрузчик репозитория, обращаясь через Интернет к внешним базам данных, загружает информацию об уязвимостях, атаках, конфигурации, «узких местах», платформах и контрмерах. Генератор спецификаций преобразует информацию о сетевых событиях, конфигурации и политиках безопасности, полученную от других компонентов или от пользователя, во внутреннее представление. Модуль моделирования нарушителя определяет индивидуальные характеристики нарушителей, их начальное местоположение, пол-

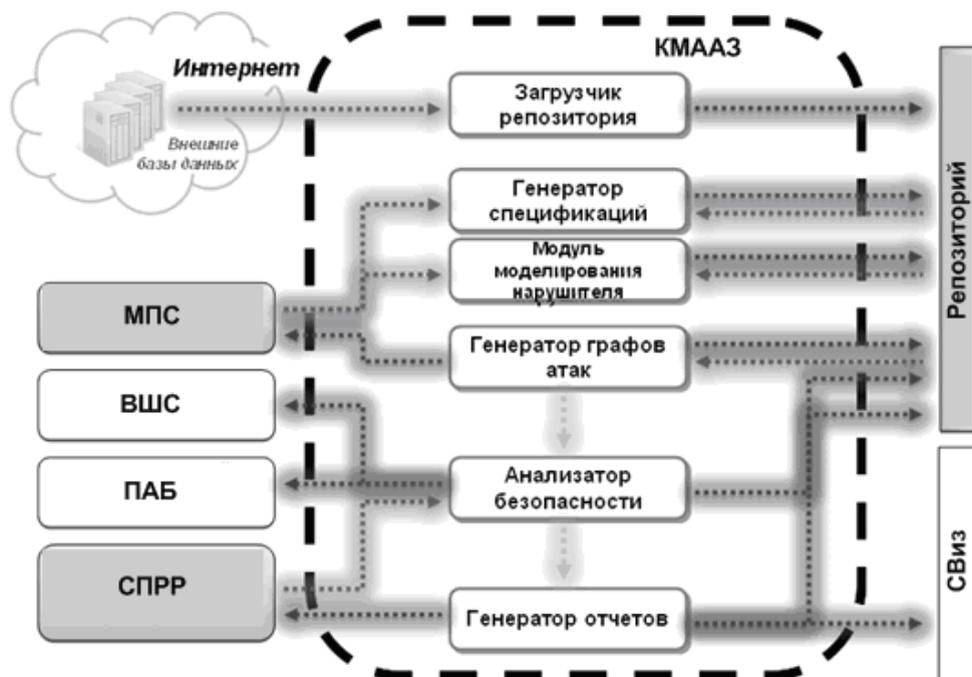


Рис. 3. Архитектура компонента КМАЗ

номочия, уже осуществленные и возможные действия, которые могут быть предсказаны, и знания об анализируемой сети. *Генератор графов атак* строит графы (деревья) атак путем моделирования последовательностей атакующих действий нарушителя в анализируемой компьютерной сети, используя информацию о различных типах возможных атак, зависимостях сервисов, конфигурации сети и использованных политиках безопасности. *Анализатор безопасности* имитирует многошаговые атаки, вычисляет метрики безопасности и оценивает эффективность контрмер. *Генератор отчетов* выдает информацию об обнаруженных уязвимостях и «узких местах», рекомендации по повышению уровня защищенности и другую релевантную информацию.

Компонент СВиз предназначен для визуального анализа информации о безопасности [24, 25]. Архитектура этого компонента включает три слоя (рис. 4): интерфейс пользователя; управляющие сервисы; графические элементы. *Интерфейс пользователя* поддерживает различные виды графических интерфейсов, начиная от простой командной строки и заканчивая сложным

многооконным интерфейсом с панелями управления. *Слой управляющих сервисов* рассматривается как модуль управления визуализацией. Исходя из выполняемых функций, в данном слое можно выделить контроллер графических элементов и менеджер сервисов. *Контроллер графических элементов* предоставляет стандартный интерфейс по работе с потоками визуализации, обеспечивающий создание и остановку графического потока, реализуемого на уровне графических элементов. *Менеджер сервисов* обеспечивает подключение сервисов мониторинга и управления безопасностью. *Слой графических элементов* включает библиотеку необходимых графических примитивов: графов, лепестковых диаграмм, гистограмм, карт деревьев, географических карт и т. д. *Графические элементы* реализуют обработку входных данных, их отображение и взаимодействие пользователя с входными данными.

Рассмотренные особенности построения и функционирования компонентов SIEM-системы нового поколения позволяют выработать предложения по ее применению в различных инфраструктурах.

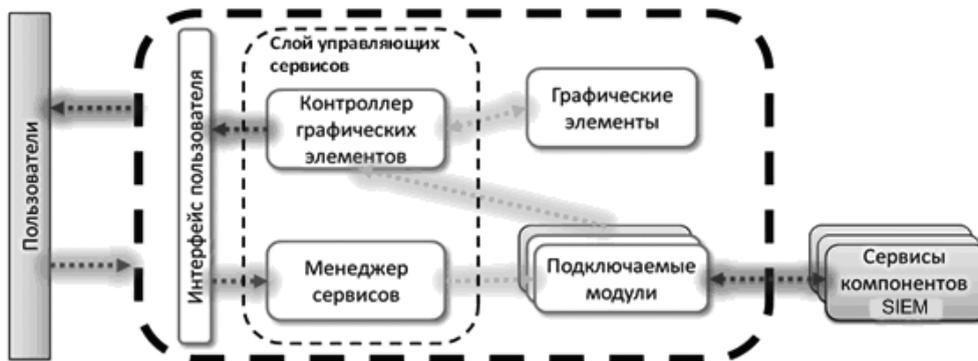


Рис. 4. Архитектура компонента визуализации

Предложения по применению

В качестве примеров областей использования, демонстрирующих преимущества SIEM-систем нового поколения, используются следующие сценарии:

компьютерная инфраструктура высокой производительности, применяющаяся для поддержки проведения Олимпийских Игр, в которой циркулируют потоки, равные сотням тысяч / миллионам событий в секунду;

распределенная компьютерная инфраструктура, в которой доставка данных о событиях безопасности от периферии к центру и передача решений по применению контрмер от центра к периферии осуществ-

ляется через коммуникационную среду, подвергающуюся многочисленным воздействиям;

критическая инфраструктура (например, дамба), в которой необходимо проведение совместной обработки данных, поступающих как от традиционных для SIEM-систем источников событий, так и от инфраструктурных датчиков (сенсоров), фиксирующих параметры состояния элементов инфраструктуры;

информационная инфраструктура для проведения мобильных денежных платежей, в которой угрозы информационной безопасности коррелируются с угрозами финансового мошенничества.

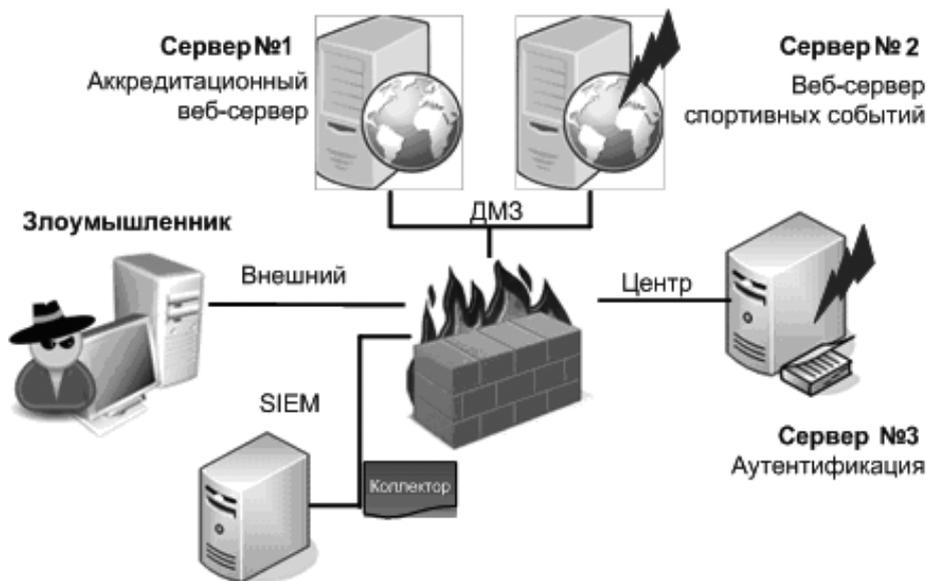


Рис. 5. Пример применения SIEM-системы в инфраструктуре Олимпийских Игр

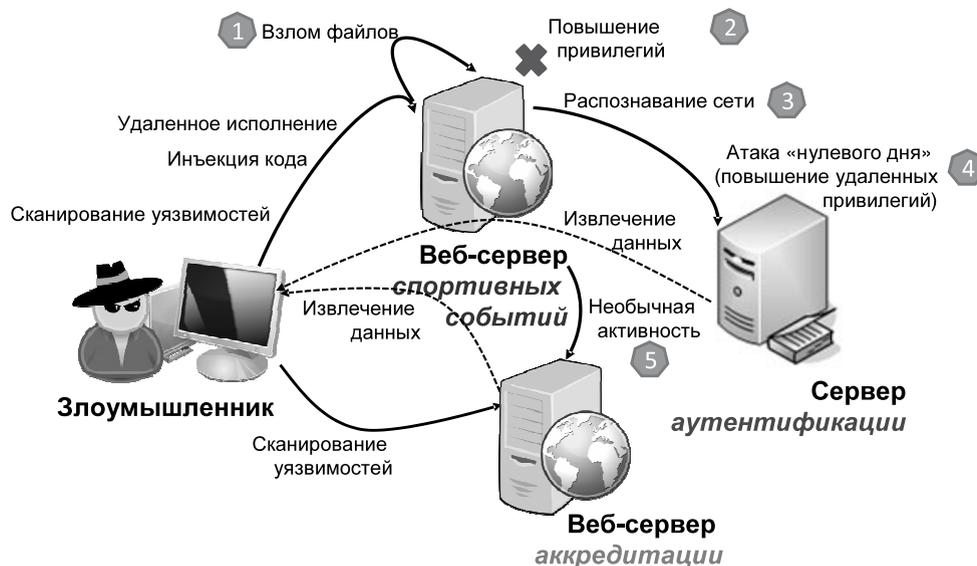


Рис. 6. Схема действий нарушителя в инфраструктуре Олимпийских Игр

Для примера остановимся на особенностях только первого сценария. С особенностями других сценариев можно ознакомиться в [1, 7].

Логическая модель примера сценария компьютерной инфраструктуры Олимпийских Игр представлена на рис. 5. Демилитаризованную зону (ДМЗ) образуют два веб-сервера: сервер аккредитации и сервер спортивных событий. К центральной части защищаемой инфраструктуры относится сервер аутентификации. В центральную часть поступают данные от периферии (удаленной части) инфраструктуры. Атакующий злоумышленник, как предполагается, также является внешним пользователем.

Схема проведения атаки злоумышленником представлена на рис. 6. Она включает пять этапов. Целью первого этапа является установление удаленного контроля над сервером спортивных событий. В ходе первого этапа злоумышленником выполняются следующие действия: сканирование уязвимостей, инъекция кода (внедрение вредоносной программы), удаленное исполнение кода и взлом файлов на сервере спортивных событий.

На втором этапе осуществляется повышение привилегий (полномочий), которыми наделяется злоумышленник. Целью этого этапа является атака «грубой силы» на

пароль локальной административной учетной записи. На третьем этапе производится исследование компьютерной сети на предмет наличия уязвимостей. Цель этого этапа заключается в определении открытых портов на сервере аутентификации. На четвертом этапе осуществляется проведение атаки «нулевого дня» (направленной на ранее неизвестную уязвимость), целью которой является удаленное повышение полномочий. Благодаря выполнению этой атаки злоумышленник получает возможность выполнения произвольных команд на удаленном сервере. Наконец, пятый этап заключается в нахождении учетной записи, обеспечивающей доступ к приложениям сервера аккредитации. Проявлением таких действий злоумышленника может быть наблюдаемая необычная активность сервера аутентификации и аккредитации.

Особенность и основная задача применения SIEM-системы в этом сценарии заключается в обеспечении ее способности выявить «необычную активность», обусловленную деятельностью злоумышленника, на фоне огромного количества прочих событий, обрабатываемых в системе.

В статье представлены основные решения по построению и функционированию нового поколения систем мониторинга и

управления безопасностью. Рассмотренные компоненты SIEM-системы нового поколения расширяют ее возможности в направлении высокой масштабируемости, возможности межуровневой корреляции событий безопасности, высокоустойчивого сбора и передачи данных в распределенной вычислительной среде и повышения безопасно-

сти информационных инфраструктур.

Работа выполняется при финансовой поддержке РФФИ (13-01-00843, 13-07-13159, 14-07-00697, 14-07-00417), программы фундаментальных исследований ОНИТ РАН (контракт №2.2), проекта ENGENSEC программы Европейского Сообщества TEMPUS и госконтракта № 14.BVV.21.0097.

СПИСОК ЛИТЕРАТУРЫ

1. Котенко И.В., Саенко И.Б. SIEM-системы для управления информацией и событиями безопасности // Защита информации. Инсайд. 2012. № 5. С. 54–65.
2. Котенко И.В., Саенко И.Б., Полубелова О.В., Чечулин А.А. Технологии управления информацией и событиями безопасности для защиты компьютерных сетей // Проблемы информационной безопасности. Компьютерные системы. 2012. № 2. С. 57–68.
3. QRadar SIEM [электронный ресурс] / URL: <http://q1labs.com/Products/QRadar-SIEM.aspx> (дата обращения 25.01.2014).
4. ArcSight ESM Enterprise Security Manager [электронный ресурс] / URL: <http://www.arcsight.com/products/products-esm> (дата обращения 25.01.2014).
5. Symantec Security Information Manager [электронный ресурс] / URL: <http://www.symantec.com/business/security-information-manager> (дата обращения 25.01.2014).
6. Sentinel Log Manager Review [электронный ресурс] / URL: <https://www.sans.org/reading-room/analysts-program/novell-whitepaper> (дата обращения 25.01.2014).
7. MASSIF FP7 Project [электронный ресурс] / URL: <http://massif-project.eu> (дата обращения 25.01.2014).
8. Service-oriented Architecture [электронный ресурс] / URL: http://en.wikipedia.org/wiki/Service-oriented_architecture (дата обращения 25.01.2014).
9. Котенко И.В., Воронцов В.В., Чечулин А.А., Уланов А.В. Проактивные механизмы защиты от сетевых червей: подход, реализация и результаты экспериментов // Информационные технологии. 2009. № 1. С. 37–42.
10. OSSIM [электронный ресурс] / URL: <http://www.alienvault.com/community> (дата обращения 25.01.2014).
11. Prelude [электронный ресурс] / URL: <http://www.prelude-ids.com> (дата обращения 25.01.2014).
12. OrBAC: Organization Based Access Control [электронный ресурс] / URL: <http://orbac.org> (дата обращения 25.01.2014).
13. Kotenko I., Polubelova O., Saenko I. Data Repository for Security Information and Event Management in Service Infrastructures // Internat. Conf. on Security and Cryptography. Rome, Italy, 2012. Pp. 308–313.
14. Полубелова О.В., Полубелова О.В., Котенко И.В., Саенко И.Б., Чечулин А.А. Применение онтологий и логического вывода для управления информацией и событиями безопасности // Системы высокой доступности. 2012. № 2. Т. 8. С.100–108.
15. Kotenko I., Polubelova O., Saenko I. The Ontological Approach for SIEM Data Repository Implementation // IEEE Internat. Conf. on Internet of Things. Bezanson, France, 2012. Los Alamitos, California. IEEE Computer Society, 2012. Pp. 761–766.
16. Kotenko I., Stepashkin M. Network Security Evaluation Based on Simulation of Malefactor's Behavior // Internat. Conf. on Security and Cryptography. Proceedings. Portugal, 2006. Pp. 339–344.
17. Kotenko I., Ulanov A. Agent Teams in Cyberspace: Security Guards in the Global Internet // Internat. Conf. on Cyber Worlds. Lausanne, Switzerland, 2006. Proceedings. IEEE Computer Society, 2006. Pp.133–140.
18. Котенко И.В., Степашкин М.В. Системы-имитаторы: назначение, функции, архитектура и подход к реализации // Изв. вузов. Приборостроение. 2006. Т. 49. № 3. С. 3–8.
19. Ruiz J.F., Harjani R., Maca A., Desnitsky V., Kotenko I., Chuchulin A. A Methodology for the Analysis and Modelling of Security Threats and Attacks for Systems of Embedded Components // Proc. of 20th Euromicro Internat. Conf. on Parallel, Distributed and Network-Based Processing. Garching, 2012. Pp. 261–268.
20. Котенко И.В., Саенко И.Б., Полубелова О.В., Чечулин А.А. Применение технологии управления информацией и событиями безопасности для защиты информации в критически важных инфраструктурах // Труды СПИИ РАН. СПб.: Наука, 2012. Вып. 1(20). С. 27–56.

21. Котенко И.В., Степашкин М.В., Котенко Д.И., Дойникова Е.В. Оценка защищенности информационных систем на основе построения деревьев социоинженерных атак // Изв. вузов. Приборостроение. 2011. Т. 54. № 12. С. 5–9.

22. Котенко И.В., Степашкин М.В., Богданов В.С. Архитектуры и модели компонентов активного анализа защищенности на основе имитации действий злоумышленников // Проблемы информационной безопасности. Компьютерные системы. 2006. № 2. С. 7–24.

23. Kotenko I., Chechulin A., Novikova E. Attack Modelling and Security Evaluation for Security Information and Event Management // Internat. Conf. on Security and Cryptography. Rome, Italy, 2012. Pp. 391–394.

24. Новикова Е.С., Котенко И.В. Механизмы визуализации в SIEM-системах // Системы высокой доступности. 2012. № 2. С. 91–99.

25. Новикова Е.С., Котенко И.В. Технологии визуализации для управления информацией и событиями безопасности // Труды СПИИ РАН. СПб.: Наука, 2012. Вып. 4(23). С. 7–29.

REFERENCES

1. Kotenko I.V., Saenko I.B. SIEM-sistemyi dlya upravleniya informatsiy i sobyitiyami bezopasnosti [SIEM-system for security information and events management], *Zaschita informatsii. Insayd [Protection of the information. Inside]*, 2012, No. 5, Pp. 54–65. (rus)

2. Kotenko I.V., Saenko I.B., Polubelova O.V., Chechulin A.A. Tehnologii upravleniya informatsiy i sobyitiyami bezopasnosti dlya zaschityi kompyuternyih setey [Security information and events management technologies to protect computer networks], *Problemy informatsionnoy bezopasnosti. Kompyuternye sistemyi [The problems of information security. Computer systems]*, 2012, No. 2, Pp. 57–68. (rus)

3. *QRadar SIEM*. Available: <http://q1labs.com/Products/QRadar-SIEM.aspx> (Accessed 25.01.2014).

4. *ArcSight ESM Enterprise Security Manager*. Available: <http://www.arcsight.com/products/products-esm> (Accessed 25.01.2014).

5. *Symantec Security Information Manager*. Available: <http://www.symantec.com/business/security-information-manager> (Accessed 25.01.2014).

6. *Sentinel Log Manager Review*. Available: <https://www.sans.org/reading-room/analysts-program/novell-whitepaper> (Accessed 25.01.2014).

7. *MASSIF FP7 Project*. Available: <http://massif-project.eu> (Accessed 25.01.2014).

8. *Service-oriented Architecture*. Available: http://en.wikipedia.org/wiki/Service-oriented_architecture (Accessed 25.01.2014).

9. Kotenko I.V., Vorontsov V.V., Chechulin A.A., Ulanov A.V. Proaktivnyie mehanizmyi zaschityi ot setevyih chervyev: podhod, realizatsiya i rezultaty eksperimentov [Proactive security mechanisms against network worms: approach, implementation and results of experiments], *Informatsionnye tehnologii [Information Technologies]*, 2009, No. 1, Pp. 37–42. (rus)

10. *OSSIM*. Available: <http://www.alienvault.com/community> (Accessed 25.01.2014).

11. *Prelude*. Available: <http://www.prelude-ids.com> (Accessed 25.01.2014).

com (Accessed 25.01.2014).

12. *OrBAC: Organization Based Access Control*. Available: <http://orbac.org> (Accessed 25.01.2014).

13. Kotenko I., Polubelova O., Saenko I. Data Repository for Security Information and Event Management in Service Infrastructures, *International Conference on Security and Cryptography*. Rome, Italy, 2012, Pp. 308–313.

14. Polubelova O.V., Kotenko I.V., Saenko I.B., Chechulin A.A. Primenenie ontologiy i logicheskogo vyivoda dlya upravleniya informatsiy i sobyitiyami bezopasnosti [The use of ontologies and inference for security information and events management], *Sistemyi vyisokoy dostupnosti [High-availability systems]*, 2012, No. 2, Vol. 8, Pp. 100–108. (rus)

15. Kotenko I., Polubelova O., Saenko I. The Ontological Approach for SIEM Data Repository Implementation, *IEEE International Conference on Internet of Things*. Bezanson, France, 2012, Los Alamitos, California, IEEE Computer Society, 2012, Pp. 761–766.

16. Kotenko I., Stepashkin M. Network Security Evaluation based on Simulation of Malefactor's Behavior, *International Conference on Security and Cryptography. Proceedings*. Portugal, 2006, Pp. 339–344.

17. Kotenko I., Ulanov A. Agent Teams in Cyberspace: Security Guards in the Global Internet, *International Conference on Cyber Worlds*. Lausanne, Switzerland, 2006, Proceedings, IEEE Computer Society, 2006, Pp. 133–140.

18. Kotenko I.V., Stepashkin M.V. Sistemyi-imitatoryi: naznachenie, funktsii, arhitektura i podhod k realizatsii [Systems-imitators: purpose, functions, architecture and the approach to realization], *Izvestiya vuzov. Priborostroenie*, 2006, Vol. 49, No. 3, Pp. 3–8. (rus)

19. Ruiz J.F., Harjani R., Maca A., Desnitsky V., Kotenko I., Chechulin A. A Methodology for the Analysis and Modelling of Security Threats and Attacks for Systems of Embedded Components, *Pro-*

ceedings of 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, Garching, 2012, Pp. 261–268.

20. **Kotenko I.V., Saenko I.B., Polubelova O.V., Chechulin A.A.** Primenenie tehnologii upravleniya informatsiy i sobyitiyami bezopasnosti dlya zaschityi informatsii v kriticheski vazhnykh infrastrukturakh [Application of security information and events management technology to protect critical information infrastructures], *Trudy SPII RAN*. St. Petersburg: Nauka Publ., 2012, Iss. 1(20), Pp. 27–56. (rus)

21. **Kotenko I.V., Stepashkin M.V., Kotenko D.I., Doynikova E.V.** Otsenka zaschishennosti informatsionnykh sistem na osnove postroeniya derev sotsioinzhenernykh atak [Security assessment of information systems based on building socio-engineering attacktrees], *Izvestiya vuzov. Priborostroenie*, 2011, Vol. 54, No. 12, Pp. 5–9. (rus)

22. **Kotenko I.V., Stepashkin M.V., Bogdanov V.S.** Arhitektury i modeli komponentov aktivnogo analiza zaschishennosti na osnove imitatsii deystviy

zloumyishlennikov [Architectures and components of active security analysis model based on simulating the action of hackers], *Problemy informatsionnoy bezopasnosti. Kompyuternye sistemy* [The problems of information security. Computer systems], 2006, No. 2, Pp. 7–24. (rus)

23. **Kotenko I., Chechulin A., Novikova E.** Attack Modelling and Security Evaluation for Security Information and Event Management, *International Conference on Security and Cryptography*, Rome, Italy, 2012, Pp. 391–394.

24. **Novikova E.S., Kotenko I.V.** Mehanizmy vizualizatsii v SIEM-sistemakh [Visualization mechanisms in SIEM systems], *Sistemy vyisokoy dostupnosti* [High-availability systems], 2012, No. 2, Pp. 91–99. (rus)

25. **Novikova E.S., Kotenko I.V.** Tehnologii vizualizatsii dlya upravleniya informatsiy i sobyitiyami bezopasnosti [Visualization technologies for security information and events management], *Trudy SPII RAN*, St. Petersburg: Nauka Publ, 2012, Iss. 4(23), Pp. 7–29. (rus)

КОТЕНКО Игорь Витальевич – заведующий лабораторией проблем компьютерной безопасности Санкт-Петербургского института информатики и автоматизации РАН, доктор технических наук, профессор.

199178, Россия, Санкт-Петербург, 14-я линия В.О., д. 39.
E-mail: ivkote@comsec.spb.ru

KOTENKO, Igor V. St. Petersburg institute for information and automation of Russian academy of sciences.

199178, Liniya 14, 39, St. Petersburg, Russia.
E-mail: ivkote@comsec.spb.ru

САЕНКО Игорь Борисович – ведущий научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского института информатики и автоматизации РАН, доктор технических наук, профессор.

199178, Россия, Санкт-Петербург, 14-я линия В.О., д. 39.
E-mail: ibsaen@comsec.spb.ru

SAENKO, Igor B. St. Petersburg institute for information and automation of Russian academy of sciences.

199178, Liniya 14, 39, St. Petersburg, Russia.
E-mail: ibsaen@comsec.spb.ru

ЮСУПОВ Рафаэль Мидхатович – директор Санкт-Петербургского института информатики и автоматизации РАН, член-корреспондент РАН.

199178, Россия, Санкт-Петербург, 14-я линия В.О., д. 39.
E-mail: yusupov@iias.spb.su

YUSUPOV, Rafael M. St. Petersburg institute for information and automation of Russian academy of sciences.

199178, Liniya 14, 39, St. Petersburg, Russia.
E-mail: yusupov@iias.spb.su

УДК 004.054

А.А. Бородин

ИССЛЕДОВАНИЕ НАГРУЗОЧНЫХ СПОСОБНОСТЕЙ КОМПЬЮТЕРНЫХ СИСТЕМ

A.A. Borodin

RESEARCH OF COMPUTER SYSTEMS LOAD CAPABILITY

Нагрузочное тестирование является одним из основных способов измерения характеристик и проверки корректности функционирования информационных систем при эксплуатационных условиях. Ввиду роста и усложнения информационных систем для их тестирования требуется применение все большей нагрузки. Для ее создания оказывается недостаточно ресурсов одного компьютера, поэтому применяются механизмы, основанные на совокупности объединенных компьютерных узлов. Приведены результаты экспериментов по определению предельных нагрузочных способностей одиночных компьютерных систем, а также факторов, оказывающих на них влияние.

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ; НАГРУЗОЧНЫЕ СПОСОБНОСТИ; ВРЕМЯ ОБРАБОТКИ ПАКЕТА ОПЕРАЦИОННОЙ СИСТЕМОЙ.

Load testing is one of the main tools used to measure parameters and determine computer system behavior under operating conditions. Constant development and extension of computer systems lead to the load increase to maintain normal condition. As the resources of a single computer are not enough to create that amount of load, there are different methods to reach that amount of load using distributed computing. This article presents the experiment results to determine load capability limits for a single computer system and factors affecting them.

LOAD TESTING; LOAD CAPABILITY; PROCESSING TIME OF A PACKET BY OPERATING SYSTEM.

Нагрузочное тестирование является важнейшим средством, с помощью которого проверяется корректность функционирования и измерение характеристик информационных систем в эксплуатационных условиях. Эффективность проведенного нагрузочного тестирования напрямую влияет на качество разрабатываемой информационной системы.

Процесс тестирования состоит из множества шагов [1], которые условно можно разделить на три стадии: подготовка к тестированию, запуск тестов (сбор результатов) и анализ результатов. Основные усилия большинства исследователей направлены на ре-

шение вопросов эффективной подготовки к тестированию [2–9] и анализа результатов [10, 11]. Однако вопросам запуска тестов уделяется значительно меньшее внимание. При тестировании используются готовые программные продукты (инструменты нагрузочного тестирования). В настоящее время появилось большое количество подобных программ – проприетарных [12] и свободных [13]. Все они при выполнении стадии запуска тестов (создания нагрузки на информационную систему) используют ресурсы обычных компьютерных узлов. В случае их нехватки для наращивания создаваемой нагрузки на тестируемую систе-

му, применяют облачные и кластерные вычисления или совокупность компьютеров, объединенных с помощью локальной сети. Однако применение облачных и кластерных вычислений требует подключения к глобальной сети, что не всегда рационально и возможно. Поэтому информационную систему в большинстве случаев тестируют при помощи компьютеров, объединенных в локальную сеть. Постоянное усложнение информационных систем требует применения все большего количества компьютерных узлов для достижения эксплуатационных условий. Это приводит к увеличению затрат и усилий на обслуживание такой системы. Именно на преодоление этих недостатков и направлено наше исследование.

В настоящее время стадия запуска тестов реализовывается в виде модуля нагрузки. Он представляет собой одну из подпрограмм инструментов нагрузочного тестирования. Сформированные на подготовительных этапах тесты передаются этому модулю. На основании тестов модуль нагрузки формирует большое количество вызовов функций интерфейса программирования приложений операционной системы. Итогом выполнения каждой такой функции является отправка в сеть сформированного модулем нагрузки запроса. После того как модуль выполнит вызов функции, задачу отправки запроса выполняет операционная система. Для выполнения этой задачи ОС проводит запуск подпрограмм, которые осуществляют подготовку запроса к отправке, помещают запрос в пакет, определяют путь его следования и проводят передачу сформированного пакета в контроллер среды передачи. Операционная система может запускать и ряд других подпрограмм, например, осуществляющих проверку допустимости отправки запроса в соответствии с настройками безопасности ОС. Исполнение этого множества подпрограмм ложится на центральный процессор. По мере увеличения количества вызовов функций интерфейса программирования, подпрограммами ОС затрачивается все большее количество ресурсов центрального процессора. Это не сильно влияло бы на количество сформированных запросов в се-

кунду, если был запущен лишь модуль нагрузки. Однако в большинстве случаев в ОС запущено большое количество сторонних прикладных приложений, которые выполняют свои функции. Одновременно с этим при выполнении шагов создания нагрузки должны быть запущены программы, осуществляющие анализ поступающих ответов от тестируемой системы. Эти приложения собирают наиболее ценные данные нагрузочного тестирования – метрики производительности. Именно они подвергаются анализу, и на основе их делаются выводы о качестве спроектированной информационной системы. Для выполнения всей этой работы ресурсов одного компьютерного узла не хватает.

Предельная интенсивность генерации сообщений компьютерного узла зависит от архитектуры сетевого адаптера – устройства, обеспечивающего связь компьютера с сетью [14]. Возможность достижения этой интенсивности зависит от операционной системы, запущенных приложений и архитектуры самого компьютера. Мы провели ряд экспериментов с целью измерения времени, необходимого операционной системе для обработки отправляемого в сеть сообщения.

Для проведения экспериментов была использована программа `tcpdump`, входящая в состав UNIX-подобных операционных систем. Она использует низкоуровневую библиотеку `pcap` (`packetcapture`), осуществляющую перехват данных на уровне драйвера в соответствии с назначенными параметрами [15]. Каждый перехваченный пакет снабжается временной меткой. Она указывает системное время ОС, когда пакет был перехвачен. Для измерения времени, необходимого ОС для обработки сообщения, выполнялись следующие действия:

- считывание системного времени;
- помещение его в пакет;
- отправка пакета через сетевой стек;
- перехват пакета;
- вычисление искомого времени, исходя из разницы между временем отправки и временем перехвата.

Специально созданная экспериментальная программа отправляла системное время

в сеть с помощью транспортного протокола UDP. Этот протокол был выбран потому, что операционной системе требуется меньшее количество вычислительных ресурсов на его обработку. В программе было предусмотрено постепенное увеличение количества передаваемых сообщений в секунду, что позволило отследить влияние интенсивности генерации сообщений на время обработки одного сообщения. Для сбора данных применялась программа `tcpdump`, встроенная в операционные системы всех экспериментальных платформ. Эксперимент состоял из следующих шагов:

- запуск программы перехвата пакетов `tcpdump`;
- запуск экспериментальной программы;
- ожидание завершения работы экспериментальной программы;
- завершение работы программы перехвата пакетов;
- сбор и анализ результатов.

На практике в ходе проведения нагрузочного тестирования применяются обычные компьютерные узлы под управлением различных операционных систем. По этой причине, для достоверности результатов, эксперименты проводились также на базе нескольких обычных компьютерных узлов с актуальным аппаратным и программным обеспечением. В нашем исследовании эти узлы называются *экспериментальными платформами* (ЭП). Во время эксперимента применялись следующие компьютерные узлы:

экспериментальная платформа 1, компьютер Intel Core i7 920 2.67 ГГц, DDR3-1066 9 Гб, CentOS 6.4;

экспериментальная платформа 2, ноутбук Samsung R580-JS03 Intel Core i5 430M 2.26 ГГц, DDR3-1066 3 Гб, Ubuntu 12.10;

экспериментальная платформа 3, ноутбук MacBook Air Intel Core i7 3667U 2 ГГц, DDR3L-1600 МГц 8 Гб, OS X 10.8.5.

Предварительные испытания показали, что наибольшей интенсивности генерации сообщений нельзя достигнуть при использовании однопоточного приложения. Несмотря на то что экспериментальное приложение формировало около ста тысяч сообщений в секунду, реальное количество

отправляемых сообщений было значительно меньше. По мере увеличения количества запущенных экземпляров экспериментальной программы количество отправляемых сообщений возрастало, поэтому в процессе эксперимента запускалось несколько экземпляров. Для определения их количества предварительно было измерено, сколько экземпляров программы можно запустить до того момента, когда потери пакетов при перехвате составят около 50 % (при помощи программы `tcpdump`). Исходя из этого, во время эксперимента выбирались такие условия, когда потери не превышают 5 %.

В ходе эксперимента было проведено более миллиона измерений на каждой из платформ. Для получения достоверных результатов эксперимент повторялся десять раз. Такое количество повторов было выбрано исходя из объема обрабатываемых данных и затрат времени. Например, один эксперимент занимает 1 мин 50 с, но при этом объем данных составляет 1,2 Гб для одной платформы.

Результаты измерения времени, необходимого экспериментальным платформам для обработки сообщения, приведены в табл. 1 и 2. На рис. 1–3 отражено изменение интенсивности генерации сообщений и среднего времени обработки одного сообщения для каждой платформы.

В табл. 1 приведены обобщенные результаты для всех испытаний. В столбцах t_{\min} и t_{\max} приведено наибольшее и наименьшее время обработки пакета, обнаруженное в ходе десяти испытаний. Детализированные результаты для каждого испытания отражены в табл. 2.

Погрешность среднего рассчитывалась с помощью следующей формулы:

$$S_x = \frac{S}{\sqrt{n}} = \sqrt{\frac{\sum_{i=0}^n (t_i - \bar{t})^2}{n(n-1)}}.$$

Для расчета погрешности среднего между испытаниями применялся коэффициент Стьюдента, равный 2,262 (число измерений 10, надежность 0,95).

На представленных графиках видно, что по мере увеличения интенсивности генерации сообщений среднее время их обработ-

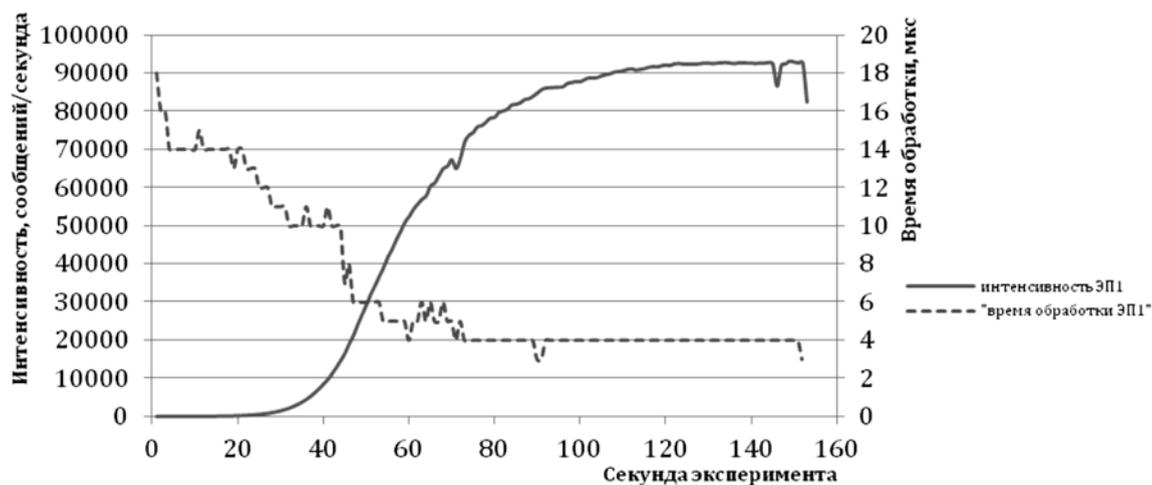


Рис. 1. Изменение интенсивности и среднего времени обработки ЭП1

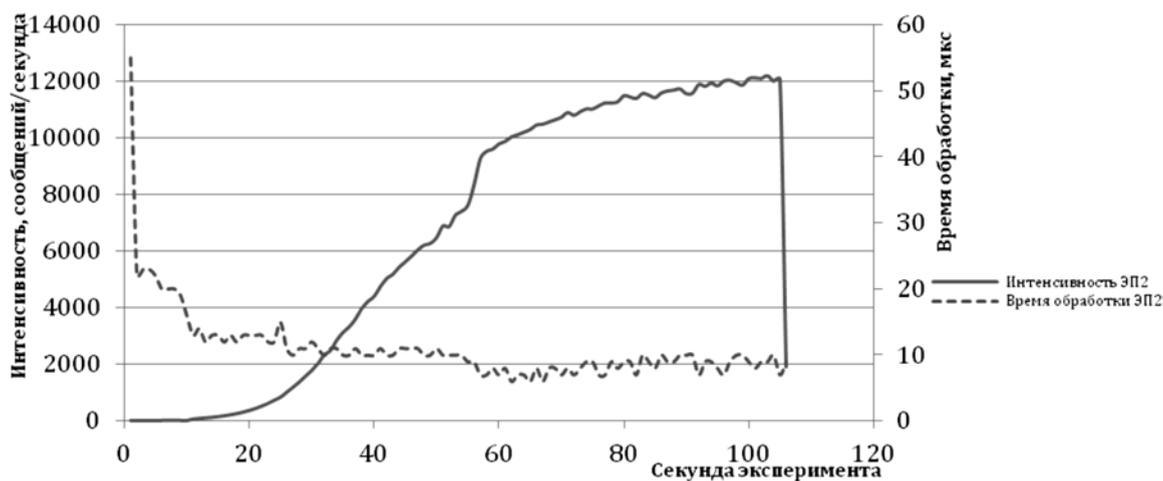


Рис. 2. Изменение интенсивности и среднего времени обработки ЭП2

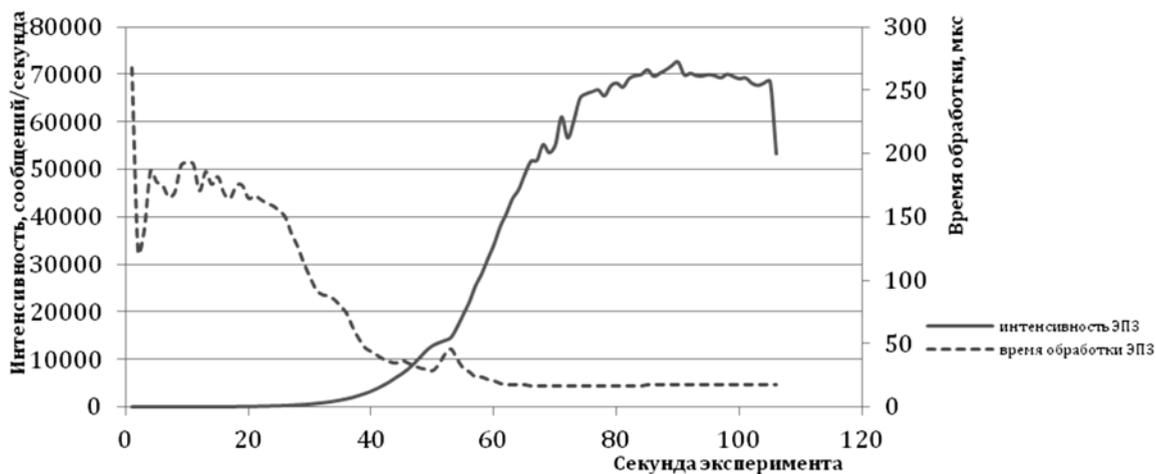


Рис. 3. Изменение интенсивности и среднего времени обработки ЭП3

Таблица 1

Обобщенные результаты измерений времени обработки

Платформа	Количество испытаний	$t_{\text{среднее}}$, мкс	t_{min} , мкс	t_{max} , мкс	Погрешность среднего между испытаниями, мкс
Общие показатели					
ЭП 1	10	9,05	2	1704	$\pm 1,6323$
ЭП 2		10,35	2	3770	$\pm 0,5810$
ЭП 3		60,23	< 0	31740	$\pm 0,9627$
Предельные показатели					
ЭП 1	10	3,83	2	835	$\pm 0,0721$
ЭП 2		8,20	2	3770	$\pm 0,0779$
ЭП 3		17,97	< 0	2030	$\pm 0,2056$

ки уменьшается. По этой причине в табл. 1 приведены результаты измерений времени обработки в тот момент, когда интенсивность генерации достигала наибольшего значения. Кроме того, для более детального изучения этого процесса были проведены дополнительные эксперименты, но из-за ограниченного объема статьи здесь их результаты не отражены.

На время обработки сообщений влияет множество факторов: запущенные приложения, настройка и реализация сетевого стека, режимы управления энергопитанием компьютерного узла и т. д. С учетом этого эксперименты проводились в условиях, когда платформы были не нагружены другими задачами. И даже в таком случае, при проведении двух одинаковых экспериментов в одинаковых условиях, полученные результаты имели различия, хотя и незначительные, связанные с внутренними процессами ОС и компьютерной системы в целом.

Следующий эксперимент был посвящен проверке предположения, что поток получаемых из сети сообщений может влиять на нагружающие способности узла. В ходе проведения нагрузочного тестирования со стороны тестируемого узла формируется поток ответов на запросы нагружающего узла. Каждый полученный ответ обрабатывается операционной системой до того, как он будет проанализирован приложением. Этот поток ответов в нашей работе мы называем *обратной нагрузкой*. Кроме того, ис-

пользование ряда протоколов сетевого стека приводит к тому, что взаимодействующие узлы обмениваются служебной информацией. Все это потребляет вычислительные ресурсы компьютера и может способствовать снижению предельной нагрузочной способности узла.

Для проверки данного предположения платформы были соединены в локальную сеть. Предварительные испытания показали, что экспериментальная платформа 3 лучше всего подходит на роль узла, показания которого будут считываться и анализироваться в ходе эксперимента. Это связано с тем, что один экземпляр экспериментальной программы на ЭП3 способен генерировать сообщения с большой интенсивностью при одновременном перехвате пакетов без потерь (в программе tcpdump).

Эксперимент состоял из следующих шагов:

- запуск перехватчика tcpdump на узлах ЭП1 и ЭП2, создающих обратную нагрузку;
- запуск экспериментальной программы на ЭП3;
- наблюдение за создаваемой нагрузкой с помощью программы мониторинга сетевой активности, встроенной в ОС ЭП3;
- при достижении ЭП3 пикового значения интенсивности генерации пакетов осуществлялся запуск программы tcpdump на ЭП3;
- запуск экспериментальной программы на ЭП1 и ЭП2;

Таблица 2

Детализированные результаты измерений времени обработки

Платформа	Номер испытания	Обработано пакетов	$t_{\text{среднее}}$, мкс	t_{min} , мкс	t_{max} , мкс	Погрешность измерений среднего, мкс
ЭП 1	1	4 636 952	10,52	2	900	$\pm 0,00287$
	2	4 623 601	8,06	2	257	$\pm 0,00185$
	3	4 607 858	8,39	2	194	$\pm 0,00199$
	4	4 631 816	7,95	2	101	$\pm 0,00178$
	5	4 674 155	15,16	2	1704	$\pm 0,00491$
	6	4 632 833	8,09	2	281	$\pm 0,00192$
	7	4 644 583	7,99	2	424	$\pm 0,00191$
	8	4 593 897	8,23	2	835	$\pm 0,00202$
	9	4 526 941	8,20	2	402	$\pm 0,00196$
	10	4 636 628	7,87	2	384	$\pm 0,00185$
ЭП 2	1	1 013 345	12,56	2	1817	$\pm 0,03862$
	2	1 013 754	9,89	2	2461	$\pm 0,03932$
	3	1 025 806	10,59	2	1838	$\pm 0,03674$
	4	1 023 268	9,84	2	1786	$\pm 0,0357$
	5	1 015 334	10,05	2	1818	$\pm 0,0401$
	6	1 029 289	9,83	2	1805	$\pm 0,03882$
	7	1 026 739	9,99	2	2123	$\pm 0,03823$
	8	1 027 369	10,17	2	3770	$\pm 0,03811$
	9	1 029 575	10,24	2	1799	$\pm 0,0393$
	10	1 027 199	10,38	2	1796	$\pm 0,03648$
ЭП 3	1	5 443 042	61,27	<0	1231	$\pm 0,01816$
	2	5 455 387	57,17	<0	447	$\pm 0,0164$
	3	5 432 302	60,44	<0	2031	$\pm 0,01777$
	4	5 434 095	61,44	<0	1290	$\pm 0,01828$
	5	5 454 387	60,99	<0	410	$\pm 0,018$
	6	5 486 672	61,51	<0	2090	$\pm 0,0182$
	7	5 598 231	59,84	<0	31740	$\pm 0,01839$
	8	5 608 325	59,63	<0	2049	$\pm 0,0174$
	9	5 617 287	59,14	<0	556	$\pm 0,01721$
	10	5 607 822	60,91	<0	458	$\pm 0,01795$

- наблюдение на ЭП3 за обратной нагрузкой с помощью программы мониторинга;
- при достижении пикового значения создаваемой обратной нагрузки осуществлялось завершение работы экспериментальной программы и перехватчика пакетов на узлах ЭП1 и ЭП2;
- завершение работы экспериментальной программы и перехватчика пакетов на ЭП3;

- сбор и анализ результатов.

Результаты проведенного эксперимента отражены на рис. 4.

На представленном графике можно видеть, что по мере увеличения обратной нагрузки происходит снижение создаваемой. После прекращения генерации обратной нагрузки интенсивность генерации пакетов исследуемой системы восстанавливается. Одновременно с этим увеличивается

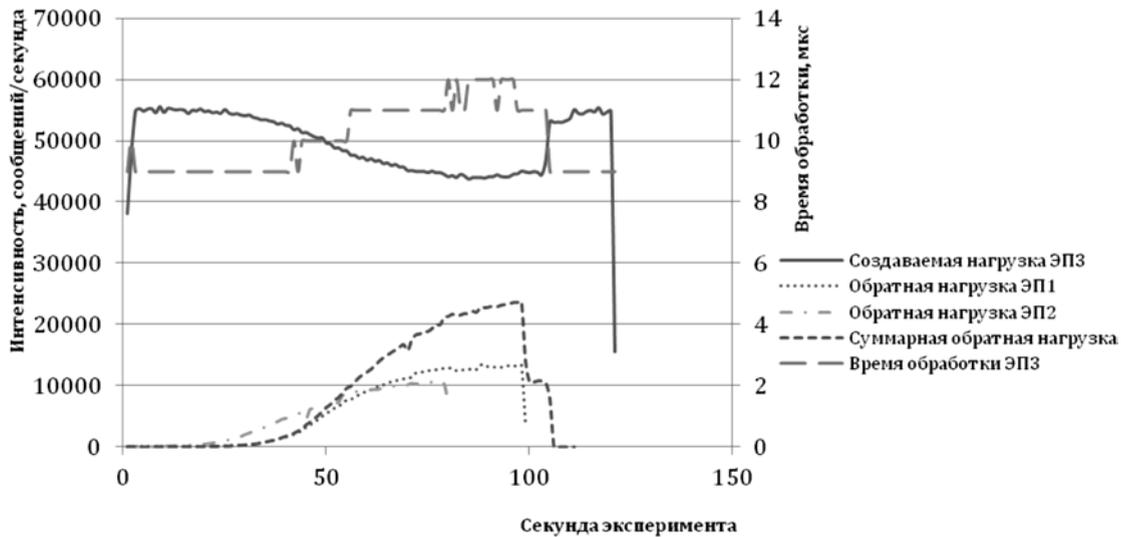


Рис. 4. Изменение интенсивности и среднего времени обработки в ходе эксперимента

среднее время обработки сообщения операционной системой. В нашем исследовании данную особенность мы назвали *эффектом обратной нагрузки*.

В описанных выше экспериментах использовались популярные современные операционные системы, за исключением Microsoft Windows. Это вызвано тем, что точность системного таймера такой ОС не позволяет измерить время обработки одного сообщения сетевым стеком, и ее измерения затруднительны.

Проведенные эксперименты позволяют оценить нагрузочные способности существующих компьютерных систем. При этом экспериментальная программа и перехватчик пакетов выступали в роли аналога модуля нагрузки и сборщика метрик производительности. Предельные нагрузочные способности каждой из экспериментальных платформ разительно отличаются друг от друга. Наибольшая интенсивность генерации сообщений, без потерь данных, составила около 90 000 в секунду на ЭП1. Используя предельные результаты измерений можно рассчитать теоретическую нагрузочную способность. Для ЭП1 она составляет 261 096, для ЭП2 — 121 951 и для ЭП3 — 55 648 сообщений в секунду. Как можно видеть, теоретическая нагрузочная способность ЭП3 меньше практической. По всей видимости, это связано с особенностями

операционной системы данной платформы.

Результаты проведенных экспериментов демонстрируют характеристики операционных систем и компьютерных узлов, используемых в настоящее время для проведения нагрузочного тестирования. Полученные данные позволяют сделать вывод о том, что имеющая место нехватка нагрузочных способностей компьютерных узлов обусловлена как внутренними, так и внешними факторами. Модули инструментов нагрузочного тестирования, анализирующие ответы от тестируемого узла, вместе с ресурсоемкими транспортными протоколами и эффектом обратной нагрузки снижают нагрузочные способности компьютера. Это свидетельствует о том, что применение компьютеров со стандартной архитектурой для решения задач нагрузочного тестирования неэффективно.

Учитывая полученные результаты, нами был разработан метод проведения нагрузочного тестирования, основанный на применении аппаратного модуля с программируемой логической интегральной схемой (ПЛИС). С его помощью может осуществляться создание нагрузки на тестируемую систему и сбор результатов тестирования. Результаты проведенной работы вместе с характеристиками модуля будут опубликованы в ближайшее время.

СПИСОК ЛИТЕРАТУРЫ

1. Meier J.D., Farre C., Bansode P., Barber Sc., Rea D. Performance Testing Guidance for Web Applications [электронный ресурс] / URL: <http://msdn.microsoft.com/en-us/library/bb924375.aspx> (дата обращения: 10.03.2014).
2. Силаков Д.В. Автоматизация тестирования web-приложений, основанных на скриптовых языках // Труды ИСП РАН. 2008. № 2. С. 159–178.
3. Сортов А., Хорошилов А. Функциональное тестирование web-приложений на основе технологии UniTesK // Труды ИСП РАН. 2004. № 8. С. 77–97.
4. Яковенко П.Н., Сапожников А.В. Инфраструктура тестирования веб-сервисов на базе технологии TTCN-3 и платформы.NET // Труды ИСП РАН. 2009. № 117. С. 63–74.
5. Zhang P., Elbaum S., Dwyer M.B. Automatic Generation of Load Tests // Proc.of the 26th IEEE / ACM Internat. Conf. on Automated Software Engineering. 2011. Pp. 43–52.
6. Zhang P., Elbaum S., Dwyer M.B. Compositional load test generation for software pipelines // Proc. of Internat. Symp. on Software Testing and Analysis. 2012. Pp. 89–99.
7. Morrison K., Haddad H.M. Converting users to testers: an alternative approach to load test script creation, parameterization and data correlation // J. of Computing Sciences in Colleges. 2012. Vol. 28. Iss. 2. Pp. 188–196.
8. Cai Yu., Grundy J., Hosking J. Synthesizing client load models for performance engineering via web crawling // Proc. of the 22nd IEEE/ACM Internat. Conf. on Automated Software Engineering. 2007. Pp. 353–362.
9. Ермакин А.А. Разработка метода построения комплекса нагрузочного тестирования распределенной информационной системы: Дис. ... канд. техн. наук. СПб., 2005.
10. Zhen Ming Jiang. Automated analysis of load testing results // Proc. of the 19th Internat. Symp. on Software Testing and Analysis. 2010. Pp. 143–146.
11. Haroon Malik. A Methodology to Support Load Test Analysis // Proc. of the 32nd ACM/IEEE Internat. Conf. on Software Engineering. 2010. Vol. 2. Pp. 421–424.
12. Top 15 Performance Testing Tools [электронный ресурс] / URL: <http://www.toolsjournal.com/tools-world/item/156-top-performance-testing-tools> (дата обращения 10.03.2014).
13. Performancetesttools [электронный ресурс] / URL: <http://www.opensourcetesting.org/performance.php> (дата обращения 10.03.2014).
14. Serpanos D., Wolf T. Architecture of Network Systems. Burlington: Morgan Kaufmann Publishers, 2011. 313 p.
15. Pcap [электронный ресурс] / URL: <http://ru.wikipedia.org/wiki/Pcap> (дата обращения 10.03.2014).

REFERENCES

1. Meier J.D., Farre C., Bansode Pr., Barber Sc., Rea D. Performance Testing Guidance for Web Applications, *Microsoft patterns & practices*, 2007, Available: <http://msdn.microsoft.com/en-us/library/bb924375.aspx> (Accessed 10.03.2014).
2. Silakov D.V. Avtomatizatsiya testirovaniya web-prilozheniy, osnovannykh na skriptovykh yazykakh, *Trudy Instituta sistemnogo programirovaniya RAN*, 2008, No. 2, Pp. 159–178. (rus)
3. Sortov A., Khoroshilov A. Funktsionalnoye testirovaniye web-prilozheniy na osnove tekhnologii UniTesK, *Trudy Instituta sistemnogo programirovaniya RAN*, 2004, No. 8, Pp. 77–97. (rus)
4. Yakovenko P.N., Sapozhnikov A.V. Infrastruktura testirovaniya web-servisov na baze tekhnologii TTCN-3 i platformy.NET, *Trudy Instituta sistemnogo programirovaniya RAN*, 2009, No. 117, Pp. 63–74. (rus)
5. Zhang P., Elbaum S., Dwyer M.B. Automatic Generation of Load Tests, *Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering*, 2011, Pp. 43–52.
6. Zhang P., Elbaum S., Dwyer M.B. Compositional load test generation for software pipelines, *Proceedings of the International Symposium on Software Testing and Analysis*, 2012, Pp. 89–99.
7. Morrison K., Haddad H.M. Converting users to testers: an alternative approach to load test script creation, parameterization and data correlation, *Journal of Computing Sciences in Colleges*, 2012, Vol. 28, Iss. 2, Pp. 188–196.
8. Cai Yu., Grundy J., Hosking J. Synthesizing client load models for performance engineering via web crawling, *Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering*, 2007, Pp. 353–362.
9. Ермакин А.А. *Razrabotka metoda postroyeniya kompleksa nagruzochnogo testirovaniya raspredelennoy informatsionnoy sistemy: dis. ... kand. tekhn. nauk*, St. Petersburg, 2005. (rus)
10. Zhen Ming Jiang. Automated analysis of load testing results, *Proceedings of the 19th International Symposium on Software Testing and Analysis*, 2010, Pp. 143–146.



11. **Haroon Malik.** A Methodology to Support Load Test Analysis, *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, 2010, Vol. 2, Pp. 421–424.

12. *Top 15 Performance Testing Tools.* Available: <http://www.toolsjournal.com/tools-world/item/156-top-performance-testing-tools> (Accessed 10.03.2014).

13. *Performance test tools.* Available: <http://www.opensourcetesting.org/performance.php> (Accessed 10.03.2014).

14. **Serpanos D., Wolf T.** *Architecture of Network Systems.* Burlington: Morgan Kaufmann Publishers, 2011, 313 p.

15. *Pcap.* Available: <http://ru.wikipedia.org/wiki/Pcap> (Accessed 10.03.2014).

БОРОДИН Антон Александрович – ассистент Московского государственного университета леса. 141005, Московская обл., г. Мытищи-5, ул. 1-я Институтская, д. 1.
E-mail: AntonioBorodin@gmail.com

BORODIN, Anton A. *Moscow State Forest University.*
141005, 1st Institutskaya Str. 1, Mytischki, Moscow region, Russia.
E-mail: AntonioBorodin@gmail.com

УДК 620.9+004.9

С.Г. Черный, А.А. Жиленков

ИДЕНТИФИКАЦИЯ ВНЕШНИХ ПАРАМЕТРОВ СИГНАЛОВ ДЛЯ ЭКСПЕРТНЫХ ПОДСИСТЕМ В СОСТАВЕ УСТРОЙСТВ СУДОВЫХ ЭЛЕКТРОЭНЕРГЕТИЧЕСКИХ СИСТЕМ

S.G. Cherny, A.A. Zhilenkov

IDENTIFICATION OF EXTERNAL PARAMETERS FOR THE EXPERT SUBSYSTEMS IN THE COMPOSITION OF THE SHIPBOARD OF ELECTRIC POWER SYSTEMS

Одной из важнейших задач для фильтро-компенсирующих устройств является точное определение изменяющихся гармонических составляющих несинусоидального сигнала. Именно это позволяет силовой части фильтра компенсировать соответствующие гармоники. В автономных электроэнергетических системах, работающих на мощную полупроводниковую нагрузку, формы сетевых токов и напряжений существенно искажены, а генераторные установки генерируют напряжение с непостоянной частотой, фазой и амплитудой. Что делает определение гармонического состава искаженных сигналов нетривиальной задачей.

МОЩНОСТЬ; НАГРУЗКА; ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ; ИДЕНТИФИКАЦИЯ; ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ.

One of the most important tasks for the filter-compensation devices is an accurate determination of changing harmonic components of a non-sinusoidal signal. It helps the power component of the filter compensate the appropriate harmonics. The form of the network currents and voltages in the autonomous electric energetic systems operating with powerful semiconductor load are rather distorted and generator sets generate voltage with non-constant frequency, phase and amplitude. Thus, the calculation of the harmonic composition of the distorted signals proves to be a sophisticated problem.

POWER LOAD; INTELLIGENT SYSTEMS; IDENTIFICATION; OPTIMAL CONTROL.

Современные судовые электроэнергетические системы (СЭЭС) характеризуются наличием в своем составе большого количества преобразовательной нагрузки, включающей преобразователи частоты, источники бесперебойного питания, инверторы, выпрямители и другие потребители, отличающиеся нелинейной вольт-амперной характеристикой.

Так, на современных судах широко используются системы электродвижения на основе гребных электрических установок (ГЭУ), которые для наиболее эффективного использования системы получают питание от единой СЭЭС, обеспечивающей энергией все общесудовые потребители. Питание таких пропульсивных установок осуществляется через полупроводниковые

преобразователи, негативно воздействующие на питающую сеть переменного тока путем генерации в нее высших гармонических составляющих токов и напряжений, значения которых превышают допустимые нормативной документацией [5].

Высшие гармонические составляющие тока и напряжения, обуславливающие потребляемую от станции мощность искажений, наряду с реактивной мощностью, увеличивают вероятность возникновения резонансных явлений в СЭЭС, нарушают работу микропроцессорной техники, устройств защиты и автоматики, вызывают ускоренное старение изоляции основного электрооборудования. То есть снижают надежность электроснабжения и работу самих потребителей автономного объекта. В свя-



зи с этим все большее внимание уделяется вопросам разработки способов и средств устранения негативного влияния высших гармоник токов и напряжений: вопросам создания автоматических систем управления процессами генерирования, передачи и распределения электрической энергии, позволяющим обеспечить требуемое качество электроэнергии, устранение взаимного влияния параллельно работающего оборудования со стороны питающей сети и т. п.

Экспериментальные исследования [1–4], проведенные на автопароме «Ейск» Керченской государственной компании «Керченская паромная переправа», показали необходимость применения на судне активных фильтро-компенсирующих устройств (ФКУ), а также модернизации систем автоматического регулирования дизель-генераторных агрегатов (САР ДГА) судна для компенсации неактивной мощности, генерируемой в СЭЭС. На пароме установлена ГЭУ постоянного тока, питающаяся от единой СЭЭС переменного тока через тиристорные преобразователи, которые являются основной нагрузкой сети. СЭЭС автопарома характеризуется как нестабильная, склонная к резонансным явлениям, качанию мощности, срыву работы систем автоматики и т. п. негативным явлениям, обусловленным значительным искажением форм токов и напряжений сети. В связи с характерными особенностями СЭЭС, применение многих хорошо зарекомендовавших себя на береговых предприятиях устройств компенсации неактивной мощности в виде пассивных резонансных фильтров, активных фильтров и т. п., является малоэффективным. В первую очередь это связано с ограниченностью мощности источников энергии СЭЭС и нестабильностью параметров электроэнергии, генерируемой ими.

В результате обширных исследований рассмотренного комплекса проблем предложен ряд инженерных решений [6–8], направленных на повышение показателей качества электроэнергии в автономных СЭЭС, основной нагрузкой которых являются полупроводниковые преобразователи, питающие мощные электроприводы, заряд-

ные устройства и т. п. судовые потребители электроэнергии. Необходимым функциональным узлом предложенных разработок, определяющим их эффективность в частности и работоспособность в общем, является устройство идентификации внешних параметров системы управления. Основная функция блока – выделение из искаженного сигнала необходимых для его анализа гармонических составляющих, определение их параметров, и использование результатов этого анализа в управлении средствами повышения значений показателей качества электроэнергии СЭЭС. При этом регулируются такие показатели, как уровни целевых гармоник токов и напряжений, значения суммарных гармонических искажений, показателей мощности искажений, реактивной мощности и т. д.

Формулирование целей. Наиболее важной задачей для ФКУ является точное определение изменяющихся гармонических составляющих несинусоидального сигнала: их величины и фазового угла. Именно это позволяет силовой части фильтра компенсировать соответствующие гармоники. Помимо быстрого преобразования Фурье (БПФ), в настоящее время активно применяются методы на основе фильтра Калмана, а также системы управления на основе искусственных нейронных сетей. Однако каждый из этих методов имеет в той или иной степени один недостаток: время адаптации к изменениям в анализируемом сигнале. В наилучшем случае, при неизменной частоте тока питающей сети и резком изменении его гармонического состава, системе управления необходимо время не менее одного периода напряжения сети. Изменение же частоты питающей сети, низкочастотные колебания (характерные для дисбаланса в трехфазной сети без нулевого провода), мощные помехи приводят, как правило, к катастрофическим последствиям. Это связано с тем, что при анализе сигнала в частотной области, изменение частоты основной гармоники на 0,4 Гц приводит к ошибке в определении амплитуды пятой гармоники более 10 %. Чтобы избавиться от этой ошибки, необходимо определять частоту основной гармоники. Для этого ис-

пользуют метод, отслеживающий частоту сигнала по его прохождению через ноль. Но даже в четырехпроводной сети этот алгоритм требует времени не менее трех периодов основной гармоники. Учитывая отсутствие в СЭЭС нулевого провода (и, как следствие, неизвестность фазных напряжений сети), искаженность линейных напряжений, а также прочих дестабилизирующих факторов, существующие системы управления обладают длительным временем регулирования, либо в принципе дают критическую ошибку.

Таким образом, при всем разнообразии существующих способов выделения гармоник мощности искажения, актуальнейшей задачей остается реализация быстрого и точного алгоритма идентификации недоступных для измерения сигналов и параметров сложной распределенной системы.

Идентификация нестационарной основной гармоники искаженного сигнала. В общей форме входной сигнал $v(t)$ системы идентификации, являющийся сигналом, пропорциональным сетевому току или напряжению СЭЭС, может быть описан функцией-рядом Фурье:

$$\hat{v}(t) = \sum_{k=1}^N A_k(t) \sin(k\omega_k(t)t + \varphi_k(t)), \quad (1)$$

где $A_k(t)$ – амплитуда; $\omega_k(t)$ – частота и $\varphi_k(t)$ – фазовый угол k -й гармоники; N – количество гармоник (в общем случае $N = \infty$).

Так как в настоящей статье рассмотрено решение задачи идентификации нестационарных параметров основной гармоники входного сигнала, то искомая функция будет иметь вид

$$y(t) = A(t) \sin(\omega(t)t + \varphi(t)), \quad (2)$$

где $A(t) = A_1(t)$ – амплитуда; $\omega(t) = \omega_1(t)$ – частота и $\varphi(t) = \varphi_1(t)$ – фазовый угол искомой функции.

Введем вектор параметров $\phi(t) = [A(t), \omega(t), \varphi(t)]^T$, который принадлежит пространству параметров $\Phi(t) = \{[A, \omega, \varphi]^T\}$.

Ошибка аппроксимации характеризуется выражением

$$\varepsilon_n = y_n - \hat{v}_n. \quad (3)$$

Энергетическая функция ошибки, выражаемая среднеквадратичной погрешностью:

$$E_n = \frac{1}{2} \varepsilon_n^2. \quad (4)$$

Функция (4) является функцией трех переменных $E(A, \omega, \varphi)$ или $E(\Phi)$. Задачей аппроксимации является поиск такого алгоритма изменения элементов вектора Φ , при котором происходит минимизация функции (4). То есть поиск такого значения $\hat{\Phi}$ вектора Φ , которому соответствует экстремальное значение функции E (минимум), когда будет выполняться условие

$$E(\phi) \geq E(\hat{\phi}); \forall E \in \Phi, \quad (5)$$

где Φ – допустимая область или область возможных значений вектора ϕ , определяемая ограничениями

$$A \in [A_{\min}, A_{\max}], \quad \omega \in [\omega_{\min}, \omega_{\max}] \quad (*)$$

$$\text{и } \varphi \in [\varphi_{\min}, \varphi_{\max}].$$

На данном этапе мы подошли к задаче поиска экстремума (минимума) энергетической функции (4). Функция является дифференцируемой по всем элементам, но выражение (2) предполагает нелинейное преобразование элементов входного вектора Φ . Последнее обстоятельство ставит под вопрос возможность применения градиентных алгоритмов, наиболее зарекомендовавших себя в экстремальных системах.

При использовании метода градиента, вектор параметров ϕ вычисляется методом градиентного спуска по правилу (6). Задаваясь некоторым ϕ , для данной точки вычисляются градиент энергетической функции (4). Затем каждый компонент вектора ϕ изменяют со скоростью, пропорциональной составляющей градиента (6) по этому компоненту:

$$\frac{d\phi(t)}{dt} = -\mu \text{grad}_{\phi(t)} E(\phi), \quad (6)$$

где $\text{grad}_{\phi(t)} E(\Phi) = \frac{\partial [E(t, \phi(t))]}{\partial \phi(t)}$; μ – диагональная матрица постоянных коэффициентов скорости аппроксимации, которые при поиске минимума должны удовлетворять условию $\mu_i > 0$.

При использовании метода наискорейшего спуска градиент измеряется непрерывно в процессе изменения ϕ , так что в каждый момент скорость изменения вектора ϕ пропорциональна градиенту. В экстремальной точке $\hat{\phi}$ все компоненты вектора градиента обращаются в нуль. Метод градиентного спуска гарантирует достижение экстремума при квадратичной форме энергетической функции. Для неквадратичных функций (каковой является и рассматриваемая в настоящей статье энергетическая функция) необходимо математически доказывать сходимость метода к точке экстремума.

Для нахождения экстремума функций, не являющихся квадратичными, используют методы сопряженных градиентов. Существенным недостатком данных методов является их чувствительность к погрешностям счета. Для обеспечения сходимости при этом применяется периодический рестарт алгоритма, метод Флетчера–Ривса, который при неквадратичности целевой функции из конечного становится итеративным, либо метод Полака–Рибьера с различными модификациями, направленными на гарантирование сходимости. Перечисленные методы итеративны, т. е. не обеспечивают гарантированной сходимости за конечное

число шагов и требуют значительной вычислительной мощности при практической реализации (особенно, когда необходимо выделять несколько гармонических составляющих). На рис. 1 приведены результаты моделирования на коротком интервале при использовании градиентного метода в поставленной задаче. На рис. 2 приведены результаты моделирования при использовании градиентного метода в поставленной задаче на большом интервале. Анализ результатов применения распространенных модификаций градиентных методов при решении поставленной задачи, аналогично рис. 2 демонстрирует расхождение процесса.

Рассмотрим математические аспекты возможности применения градиентных методов в поставленной задаче. Выпишем градиенты для восстановленных параметров вектора ϕ :

$$\begin{aligned} \frac{d\hat{A}(t)}{dt} &= -\mu_A \frac{\partial[E(t, \phi(t))]}{\partial\hat{A}(t)}; \\ \frac{d\hat{\omega}(t)}{dt} &= -\mu_\omega \frac{\partial[E(t, \phi(t))]}{\partial\hat{\omega}(t)}; \\ \frac{d\hat{\phi}(t)}{dt} &= -\mu_\phi \frac{\partial[E(t, \phi(t))]}{\partial\hat{\phi}(t)}. \end{aligned} \quad (7)$$

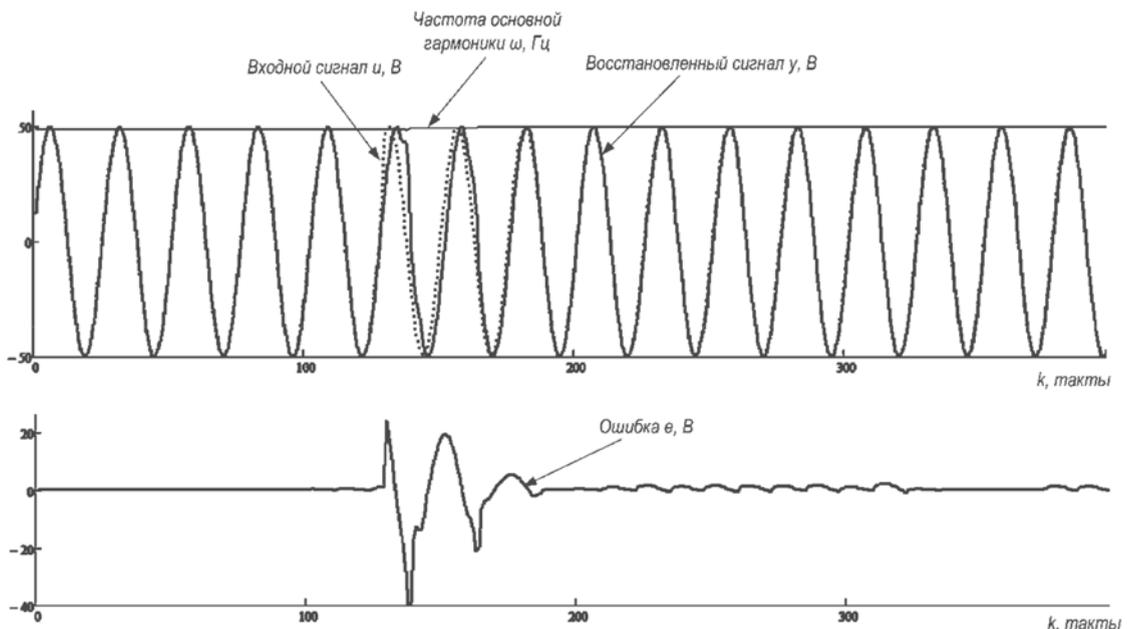


Рис. 1. Отслеживание основной гармоники $y(k)$ искаженного сигнала $v(k)$ с изменяющейся частотой $\omega(k)$ и ошибка слежения $\varepsilon(k)$

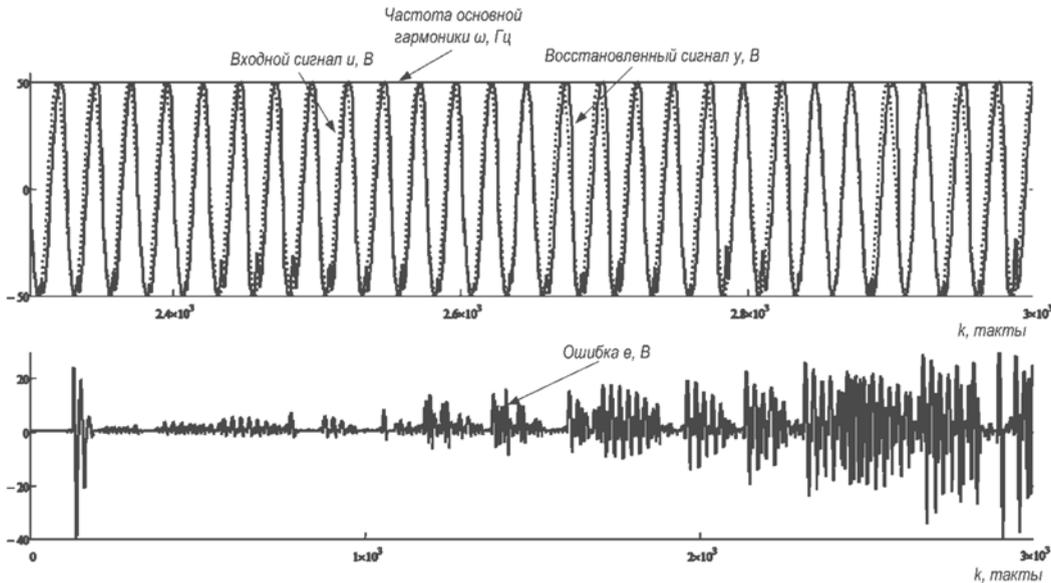


Рис. 2. Расхождение процесса отслеживания основной гармоники $y(k)$ искаженного сигнала $v(k)$ с изменяющейся частотой $\omega(k)$ на большом временном интервале

Энергетическую функцию представим как

$$E(t, \phi(t)) = \frac{1}{2} \varepsilon(t, \phi(t))^2 = \frac{1}{2} [v(t) - \hat{A}(t) \sin(\hat{\omega}(t)t + \hat{\phi}(t))]^2.$$

Подставив ее в (7) и продифференцировав правую часть, получим:

$$\begin{aligned} \frac{d\hat{A}(t)}{dt} &= -\mu_A \varepsilon(t) \sin(\hat{\omega}(t)t + \hat{\phi}(t)); \\ \frac{d\hat{\omega}(t)}{dt} &= -\mu_\omega \varepsilon(t) \hat{A}(t) t \cos(\hat{\omega}(t)t + \hat{\phi}(t)); \quad (8) \\ \frac{d\hat{\phi}(t)}{dt} &= -\mu_\phi \varepsilon(t) \hat{A}(t) \sin(\hat{\omega}(t)t + \hat{\phi}(t)). \end{aligned}$$

Как видно, во втором уравнении системы (8) появляется параметр времени вне знака косинуса. Таким образом, для данной энергетической функции выражение градиента по угловой скорости содержит комбинированный, т. е. секулярный (вековой) и нелинейный одновременно, член. При применении методов Ньютона–Рафсона (или подобных), использующих гессиан, таких членов появится несколько.

Этот факт объясняет неудачи в попытках реализовать подобные алгоритмы на основании общих подходов адаптивных систем и градиентных методов в частно-

сти, практическая реализация которых, в описываемых применениях, сталкивается с расходимостью процесса при отслеживании изменения фазы и частоты сигнала. Результаты моделирования также показали, что сходимость градиентных алгоритмов в данном случае критически зависит от значений коэффициентов скорости аппроксимации μ , малейшее отклонение которых от оптимума приводит к немедленному расхождению процесса. Аналогично ведет себя алгоритм и при изменении параметров аппроксимируемого сигнала.

Решение поставленной задачи. Так как частота $\omega(t)$ и фазовый угол $\phi(t)$ искомой функции переменны, то перейдем к понятию полной фазы $\psi(t) = \omega(t)t + \phi(t)$, учитывающей все изменения фазы функции. Тогда систему (8) приведем к виду:

$$\begin{aligned} \frac{d\hat{A}(t)}{dt} &= -\mu_A \varepsilon(t) \sin \hat{\psi}(t); \\ \frac{d\hat{\omega}(t)}{dt} &= -\mu_\omega \varepsilon(t) \hat{A}(t) t \cos \hat{\psi}(t); \quad (9) \\ \frac{d\hat{\psi}(t)}{dt} &= \hat{\omega}(t) + \mu_\psi \frac{d\hat{\omega}(t)}{dt}. \end{aligned}$$

Энергетическую функцию представим как $E(t) = \frac{1}{2} [u(t) - \hat{A}(t) \sin \hat{\psi}(t)]^2$, то есть

для ошибки аппроксимации будем иметь $\varepsilon(t) = u(t) - \hat{A}(t) \sin \hat{\psi}(t)$.

При этом входной сигнал системы можно представить как $v(t) = A(t) \sin \psi(t) + \zeta(t)$.

Идея решения задачи заключается в замене переменной времени на постоянный коэффициент α , а изменяющихся во времени параметров искомой функции – на постоянные величины. Аналогичный подход используется в теории гармонической линеаризации или стационаризации. В нашем случае замена опирается на предположение, что в данный момент времени входной сигнал содержит синусоидальную составляющую с постоянной угловой частотой ω_0 и дополнительными гармоническими составляющими. Причем в окрестности (*) не имеется резонансных частот, что очевидно исходя из формы выражения (1). Аналогично происходит замена начальной фазы искомой составляющей. И в конечном итоге полная фаза будет представлена выражением $\psi_0(t) = \omega_0 t + \varphi_0$. Таким образом, мы рассматриваем входной сигнал как сумму периодической функции с фиксированным периодом T_0 и дополнительных периодических составляющих (ξ), не имеющих частотных составляющих в данном периоде.

Опуская математические преобразования, система для отыскания параметров

восстанавливаемого сигнала принимает следующий вид:

$$\begin{aligned} \frac{d\hat{A}(t)}{dt} &= \alpha_A \varepsilon(t) \int_{t_0}^{t_0+T_0} \sin \hat{\psi}(t) dt; \\ \frac{d\hat{\omega}(t)}{dt} &= \alpha_\omega \varepsilon(t) \hat{A}(t) \int_{t_0}^{t_0+T_0} \cos \hat{\psi}(t) dt; \\ \frac{d\hat{\psi}(t)}{dt} &= \hat{\omega}(t) + \alpha_\psi \frac{d\hat{\omega}(t)}{dt}. \end{aligned} \quad (10)$$

Моделирование по предложенному алгоритму показало сходимость решения без каких-либо ограничений по интервалу моделирования. Результаты моделирования представлены на рис. 3 и 4. Являясь компромиссом между скоростью сходимости и точностью, коэффициент α определяет основные свойства предложенного алгоритма в каждом конкретном применении. Из результатов моделирования видно, что при изменении частоты и начальной фазы отслеживаемого сигнала, ошибка превысила значение, полученное на рис. 1. Однако в установившемся режиме ошибка не превышает 0,01 %, что на несколько порядков превысило точность слежения, приведенную в результатах моделирования на рис. 2.

Очевидно, что дальнейшее развитие предложенного алгоритма может идти по пути реализации адаптивного подбора ко-

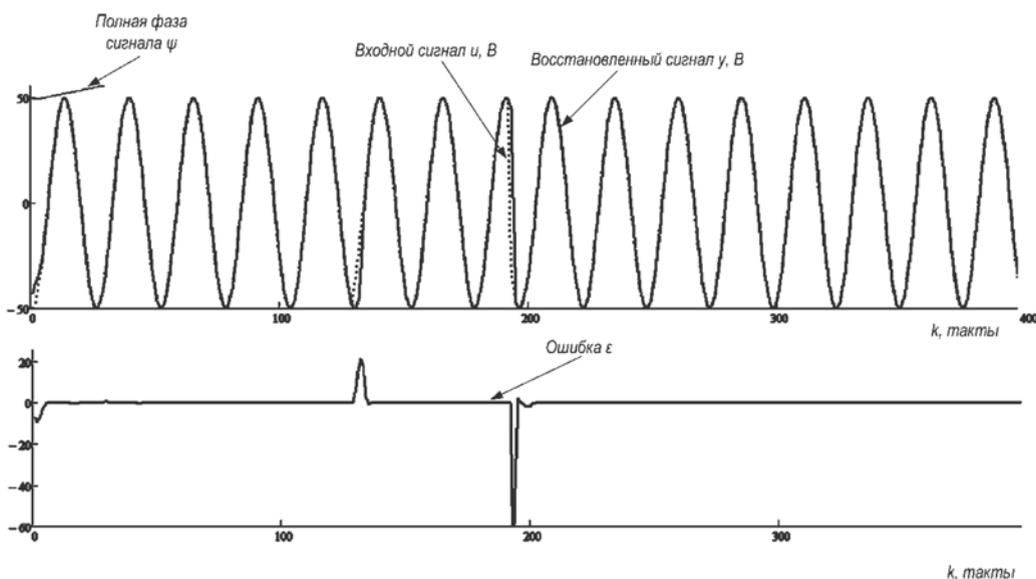


Рис. 3. Отслеживание основной гармоники $y(k)$ искаженного сигнала $v(k)$ с изменяющейся частотой $\omega(k)$ и ошибка слежения $\varepsilon(k)$ по предложенному алгоритму

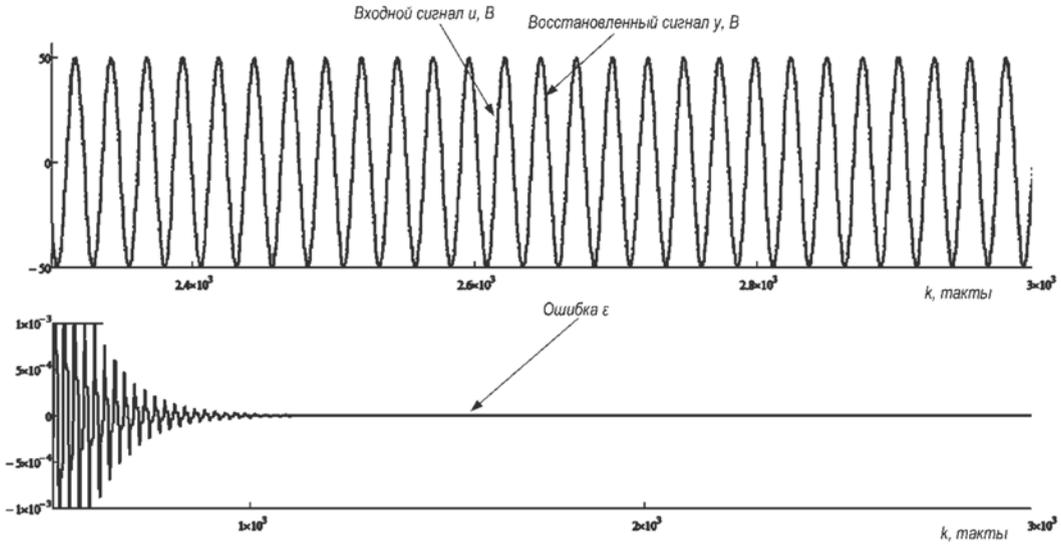


Рис. 4. Процесс отслеживания основной гармоники $u(k)$ искаженного сигнала $v(k)$ на большом временном интервале

эффицентом α в зависимости от текущего значения ошибки ϵ . Это позволит уменьшить ошибку в переходных процессах, не ухудшив показателей качества системы в установившихся.

Математически сходимость, стабильность и уникальность решения предложенного алгоритма может быть доказана с использованием теоремы карт Пуанкаре или доказательств, применяемых при использовании методов гармонической линеаризации, например, метода Крылова—

Боголюбова. Проведенные исследования показали, что система (10) представляет уникальную асимптотическую траекторию, периодическую и лежащую в окрестности аппроксимируемой функции основной гармоники сигнала.

Действительно, если преобразовать систему (10) соответствующим образом, то мы получим выражение вида $\psi(t) = q(\Phi)\phi(t) + q'(\Phi)\phi'(t)$, на что указывают синусные и косинусные составляющие в уравнениях системы. Данное выражение

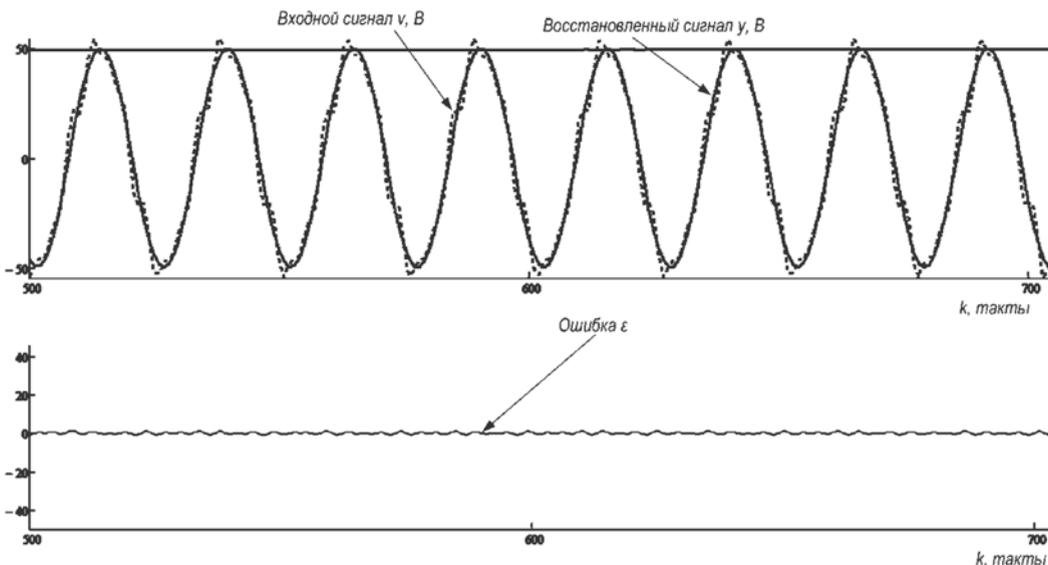


Рис. 5. Аппроксимация основной гармоники $u(k)$ искаженного сигнала $v(k)$ при наличии в нем пятой и седьмой гармоник



описывает траекторию динамической системы (10) вокруг фиксированной точки, находящейся в окрестности (*). А коэффициент α определяет скорость ее движения из точки с координатами $(A_0, \omega_0, \varphi_0)$ в точку $(A_1, \omega_1, \varphi_1)$.

На рис. 5 приведены результаты моделирования процесса слежения за основной гармоникой сигнала $v(t)$, при наличии в нем пятой и седьмой гармоник с уровнями, соответственно, 10 и 5 %.

Предложенный метод слежения за параметрами основной (в принципе, любой) гармоники искаженного сигнала позволяет обеспечить единственное, стабильное и схо-

дящееся решение нелинейной динамической системы. Полученный алгоритм обеспечивает уникальную асимптотическую траекторию, являющуюся периодической и лежащей в окрестности аппроксимируемой функции целевой гармоники сигнала. При этом в установившемся режиме абсолютная ошибка аппроксимации не превышает 0,01 %.

Разработанный метод может с успехом использоваться при построении многомерных систем идентификации внешних параметров как подсистемы экспертных блоков в автоматизированных системах управления качеством электроэнергии на водном транспорте и в других автономных электроэнергетических системах.

СПИСОК ЛИТЕРАТУРЫ

1. **Жиленков А.А.** Влияния мощных тиристорных выпрямителей на питающую их автономную электростанцию // Восточно-Европейский журнал передовых технологий. Харьков, 2012. Вып. 5/8(59). С. 14–19.

2. **Жиленков А.А.** Определение критериев оценки устойчивости работы управляемого выпрямителя в сети ограниченной мощности // Матер. V междунар. науч.-практич. конф. Современные информационные и инновационные технологии на транспорте. Херсон, 2013. Т. 2. С. 108–109.

3. **Жиленков А.А.** Анализ работы тиристорного привода постоянного тока, работающего в автономной сети // Восточно-Европейский журнал передовых технологий. Харьков, 2012. Вып. 6/8(60). С. 12–14.

4. **Жиленков А.А.** Оценка устойчивости управляемого выпрямителя в сети ограниченной мощности // Матер. II междунар. науч.-практич. конф. Обчислювальний інтелект (результати, проблеми, перспективи). Черкаassy, 2013. С. 358–359.

5. ГОСТ 13109-97. Электрическая энергия. Совместимость технических средств электромагнитная. Нормы качества электрической энергии в системах электроснабжения общего назначения. Минск: Межгосударственный Совет по стандартизации, метрологии и сертификации, 1998. 58 с.

6. **Жиленков А.А., Черный С.Г.** Адаптивная система управления активного фильтра с мо-

дульной топологией // Вісник Чернігівського державного технологічного університету. Сер. Технічні науки. Чернігів: Изд-во Черніг. держ. технол. ун-та, 2013. № 2 (65). С. 230–235.

7. **Жиленков А.А., Черный С.Г.** Моделирование адаптивного управления в сложных распределенных системах с идентификацией параметров // Вісник Хмельницького національного університету. 2013. № 6. С. 253–260.

8. **Zhilentkov A., Cherney S.** Adaptive control in complex distributed systems with the identification of parameters that can't be measured // Pressing issues and priorities in development of the scientific and technological complex: research articles. San Francisco, California: B&M Publishing, 2013. Pp. 17–20.

9. **Gyugyi L., Strycula E.** Active AC Power Filter // IEEE Trans, on Industry Applications. 1976. Pp. 529–535.

10. **Jintakosonwit P., Fujita H., Akagi H.** Control and Performance of a Fully-Digital-Controlled Shunt Active Filter for Installation on a Power Distribution System // IEEE Trans, on Power Electronics. 2002. Vol. 17, No. 1. Pp. 132–140.

11. **Kim S., Enjeti P.N.** A New Hybrid Active Power Filter (APF) Topology // IEEE Trans, on Power Electronics. 2002. Vol. 17. No. 1. Pp. 48–54.

12. **Nunez-Zuniga T.E., Pomilio J.A.** Shunt Active Power Filter Synthesizing Resistive Loads // IEEE Trans, on Power Electronics. 2002. Vol. 17. No. 2. Pp. 273–278.

REFERENCES

1. **Zhilentkov A.A.** Vliyaniya moshchnykh tiristornykh vypryamiteley na pitayushchuyu ikh avtonomnyuyu elektrostantsiyu, *Vostochno-Yevropeyskiy zhurnal peredovykh tekhnologiy*, Kharkov, 2012, No. 5/8(59), Pp. 14–19.

2. **Zhilentkov A.A.** Opredeleniye kriteriyev otsenki ustoychivosti raboty upravlyayemogo vypryamitelya v seti ogranichennoy moshchnosti, *Materialy V mezhdunarodnoy nauchno-prakticheskoy konferentsii Sovremennyye informatsionnyye i innovatsionnyye*

tehnologii na transporte. Kherson, 2013, Vol. 2, Pp. 108–109.

3. **Zhilentov A.A.** Analiz raboty tiristornogo privoda postoyannogo toka, robotayushchego v avtonomnoy seti, *Vostochno-Yevropeyskiy zhurnal peredovykh tekhnologiy*. Kharkov, 2012, No. 6/8(60), Pp. 12–14.

4. **Zhilentov A.A.** Otsenka ustoychivosti upravlyayemogo vypryamitelya v seti ogranichennoy moshchnosti, *Obchislyvalniy intelekt (rezultati, problemi, perspektivi): materialy II mezhdunarodnoy nauchno-prakticheskoy konferentsii*, Cherkassy, 2013, Pp. 358–359.

5. GOST 13109-97. *Elektricheskaya energiya. Sovmestimost tekhnicheskikh sredstv elektromagnitnaya. Normy kachestva elektricheskoy energii v sistemakh elektrosnabzheniya obshchego naznacheniya*. Minsk: Mezhdunarodnyy Sovet po standartizatsii, metrologii i sertifikatsii, 1998, 58 p.

6. **Zhilentov A.A., Cherny S.G.** Adaptivnaya sistema upravleniya aktivnogo filtra s modulnoy topologiyey, *Visnik Chernigivskogo derzhavnogo tekhnologichnogo universitetu. Ser. Tekhnichni nauki*, Chernigiv: Chernig. derzh. tekhnol. un-t Publ., 2013, No. 2(65), Pp. 230–235.

7. **Zhilentov A.A., Cherny S.G.** Modelirovaniye

adaptivnogo upravleniya v slozhnykh raspredelennykh sistemakh s identifikatsiyey parametrov, *Visnik Khmel'nitskogo natsionalnogo universitetu*, 2013, No. 6, Pp. 253–260.

8. **Zhilentov A., Cherny S.** Adaptive control in complex distributed systems with the identification of parameters that can't be measured, *Pressing issues and priorities in development of the scientific and technological complex: research articles*, B&M Publishing, San Francisco, California, 2013, Pp. 17–20.

9. **Gyugyi L., Strycula E.** Active AC Power Filter, *IEEE Trans, on Industry Applications*, 1976, Pp. 529–535.

10. **Jintakosonwit P., Fujita H., Akagi H.** Control and Performance of a Fully-Digital-Controlled Shunt Active Filter for Installation on a Power Distribution System, *IEEE Trans, on Power Electronics*, 2002, Vol. 17, No. 1, Pp. 132–140.

11. **Kim S., Enjeti P.N.** A New Hybrid Active Power Filter (APF) Topology, *IEEE Trans, on Power Electronics*, 2002, Vol. 17, No. 1, Pp. 48–54.

12. **Nunez-Zuniga T.Ye., Pomilio J.A.** Shunt Active Power Filter Synthesizing Resistive Loads, *IEEE Trans, on Power Electronics*, 2002, Vol.17, No.2, Pp. 273–278.

ЧЕРНЫЙ Сергей Григорьевич — доцент кафедры электрооборудования судов и автоматизации производства Керченского государственного морского технологического университета, кандидат технических наук.

298300, Россия, Крым, г. Керчь, ул. Орджоникидзе, д. 82.

E-mail: sergiiblack@gmail.com

CHERNY, Sergey G. *Kerch State Marine Technological University.*

298300, Orgenikidse Str. 82, Kerch, Russia.

E-mail: sergiiblack@gmail.com

ЖИЛЕНКОВ Антон Александрович — старший преподаватель Керченского государственного морского технологического университета.

298300, Россия, Крым, г. Керчь, ул. Орджоникидзе, д. 82.

E-mail: zhilentovanton@gmail.com

ZHILENKOV, Anton A. *Kerch State Marine Technological University.*

298300, Orgenikidse Str. 82, Kerch, Russia.

E-mail: zhilentovanton@gmail.com

УДК 004.932.2:519.254

Л.А. Демидова, Н.И. Нестеров, Р.В. Тишкин

**ВОЗМОЖНОСТНО-НЕЧЕТКАЯ СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ
ЗЕМНОЙ ПОВЕРХНОСТИ С ПРИМЕНЕНИЕМ ГЕНЕТИЧЕСКИХ
АЛГОРИТМОВ И ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ**

L.A. Demidova, N.I. Nesterov, R.V. Tishkin

**POSSIBILISTIC-FUZZY SEGMENTATION
OF EARTH SURFACE IMAGES BY MEANS OF GENETIC ALGORITHMS
AND ARTIFICIAL NEURAL NETWORKS**

Рассмотрена задача сегментации изображений земной поверхности с применением интеллектуальных информационных технологий. Показана целесообразность применения генетических алгоритмов для получения субоптимальных результатов сегментации изображений с использованием алгоритмов кластеризации в условиях неопределенности. Предложено для уточнения результатов сегментации, полученных с применением алгоритмов кластеризации в условиях неопределенности, использовать инструментальный искусственных нейронных сетей.

ДИСТАНЦИОННОЕ ЗОНДИРОВАНИЕ ЗЕМЛИ; ГИПЕРСПЕКТРАЛЬНАЯ СЪЁМКА; СЕГМЕНТАЦИЯ; КЛАСТЕРИЗАЦИЯ; АЛГОРИТМ НЕЧЕТКИХ С-СРЕДНИХ; ГЕНЕТИЧЕСКИЙ АЛГОРИТМ; ИСКУССТВЕННАЯ НЕЙРОННАЯ СЕТЬ.

We have focused on the problem of image segmentation of earth surface using intelligent information technologies. The application expediency of genetic algorithms to receive suboptimum results of the image segmentation on the basis of clustering algorithms under uncertainty has been shown. We have offered to use artificial neural networks to specify segmentation results received on the base of clustering algorithms under uncertainty.

EARTH REMOTE SENSING; HYPER SPECTRAL SHOOTING; SEGMENTATION; CLUSTERING; FUZZY C-MEANS ALGORITHM; GENETIC ALGORITHM; ARTIFICIAL NEURAL NETWORK.

Сегментация изображений — одна из важных задач, решаемых системами дистанционного зондирования Земли, применяется для реализации распознавания сцен и заключается в упрощении описания изображений посредством перехода от представления в виде множества точек к представлению в виде множества классов объектов. Этот переход основан на выделении областей, имеющих схожие визуальные характеристики.

В последнее время особый интерес

представляет решение задачи сегментации гиперспектральных изображений (ГСИ), получаемых от систем дистанционного зондирования Земли в процессе гиперспектральной съемки, предполагающей одновременное формирование сотен и тысяч изображений одной и той же сцены, зафиксированных в очень узких соприкасающихся диапазонах спектра. При этом ГСИ представляется как куб данных, включающий в себя пространственную информацию (2D) об объекте, дополненную

спектральной информацией (1D) по каждой пространственной координате. Таким образом, каждой точке ГСИ соответствует спектр, полученный в этой точке анализируемой сцены.

Одной из главных задач, возникающих при сегментации ГСИ, является задача распознавания принадлежности тому или иному известному классу выявленных неопределенных зон местности. Как правило, для решения подобных задач распознавания используются технологии, основанные на применении аппарата линейного дискриминантного анализа [1], алгоритма AdaBoost [2], метода опорных векторов [2], инструментария искусственных нейронных сетей [3]. Однако, как показывает анализ, известные на данный момент технологии распознавания неопределенных зон местности являются недостаточно эффективными при решении задачи сегментации ГСИ [4].

В связи с этим можно говорить о целесообразности разработки новой интеллектуальной технологии сегментации ГСИ, основанной на комплексном использовании алгоритмов кластеризации в условиях неопределенности, генетических алгоритмов (ГА) и искусственных нейронных сетей (ИНС).

Так как задача распознавания неопределенных зон местности на ГСИ предполагает проведение анализа данных в условиях неопределенности, то для ее решения обоснованным является использование т. н. *алгоритмов кластеризации в условиях неопределенности* (в частности, FCM-, РСМ- и PFCM-алгоритмов) [4–10], применение которых позволит адекватно разделить спектры различных поверхностей местности, соответствующие материалам искусственного и природного происхождения. При этом совместное использование алгоритмов кластеризации в условиях неопределенности и ГА обеспечит получение субоптимального решения о результатах сегментации, которое в дальнейшем может быть уточнено с использованием инструментария ИНС.

Теоретическая часть. Пусть R – множество точек ГСИ, подлежащих сегментации.

Каждой такой точке соответствует вектор числовых значений спектральных характеристик $r_i = (r_i^1, r_i^2, \dots, r_i^l)$, где r_i^j – числовое значение j -й спектральной характеристики для i -й точки ($i = \overline{1, n}; j = \overline{1, l}$).

В качестве функции расстояния $d(r_i, r_q)$ в l -мерном пространстве спектральных характеристик может использоваться функция, основанная на вычислении евклидовой метрики:

$$d(r_i, r_q) = \sqrt{\sum_{j=1}^l (r_i^j - r_q^j)^2}. \quad (1)$$

Поскольку задаче сегментации ГСИ присуща неопределенность, обусловленная невозможностью однозначной классификации изображенных объектов, то целесообразно применять алгоритмы кластеризации, основанные на применении инструментария теории нечетких множеств, реализующие ту или иную интерпретацию неопределенности и допускающие принадлежность точки ГСИ к нескольким кластерам одновременно, что обеспечивает гибкость анализа ГСИ. Наиболее известны такие алгоритмы кластеризации в условиях неопределенности, как алгоритм нечетких c -средних (FCM-алгоритм, fuzzy c -means), алгоритм вероятностных c -средних (PCM-алгоритм, possibilistic c -means) и алгоритм вероятностно-нечетких c -средних (PFCM-алгоритм, possibilistic fuzzy c -means).

FCM-алгоритм реализует нечеткую интерпретацию неопределенности и является итерационным алгоритмом, вычисляющим значения функций принадлежности точек ГСИ кластерам и координаты центров кластеров в соответствии со значениями функций принадлежности [6, 8, 11].

FCM-алгоритм выполняет минимизацию целевой функции:

$$J(U, V) = \sum_{k=1}^c \sum_{i=1}^n (u_k(r_i))^m \cdot d^2(v_k, r_i) \quad (2)$$

при

$$\sum_{k=1}^c u_k(r_i) = 1 \quad (c \in N \text{ и } c > 1; i = \overline{1, n}), \quad (3)$$

где $U = [u_k(r_i)]$ – нечеткое c -разбиение множества точек ГСИ R на основе функций принадлежности $u_k(r_i)$, определяющих сте-

пень принадлежности i -й точки ГСИ k -му кластеру; $V = (v_1, \dots, v_c)$ – центры кластеров; $d(v_k, r_i)$ – расстояние между центром кластера v_k и точкой ГСИ r_i в соответствии с формулой (1); m – фаззификатор ($m \in R, m > 1$); c – количество кластеров; n – количество точек ГСИ; $i = \overline{1, n}$; $k = \overline{1, c}$.

Функции принадлежности $u_k(r_i)$ играют роль весовых коэффициентов, определяя степень принадлежности i -й точки ГСИ k -му кластеру, а также вклад i -й точки ГСИ в оценку координат центра k -го кластера.

ФСМ-алгоритм предполагает выполнение следующих шагов [6, 8, 11].

Шаг 1. Инициализация начального нечеткого разбиения $U = [u_k(r_i)]$, удовлетворяющего условию (3).

Шаг 2. Вычисление координат центров кластеров:

$$v_k^j = \frac{\sum_{i=1}^n u_k(r_i)^m \cdot r_i^j}{\sum_{i=1}^n u_k(r_i)^m}. \quad (4)$$

Шаг 3. Вычисление новых значений функций принадлежности:

$$u_k(r_i) = \frac{1}{\sum_{t=1}^c \left(\frac{d(v_k, r_i)}{d(v_t, r_i)} \right)^{\frac{2}{m-1}}}. \quad (5)$$

Шаг 4. Шаги 2 и 3 повторяются до тех пор, пока не будет выполнено заданное количество итераций s или не будет достигнута заданная точность $|J(U, V) - J'(U, V)| \leq \varepsilon$, где $J(U, V)$, $J'(U, V)$ – значения целевой функции на двух последовательных итерациях.

ФСМ-алгоритм не всегда точно оценивает координаты центров кластеров по причине используемого в нем ограничения (3), устанавливающего, что степени принадлежности кластерам для любой i -й точки ГСИ в сумме должны давать единицу [5, 7]. Поэтому атипичные точки ГСИ (точешумы), расположенные на одинаковом расстоянии от реальных центров кластеров, могут сильно влиять на оценки координат центров кластеров и, следовательно, на окончательный результат кластеризации. Для решения проблемы, связанной с нега-

тивным влиянием атипичных точек ГСИ, следует отказаться от ограничения (3) и модифицировать целевую функцию.

Для уменьшения влияния атипичных точек ГСИ на результаты сегментации может использоваться РСМ-алгоритм [12], реализующий возможность интерпретацию неопределенности и являющийся итерационным алгоритмом, вычисляющим значения функций типичности точек ГСИ кластерам и координаты центров кластеров в соответствии со значениями функций типичности.

РСМ-алгоритм выполняет минимизацию целевой функции [6, 8, 12]:

$$J(W, V) = \sum_{k=1}^c \sum_{i=1}^n (w_k(r_i))^m \cdot d^2(v_k, r_i) + \sum_{k=1}^c \eta_k^2 \cdot \sum_{i=1}^n (1 - w_k(r_i))^m, \quad (6)$$

где $W = [w_k(r_i)]$ – возможностное c -разбиение множества точек ГСИ R на основе функций типичности $w_k(r_i)$, определяющих степень типичности i -й точки ГСИ k -му кластеру; $V = (v_1, \dots, v_c)$ – центры кластеров; $d(v_k, r_i)$ – расстояние между центром кластера v_k и точкой ГСИ r_i в соответствии с формулой (1); m – фаззификатор ($m \in R, m > 1$); η_k ($k = \overline{1, c}$) – «ширина зоны», определяющая расстояние, на котором значение функции типичности точки ГСИ k -му кластеру равно 0,5; c – количество кластеров; n – количество точек ГСИ; $i = \overline{1, n}$; $k = \overline{1, c}$.

В РСМ-алгоритме отбрасывается ограничение (3), что снижает влияние точешумов ГСИ, расположенных на равном расстоянии от центров кластеров, на результаты кластеризации.

Функции типичности могут быть вычислены как [6, 8, 12]

$$w_k(r_i) = \frac{1}{1 + \left(\frac{d(v_k, r_i)}{\eta_k} \right)^{\frac{2}{m-1}}}, \quad (7)$$

при этом для каждого k -го кластера ($k = \overline{1, c}$) должно выполняться условие:

$$\sum_{i=1}^n w_k(x_i) = 1. \quad (8)$$

В РСМ-алгоритме координаты центров кластеров находятся как

$$v_k^j = \frac{\sum_{i=1}^n w_k(r_i)^m \cdot r_i^j}{\sum_{i=1}^n w_k(r_i)^m}. \quad (9)$$

Так как обычно РСМ-алгоритм инициализируется с помощью одного шага РСМ-алгоритма, то значения «ширины зоны» η_k ($k = \overline{1, c}$) могут быть определены как [6, 8, 12]

$$\eta_k = \left(\frac{\sum_{i=1}^n (u_k(r_i))^m \cdot d^2(v_k, r_i)}{\sum_{i=1}^n (u_k(r_i))^m} \right)^{0.5}. \quad (10)$$

Применение РСМ-алгоритма является эффективным, если множество точек ГСИ содержит точки-шумы, что является характерным для ГСИ. Однако применение РСМ-алгоритма может привести к формированию совпадающих кластеров, если центры кластеров (и, следовательно, спектральные характеристики выделяемых классов объектов) близки друг к другу, т. к. функции типичности точек ГСИ некоторому кластеру не зависят от расстояний точек ГСИ до центров других кластеров. В этом случае целесообразным является использование PFCM-алгоритма, реализующего одновременный учет свойств кластерной относительности и кластерной типичности точек ГСИ [6, 13].

PFCM-алгоритм реализует возможно-нечеткую интерпретацию неопределенности и является итерационным алгоритмом, вычисляющим значения функций принадлежности (5) и типичности (7) точек ГСИ кластерам, а также координаты центров кластеров в соответствии со значениями функций принадлежности и типичности.

PFCM-алгоритм выполняет минимизацию целевой функции [6, 13]:

$$J(U, W, V) = \sum_{k=1}^c \sum_{i=1}^n ((u_k(r_i))^m + (w_k(r_i))^{m'}) \cdot d^2(v_k, r_i) \rightarrow \min, \quad (11)$$

где $U = [u_k(r_i)]$ – нечеткое c -разбиение

множества точек ГСИ R на основе функций принадлежности $u_k(r_i)$, определяющих степень принадлежности i -й точки ГСИ k -му кластеру; $W = [w_k(r_i)]$ – возможностное c -разбиение множества точек ГСИ R на основе функций типичности $w_k(r_i)$, определяющих степень типичности i -й точки ГСИ k -му кластеру; $V = (v_1, \dots, v_c)$ – центры кластеров; $d(v_k, r_i)$ – расстояние между центром кластера v_k и точкой ГСИ r_i в соответствии с формулой (1); m и m' – фаззификаторы ($m, m' \in R$; $m, m' > 1$; обычно $m = m' = 2$); c – количество кластеров; n – количество точек ГСИ; $i = \overline{1, n}$; $k = \overline{1, c}$.

При реализации PFCM-алгоритма координаты центров кластеров находятся как [6, 13]

$$v_k^j = \frac{\sum_{i=1}^n ((u_k(r_i))^m + (w_k(r_i))^{m'}) r_i^j}{\sum_{i=1}^n ((u_k(r_i))^m + (w_k(r_i))^{m'})}, \quad (12)$$

и для каждой точки ГСИ r_i ($i = \overline{1, n}$) должно выполняться ограничение (3), а для каждого k -го ($k = \overline{1, c}$) кластера должно выполняться ограничение (8).

Как и РСМ-алгоритм, PFCM-алгоритм обычно инициализируется с помощью одного шага РСМ-алгоритма с последующим вычислением значений «ширины зоны» η_k ($k = \overline{1, c}$) по формуле (10) [6, 13], а затем осуществляется итерационное уточнение значений функций принадлежности $u_k(r_i)$, функций типичности $w_k(r_i)$ и расчет координат центров кластеров по формулам (5), (7) и (12) соответственно.

PFCM-алгоритм предполагает выполнение следующих шагов [6, 13].

Шаг 1. Инициализация начального нечеткого разбиения $U = [u_k(r_i)]$, удовлетворяющего условию (3) с использованием одного шага РСМ-алгоритма.

Шаг 2. Вычисление значений «ширины зоны» η_k ($k = \overline{1, c}$) в соответствии с формулой (10).

Шаг 3. Вычисление значений функций принадлежности и типичности в соответствии с формулами (5) и (7).

Шаг 4. Вычисление координат центров кластеров в соответствии с формулой (12).

Шаг 5. Шаги 3 и 4 повторяют-

ся до тех пор, пока не будет выполнено заданное количество итераций s или не будет достигнута заданная точность $|J(U, W, V) - J'(U, W, V)| \leq \varepsilon$, где $J(U, W, V)$, $J'(U, W, V)$ — значения целевой функции на двух последовательных итерациях.

При применении алгоритмов кластеризации в условиях неопределенности определяются локально-оптимальные разбиения точек ГСИ на кластеры, описываемые совокупностью функций принадлежности или/и функций типичности, и координаты центров кластеров. Для принятия адекватного решения о результатах кластеризации необходимо многократное выполнение данных алгоритмов при заданном количестве кластеров для различных исходных разбиений точек ГСИ на кластеры [6, 14].

Применение ГА при реализации такого алгоритма кластеризации позволяет значительно сократить время поиска субоптимального разбиения (разбиения, для которого значение функции соответствия ГА, в качестве которой используется, например, целевая функция алгоритма кластеризации, будет минимальным) на кластеры [4–8]. Так как количество кластеров обычно заранее неизвестно, то целесообразно использовать ГА с хромосомой переменной длины, что позволит одновременно найти как самое оптимальное количество кластеров, так и соответствующее ему субоптимальное разбиение на кластеры. При этом хромосому следует кодировать координатами центров всех кластеров [6], в качестве которых в контексте задачи сегментации ГСИ будут выступать спектральные характеристики эталонных классов объектов. Если искомое количество кластеров заранее неизвестно, а c_{\min} и c_{\max} — минимально и максимально возможное количество кластеров, то длина хромосомы может принимать значения от $c_{\min} \cdot l$ до $c_{\max} \cdot l$ (кратные l), где l — количество спектральных каналов точки ГСИ.

В общем случае хромосома, закодированная координатами центров кластеров, может быть представлена как

$$s = (v_1^1, v_1^2, \dots, v_1^l, \dots, v_c^1, v_c^2, \dots, v_c^l), \quad (13)$$

где v_k^j — j -я координата k -го центра кластера; c — количество кластеров; l — количе-

ство спектральных характеристик; $k = \overline{1, c}$; $j = \overline{1, l}$.

При этом первые l элементов хромосомы соответствуют координатам центра первого кластера, вторые l элементов — координатам центра второго кластера и т. п.

В контексте решения задачи сегментации ГСИ диапазон изменения значения каждого гена (некоторой координаты центра кластера) определяется интервалом $[d_{\min}, d_{\max}]$, где d_{\min} и d_{\max} — минимальное и максимальное значения яркости одного спектрального канала.

При создании начальной популяции хромосом размером G необходимо обеспечить формирование такой популяции, которая содержала бы только «жизнеспособные» хромосомы. В данном случае под «жизнеспособностью» понимается способность хромосомы обеспечить разбиение точек ГСИ на заданное количество кластеров c^* . Так как начальная популяция хромосом создается случайным образом, то могут быть сформированы хромосомы, реально обеспечивающие кластеризацию на количество кластеров \tilde{c} , меньшее, чем «назначенное» для данной хромосомы количество кластеров c^* ($\tilde{c} < c^*$; $c_{\min} \leq c^* \leq c_{\max}$). «Жизнеспособность» популяции хромосом должна поддерживаться и при смене поколений ГА. Использование популяции хромосом, состоящей только из «жизнеспособных» хромосом, позволяет повысить эффективность применения ГА.

При реализации ГА для выбора хромосом-родителей предлагается использовать процедуру вероятностного отбора [3, 6]. В контексте решения задачи сегментации ГСИ следует использовать 3–5 точек скрещивания и мутации ввиду работы с хромосомами существенно большей длины, чем при решении большинства аналогичных задач кластеризации данных, при этом значения коэффициентов скрещивания R_c и мутации R_m следует выбирать из отрезков $[0,7; 0,8]$ и $[0,01; 0,02]$ соответственно [3, 6].

В качестве условия завершения ГА наиболее часто используется условие генерации наперед заданного количества поколений W . Однако могут применяться и другие

условия завершения, в частности, условие сходимости «среднего здоровья» популяции к «лучшему (максимальному) здоровью» популяции, под которыми в данном случае понимаются соответственно усредненная сумма значений функций соответствия хромосом текущей популяции и значение функции соответствия лучшей (первой) хромосомы, являющееся минимальным в текущей популяции хромосом. В связи с тем, что могут использоваться различные условия завершения (продолжения) ГА, в дальнейшем при описании шагов реализации ГА будет говориться об этих условиях в общем виде.

В качестве функции соответствия ГА обычно выбирается целевая функция алгоритма кластеризации в условиях неопределенности (или сопоставленный этому алгоритму кластеризации показатель кластеризации, например, индекс Се–Бени для FCM-алгоритма) [6].

Комбинированный алгоритм кластеризации, реализующий совместное использование одного из алгоритмов кластеризации в условиях неопределенности и ГА, обеспечивает поиск субоптимального разбиения на кластеры и может быть описан следующей последовательностью шагов [6].

Шаг 1. Для формирования начальной популяции хромосом размером G выполняется один шаг алгоритма кластеризации в условиях неопределенности (FCM-, РСМ- или PFCM-алгоритма).

Шаг 2. При выполнении условия продолжения ГА осуществляется один шаг ГА с реализацией операций скрещивания и мутации и вычислением значений функции соответствия для хромосом популяции размером $(G + R_c \cdot G)$, состоящей из хромосом-родителей и хромосом-потомков.

Шаг 3. Для популяции размером $(G + R_c \cdot G)$ выполняется один шаг алгоритма кластеризации в условиях неопределенности (FCM-, РСМ- или PFCM-алгоритма) с вычислением значений функций принадлежности (или/и функций типичности) точек ГСИ центрам кластеров и координат центров кластеров. Затем осуществляется уточняющий пересчет значений функций принадлежности (или/и функций типично-

сти) точек ГСИ центрам кластеров и вычисление значений функции соответствия.

Шаг 4. Из расширенной популяции хромосом размером $(2G + R_c \cdot G)$, полученной путем объединения популяции размером G предыдущего поколения и популяции размером $(G + R_c \cdot G)$ текущего поколения, удаляются «нежизнеспособные» $(G + R_c \cdot G)$ хромосомы с худшими (максимальными) значениями функции соответствия. Если выполняется условие продолжения ГА, осуществляется переход к шагу 2. Если выполняется условие завершения ГА, то работа ГА завершается и осуществляется переход к шагу 5.

Шаг 5. Выбирается лучшая хромосома, которая минимизирует функцию соответствия. В качестве искомым координат центров кластеров берутся координаты центров кластеров, соответствующие лучшей хромосоме. В качестве искомым степеней принадлежности (или/и степеней типичности) точек ГСИ центрам кластеров полагаются степени принадлежности (или/и степени типичности) точек ГСИ центрам кластеров, соответствующие лучшей хромосоме.

Как показывают экспериментальные исследования, совместное использование алгоритма кластеризации в условиях неопределенности и ГА позволяет получить адекватные субоптимальные результаты кластеризации точек ГСИ с приемлемыми временными затратами.

По результатам экспериментальных исследований установлено, что в контексте решения задачи сегментации ГСИ при совместном применении алгоритмов кластеризации в условиях неопределенности и ГА в большинстве случаев для обеспечения нахождения субоптимального решения с приемлемыми временными затратами целесообразно работать с популяцией размером в 40–60 хромосом при смене 500–1000 поколений.

Сегментация точек ГСИ с использованием рассмотренных алгоритмов кластеризации в условиях неопределенности основана на предположении, что кластеры, образуемые точками ГСИ, имеют гиперсферическую форму или форму, близкую к ней. Поскольку в реальности кластеры

могут иметь произвольную форму, а в анализируемых данных могут присутствовать точки-шумы, то полученные с помощью того или иного алгоритма кластеризации в условиях неопределенности (FCM-, РСМ- или PFCM-алгоритма) результаты сегментации, даже если они признаны удовлетворительными, могут содержать те или иные ошибки относительно кластерной принадлежности некоторых точек ГСИ (не говоря уже о возможных погрешностях оценки координат центров кластеров).

В связи с этим целесообразно использовать алгоритмы уточнения результатов сегментации, полученных с помощью алгоритмов кластеризации в условиях неопределенности, для разработки которых предлагается использовать ИНС, хорошо зарекомендовавшие себя в качестве универсальных аппроксиматоров при решении различных прикладных задач.

В контексте задачи уточнения результатов сегментации предлагается на основе уже имеющихся данных о принадлежности точек ГСИ к некоторому кластеру сформировать обучающую выборку данных, выбрав в каждом кластере 50 % точек ГСИ, наиболее близко расположенных к центру кластера. Принадлежность к кластеру для таких близко расположенных к центру кластера точек ГСИ полагается эталонной. Обучив ИНС на выборке точек ГСИ, чья принадлежность к своим кластерам является эталонной, и, оценив весовые коэффициенты ИНС, можно выполнить уточнение принадлежности к кластерам для других точек ГСИ, не попавших в обучающую выборку.

При разработке ИНС предлагается использовать нейронную сеть с тремя или более слоями [3]. Использование скрытых слоев вызвано необходимостью более точного приближения набора сигналов на входе множеству сигналов на выходе. При этом увеличение количества скрытых слоев позволяет выделить статистические закономерности высших порядков. Количество входных и выходных нейронов в многослойной ИНС соответствует размерностям входного и выходного векторов соответственно.

В контексте задачи уточнения результатов сегментации размерность входного вектора равна количеству спектральных характеристик, а выходного — количеству кластеров. При этом в идеале только один выходной нейрон может иметь значение, равное единице (и именно номер этого нейрона определяет номер искомого кластера принадлежности точки ГСИ), в то время как все остальные выходные нейроны имеют значения, равные нулю. Так как на практике обычно все выходные нейроны имеют значения, отличные от нуля, то номер искомого кластера принадлежности точки ГСИ определяют по номеру выходного нейрона с максимальным вычисленным выходным значением.

Процедура функционирования ИНС может быть описана следующей последовательностью шагов.

Шаг 1. На каждый входной нейрон (вход) ИНС подается определенный сигнал, представляющий собой спектральную характеристику: яркость точки ГСИ в одном из спектральных каналов.

Шаг 2. Входные нейроны ИНС передают их нейронам скрытого слоя.

Шаг 3. Каждый нейрон скрытого слоя принимает выходные значения с предшествующего слоя и модифицирует их с учетом весовых коэффициентов, которые инициализируются случайным образом и уточняются в процессе обучения.

Шаг 4. Находится сумма скорректированных с учетом весовых коэффициентов значений, на основании которой рассчитывается выходное значение нейрона скрытого слоя с учетом вида функции активации.

Шаг 5. Посчитанные таким образом выходные значения нейронов одного скрытого слоя передаются нейронам следующего скрытого слоя.

Шаг 6. Выходные нейроны последнего (выходного) слоя формируют выход ИНС, который трактуется по правилу «победитель забирает всё», в результате применения которого точка ГСИ, чьи значения спектральных характеристик поданы на вход ИНС, интерпретируется как принадлежащая к кластеру с номером, соответствующим самому возбужденному (имею-

щему максимальное значение) выходному нейрону ИНС.

Данная процедура функционирования ИНС называется *прямым ходом* и используется для определения выходов ИНС при обучении и при функционировании обученной ИНС.

Во время обучения ИНС необходимо проводить процедуру обратного хода для изменения весов в соответствии с подсчитанным значением ошибки выхода ИНС. Модификация весов выполняется в соответствии с некоторым алгоритмом обучения, обеспечивающим уменьшение ошибки по всем выходам ИНС.

Процесс обучения ИНС в данном случае представляет собой, в частности, процедуру настройки весовых коэффициентов. При этом последовательная настройка весовых коэффициентов осуществляется при использовании данных обучающей выборки и предполагается, что реализуется парадигма обучения с учителем, когда для каждого входного примера известно значение выхода ИНС.

Таким образом, при разработке алгоритмов уточнения результатов сегментации, полученных с помощью алгоритмов кластеризации в условиях неопределенности, для каждой точки ГСИ с некоторыми известными значениями спектральных характеристик, содержащейся в обучающей выборке, должен быть известен номер кластера, которому принадлежит эта точка ГСИ. При этом веса подстраиваются так, чтобы значения выходов ИНС были как можно ближе к требуемым результатам кластеризации.

В частности, обучение ИНС может быть основано на коррекции ошибки между требуемым (обучающим) и реальным выходами сети посредством изменения весов ИНС таким образом, чтобы минимизировать эту ошибку.

Комплексное использование алгоритмов кластеризации в условиях неопределенности, ГА и ИНС при решении задачи сегментации ГСИ позволяет говорить о создании новой интеллектуальной технологии сегментации ГСИ.

Экспериментальные исследования. Аprobация предлагаемой интеллектуальной тех-

нологии сегментации ГСИ была выполнена на реальных данных, в качестве которых использовались ГСИ от гиперспектральной аппаратуры Hypersion (220 спектральных каналов), AVIRIS (224 спектральных канала) и Сокол-ГЦП (75 спектральных каналов).

При проведении экспериментальных исследований по оценке ошибок ложных идентификаций точек ГСИ в качестве эталонных изображений использовались изображения, сегментация которых была выполнена экспертом вручную.

При этом для всех рассматриваемых алгоритмов кластеризации в условиях неопределенности были получены результаты, подтверждающие целесообразность использования ГА и ИНС для повышения качества сегментации ГСИ. Однако результаты сегментации с применением PFCM-алгоритма в большинстве случаев оказались несколько лучше (на 3–7 %), результатов сегментации с применением FCM- и РСМ-алгоритмов, что объясняется реализацией одновременного сбалансированного учета свойств кластерной относительности и кластерной типичности точек ГСИ. В связи с этим ниже приведен пример, демонстрирующий эффективность использования интеллектуальной технологии сегментации ГСИ, реализующей кластеризацию точек ГСИ именно с применением PFCM-алгоритма.

На рис. 1 приведено цветосинтезированное изображение размером 550×460 пикселей, полученное на основе ГСИ от датчика AVIRIS. На рис. 2 показаны соответствующие ему результаты ручной сегментации исходного ГСИ. На рис. 3, 4 и 5 приведены соответственно результаты сегментации исходного ГСИ с применением PFCM-алгоритма, результаты сегментации с применением PFCM-алгоритма и ГА, результаты сегментации с применением PFCM-алгоритма, ГА и ИНС. Полученные результаты свидетельствуют о наличии трех кластеров, группирующих точки ГСИ в три класса: вода, растительность и земля (соответствующие трем оттенкам цвета от черного до светло-серого). Значения ошибок ложных идентификаций точек ГСИ с применением PFCM-алгоритма, с применением PFCM-алгоритма и ГА, с применением

Результаты классификации точек ГСИ

Вариант сегментации	Точки ГСИ			Точки ГСИ, классифицированные, %	
	Всего	Классифицированные		верно	ошибочно
		верно	ошибочно		
PFCM-алгоритм	253 000	217 148	35 852	85,83	14,17
PFCM-алгоритм и ГА	253 000	224 961	28 039	88,92	11,08
PFCM-алгоритм, ГА и ИНС	253 000	230 663	22 337	91,17	8,83

PFCM-алгоритма, ГА и ИНС приведены в таблице, из которой видно, что комбинирование PFCM-алгоритма с ГА, а затем использование ИНС для уточнения полученных результатов кластеризации позво-

ляет существенно снизить значения ошибок ложных идентификаций точек ГСИ.

В данном примере использование именно PFCM-алгоритма позволило получить более высокое качество сегментации (ко-

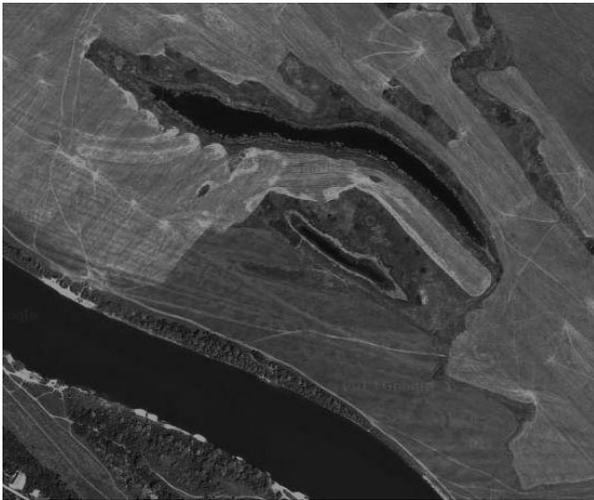


Рис. 1. Цветосинтезированное изображение района от датчика AVIRIS

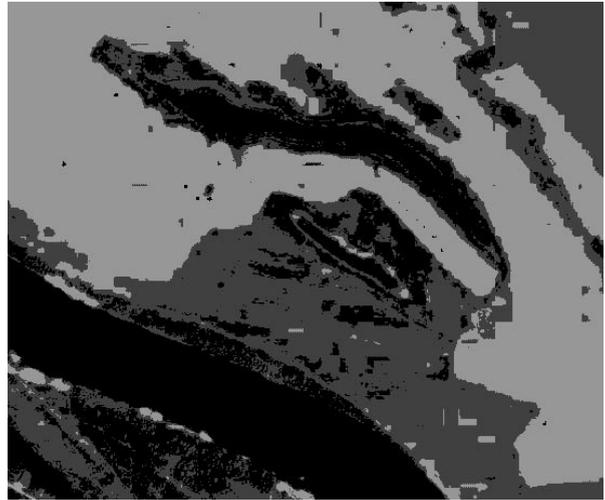


Рис. 3. Сегментация с применением PFCM-алгоритма



Рис. 2. Результаты ручной сегментации изображения

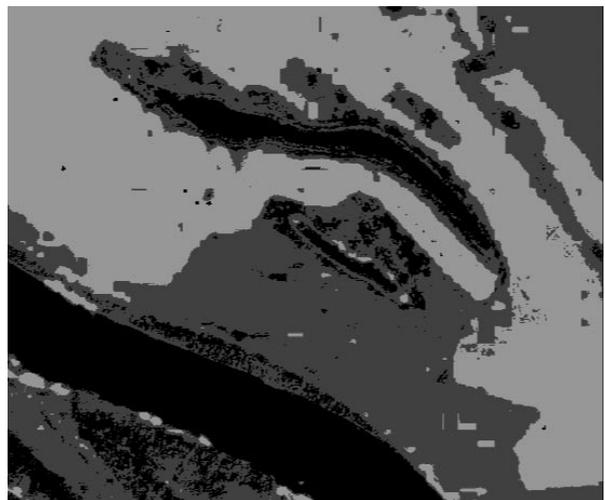


Рис. 4. Сегментация с применением PFCM-алгоритма и ГА

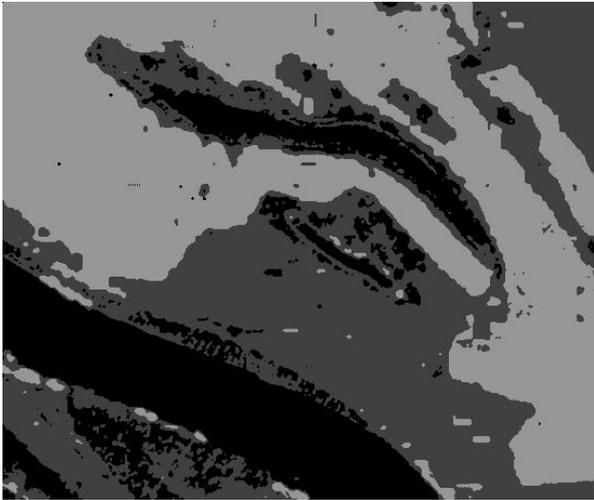


Рис. 5. Сегментация с применением PFCM-алгоритма, ГА и ИНС

личество ошибочно классифицированных точек ГСИ с использованием интеллектуальной технологии сегментации ГСИ с применением PFCM-алгоритма оказалось на 4 и 2 % меньше, чем с применением FCM- и РСМ-алгоритмов соответственно).

Тем не менее, несмотря на полученные для данного ГСИ результаты, нельзя однозначно говорить о явном преимуществе PFCM-алгоритма перед FCM- и РСМ-алгоритмами. Как показывают экспериментальные исследования, в каждом конкретном случае при сегментации того или иного ГСИ «лучшим» в смысле обеспечения более высокого качества сегментации может оказаться любой из трех рассмотренных алгоритмов кластеризации в условиях неопределенности (FCM-, РСМ- или PFCM-алгоритм), что может быть объяснено невозможностью точного

априорного выявления кластерных свойств (в частности, свойств кластерной относительности и кластерной типичности), присущих большинству точек анализируемого ГСИ. В связи с этим целесообразным является применение кластерных ансамблей [4, 15], обеспечивающих высококачественную консолидацию частных результатов сегментации ГСИ, полученных с применением алгоритмов кластеризации в условиях неопределенности (FCM-, РСМ- и PFCM-алгоритмов), ГА и ИНС.

Результаты экспериментальных исследований подтверждают эффективность интеллектуальной технологии сегментации ГСИ, основанной на комплексном использовании одного из алгоритмов кластеризации в условиях неопределенности, ГА и ИНС, и позволяют говорить о перспективности дальнейшей проработки предложенного подхода к сегментации ГСИ.

По результатам выполненных исследований предлагается использовать интеллектуальную технологию сегментации ГСИ, реализующую:

применение к ГСИ алгоритмов кластеризации в условиях неопределенности (FCM-, РСМ- и PFCM-алгоритмов) совместно с соответствующими ГА, позволяющими получить субоптимальные результаты кластеризации с приемлемыми временными затратами;

уточнение полученных с применением алгоритмов кластеризации в условиях неопределенности (FCM-, РСМ- и PFCM-алгоритмов) результатов сегментации посредством использования алгоритмов, основанных на применении ИНС.

СПИСОК ЛИТЕРАТУРЫ

1. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Классификации и снижение размерности. М.: Финансы и статистика, 1989. 607 с.
2. Вьюгин В.В. Элементы математической теории машинного обучения: учеб. пособие. М.: Изд-во МФТИ, 2010. 252 с.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы, нечеткие системы: Пер. с польск. И.Д. Рудинского. М.: Горячая линия – Телеком,

2004. 452 с.

4. Демидова Л.А., Тишкин Р.В., Юдаков А.А. Разработка ансамбля алгоритмов кластеризации на основе матриц подобия меток кластеров и алгоритма спектральной факторизации // Вестник Рязанского государственного радиотехнического университета. 2013. № 4–1 (46). С. 9–17.

5. Демидова Л.А., Кираковский В.В. Методы кластеризации объектов на основе нечетких множеств второго типа и генетического алгоритма // Научно-технические ведомости СПбГПУ.

Информатика. Телекоммуникации. Управление. 2008. № 6 (69). С. 136–142.

6. **Демидова Л.А., Кираковский В.В., Пылькин А.Н.** Принятие решений в условиях неопределенности. М.: Горячая линия – Телеком, 2012. 288 с.

7. **Демидова Л.А., Коняева Е.И.** Кластеризация объектов с использованием FCM-алгоритма на основе нечетких множеств второго типа и генетического алгоритма // Вестник Рязанского государственного радиотехнического университета. 2008. № 26. С. 46–54.

8. **Демидова Л.А., Нестеров Н.И., Тишкин Р.В.** Сегментация спутниковых изображений с применением аппарата теории нечетких множеств // Вестник Рязанского государственного радиотехнического университета. 2012. № 3(41). С. 11–17.

9. **Демидова Л. А., Титов С.Б.** Подход к проблеме нечеткой кластеризации в условиях неопределенности выбора целевой функции // Вестник Рязанского государственного радиотех-

нического университета. 2009. № 29. С. 54–60.

10. **Тишкин Р.В.** Мягкие вычисления в задачах сегментации космических изображений // Цифровая обработка сигналов. 2010. № 3. С. 25–29.

11. **Bezdek J.C., Ehrlich R., Full W.** FCM: Fuzzy C-Means Algorithm // Computers and Geoscience. 1984. No. 2. Vol. 10. Pp. 191–203.

12. **Bezdek J.C., Keller J.M., Pal N.R., Pal K.** A Possibilistic Fuzzy c-Means Clustering Algorithm // IEEE Transactions on Fuzzy Systems. 2005. Vol. 13 (4). Pp. 517–530.

13. **Krishnapuram R., Keller J.** A possibilistic approach to clustering // IEEE Transactions on Fuzzy Systems. 1993. Vol. 1. Pp. 98–109.

14. **Леоненков А.** Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2003. 736 с.

15. **Strehl A., Ghosh J.** Cluster Ensembles A Knowledge Reuse Framework for Combining Multiple Partitions // J. of Machine Learning Research. 2002. No. 3. Pp. 583–617.

REFERENCES

1. **Ayvazyan S.A., Bukhshtaber V.M., Yenyukov I.S., Meshalkin L.D.** *Prikladnaya statistika. Klassifikatsii i snizheniye razmernosti.* Moscow: Finansy i statistika Publ., 1989, 607 p. (rus)

2. **Vyugin V.V.** *Elementy matematicheskoy teorii mashinnogo obucheniya: uchebnoye posobiye.* Moscow: MFTI Publ., 2010, 252 p. (rus)

3. **Rutkovskaya D., Pilinskiy M., Rutkovskiy L.** *Neyronnyye seti, geneticheskiye algoritmy, nechetkiye sistemy,* Moscow: Goryachaya liniya – Telekom Publ., 2004, 452 p. (rus)

4. **Demidova L.A., Tishkin R.V., Yudakov A.A.** Razrabotka ansamblya algoritmov klasterizatsii na osnove matrits podobiya metok klasterov i algoritma spektralnoy faktorizatsii, *Vestnik RGRTU.* Ryazan, 2013, No. 4, Pp. 9–17. (rus)

5. **Demidova L.A., Kirakovskiy V.V.** Metody klasterizatsii obyektov na osnove nechetkikh mnozhestv vtorogo tipa i geneticheskogo algoritma, *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye,* St. Petersburg: SPbGPU Publ., 2008, No. 6(69), Pp. 136–142. (rus)

6. **Demidova L.A., Kirakovskiy V.V., Pylkin A.N.** *Prinyatiye resheniy v usloviyakh neopredelennosti.* Moscow: Goryachaya liniya – Telekom Publ., 2012, 288 p. (rus)

7. **Demidova L.A., Konyayeva Ye.I.** Klasterizatsiya obyektov s ispolzovaniyem FCM-algoritma na osnove nechetkikh mnozhestv vtorogo tipa i geneticheskogo algoritma, *Vestnik RGRTU,* Ryazan,

2008, No. 4(26), Pp. 46–54. (rus)

8. **Demidova L.A., Nesterov N.I., Tishkin R.V.** Segmentatsiya sputnikovykh izobrazheniy s primeneniym apparata teorii nechetkikh mnozhestv, *Vestnik RGRTU,* Ryazan, 2012. No. 3(41), Pp. 11–17. (rus)

9. **Demidova L.A., Titov S.B.** Podkhod k probleme nechetkoy klasterizatsii v usloviyakh neopredelennosti vybora tselevoy funktsii, *Vestnik RGRTU,* Ryazan, 2009. No. 3(29), Pp. 54–60. (rus)

10. **Tishkin R.V.** Myagkiye vychisleniya v zadachakh segmentatsii kosmicheskikh izobrazheniy, *Tsifrovaya obrabotka signalov,* 2010, No. 3, Pp. 25–29. (rus)

11. **Bezdek J.C., Ehrlich R., Full W.** FCM: Fuzzy C-Means Algorithm, *Computers and Geoscience,* 1984. No. 2, Vol. 10, Pp. 191–203.

12. **Bezdek J.C., Keller J.M., Pal N.R., Pal K.** A Possibilistic Fuzzy c-Means Clustering Algorithm, *IEEE Transactions on Fuzzy Systems,* 2005, Vol. 13 (4), Pp. 517–530.

13. **Krishnapuram R., Keller J.** A possibilistic approach to clustering, *IEEE Transactions on Fuzzy Systems,* 1993, Vol. 1, Pp. 98–109.

14. **Leonenkov A.** *Nechetkoye modelirovaniye v srede MATLAB i fuzzyTECH,* St. Petersburg: BKhV-Petersburg Publ., 2003, 736 p. (rus)

15. **Strehl A., Ghosh J.** Cluster Ensembles A Knowledge Reuse Framework for Combining Multiple Partitions, *Journal of Machine Learning Research,* 2002, No. 3, Pp. 583–617.

ДЕМИДОВА Лилия Анатольевна – профессор кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета, доктор технических наук.

390005, Россия, г. Рязань, ул. Гагарина, д. 59/1.

E-mail: liliya.demidova@rambler.ru

DEMIDOVA, Liliya A. *Ryazan State Radio Engineering University.*

390005, Gagarina Str. 59/1, Ryazan, Russia.

E-mail: liliya.demidova@rambler.ru

НЕСТЕРОВ Николай Игоревич – инженер кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета.

390005, Россия, г. Рязань, ул. Гагарина, д. 59/1.

E-mail: nikolaynesterov@gmail.com

NESTEROV, Nikolay I. *Ryazan State Radio Engineering University.*

390005, Gagarina Str. 59/1, Ryazan, Russia.

E-mail: nikolaynesterov@gmail.com

ТИШКИН Роман Валентинович – доцент кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета, кандидат технических наук.

390005, Россия, г. Рязань, ул. Гагарина, д. 59/1.

E-mail: roman.tishkin@mail.ru

TISHKIN, Roman V. *Ryazan State Radio Engineering University.*

390005, Gagarina Str. 59/1, Ryazan, Russia.

E-mail: roman.tishkin@mail.ru



Математическое моделирование: методы, алгоритмы, технологии

УДК 519.6:527

В.А. Ботнев, С.М. Устинов

МЕТОДЫ РЕШЕНИЯ ПРЯМОЙ И ОБРАТНОЙ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ С ВЫСОКОЙ ТОЧНОСТЬЮ

V.A. Botnev, S.M. Ustinov

METHODS FOR DIRECT AND INVERSE GEODESIC PROBLEMS SOLVING WITH HIGH PRECISION

Предложены алгоритмы для навигационных расчетов, которые обеспечивают предельно малую погрешность во всем диапазоне входных параметров и позволяют избежать характерной для этих задач потери точности. Методы демонстрируют приемлемое для практических приложений быстродействие и могут быть рекомендованы для использования в коммерческих навигационных программных продуктах, удовлетворяющих международным стандартам.

МЕТОДЫ МОДЕЛИРОВАНИЯ; ПОТЕРЯ ТОЧНОСТИ; МАШИННОЕ ЭПСИЛОН; АСИМПТОТИЧЕСКОЕ РАЗЛОЖЕНИЕ; ГЕОДЕЗИЧЕСКИЕ ЗАДАЧИ; ЛОКСОДРОМИЯ.

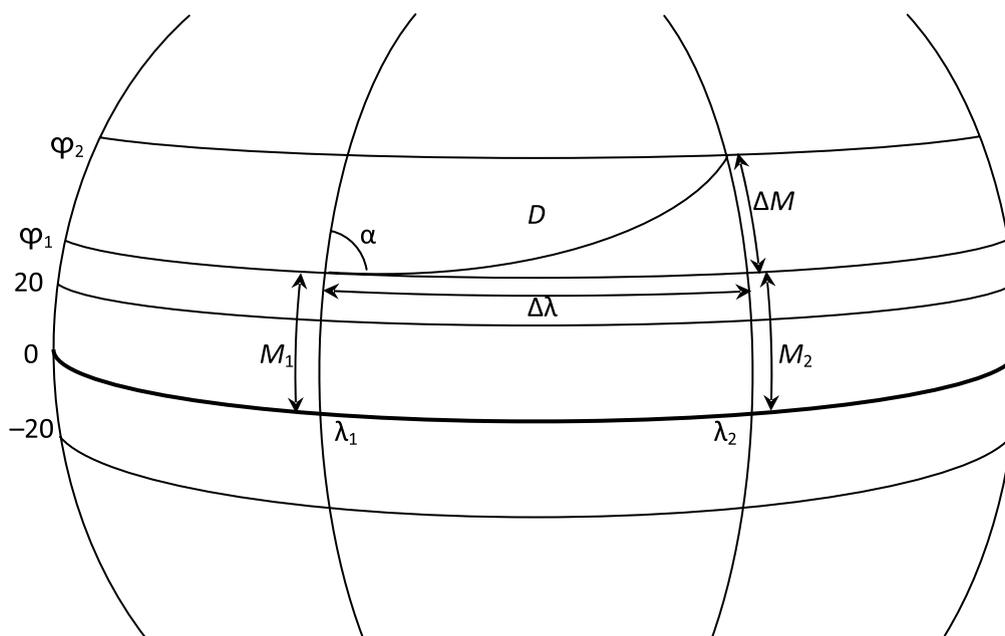
Proposed algorithms for navigational calculations provide an extremely low error over the entire range of input parameters and allow avoiding the loss of accuracy typical for these tasks. These methods demonstrate an acceptable performance for practical applications and we can recommend using them in commercial navigational software products meeting international navigational standards.

MODELLING METHODS; LOSS OF ACCURACY; MACHINE EPSILON; ASIMPTOTIC EXPANSIONS; GEODESIC PROBLEMS; LOXODROME.

Практически ни одно приложение геодезии и навигации не обходится без обращения к прямой и обратной геодезической задаче. Поэтому их эффективное решение имеет первостепенное значение. Суть прямой задачи заключается в следующем. По заданной широте φ_1 и долготе λ_1 первой точки требуется найти географические координаты φ_2 и λ_2 второй точки, если известны начальный азимут α и расстояние D между этими точками (см. рисунок). В свою очередь, обратная задача ставит своей целью найти расстояние D между двумя заданными точками и азимут α из первой точки на вторую. Обычно эти задачи решаются в двух постановках в зависимости от того, по какой линии происходит переход из одной точки в другую.

В первом случае линия (ортодрома) является кратчайшим путем по поверхности Земли. Во втором случае в силу ряда практических потребностей и исторических причин (особенно в задачах навигации) [1, 2] этой линией между точками является линия постоянного азимута α , т. е. такая линия (локсодрома) отвечает траектории с постоянным курсом. Если расстояние между точками (не в приполярной области) значительно меньше размеров Земли, то решения обеих задач оказываются относительно близкими.

Методы решения обоих вариантов задач хорошо известны и на их основе реализованы многочисленные программные продукты, успешно используемые на практике. Первый вариант задачи (ортодромия)



Локсодрома на эллипсоиде

глубже исследован, а соответствующее программное обеспечение хорошо апробировано и имеется в открытом доступе [3].

Методики решения задач во второй постановке (локсодромии) отличаются большим разнообразием [4–7] и среди них нет единой общепризнанной и всесторонне проверенной на практике.

Так, например, в работах [5, 7] указывается на то, что существует некоторое многообразие различных методов и формул, часть из которых появилась относительно недавно. В [5] отмечается также, что в попытке создать простые и эффективные алгоритмы некоторые авторы предлагают методы расчета с гигантскими ошибками. С другой стороны, даже подзадачи являются предметом отдельных статей [8, 9], и для них также не существует единого метода решения. Так, например, в работе [8] разбираются несколько приемов для нахождения меридионального расстояния (эта подзадача является одним из составных элементов прямой и обратной задач). Авторы статьи [8] не останавливают свой выбор на каком-то одном из рассматриваемых ими методов, а только лишь забраковывают некоторые из них.

В этих и в ряде других публикаций указывается на целый ряд проблем, возника-

ющих при решении геодезических задач. Так, например, отмечается заметная потеря точности, связанная с вычитанием близких чисел для случая, когда широты обеих точек примерно равны. Известные проблемы возникают также, когда хотя бы одна из исходных точек находится в приполярных областях.

В настоящей статье предлагается свободный от этих недостатков эффективный алгоритм решения прямой и обратной задач для локсодромы с высокой точностью. Под высокой точностью здесь понимается сокращение погрешности вплоть до величины порядка нескольких машинных эпсилон ε . Здесь ε — это минимальное число, которое при сложении с единицей дает результат, больший единицы для данного компьютера.

Такие жесткие требования к погрешности вызваны следующим. Существует целый ряд практических задач (например, нахождение на локсодромической траектории точки, ближайшей к заданной), которые содержат прямую или обратную геодезическую задачу как отдельные фрагменты. Здесь требуется найти решение с относительной погрешностью $10^{-7} - 10^{-8}$, что сводится к ошибкам порядка 1 метра. Так как

используемые алгоритмы ориентированы на поиск различного рода экстремальных точек, требования к точности решений геодезических задач ужесточаются. Необходимая погрешность в них не должна превышать 10^{-15} . Для арифметики с двойной точностью, где машинное эpsilon порядка 10^{-16} , это достижимо, но труднореализуемо. Учетверенная же точность в настоящее время доступна на аппаратном уровне только на весьма ограниченном круге платформ, а надежные кроссплатформенные библиотеки для работы с числами с плавающей точкой учетверенной точности отсутствуют. К тому же из-за программной реализации скорость работы с числами учетверенной точности не очень велика, в то время как предлагаемый в работе алгоритм для чисел двойной точности в несколько раз опережает по скорости максимально упрощенный вариант алгоритма с числами учетверенной точности.

Кроме сказанного выше имеется целый ряд широко распространенных устройств (планшетные компьютеры, встраиваемые контроллеры и пр.) с величиной машинного эpsilon порядка 10^{-7} . При этом существуют многочисленные приложения, в которых такая точность допустима. Предлагаемые ниже алгоритмы актуальны и в этом случае. Они обеспечивают погрешность, немногим превышающую машинное эpsilon, независимо от того, какая точность ($\varepsilon \approx 10^{-7}$ или $\varepsilon \approx 10^{-16}$) обеспечивается на аппаратном уровне. В случае $\varepsilon \approx 10^{-7}$ учитывается меньшее число слагаемых соответствующих разложений.

Методика решения прямой геодезической задачи

Рассматривается движение из одной точки в другую под постоянным углом α между направлением на северный полюс и траекторией.

Заданы точка с широтой φ_1 и долготой λ_1 , азимут α и расстояние D до второй точки. Требуется найти широту φ_2 и долготу λ_2 второй точки.

Эта геодезическая задача для локсодромии может быть разделена на следующие этапы:

1. Вычисление меридионального расстояния M_1 для первой точки (см. рисунок).
2. Вычисление разности меридиональных расстояний ΔM и географической широты φ_2 второй точки.
3. Вычисление разности изометрических широт.
4. Вычисление разности долгот второй и первой точек $\Delta \lambda$.

Вычисление меридионального расстояния для первой точки. Меридиональное расстояние M_1 первой точки рассчитывается по хорошо известной формуле:

$$M_1 = \int_0^{\varphi_1} \frac{a(1 - e^2)d\varphi}{(1 - e^2 \sin^2 \varphi)^{3/2}}. \quad (1)$$

Известные методы вычисления интеграла (1) условно разделяются на три группы.

1) *Использование различных квадратурных формул.* Такой подход не способен обеспечить точность до нескольких машинных эpsilon, т. к. сводится к сложению нескольких слагаемых одного порядка, что приводит к накоплению ошибок округления. И чем больше значений функций вычисляется на промежутке для повышения точности, тем сильнее сказывается влияние ошибок округлений при многократном сложении.

2) *Разложение подынтегрального выражения в степенной ряд по различным параметрам с последующим интегрированием этого ряда.* Простейший вариант – выбрать в качестве малого параметра $e^2 = f(2 - f)$ (где f – это коэффициент сжатия земного эллипсоида), которое действительно мало и для модели Земли, принятой для международного мореплавания (WGS84), составляет около 0,0067. Но существует и более удобная для этой цели величина:

т. н. третий эксцентриситет $n = \frac{f}{2 - f}$, который еще меньше и приблизительно равен 0,0017. Важно отметить, что выбор этого параметра сокращает число слагаемых ряда и не приводит к усложнению вычисления коэффициентов. Как следствие, это позволяет уменьшить время расчета и снизить накопление ошибок. Используемое разложение в ряд имеет вид:

$$M(\varphi) = a(1 - n)(1 - n^2)S(\varphi), \quad (2)$$

где

$$S(\varphi) \approx a_0 \varphi + \sum_{k=1}^p (-1)^k a_{2k} \sin(2k\varphi), \quad p = 6 \quad (3)$$

$$a_0 = 1 + \frac{9}{4} n^2 + \frac{225}{64} n^4 + \frac{1225}{256} n^6 + \dots$$

$$a_2 = \frac{3}{2} n + \frac{45}{16} n^3 + \frac{525}{128} n^5 + \dots$$

$$a_4 = \frac{15}{16} n^2 + \frac{105}{64} n^4 + \frac{4725}{2048} n^6 + \dots$$

$$a_6 = \frac{35}{48} n^3 + \frac{315}{256} n^5 + \dots$$

$$a_8 = \frac{315}{512} n^4 + \frac{2079}{2048} n^6 + \dots$$

$$a_{10} = \frac{693}{1280} n^5 + \dots$$

$$a_{12} = \frac{1001}{2048} n^6 + \dots$$

Разложение этого вида было впервые предложено Гельмертом с точностью до a_8 [10] и уточнено до a_{10} в работе [7]. Для достижения необходимой точности нужно было продолжить представленный ряд до a_{12} ($p = 6$ в формуле (3)) и вычислить необходимые коэффициенты, содержащие слагаемые с n^6 .

Введение a_{12} влияет на точность результата не слишком сильно, в то время как учет слагаемых с n^6 в коэффициентах a_0 , a_4 и a_8 крайне важен и позволяет ограничить максимальную ошибку сверху величиной 2ε . В противном случае она достигает 4ε .

Суммирование ряда (2) осуществляется по технике Кленшоу [11], что позволяет сократить объем вычислений и получить более точный результат. На практике эта процедура сводится к следующему. Зададим

$$c_{p+1}(\varphi) = c_{p+2}(\varphi) = 0.$$

Тогда, вычисляя последовательно по мере уменьшения индексов все $c_p(\varphi)$ по формуле

$$c_k(\varphi) = a_{2k} + 2c_{k+1}(\varphi) \cos(2\varphi) - c_{k+2}(\varphi), \quad 1 \leq k \leq p, \quad (4)$$

получим $c_1(\varphi)$ и

$$S(\varphi) = a_0 \varphi + c_1(\varphi) \sin(2\varphi). \quad (5)$$

Как показано в работе [11], формула

(5) дает для $S(\varphi)$ тот же результат, что и (3). Эта техника подобна схеме Горнера и позволят свести вычисление суммы к вычислению только двух тригонометрических функций ($\sin(2\varphi)$ и $\cos(2\varphi)$). Таким образом, расчеты значительно ускоряются. Кроме того, т. к. первое слагаемое в формуле (4) будет, исходя из выражений для коэффициентов a_{2k} , значительно доминировать над остальными, ошибка расчетов не будет накапливаться на каждом шаге применения (4). В итоге конечный результат будет точнее, чем при применении исходной формулы (3), где смежные слагаемые могут быть разных знаков и примерно одинаковыми по модулю для некоторых значений широты φ_1 , что приводит к катастрофической потере точности.

3) *Представление интеграла (1) в виде комбинации легко вычисляемых эллиптических интегралов.* Исходный интеграл (1) представляется как комбинация интегралов в форме Карлсона [12, 13]:

$$M(\varphi) = a \sin(\varphi) \left(R_\varphi(\cos^2 \varphi, 1 - e^2 \sin^2 \varphi, 1) - \frac{e^2 \sin^2 \varphi R_D(\cos^2 \varphi, 1 - e^2 \sin^2 \varphi, 1)}{3} + \frac{\cos(\varphi)}{\sqrt{1 - e^2 \sin^2 \varphi}} \right), \quad (6)$$

где R_φ и R_D – базовые эллиптические интегралы в форме Карлсона, которые могут быть вычислены эффективно по специальным алгоритмам. Погрешность данного метода чуть больше 2ε . Этот результат оказывается чуть менее точным, чем разложение в ряд с использованием техники Кленшоу.

Подводя итоги этого раздела, можно отметить, что средняя погрешность вычислений при представлении интеграла (1) в виде комбинации эллиптических интегралов (6) оказывается несколько меньше, чем расчеты по асимптотической формуле (2). Однако объем вычислений в первом случае оказывается на порядок больше, чем во втором. Окончательный выбор диктуется спецификой решаемой задачи.

Вычисление разности меридиональных

расстояний ΔM и географической широты φ_2 второй точки. Нужно найти такое φ_2 , что выполнено равенство:

$$M_2 = \int_0^{\varphi_2} \frac{a(1 - e^2)d\varphi}{(1 - e^2 \sin^2 \varphi)^{3/2}}, \quad (7)$$

где M_2 можно получить из разности изометрических расстояний $\Delta M = D \cos(\alpha)$, $M_2 = M_1 + \Delta M$. Для определения $\varphi_2 = \varphi_1 + \Delta\varphi$ найдем сначала разность широт $\Delta\varphi$. Использование в дальнейших расчетах величины $\Delta\varphi$ вместо φ_2 вызвано тем, что при близких значениях φ_1 и φ_2 происходит катастрофическая потеря точности в операторе $\varphi_2 - \varphi_1$. С этой целью обратим ряд (3) и вместо $M(\varphi)$ построим другой ряд $\varphi(M)$, воспользовавшись техникой, изложенной в [7]. При этом, так же как и в предыдущем подразделе, необходимо в коэффициентах разложения учитывать слагаемые с n^5 и n^6 , чтобы не использовать метод Ньютона, как это было предложено в [7].

Первоначально строится ряд:

$$\begin{aligned} \varphi(M) = & S + b_2 \sin 2S + \\ & + b_4 \sin 4S + b_6 \sin 6S + b_8 \sin 8S + \\ & + b_{10} \sin 10S + b_{12} \sin 12S + \dots \end{aligned} \quad (8)$$

где

$$\begin{aligned} b_2 &= \frac{3}{2}n - \frac{27}{32}n^3 + \frac{269}{512}n^5 + \dots \\ b_4 &= \frac{21}{16}n^2 - \frac{55}{32}n^4 + \frac{6759}{4096}n^6 + \dots \\ b_6 &= \frac{151}{96}n^3 - \frac{417}{128}n^5 + \dots \\ b_8 &= \frac{1097}{512}n^4 - \frac{15543}{2560}n^6 + \dots \\ b_{10} &= \frac{8011}{2560}n^5 - \dots \\ b_{12} &= \frac{293393}{61440}n^6 - \dots \\ S &= \frac{1}{a(1-n)(1-n^2)} \times \\ & \times \frac{M}{\left(1 + \frac{9}{4}n^2 + \frac{225}{64}n^4 + \frac{1225}{256}n^6 + \dots\right)}. \end{aligned}$$

Если не учитывать коэффициент b_{12} , то погрешность может достигать семи машин-

ных эпсилон, в то время как с его учетом мы получаем ошибку не более 2ε .

Далее,

$$\begin{aligned} \Delta\varphi &= \varphi(M + \Delta M) - \varphi(M) = \\ &= \Delta S + 2b_2 \sin \Delta S \cdot \cos\left(S + \frac{\Delta S}{2}\right) + \\ &+ 2b_4 \sin 2\Delta S \cdot \cos 2\left(S + \frac{\Delta S}{2}\right) + \\ &+ 2b_6 \sin 3\Delta S \cdot \cos 3\left(S + \frac{\Delta S}{2}\right) + \\ &+ 2b_8 \sin 4\Delta S \cdot \cos 4\left(S + \frac{\Delta S}{2}\right) + \\ &+ 2b_{10} \sin 5\Delta S \cdot \cos 5\left(S + \frac{\Delta S}{2}\right) + \\ &+ 2b_{12} \sin 6\Delta S \cdot \cos 6\left(S + \frac{\Delta S}{2}\right). \end{aligned} \quad (9)$$

Суммирование этого ряда также осуществляется по технике Кленшоу [11]. Следует при этом отметить, что относительная погрешность для $\Delta\varphi$ не превышает 2ε .

После вычисления $\Delta\varphi$ легко определить $\varphi_2 = \varphi_1 + \Delta\varphi$. В итоге ошибка по широте не превышает 15 нм в терминах расстояния вдоль меридиана.

Вычисление разности изометрических широт. Изометрическая широта Q является функцией обычной широты:

$$Q = \ln \left[\tan \left(\frac{\pi}{4} + \frac{\varphi}{2} \right) \left(\frac{1 - e \sin \varphi}{1 + e \sin \varphi} \right)^{\frac{e}{2}} \right]. \quad (10)$$

В дальнейших расчетах потребуется значение разности изометрических широт $\Delta Q = Q_2 - Q_1$.

Чтобы не потерять точность, необходимо избегать непосредственного вычитания выражений, полученных в результате промежуточных вычислений [14]. С этой целью вместо φ_2 будем использовать выражение $\varphi_1 + \Delta\varphi$:

$$\Delta Q = \ln \left[\frac{\tan \left(\frac{\pi}{4} + \frac{\varphi_1 + \Delta\varphi}{2} \right)}{\tan \left(\frac{\pi}{4} + \frac{\varphi_1}{2} \right)} \right] + \quad (11)$$

$$+ \frac{e}{2} \ln \left[\frac{(1 - e \sin(\varphi_1 + \Delta\varphi))(1 + e \sin \varphi_1)}{(1 + e \sin(\varphi_1 + \Delta\varphi))(1 - e \sin \varphi_1)} \right]. \quad (11)$$

При малых значениях $\Delta\varphi$ числители и знаменатели обоих логарифмов в (11) близки друг к другу, и здесь также может происходить потеря точности при вычитании близких чисел. Поэтому перспективным представляется разложение этих логарифмов в ряд как по самому параметру $\Delta\varphi$, так и какому-либо другому малому параметру, зависящему от $\Delta\varphi$.

Наибольшую проблему представляет первое слагаемое в (11), т. к. выражение под логарифмом может быть сколь угодно близким к нулю и сколь угодно большим в допустимом диапазоне изменения широт. Задача вычисления второго логарифма заметно проще, т. к. его аргументом является выражение, отстоящее от единицы на величину порядка $2e$ или $0,16$ для земного эллипсоида. Рассмотрим четыре варианта решения возникших трудностей.

Вариант 1. Воспользуемся известным разложением для функции $\ln(1+x)$ при малом значении x :

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots \quad (12)$$

Предварительно преобразуем первое слагаемое в (11):

$$\begin{aligned} R(\varphi_1, \Delta\varphi) &= \ln \left[\frac{\tan\left(\frac{\pi}{4} + \frac{\varphi_1 + \Delta\varphi}{2}\right)}{\tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right)} \right] = \\ &= \ln \left[\frac{\tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right) + \tan\left(\frac{\Delta\varphi}{2}\right)}{\tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right) \left(1 - \tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right) \tan\left(\frac{\Delta\varphi}{2}\right)\right)} \right] = \\ &= \ln \left(\frac{1 + \frac{z}{a}}{1 - a \cdot z} \right), \quad (13) \end{aligned}$$

где $a = \tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right)$ и $z = \tan\left(\frac{\Delta\varphi}{2}\right)$. С учетом диапазона изменения географической широты $\varphi_1 \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ параметр a не может

быть отрицательным.

Теперь, руководствуясь (12), получаем ряд по степеням z . Ему свойственна относительно медленная сходимость. Наиболее эффективный расчет будет получаться только в окрестности экватора, где $a = 1$, что приведет к сокращению всех четных степеней z в разложении:

$$\ln\left(\frac{1+z}{1-z}\right) = 2z \left(1 + \frac{z^2}{3} + \frac{z^4}{5} + \frac{z^6}{7} + \dots\right)$$

Для упрощения дальнейшего анализа заметим, что

$$R(\varphi_1, \Delta\varphi) = R(-\varphi_1, -\Delta\varphi). \quad (14)$$

Это позволяет ограничиться рассмотрением только положительных значений φ_1 , что гарантирует неравенство $a \geq 1$. Так как в этом случае $\left|\frac{z}{a}\right| \leq |az|$, наибольшие проблемы вызывает разложение знаменателя дроби под логарифмом в правой части (13). Численные эксперименты показали, что использовать разложение в ряд выгоднее из соображений погрешности только при $|az| \leq 0,5$. В противном случае без потери точности можно использовать непосредственное вычисление логарифма.

Этот критерий стал определяющим для стратегии вычисления $R(\varphi_1, \Delta\varphi)$. При граничном значении $|az| = 0,5$ для вычисления $R(\varphi_1, \Delta\varphi)$ с максимальной точностью в 3ϵ необходимо учесть 48 слагаемых разложения (13), что не очень эффективно. Реально эта погрешность будет несколько выше, т. к. имеется погрешность в вычислении $\Delta\varphi$. Второе слагаемое в (11) во всех случаях разлагалось в ряд (12), т. к. его сходимость не представляла проблемы при любых значениях входных аргументов. В итоге погрешность вычисления только ΔQ (считая аргументы абсолютно точными) достигала 6ϵ .

Вариант 2. Второй метод заключается в использовании формулы:

$$\ln(1+x) = \frac{x \cdot \ln(1+x)}{(1+x) - 1}, \quad (15)$$

применимость которой оправдана при $|x| \leq 0,5$ [15]. При этом если сложение $1+x$ дает 1 (x меньше машинного эпсилон), то достаточно в качестве ответа вернуть само

значение x (так как $\ln(1+x) \approx x$).

Погрешность данного метода не превышает 2ϵ с приемлемой эффективностью расчета. Проблема лишь только в том, что нужно явно представить выражение для x , применительно к формуле (13). Это выражение имеет сложный вид и само порождает погрешность в несколько ϵ . Другая проблема связана с тем, что необходимо обеспечить кроссплатформенность реализации, которую в данном случае нельзя гарантировать, т. к. некоторые оптимизирующие компиляторы произведут сокращение дроби (15), и логарифм будет вычисляться непосредственно, а не по формуле (15).

Вариант 3. В широко распространенной библиотеке Boost Math [15] для вычисления $\ln(1+x)$ при малом значении x , если компилятор не позволяет реализовать подход, описанный в варианте 2, предлагается использовать Паде-аппроксимацию. Многочисленные эксперименты с этим алгоритмом показали, что погрешность достигает 6ϵ , что даже хуже варианта 1.

Вариант 4. Наилучшие показатели обеспечил следующий прием:

$$\begin{aligned} \ln x &= \ln\left(\frac{1+y}{1-y}\right) = \\ &= 2y\left(1 + \frac{y^2}{3} + \frac{y^4}{5} + \frac{y^6}{7} + \dots\right), \end{aligned} \quad (16)$$

где $y = \frac{x-1}{x+1}$ и $x > 0$. Погрешность этого метода не превышает 2ϵ , а проигрыш по времени варианту 2 (самому быстрому из всех методов) составляет около 14 %.

Существенная потеря точности обычного логарифма, как уже отмечалось, происходит в диапазоне $x \in [0, 5; 1, 5]$, что для переменной y отвечает диапазону $y \in \left[-\frac{1}{3}; \frac{1}{5}\right]$. Таким образом, наблюдается двойной выигрыш. Во-первых, меньшая величина y обеспечивает сокращение учитываемых слагаемых разложения. Во-вторых, получающийся ряд (16) быстрее сходится, т. к. в нем присутствуют только четные степени y . В итоге быстрое действие этого метода в два раза выше по сравнению с вариантом 1.

Реализация алгоритма сводится к следующему. Чтобы для первого слагаемого в (11) обеспечить вид

$$\ln\left[\frac{\tan\left(\frac{\pi}{4} + \frac{\varphi_1 + \Delta\varphi}{2}\right)}{\tan\left(\frac{\pi}{4} + \frac{\varphi_1}{2}\right)}\right] = \ln\left(\frac{1+y}{1-y}\right),$$

следует выбрать $y = \frac{\sin\frac{\Delta\varphi}{2}}{\cos\left(\varphi_1 + \frac{\Delta\varphi}{2}\right)}$. Аналогично, второе слагаемое в (11) приобретает вид:

$$\begin{aligned} \ln\left[\frac{(1 - e \sin(\varphi_1 + \Delta\varphi))(1 + e \sin \varphi_1)}{(1 + e \sin(\varphi_1 + \Delta\varphi))(1 - e \sin \varphi_1)}\right] = \\ = \ln\left(\frac{1+y}{1-y}\right), \end{aligned}$$

$$\text{если } y = \frac{-2e \sin\frac{\Delta\varphi}{2} \cos\left(\varphi_1 + \frac{\Delta\varphi}{2}\right)}{1 - e^2 \sin \varphi_1 \sin\left(\varphi_1 + \frac{\Delta\varphi}{2}\right)}.$$

Модуль второго слагаемого в (11) почти на два порядка меньше, чем модуль первого. Это позволяет складывать их непосредственно, т. к. потеря точности при этом возникнуть не может. Для суммирования знакопостоянного ряда (16) применялась схема Горнера, сокращающая объем вычислений и минимизирующая погрешность.

Вычисление разности долгот. Разность долгот вычисляется по формуле:

$$\Delta\lambda = \Delta Q \cdot \tan \alpha. \quad (17)$$

Отсюда легко получить $\lambda_2 = \lambda_1 + \Delta\lambda$. Относительная погрешность $\Delta\lambda$ не превышает нескольких машинных эпсилон, что подтвердилось всесторонними экспериментами во всем допустимом диапазоне входных параметров. Вместе с тем выявился набор исходных данных, который отвечает движению к околополярной точке с многочисленным количеством витков вокруг полюса. В этом случае погрешность в определении второй точки могла составить до 360 нм. Справедливости ради следует отметить, что такая траектория лишена какого-либо практического смысла.

Обратная геодезическая задача

Даны: широта φ_1 и долгота λ_1 первой точки и широта φ_2 и долгота λ_2 второй точки. Нужно определить азимут α и дистанцию, которую нужно пройти, двигаясь постоянно под углом α к направлению на северный полюс, чтобы из первой точки попасть во вторую.

Основные этапы решения обратной геодезической задачи для локсодромии:

1) Вычисление разности изометрических широт ΔQ .

2) Вычисление азимута α .

3) Вычисление меридионального расстояния ΔM .

4) Вычисление расстояния между точками D .

Вычисление разности изометрических широт. Эта величина определяется по методике варианта 4, только вместо $\varphi_1 + \Delta\varphi$ записываем φ_2 :

$$\Delta Q = \ln \left[\frac{\tan \left(\frac{\pi}{4} + \frac{\varphi_2}{2} \right)}{\tan \left(\frac{\pi}{4} + \frac{\varphi_1}{2} \right)} \right] + \frac{e}{2} \ln \left[\frac{(1 - e \sin(\varphi_2))(1 + e \sin \varphi_1)}{(1 + e \sin(\varphi_2))(1 - e \sin \varphi_1)} \right]. \quad (18)$$

Первое слагаемое принимает требуемый

вид для $y = \frac{\sin \frac{\varphi_2 - \varphi_1}{2}}{\cos \frac{\varphi_2 + \varphi_1}{2}}$, а второе слага-

емое — при $y = \frac{-2 \sin \frac{\varphi_2 - \varphi_1}{2} \cos \frac{\varphi_2 + \varphi_1}{2}}{1 - e^2 \sin \varphi_1 \sin \frac{\varphi_2 + \varphi_1}{2}}$.

Вычисление азимута:

$$\alpha = \arctan \left(\frac{\lambda_2 - \lambda_1}{\Delta Q} \right). \quad (19)$$

С учетом знаков числителя и знаменателя α однозначно определяется в диапазоне $[-\pi; \pi]$.

Погрешность при определении азимута, как показали многочисленные эксперименты, не превышает 3ε .

Вычисление меридионального расстояния:

$$\Delta M = \int_{\varphi_1}^{\varphi_2} \frac{a(1 - e^2)d\varphi}{(1 - e^2 \sin^2 \varphi)^{3/2}}. \quad (20)$$

Это выражение записывается как $M(\varphi_2) - M(\varphi_1)$. Уменьшаемое и вычитаемое раскладываются в ряд (2) по степеням n , затем ряды вычитаются подобно тому, как это делалось в предыдущем разделе.

Возможен и другой подход. В этом случае применяется метод Карлсона, описанный выше.

Вычисление расстояния:

$$D = \frac{\Delta M}{\cos \alpha}. \quad (21)$$

Специальный случай возникает при $\alpha = \pm \frac{\pi}{2}$ и $\cos \alpha = 0$, что отвечает движению вдоль параллели.

Здесь можно легко определить D , зная радиус параллели

$R = \frac{\alpha \cos \varphi_1}{\sqrt{1 - e^2 \sin^2 \varphi_1}}$ и радианную меру дуги

по известным долготам λ_1 и λ_2 . Погрешность при определении расстояния не превышает 5ε .

Предложена методика решения прямой и обратной геодезических задач с высокой точностью. При этом на отдельных ее этапах получены следующие результаты.

- На основе всестороннего анализа различных подходов к вычислению меридионального расстояния выявлены два, удовлетворяющие заявленным требованиям по точности. Для первого из них (разложение в ряд) показано, что традиционно используемого количества членов разложения недостаточно для достижения заданной точности. Асимптотическое разложение было продолжено с получением выражения для последующих коэффициентов. Для второго подхода был предложен метод вычисления меридионального расстояния в виде суммы эллиптических интегралов специального вида.

- Для вычисления широты конечной точки получены коэффициенты асимптотического разложения, позволяющие обеспечить вычисления с требуемой погрешностью. При этом для вычисления разности широт предложенный алгоритм обеспечил

требуемую высокую точность во всем диапазоне входных параметров, гарантируя отсутствие вычитания близких чисел.

• Для вычисления разности изометрических широт предложены два метода (вариант 1 и вариант 4) и выполнено их сравнение с используемыми (вариант 2 и вариант 3). Вариант 3 (Паде-аппроксимация) оказался заметно хуже всех по точности. Вариант 4, хотя и уступил по объему вычислений варианту 2 на 14 %, однако, в отличие от последнего, гарантированно работает на всех платформах. Еще один предложенный метод (вариант 1) также может быть рекомендован для решения поставленной задачи. При этом, если одна из точек расположена на экваторе, объем вычислений и точность методов (варианты 1 и 4) совпадают, а в остальных случаях эти характеристики для варианта 4 несколько лучше.

• Для вычисления меридионального расстояния в обратной геодезической задаче предложены два алгоритма. Для первого из них построено отвечающее требуемой точности асимптотическое разложение. Во втором случае получено представление исходного интеграла в виде комбинации базовых эллиптических интегралов в форме Карлсона. Оба метода продемонстрировали соизмеримую точность, но второй метод уступает по быстрдействию.

Предложенная методика решения прямой и обратной геодезической задачи и составляющие ее отдельные методы позволяют с требуемой точностью и за приемлемое время обслуживать многочисленные приложения, формирующие библиотеку навигационных расчетов. Она может использоваться в коммерческих навигационных программных продуктах, удовлетворяющих международным стандартам.

СПИСОК ЛИТЕРАТУРЫ

1. **Alexander J.** Loxodromes: A Rhumb Way to Go // *College Mathematics Journal*. 2004. No. 77(5). Pp. 349–356.

2. **Weintrit A., Kopacz P.** A Novel Approach to Loxodrome (Rhumb-Line), Orthodrome (Great Circle) and Geodesic Line in ECDIS and Navigation in General // *Methods and Algorithms in Navigation, Marine Navigation and Safety of Sea Transportation*. Leiden: A Balkema Book, CRC Press, Taylor & Francis Grup, 2011. Pp. 123–132.

3. **Karney C.F.F.** Algorithms for geodesics // *J. Geodesy*. 2013. No. 87(1). Pp. 43–55.

4. **Petrović M.** Differential Equation of a Loxodrome on the Spheroid // *Naše more*. 2007. No. 54(3-4). Pp. 87–89.

5. **Pallikaris A., Paradassis D., Tsoulos L.** New calculation algorithms for GIS navigational systems and receivers // *European Navigational Conf. ENC-GNSS*. Naples, Italy, 2009.

6. [Электронный ресурс]// URL: <http://koti.mbnet.fi/jukaukor/loxodrome.pdf>

7. **Deakin R.E., Hunter M.N.** Geometric Geodesy. Part A // *Lecture Notes*. School of Mathematical & Geospatial Sciences, RMIT University, Melbourne, Australia, 2008. 140 p.

8. **Kos S., Pogany T.K.** On the mathematics

of navigational calculations for meridian sailing, Solstice // *J. of Geography and Mathematics*. 2012. No. 23(2).19 p.

9. **Weintrit A., Kopacz P.** On Computational Algorithms Implemented in Marine Navigational Software Used in Marine Navigation Electronic Devices and Systems // *Annual of Navigation*. 2012. No. 19. Part 2. Pp. 171–184.

10. **Helmert F.R.** Die mathematischen und physikalischen Theorem der höheren Geodäsie // *Die mathematischen Theorem*. Leipzig, 1880. Vol. 1.

11. **Clenshaw C.W.** A note on the summation of Chebyshev series // *Math. Tables Aids Comput.* 1955. No. 9. Pp. 118–120.

12. **Carlson B.C.** Numerical computation of real or complex elliptic integrals // *Numerical Algorithms*. 1995. No. 10. Pp. 13–26.

13. **Carlson B.C.** A table of elliptic integrals of second kind // *Math. Comp.* 1987. No. 49. Pp. 595–606.

14. **Устинов С.М., Зимницкий В.А.** Вычислительная математика. СПб.: БХВ-Петербург, 2009. 336 с.

15. [Электронный ресурс]// URL: <http://boost.org>

REFERENCES

1. **Alexander J.** Loxodromes: A Rhumb Way to Go, *College Mathematics Journal*, 2004, No. 77(5), Pp. 349–356.

2. **Weintrit A., Kopacz P.** A Novel Approach to Loxodrome (Rhumb-Line), Orthodrome (Great Circle) and Geodesic Line in ECDIS and Navigation

in General, *Methods and Algorithms in Navigation, Marine Navigation and Safety of Sea Transportation*. Leiden: A Balkema Book, CRC Press, Taylor & Francis Grup, 2011, Pp. 123–132.

3. **Karney C.F.F.** Algorithms for geodesics, *Journal Geodesy*, 2013, No. 87(1), Pp. 43–55.

4. **Petrović M.** Differential Equation of a Loxodrome on the Spheroid, *Naše more*, 2007, No. 54(3-4), Pp. 87–89.

5. **Pallikaris A., Paradissis D., Tsoulos L.** New calculation algorithms for GIS navigational systems and receivers, *European Navigational Conference ENC-GNSS*. Naples, Italy, 2009.

6. Available: <http://koti.mbnet.fi/jukaukor/loxodrome.pdf>

7. **Deakin R.E., Hunter M.N.** Geometric Geodesy. Part A, *Lecture Notes*. School of Mathematical & Geospatial Sciences, RMIT University, Melbourne, Australia, 2008, 140 p.

8. **Kos S., Pogany T.K.** On the mathematics of navigational calculations for meridian sailing, Solstice, *Journal of Geography and Mathematics*,

2012, No. 23(2), 19 p.

9. **Weintrit A., Kopicz P.** On Computational Algorithms Implemented in Marine Navigational Software Used in Marine Navigation Electronic Devices and Systems, *Annual of Navigation*, 2012, No. 19, Part 2, Pp. 171–184.

10. **Helmert F.R.** Die mathematischen und physikalischen Theorem der höheren Geodäsie, *Die mathematischen Theorem*, Leipzig, 1880, Vol. 1.

11. **Clenshaw C.W.** A note on the summation of Chebyshev series, *Math. Tables Aids Comput.*, 1955, No. 9, Pp. 118–120.

12. **Carlson B.C.** Numerical computation of real or complex elliptic integrals, *Numerical Algorithms*, 1995, No. 10, Pp. 13–26.

13. **Carlson B.C.** A table of elliptic integrals of second kind, *Math. Comp.*, 1987, No. 49, Pp. 595–606.

14. **Ustinov S.M., Zimnitskiy V.A.** *Vychislitel'naya matematika*, St. Petersburg: BKhV-Petersburg Publ., 2009, 336 p. (rus)

15. Available: <http://boost.org>

БОТНЕВ Виктор Александрович – старший инженер-программист ЗАО «Транзас». 199178, Россия, Санкт-Петербург, Малый пр. В.О., д. 54, корп. 4, лит. В.
E-mail: Botnev.Victor@transas.com

BOTNEV, Victor A. *Transas*. 199178, Maly Ave. 54-4, V.O., St. Petersburg, Russia.
E-mail: Botnev.Victor@transas.com

УСТИНОВ Сергей Михайлович – профессор Санкт-Петербургского государственного политехнического университета, доктор технических наук. 195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: usm50@yandex.ru

USTINOV, Sergey M. *St. Petersburg State Polytechnical University*. 195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: usm50@yandex.ru



Приборы, информационно-измерительные системы

УДК 550.388.2

С.Ф. Четвериков

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ ИЗУЧЕНИЯ СТАТИСТИЧЕСКИХ СВОЙСТВ И СУТОЧНО-СЕЗОННЫХ ВАРИАЦИЙ ИНТЕНСИВНОСТИ ЕСТЕСТВЕННОГО ОНЧ РАДИОШУМА

S.F. Chetverikov

USING THE INFORMATION AND MEASUREMENT SYSTEM TO STUDY STATISTICAL CHARACTERISTICS AND DAILY AND SEASONAL VARIATION IN THE INTENSITY OF VLF ATMOSPHERIC RADIO NOISE

Рассмотрены и изучены физические условия происхождения, статистические свойства и суточно-сезонные вариации естественных радишумов ОНЧ диапазона. Приведены основные закономерности суточного хода огибающей напряженности поля атмосферных радиопомех. Описано функционирование измерительно-вычислительной системы.

ГРОЗОВЫЕ РАЗРЯДЫ; АТМОСФЕРНЫЕ РАДИОШУМЫ; СТАТИСТИЧЕСКИЕ СВОЙСТВА; СУТОЧНО-СЕЗОННЫЕ ВАРИАЦИИ; ОБОБЩАЮЩАЯ ЭМПИРИЧЕСКАЯ МОДЕЛЬ; АТМОСФЕРИК.

This paper discusses physical conditions of the origin of VLF Atmospheric Radio Noise, summarizes its statistical characteristics and daily and seasonal variation in its intensity. It also examines general regularities in the diurnal variation of the envelope of Atmospheric Radio Noise field strength and describe the information and measurement system operates.

LIGHTNING DISCHARGES; ATMOSPHERIC RADIO NOISE; STATISTICAL CHARACTERISTICS; DAILY AND SEASONAL VARIATION; GENERALIZED EMPIRICAL MODEL; ATMOSPHERIC.

Известно, что основным источником естественных очень низкочастотных (ОНЧ) радишумов (частоты 3÷30 кГц) являются грозные разряды, число которых на земле в течение суток колеблется от 80 до 120 в единицу времени [1].

Радиоволны, излученные в моменты грозных разрядов, занимают широкий спектр частот (от долей герц до десятков мегагерц). При этом основную долю разрядов составляют вертикальные (между облаками и землей и между близлежащими облаками), т. е. излучаются радиоволны в основном с вертикальной составляющей

E_z электрической компоненты электромагнитного поля. Излученные радишумы представляют собой в точке приема пуассоновский поток со случайными моментами прихода радиоимпульсов, их амплитудами и длительностями. Накладываясь друг на друга, при узкополосном приеме они формируют функцию распределения, которая подчиняется нормальному закону при малых пороговых уровнях поля, куда поступает основная часть импульсов, и переходит в чисто пуассоновский поток при высоких пороговых уровнях поля, где каждый импульс принимается отдельно. Это легко ви-

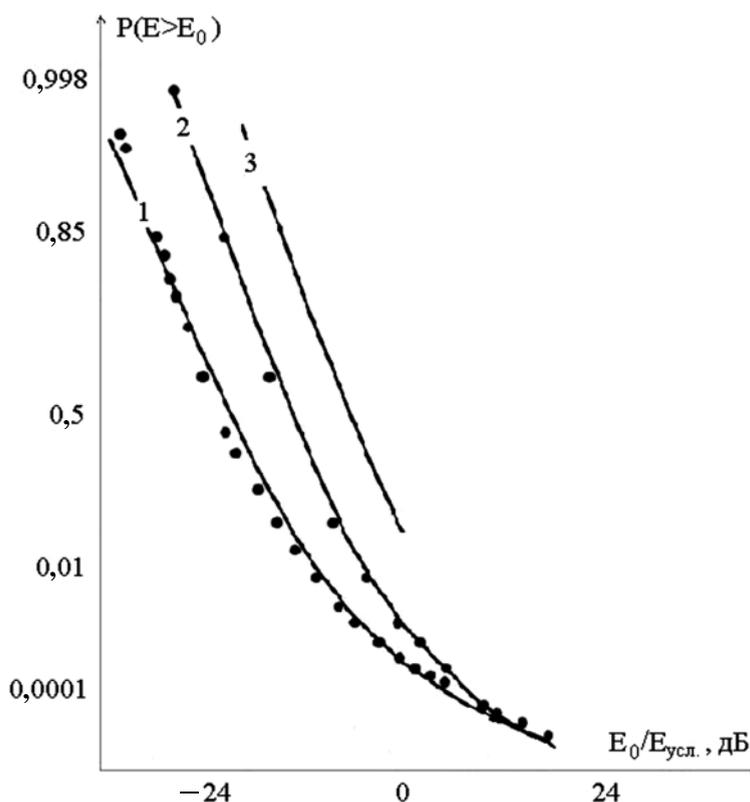


Рис. 1. Измеренная (.....) и рассчитанная (—) интегральные функции распределения для атмосферного радишума (1), (2) и нормального узкополосного шума (3)

деть из поведения функции распределения вероятностей огибающей напряженности поля $P(E > E_0)$, представленной на рис. 1.

Экспериментальная часть

Функция распределения важна при оценке помехоустойчивости радиосистем в присутствии неустраняемых импульсных грозных разрядов. Однако важно подчеркнуть, что условия приема в точке регистрации определяются не абсолютным значением интенсивности полезного сигнала, а его отношением к интенсивности шума. Поэтому возникает практическая необходимость в исследовании среднего уровня радишумов естественной природы.

На рис. 2 а для города Магадан [2] представлены в условных единицах суточно-сезонные изменения медианного уровня E_{50} атмосферных радиопомех для зимы, весны и лета на частоте приема 12,5 кГц при полосе пропускания 200 Гц.

Как видно из рис. 2, основные зако-

номерности в форме суточного хода огибающей напряженности поля атмосферных радиопомех следующие:

1) максимальная интенсивность шума летом, а минимальная — зимой;

2) в летнее время имеется глубокий минимум, приходящийся на 14 ч 30 мин местного времени.

Исследования проводились с помощью измерительно-вычислительного комплекса, представленного на рис. 3.

Интересно отметить, что измерения естественных радишумов очень низкочастотного диапазона в одно и то же время проводились как на штыревую антенну, так и на магнитную антенну, пермалловый стержень которой был ориентирован в направлении север-юг. Диаграмма направленности этой антенны по электрической составляющей принимаемого сигнала в горизонтальной плоскости представляла восьмерку, а по пространству — тороид со стянутым в точку центральным отверстием,

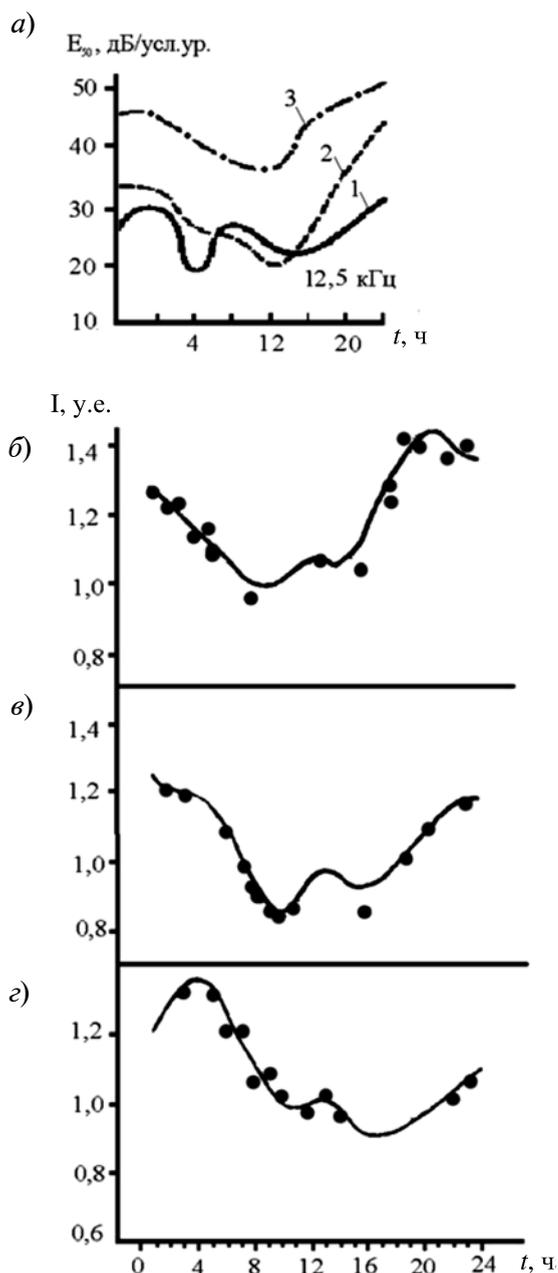


Рис. 2. Сезонно-суточные изменения медианного уровня E_{50} огибающей атмосферных радиопомех в Магадане (а), усредненный суточный ход очень низкочастотного регулярно шумового фона (I, у. е.) в магнитоспокойные периоды летом (б), в равноденствие (в) и зимой (г)

половина которого находилась на поверхности Земли с максимумом диаграммы направленности в меридиальной плоскости, рассекающей тороид в направлении запад-восток.

На рис. 2 [4] представлена усредненная интенсивность суточного хода очень низкочастотного регулярно шумового фона (ОНЧ РШФ) в магнитоспокойные периоды для лета (б), равноденствия (в) и зимы (г). Отметим, что установка на рис. 3 соответствует по своим параметрам аппаратному комплексу, применяемому для этих целей в материалах Отчета № 322 Международного консультативного комитета по радиосвязи (МККР) [3]. Сигнал принимается на штыревую антенну длиной 5 м, подключенную через антенный усилитель с полосой пропускания 3–150 кГц к узкополосному приемнику. Далее сигнал поступает на статистический анализатор и передает информацию об измеренном напряжении случайного процесса в полосе частот 30 кГц и количестве превышений огибающей случайного процесса для определенных конкретными условиями пороговых уровней анализатора на аналого-цифровой преобразователь ЭВМ, что позволяет исследовать статистические свойства естественного ОНЧ радишума по разработанным алгоритмам.

Для описания статистических свойств амплитудных шумов с помощью дополнительной интегральной функции распределения вероятностей используется обобщающая эмпирическая модель (ОЭМ) [5]:

$$P(E) = \begin{cases} \left[1 + \left(\frac{E_0}{E_{50q1}} \right)^{q_1} \right]^{-1}, & 0 \leq E_0 \leq E'_0 \\ \left[1 + \left(\frac{E_0}{E_{50q2}} \right)^{q_2} \right]^{-1}, & E'_0 \leq E_0 \leq \infty, \end{cases} \quad (1)$$

где E – напряжение огибающей поля атмосферного радишума; E_0 – некоторый пороговый уровень напряженности поля; q_1, q_2, b, E'_0 – некоторые параметры модели, причем $q_2 > q_1$.

В данной работе предложен алгоритм (рис. 4) расчета статистических амплитудных распределений вероятностей превышения порогового уровня для атмосферного узкополосного шума: 1 – ввод количества пороговых уровней i ; 2 – цикл от 1 до k с шагом 1; 3 – массивы распределения плот-

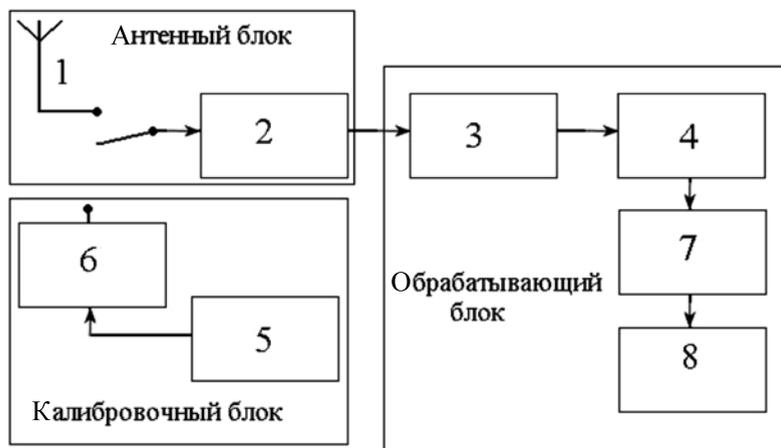


Рис. 3. Измерительно-вычислительный комплекс для исследования статистических свойств атмосферных радишумов:

- 1 – штыревая антенна; 2 – антенный усилитель; 3 – узкополосный приемник;
4 – статистический анализатор; 5 – калибровочный генератор нормального шума;
6 – эквивалент антенны; 7 – АЦП; 8 – ЭВМ

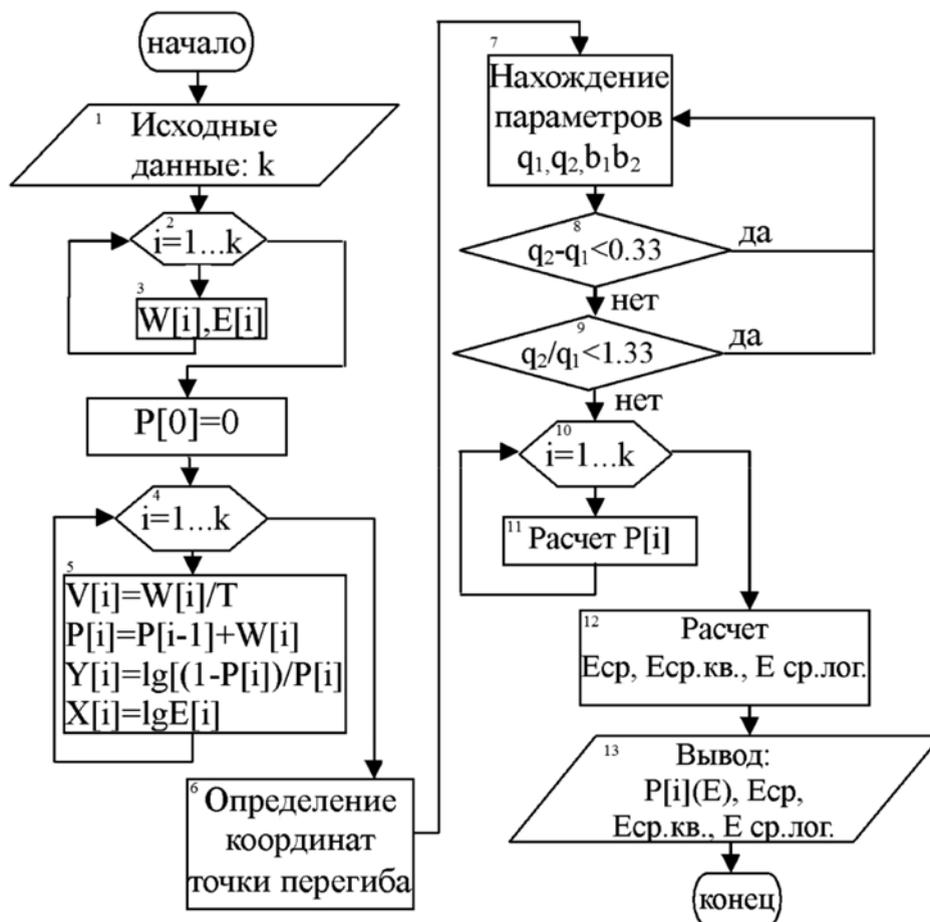


Рис. 4. Алгоритм расчета распределения вероятностей по ОЭМ

ности вероятностей W_i огибающей превышения напряженности E_i каждого порогового уровня i ; 4 – цикл от 1 до k с шагом 1; 5 – расчет распределения вероятностей P_i огибающей, пересчет распределения вероятностей в необходимую систему координат; 6 – определение координат точки перегиба; 7 – нахождение параметров q_1, q_2, b_1, b_2 , с помощью двух прямых аппроксимирующих распределения вероятностей; 8, 9 – проверка условий на описание распределения вероятностей; 10 – цикл от 1 до k с шагом 1; 11 – расчет распределения вероятностей по ОЭМ; 12 – расчет статистических параметров E ср., E ср. кв. и E ср. лог.; 13 – вывод функции вероятно-

стей огибающей и среднего, среднеквадратичного и среднелогарифмического значений напряженности поля.

Для представления статистических свойств узкополосных радишумов грозового происхождения в работе [5] предложена ОЭМ, хорошо описывающая экспериментальные результаты (см. рис. 1) в диапазоне частот от 3 кГц до 10 МГц. Плюсом ОЭМ является возможность описания поля атмосферных помех как в присутствии, так и в отсутствии близких гроз, что позволяет широко использовать данную модель для практического применения [3, 4].

СПИСОК ЛИТЕРАТУРЫ

1. **Watt A.D., Maxwell E.L.** Measured statistical characteristics of VLF atmospheric noise // *Proc. JRE*, 1957, Vol. 45, Pp. 55–62.

2. **Осинин В.Ф., Осинин И.В., Подлесных Д.А., Шарапов С.И.** Методика и некоторые результаты исследований статистических свойств естественных ОНЧ радиополей в центральном Черноземье // *Информационные технологии в процессе подготовки современного специалиста: Межвузовский сб.* Липецк: Изд-во ЛГПУ, 2006. Вып. 9. Т. 2. С. 46–54.

3. **Осинин В.Ф.** Статистические характеристики атмосферных радиопомех в Магадане // *Геофизические явления верхней атмосферы и земной коры. Труды СВКНИИ ДВНЦ АН СССР.* Магадан, 1973. Вып. 47. С. 28–39.

4. World distribution and characteristics of atmospheric radio noise // 10th Plenary Assembly, Int. Telecommun. Union. Int. Rad. Consult. Comm. Geneva, 1964. Report 322.

5. **Горшков Ю.Н.** О связи ОНЧ регулярного шумового фона с ионосферой и магнитной активностью // *Геофизические явления верхней атмосферы и земной коры. Труды СВКНИИ ДВНЦ АН СССР.* Магадан, 1973. Вып. 47. С. 59–66.

6. **Осинин В.Ф., Четвериков С.Ф., Шарапов С.И., Гамова Л.Г., Коваленко И.А.** Математическая модель представления функции распределения атмосферного радишума // *Вести высших учебных заведений Черноземья.* Липецк: Изд-во ЛГТУ, 2013. № 2. С. 52–59.

REFERENCES

1. **Watt A.D., Maxwell E.L.** Measured statistical characteristics of VLF atmospheric noise. *Proc. JRE*, 1957, Vol. 45, Pp. 55–62.

2. **Osinin V.F., Osinin I.V., Podlesnykh D.A., Sharapov S.I.** Metodika i nekotorye rezul'taty issledovaniy statisticheskikh svoystv estestvennykh ONCh radiopolei v tsentral'nom Chernozem'e, *Informatsionnye tekhnologii v protsesse podgotovki sovremenogo spetsialista: Mezhvuzovskii sbornik.* Lipetsk: LGPU Publ., 2006, Vol. 9-2, Pp. 46–54. (rus)

3. **Osinin V.F.** Statisticheskie kharakteristiki atmosferykh radiopomekh v Magadane, *Geofizicheskie iavleniia verkhnei atmosfery i zemnoi kory. Trudi SVKNII DVNTs AN SSSR.* Magadan, 1973, Vol. 47, Pp. 28–39. (rus)

4. World distribution and characteristics of atmospheric radio noise, *10th Plenary Assembly, Int. Telecommun. Union. Int. Rad. Consult. Comm. Geneva*, 1964, Report 322.

5. **Gorshkov Yu. N.** O sviazi ONCh reguliarnogo shumovogo fona s ionosferoi i magnitnoi aktivnost'iu, *Geofizicheskie iavleniia verkhnei atmosfery i zemnoi kory. Trudi SVKNII DVNTs AN SSSR.* Magadan, 1973, Vol. 47, Pp. 59–66. (rus)

6. **Osinin V.F., Chetverikov S.F., Sharapov S.I., Gamova L.G., Kovalenko I.A.** Matematicheskaia model' predstavleniia funktsii raspredeleniia atmosfernogo radioshuma, *Vesti vysshikh uchebnykh zavedenii Chernozem'ia.* Lipetsk: LGTU Publ., 2013, No. 2, Pp. 52–59. (rus)

ЧЕТВЕРИКОВ Сергей Федорович – аспирант кафедры физики и биомедицинской техники Липецкого государственного технического университета.

398600, Россия, г. Липецк, ул. Московская, д. 30.

E-mail: tschetserg@yandex.ru

CHETVERIKOV, Sergey F. *Lipetsk State Technical University.*

398600, Moskovskaya Str. 30, Lipetsk, Russia.

E-mail: tschetserg@yandex.ru

УДК 621.391

*Н.И. Червяков, М.Г. Бабенко, П.А. Ляхов,
И.Н. Лавриненко, А.М. Лягин*

**УМНОЖЕНИЕ И ДЕЛЕНИЕ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ
С ИСПОЛЬЗОВАНИЕМ ПОЛЕЙ ГАЛУА $GF(p)$**

*N.I. Chervyakov, M.Gr. Babenko, P.A. Lyakhov,
I.N. Lavrinenko, A.M. Lyagin*

**MULTIPLICATION AND DIVISION IN THE RESIDUE NUMBER SYSTEM
USING GALOIS FIELDS $GF(p)$**

Предложен алгоритм умножения и деления в системе остаточных классов, основанный на теории полей Галуа $GF(p)$. Применение полей Галуа $GF(p)$ для решения проблем арифметического умножения и деления устраняет многие ограничения существующих алгоритмов. Преимущество предложенного алгоритма заключается в том, что он не имеет ограничений на делимое и делитель, не использует обобщенную позиционную систему счисления и расширение системы остаточных классов.

КОМПЬЮТЕРНАЯ АРИФМЕТИКА; СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ; МОДУЛЯРНАЯ АРИФМЕТИКА; ПОЛЯ ГАЛУА; ТЕОРИЯ ЧИСЕЛ; ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ.

The current paper presents an algorithm of multiplication and division in the residual classes based on the theory of Galois fields $GF(p)$. The use of Galois fields $GF(p)$ to solve the problems of arithmetic multiplication and division eliminates a lot of limitations of existing algorithms. The advantage of the proposed algorithm is that it has no restrictions on the dividend and the divisor and it does not use the generalized positional notation and the expansion of the residue number systems.

COMPUTER ARITHMETIC; RESIDUE NUMBER SYSTEM; MODULAR ARITHMETIC; GALOIS FIELD; NUMBER THEORY; PARALLEL COMPUTING.

Проблема деления в СОК в общем виде привлекает внимание многих исследователей при разработке высокопроизводительных многомодульных арифметикологических устройств (АЛУ). Цифровые системы, построенные на основе арифметики СОК, могут сыграть важную роль в высокоскоростных системах обработки данных в режиме реального времени, с поддержкой параллельной обработки целочисленных данных [1].

Некоторые алгоритмы деления в СОК

в общем случае были разработаны ранее. Эти алгоритмы можно разделить на две категории: с использованием умножения и с использованием вычитания [2, 3]. Большинство алгоритмов на основе умножения предварительно вычисляют обратную величину делителя, после чего эта величина умножается на делимое. Алгоритмы на основе вычитания используют вычитание кратных делителя из делимого до тех пор, пока полученный результат не будет меньше делителя. Некоторые известные алго-

ритмы деления в СОК на основе умножения изложены в [4–8]. Все эти алгоритмы используют преобразование в обобщенную позиционную систему счисления (ОПСС) для нахождения обратной величины делителя и сравнения чисел. Кроме того, эти алгоритмы являются медленными, т. к. требуют выполнения большого количества арифметических действий. Некоторые алгоритмы на основе вычитания представлены в [9, 10]. Эти алгоритмы не требуют вычислений в ОПСС, однако используют некоторые другие немодульные операции [11, 12]. Алгоритм с вычитанием, представленный в [10], кажется наиболее привлекательным для применения на практике, поскольку использует эффективный метод проверки четности для сравнения и обнаружения переполнения диапазона. Большинство существующих алгоритмов обладают различными недостатками, что делает их менее пригодными для решения задачи деления в СОК.

Анализ известных алгоритмов деления в СОК показывает, что большинство из них работает на основе метода спуска Ферма и итераций Ньютона, каждая из которых содержит трудные операции: масштабирования, расширения, округления и другие, включающие большое количество арифметических действий.

Так как эти методы являются итерационными, то их сравнение можно провести по двум критериям:

по общему количеству выполненных итераций;

по количеству операций, выполненных за одну итерацию

Сравнительная оценка по количеству итераций, выполненных, в алгоритмах Ферма и Ньютона приведена на основе компьютерного моделирования, результаты которого показаны на рис. 1 и 2.

Вычислительные средства, построенные на основе арифметики системы остаточных классов (СОК), могут сыграть важную роль в высокоскоростных и надежных системах обработки данных в режиме реального времени, с поддержкой параллельной обработки целочисленных данных. Операции сложения, вычитания и умножения,

называемые *модульными операциями*, могут быть реализованы очень быстро, без распространения межразрядных переносов. К немодульным относятся операции деления, сравнения чисел, определения знака, определения переполнения динамического диапазона и др. Любое улучшение скорости этих медленных операций значительно улучшит производительность многомодульных арифметико-логических устройств.

В десятичной арифметике логарифмы часто используются для умножения и деления. В СОК используется аналогичный метод, называемый *индексными вычислениями* [13]. Использование индексного преобразования над полем Галуа $GF(p)$ сводит операции умножения и деления к операциям сложения и вычитания, соответственно. Операция умножения является модульной и поэтому может быть реализована в СОК как сложение. С точки зрения аппаратной реализации, сложение в СОК реализуется легче, чем умножение [13, 14]. Деление, напротив, является немодульной операцией в СОК, поэтому использование индексных преобразований над $GF(p)$ значительно улучшит время выполнения операции и снизит аппаратные затраты.

При построении умножителей, работающих в модулярной арифметике, можно воспользоваться индексным или «дискретно-логарифмическим» представлением операндов и заменить операцию модулярного умножения операцией модулярного сложения индексов или дискретных логарифмов. Индексное представление модулярного числа основывается на понятии первообразного «корня по простому модулю». Таким корнем является целое число, возведение которого в степень $1, 2, \dots, p-1$ дает неповторяющиеся вычеты по модулю [13].

Индексный метод позволяет получить очень простую реализацию с помощью метода, аналогичного методу вычисления логарифмов. Умножители по модулю p_i реализуются с помощью табличного поиска при малом p_i (5 бит и менее) и индексного сложения при больших p_i (6–10 бит) [14]. Поэтому в тех случаях, когда диапазон обрабатываемых данных лежит в преде-

лах 6–10 бит, целесообразно использовать умножители и деления на основе индексного исчисления. Однако индексное исчисление может быть использовано только в том случае, когда модуль p_i – простое число.

На практике встречаются также случаи, когда модуль p_i – произвольное число, поэтому появляется необходимость разработать эффективные методы умножения, пригодные при любом модуле, независимо от того, является ли последний простым числом или нет. Так, например, если реализовать умножитель по модулю $p_i = 249$, который не является простым числом, т. к. $249 = 83 \cdot 3$, то непосредственное использование индексного счисления невозможно. Так как числа 83 и 3 являются простыми, то возможно индексное исчисление по разложенным модулям.

При умножении (делении) по модулю p_i сначала определяются индексы чисел, затем эти индексы складываются (вычитаются) по модулю $p_i - 1$ и по обратной таблице определяется окончательный результат [11, 12, 15].

Если модуль не является простым, то необходимо разложить его на меньшие модули, провести вычисления с помощью обычных методов, а затем вновь перейти к

исходному модулю. Метод индексного исчисления эффективен при обработке данных 6–10 бит и когда модуль представляет собой простое число [14].

Сравнительная оценка общего количества итераций, выполняемых в алгоритмах Ферма и Ньютона, приведена на рис. 1 и 2.

На рис. 1 приведена сравнительная оценка в случае, когда значение делителя фиксировалось, а значение делимого менялось. На рис. 2 наоборот, неизменным оставалось значение делимого, делитель принимал различные значения.

Проведенная сравнительная оценка показала, что количество итераций метода Ферма растет вместе с ростом разницы между разрядностями делимого и делителя. Количество итераций метода Ньютона от этого не зависит. Поэтому, если разница между разрядностями делимого и делителя является достаточно большой, то в этом случае хороший результат дает применение метода Ньютона. Однако если разница небольшая, то выгоднее применять для выполнения операции деления метод Ферма.

Количество операций в каждой итерации в основном определяются операциями расширения и масштабирования.

Операция масштабирования состоит в выполнении деления с нулевым остатком

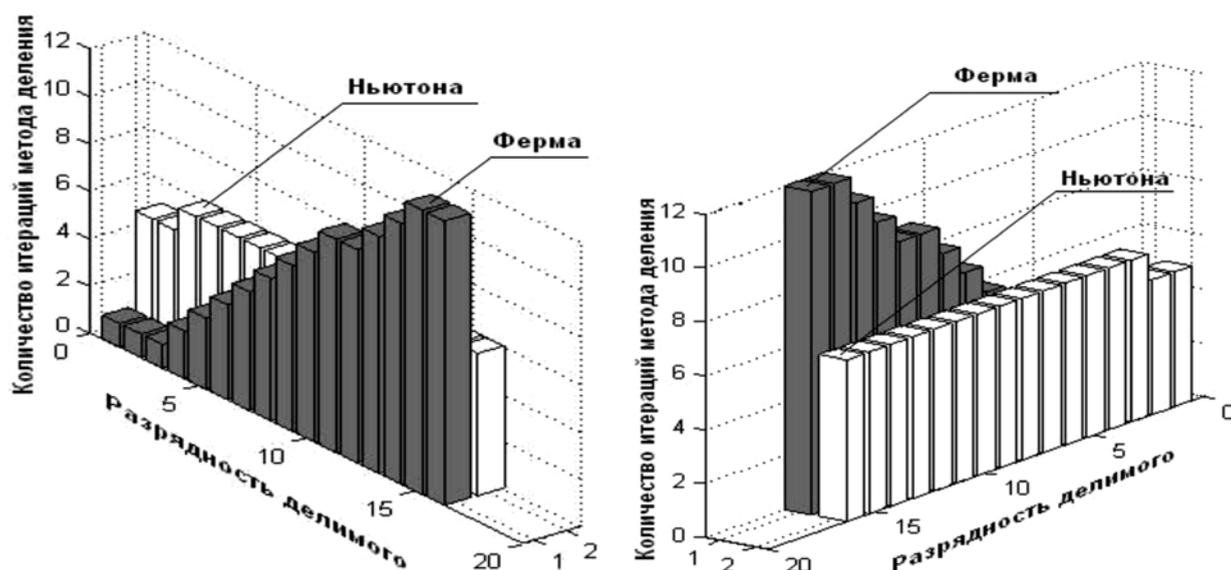


Рис. 1. Оценка количества итераций методов Ферма и Ньютона в зависимости от разрядности делимого

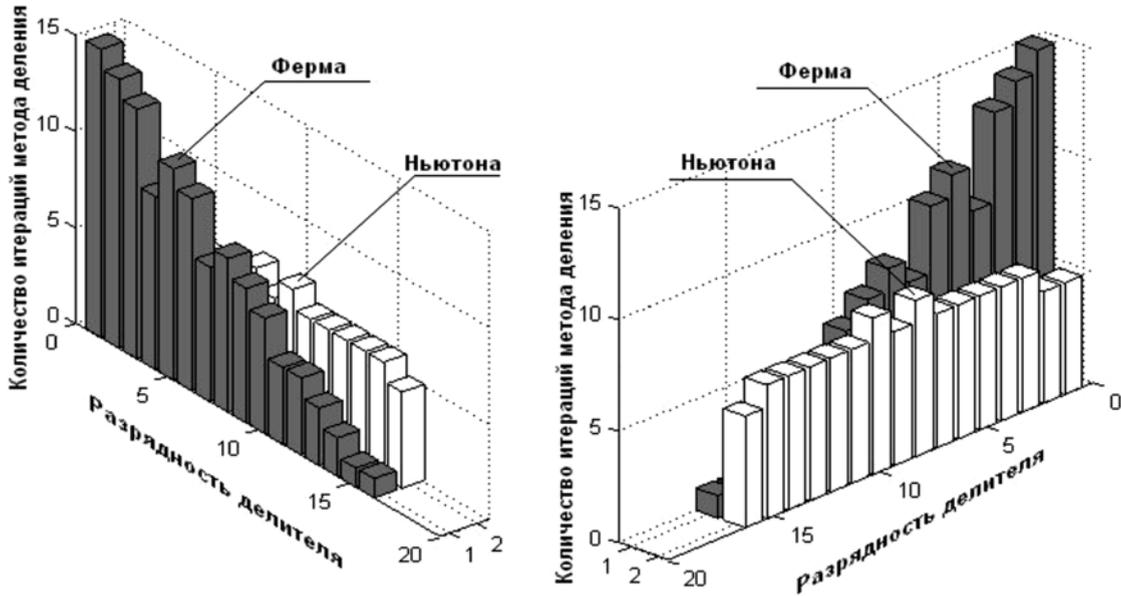


Рис. 2. Оценка количества итераций методов Ферма и Ньютона в зависимости от разрядности делителя

(Z вычитаний и Z сложений, где Z – количество модулей СОК, по которым проводится масштабирование).

Операции расширения выполняются на основе метода Гарнера, который содержит r операций вычитания и r операций умножения, где r – количество модулей исходной СОК.

Трудность выполнения операций модулярного деления, связанная с итерационными процессами, наводит на мысль об исключении итераций при делении изоморфизмом, позволяющем операцию деления заменить операцией вычитания остатков делимого и делителя как разность их остатков.

В настоящей статье рассматривается очень быстрый алгоритм деления в СОК с использованием индексов в $GF(p)$. Улучшенный алгоритм обладает следующими свойствами: очень быстр по сравнению с известными алгоритмами, не использует предварительную оценку частного, обратную величину для делителя и операцию расширения базы СОК.

Постановка задачи

Модулярная арифметика обладает большими возможностями обработки данных

за счет дробления операндов на несколько более мелких остатков и независимого параллельного выполнения арифметических операций сложения, вычитания и умножения с остатками.

Большой интерес к системе остаточных классов в последнее время обусловлен внедрением в цифровых системах микроэлектроники с высокой синхронизирующей частотой и низким потреблением энергии. Поскольку система остаточных классов не является взвешенной системой счисления, то операция деления, включающая такие операции, как сравнение, расширение, масштабирование и другие, не может считаться простой. Все известные алгоритмы деления характеризуются использованием итерационных принципов, что существенно снижает способность обработки данных. Для сокращения времени выполнения операции деления желательно исключить итерации. Это можно реализовать с помощью полей Галуа. Для этого необходимо использовать изоморфизм, который позволяет операцию деления заменить операцией вычитания остатков делимого и делителя как разность их остатков. В данной работе предлагается алгоритм модулярного деления без использования итерационных процессов.

Индексы в полях Галуа GF(p)

Таблица 1

Число I , являющееся решением сравнения $q^x \equiv A \pmod{p}$, называется индексом числа A и обозначается $I = \text{ind } A$. Первообразный корень q называется основанием индекса. Для нахождения индекса числа A по модулю p надо найти первообразный корень q и затем найти решение этого сравнения для данного первообразного корня.

Пример 1. Вычислить индексы по модулю 7 чисел 0, 1, 2, 3, 4, 5, 6. Первообразные корни числа 7 суть 3 и 5.

Решение. Примем основание $q = 3$. Вычислим 3^x для $x = 0, 1, 2, 3, 4, 5, 6$.

$$\begin{aligned} 3^0 &\equiv 1 \pmod{7}, & 3^1 &\equiv 3 \pmod{7}, & 3^2 &\equiv 2 \pmod{7}, \\ 3^3 &\equiv 6 \pmod{7}, & 3^4 &\equiv 4 \pmod{7}, & 3^5 &\equiv 5 \pmod{7}, \\ & & 3^6 &\equiv 6 \pmod{7}. \end{aligned}$$

Из этих сравнений следует, что

$$\begin{aligned} \text{ind } 1 &= 0, & \text{ind } 2 &= 2, & \text{ind } 3 &= 1, & \text{ind } 4 &= 4, \\ & & \text{ind } 5 &= 5, & \text{ind } 6 &= 3. \end{aligned}$$

Аналогично, индексы по первообразному корню 5:

$$\begin{aligned} \text{ind } 1 &= 0, & \text{ind } 2 &= 4, & \text{ind } 3 &= 5, & \text{ind } 4 &= 2, \\ & & \text{ind } 5 &= 1, & \text{ind } 6 &= 3. \end{aligned}$$

Антииндексом числа I называется число a , такое что

$$I = \text{ind } a \text{ или } a = \text{ind}^{-1}I.$$

Если антииндекс обозначить через $N(I)$, то $N(\text{ind } a) = a$.

Пример 2. Вычислить антииндексы по модулю 7 чисел 0, 1, 2, 3, 4, 5.

Решение. В предыдущем примере приведены индексы чисел от 0 до 6, которые равны 0, 1, 2, 3, 4, 5. Естественно, что число нуль не может иметь индекса, потому что нет такого показателя степени, возведя в которую конечное, отличное от нуля основание, можно было бы получить нуль.

Первообразный корень 3:

$$\begin{aligned} N(0) &= 1, & N(1) &= 3, & N(2) &= 2, & N(3) &= 6, \\ & & N(4) &= 4, & N(5) &= 5. \end{aligned}$$

Первообразный корень 5:

$$\begin{aligned} N(0) &= 1, & N(1) &= 5, & N(2) &= 4, & N(3) &= 6, \\ & & N(4) &= 2, & N(5) &= 3. \end{aligned}$$

При помощи индексов можно вычис-

Индексы по модулю 7

I	q	
	3	5
0	—	—
1	0	0
2	2	4
3	1	5
4	4	2
5	5	1
6	3	3

лять умножение, деление, возведение в степень.

Пример 3. Вычислить выражение

$$c = \frac{ab^3}{k^2d} \pmod{p},$$

где $a = 2, b = 5, k = 3, d = 4, p = 7$.

Решение.

1. Находим индексы величин, входящих в вычисляемое выражение, приняв первообразный корень равным 5.

Из табл. 1 индексов по модулю 7 находим:

$$\text{ind } 2 = 4, \text{ ind } 5 = 1, \text{ ind } 3 = 5, \text{ ind } 4 = 2.$$

2. Вычисляем индекс результата:

$$\text{ind } c = 4 + 1 \cdot 3 - 5 \cdot 2 - 2 \pmod{6} \equiv 1 \pmod{6}.$$

3. Находим антииндекс 1: $N(1) = 5$.

Непосредственное вычисление показывает, что

$$c = \frac{2 \cdot 5^3}{3^2 \cdot 4} \pmod{7} = 5.$$

Конечные поля Галуа бывают двух типов: простые поля $\text{GF}(p)$ и полиномиальные поля $\text{GF}(p^n)$, где p — простое число, n — целое положительное. Все поля Галуа обладают свойством: все ненулевые элементы могут быть построены с использованием примитивного элемента, который будем обозначать g . Это свойство может быть использовано при делении над полем $\text{GF}(p)$. Свойство определяется следующим образом [16]:

если p — любое простое число, а g — любой примитивный корень в $\text{GF}(p)$, то для

каждого целого a , взаимно простого с p , существует единственное целое i , обозначаемое $i = \text{ind}_g a$, такое что

$$a = |g^i|_p, 0 \leq i \leq p-1. \quad (1)$$

Индексы над полем $\text{GF}(p)$ обладают следующими важными свойствами:

- (1) $\text{ind}_g 1 = 0$,
- (2) $\text{ind}_g(ab) = |\text{ind}_g a + \text{ind}_g b|_{p-1}$,
- (3) $\text{ind}_g a^k = |k \cdot \text{ind}_g a|_{p-1}$,
- (4) $\text{ind}_g a = |\text{ind}_g g' + \text{ind}_g a|_{p-1}$, где g' – любой другой примитивный корень.

В тех случаях, когда сумма индексов превышает самое большое значение в $\text{GF}(p)$ используется теорема Ферма. При этом:

если p – простое число, то

$$|a^p|_p = |a|_p, \text{ для всех целых } a; \quad (2)$$

если p простое число и a – целое, то

$$|a^{p-1}|_p = 1; \quad (3)$$

если p – простое число, k и a – целые числа, то

$$|a^k|_p = |a^{|k|_{p-1}}|_p. \quad (4)$$

В СОК проблема арифметического деления делится на три категории:

- деление с нулевым остатком (ДН);
- масштабирование;
- деление в общем случае.

Деление с нулевым остатком и масштабирование применяются для ограниченного круга приложений [17]. Для деления с нулевым остатком необходимо знать априори, что остаток равен нулю. Масштабирование реализует деление только на фиксированный делитель, при этом делитель может представляться в виде произведения нескольких модулей. Таким образом, общее деление используется, когда неизвестно априори, что делимое делится нацело на делитель, и когда неизвестно, что делитель принадлежит множеству пригодных для масштабирования значений. Эффективный алгоритм деления на основе приближенного метода рассмотрен в работах [18, 19].

Алгоритм деления с нулевым остатком над полем Галуа $\text{GF}(p)$

Деление с нулевым остатком требует

вычисления частного $\left| \frac{x}{y} \right|_p$, когда известно априори, что остаток равен нулю. Поле Галуа $\text{GF}(p)$ может быть использовано весьма эффективно при решении проблемы деления с нулевым остатком. Так, если $\{q_n\} = \{1, 2, \dots, p-1\}$ – мультипликативная группа $\text{GF}(p)$, $\{i_n\} = \{0, 1, \dots, p-2\}$ – ассоциативная изоморфная группа с обозначением $q_n = |q^{i_n}|_p$, где g – примитивный элемент $\text{GF}(p)$, а q_x и q_y – два целых числа, таких что q_y делит q_x , тогда деление с нулевым остатком $\left| \frac{q_x}{q_y} \right|_p$ может быть определено как

$$\left| \frac{q_x}{q_y} \right|_p = \left| g^{|\text{ind}_g q_x - \text{ind}_g q_y|_{p-1}} \right|_p = \left| g^{|\text{ind}_g q_x + (p-1 - \text{ind}_g q_y)|_{p-1}} \right|_p. \quad (5)$$

Если $p-1 = \prod_{j=0}^{r-1} p_j$, так что $(p_0, p_1, \dots, p_{r-1})$ взаимно простые, $\text{ind}_g x = (x_1, x_2, \dots, x_r)$ и $\text{ind}_g y = (y_1, y_2, \dots, y_r)$ и y делит x , арифметическое деление $\left| \frac{x}{y} \right|_p$ может быть определено как

$$\left| \frac{x}{y} \right|_p = \left| g^{(|x_1 - y_1|_{p_1}, |x_2 - y_2|_{p_2}, \dots, |x_r - y_r|_{p_r})} \right|_p. \quad (6)$$

Пример 4. Пусть простое число $p = 43$ и $g = 3$. Группа $\{q_n\} = \{1, 2, \dots, 42\}$ – ассоциативная изоморфная группа $\{i_n\} = \{0, 1, \dots, 41\}$. Так как $p-1 = 42$, возьмем модули $p_1 = 2, p_2 = 3$ и $p_3 = 7$. Любое целое i_n может быть единственным образом представлено в виде тройки $r_n = (|i_n|_2, |i_n|_3, |i_n|_7)$. Множество всех троек образует группу $\{r_n\}$. Изоморфизм между группами $\{q_n\}$ и $\{r_n\}$ показан в табл. 2.

В табл. 3 показан обратный переход к $\text{GF}(43)$.

Вычисление табличных значений покажем для $q_n = 27$. Первый индекс вычисляется следующим образом:

$$i_n = \text{ind}_3 27 = 3, \text{ то есть } 27 = |3^3|_{43} = |3^3|_{43}.$$

Тройка, полученная из i_n :

$$r_n = (|i_n|_2, |i_n|_3, |i_n|_7) = (|3|_2, |3|_3, |3|_7) = (1, 0, 3).$$

В табл. 2 $q_n = 27$ соответствует запись $(1, 0, 3) = (1, 00, 011)_2$. Результат записан

Таблица 2

Изоморфизм между группами $\{q_n\}$ и $\{r_n\}$
 для $GF(43)$, $p = 43$, $g = 3$, $p_1 = 2$, $p_2 = 3$ и $p_3 = 7$

q_n	r_n	q_n	r_n	q_n	r_n	q_n	r_n	q_n	r_n
1 000001	(0, 0, 0) 0 00 000	10 001010	(0, 1, 3) 0 01 011	19 010011	(1, 1, 6) 1 01 110	28 011100	(1, 2, 5) 1 10 101	37 100101	(1, 1, 0) 1 01 000
2 000010	(1, 0, 6) 1 00 110	11 001011	(0, 0, 2) 0 00 010	20 010100	(1, 1, 2) 1 01 010	29 011101	(1, 2, 6) 1 10 110	38 100110	(0, 1, 4) 0 01 100
3 000011	(1, 1, 1) 1 01 001	12 001100	(1, 1, 5) 1 01 101	21 010101	(0, 0, 1) 0 00 001	30 011110	(1, 2, 4) 1 10 100	39 100111	(1, 0, 5) 1 00 101
4 000100	(0, 0, 5) 0 00 101	13 001101	(0, 2, 4) 0 10 100	22 010110	(1, 0, 1) 1 00 001	31 011111	(0, 1, 6) 0 01 110	40 101000	(0, 1, 1) 0 01 001
5 000101	(1, 1, 4) 1 01 100	14 001110	(0, 2, 6) 0 10 110	23 010111	(0, 1, 2) 0 01 010	32 100000	(1, 0, 2) 1 00 010	41 101001	(0, 0, 6) 0 00 110
6 000110	(0, 1, 0) 0 01 000	15 001111	(0, 2, 5) 0 10 101	24 011000	(0, 1, 5) 0 01 101	33 100001	(1, 1, 3) 1 01 011	42 101010	(1, 0, 0) 1 00 000
7 000111	(1, 2, 0) 1 10 000	16 010000	(0, 0, 3) 0 00 011	25 011001	(0, 2, 1) 0 10 001	34 100010	(1, 2, 2) 1 10 010		
8 001000	(1, 0, 4) 1 00 100	17 010001	(0, 2, 3) 0 10 011	26 011010	(1, 2, 3) 1 10 101	35 100011	(0, 0, 4) 0 00 100		
9 001001	(0, 2, 2) 0 10 010	18 010010	(1, 2, 1) 1 10 001	27 011011	(1, 0, 3) 1 00 011	36 100100	(0, 2, 0) 0 10 000		

Таблица 3

Обратное преобразование в $GF(43)$, $p = 43$, $g = 3$, $p_1 = 2$, $p_2 = 3$ и $p_3 = 7$

r_n	q_n	r_n	q_n	r_n	q_n	r_n	q_n	r_n	q_n
(0, 0, 0) 0 00 000 = 0	1	(0, 1, 2) 0 01 010 = 10	23	(0, 2, 4) 0 10 100 = 20	13	(1, 0, 6) 1 00 110 = 38	2	(1, 2, 1) 1 10 001 = 49	18
(0, 0, 1) 0 00 001 = 1	21	(0, 1, 3) 0 01 011 = 11	10	(0, 2, 5) 0 10 101 = 21	15	(1, 1, 0) 1 01 000 = 40	37	(1, 2, 2) 1 10 010 = 50	34
(0, 0, 2) 0 00 010 = 2	11	(0, 1, 4) 0 01 100 = 12	38	(0, 2, 6) 0 10 110 = 22	14	(1, 1, 1) 1 01 001 = 41	3	(1, 2, 3) 1 10 101 = 51	26
(0, 0, 3) 0 00 011 = 3	16	(0, 1, 5) 0 01 101 = 13	24	(1, 0, 0) 1 00 000 = 32	42	(1, 1, 2) 1 01 010 = 42	20	(1, 2, 4) 1 10 100 = 52	30
(0, 0, 4) 0 00 100 = 4	35	(0, 1, 6) 0 01 110 = 14	31	(1, 0, 1) 1 00 001 = 33	22	(1, 1, 3) 1 01 011 = 43	33	(1, 2, 5) 1 10 101 = 53	28
(0, 0, 5) 0 00 101 = 5		(0, 2, 0) 0 10 000 = 16	6	(1, 0, 2) 1 00 010 = 34	2	(1, 1, 4) 1 01 100 = 44		(1, 2, 6) 1 10 110 = 54	9
(0, 0, 6) 0 00 110 = 6	1	(0, 2, 1) 0 10 001 = 17	5	(1, 0, 3) 1 00 011 = 35	7	(1, 1, 5) 1 01 101 = 45	2		
(0, 1, 0) 0 01 000 = 8		(0, 2, 2) 0 10 010 = 18		(1, 0, 4) 1 00 100 = 36		(1, 1, 6) 1 01 110 = 46	9		
(0, 1, 1) 0 01 001 = 9	0	(0, 2, 3) 0 10 011 = 19	7	(1, 0, 5) 1 00 101 = 37	9	(1, 2, 0) 1 10 000 = 48			

в двоичной форме и содержит 1 бит, 2 бита и 3 бита для записи значения по mod 2, mod 3 и mod 7 соответственно.

Вычитание по модулю p выполняется одновременно по всем модулям

$$p - 1 = \prod_{i=1}^r p_i.$$

Существует множество вариантов, из которых можно выбрать набор модулей для СОК систем. Скорость и эффективность деления нацело увеличивается, если подбирать модули на основе следующих критериев:

- 1) $p - 1 = \prod_{i=1}^r p_i$;
- 2) они должны быть взаимно простыми $(p_i, p_j) = 1$, если $i \neq j$;
- 3) $\sum_{j=1}^r \log(p_j - 1) \rightarrow \min$.

Пример 5. При $p = 43$, $g = 3$, $p_1 = 2$, $p_2 = 3$ и $p_3 = 7$. Найти $\frac{36}{2}$. Это задача деления с нулевым остатком.

По табл. 2 получаем:

$$\begin{aligned} \text{для } 36: \text{ind}_3 36 &= (0, 2, 0); \\ \text{для } 2: \text{ind}_3 2 &= (1, 0, 6). \end{aligned}$$

Используя вычитание по модулям 2, 3 и 7, получаем

$$\text{ind}_3 \left(\frac{36}{2} \right) = (0, 2, 0) - (1, 0, 6) = (1, 2, 1),$$

Из табл. 3 следует, что результат деления чисел $\frac{36}{2}$ равен

$$(1, 2, 1) = (1 \ 10 \ 001)_2 = 49.$$

Схема для реализации процесса с нулевым остатком представлена на рис. 3. Реализация операции сложения проще, чем операции вычитания, поэтому используется дополнительный код делителя, позволяющий заменить операцию вычитания операцией сложения.

На вход $LUT_1 p_i$ поступает двоичный код делимого, а на вход $LUT_2 p_i$ — двоичный код делителя. На выходе $LUT_1 p_i$ формируется прямой код индекса делимого, а на вы-

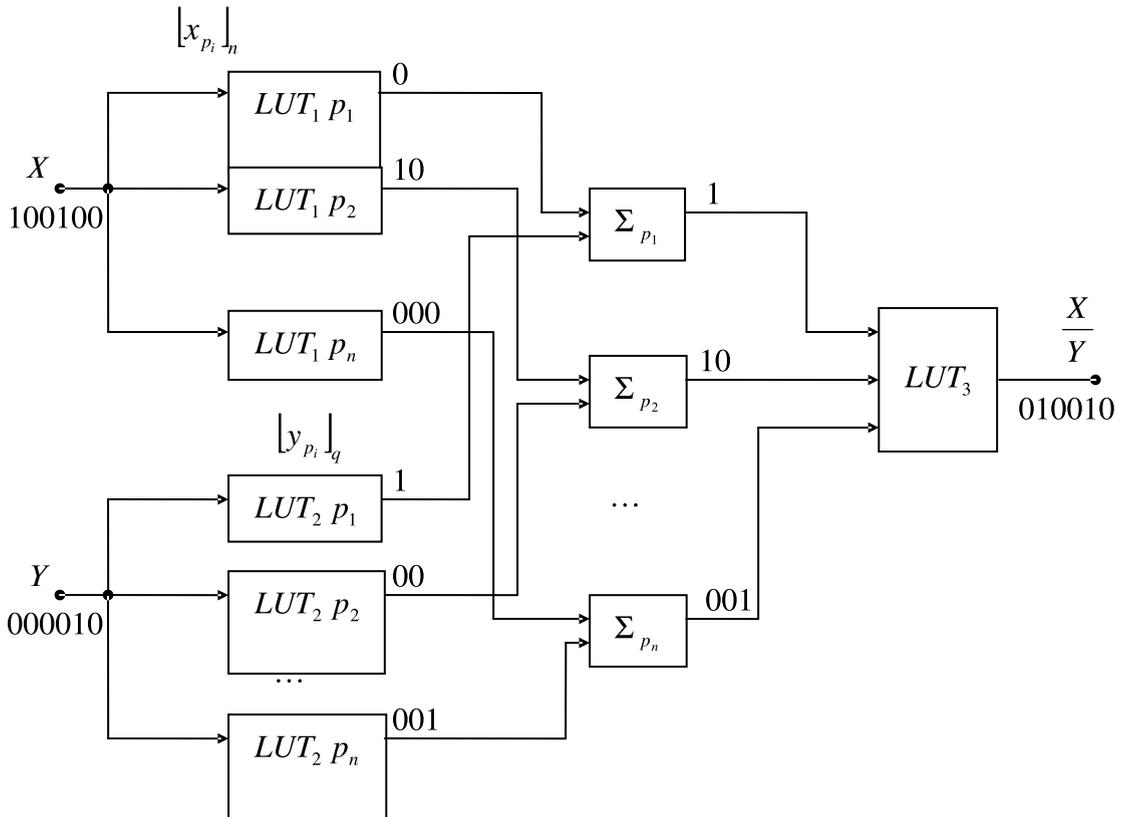


Рис. 3. Деление без остатка по модулям СОК в $GF(p)$ при n двоичных разрядах

ходе $LUT_2 p_i$ формируется дополнительный код индекса делителя. Индексы делимого и делителя суммируются модульными сумматорами Σ_{p_i} и поступают на вход $LUT_3 p_i$, где формируется частное $\frac{X}{Y}$.

На рис. 3 показаны конкретные значения индексов и антииндексов для примера 5.

Масштабирование чисел может быть выполнено как последовательность решения задач деления с нулевым остатком, если делитель является произведением некоторых модулей, причем порядок деления с нулевым остатком не имеет значения.

Общее арифметическое деление над полем Галуа $GF(p)$ используется в случае, когда заранее неизвестно, какой делитель представляет кратное число делимого, или когда делимое есть произведение некоторых модулей. Общая проблема деления $\frac{X}{Y}$ может быть сведена к процессу масштабирования $\frac{X}{Y'}$, где Y' удовлетворяет следующим условиям:

1. $Y' > Y$.

2. Y' является произведением некоторых модулей СОК.

Выбор Y' подробно рассмотрен в [11].

Оценим вычислительную сложность предлагаемого алгоритма и сравним с известным алгоритмом, предложенным в работе [4]. Анализ сложности алгоритма базируется на числе модулярных вычислений и логической глубине (количество последовательно соединенных модульных операций). Вычислительная сложность определяется вычислением в полях Галуа $GF(p)$, которые выполняются параллельно. В представленном алгоритме логическая глубина равна трем и имеет представление $LUT_{(1,2)} P \rightarrow \sum p_i \rightarrow LUT_3 p$, при этом длительность выполнения операции деления $t_g = t'_1 + t_2 + t''_1$, где $t'_1 = t''_1 = O(\log p)$ – такт синхронизации доступа к памяти, $t_2 = O\left(\sum_{i=1}^r \lceil \log p_i \rceil\right)$ – такт синхронизации модульного сложения. Разрядность данных $R = O(\lceil \log p_i \rceil)$. Реализация таблиц $LUT_{(1,2)}$ требует памяти объемом $M = n2^n$.

В известном алгоритме [4] используется одна операция сложения, одна опе-

рация деления и процедура расширения базы, состоящая в нахождении остатков представления числа по модулю делителя. Эта проблема по сути межмодульная и в классическом подходе ее сложность линейно зависит от числа модулей, т. к. она включает все остатки текущего представления и вычисляется последовательно по алгоритму Гарнера в котором необходимы ряд шагов того же порядка, что и r , где r – количество модулей СОК. При этом каждый шаг состоит из операций сложения и умножения, сложность которых составляют $O\left(2 + 2r \cdot \sum_{i=1}^r \lceil \log p_i \rceil\right)$. Логическая глубина равна $2r + 2$.

Вычислительная сложность предложенного метода без учета доступа к памяти примерно в n раз меньше, чем известного. Логическая глубина представленного метода равна трем, а известного – $2r + 2$. Современная вычислительная база позволяет приблизить время доступа к времени вычисления выполнения модульных операций в СОК.

В современных ПЛИС Xilinx время распространения сигнала через LUT -таблицу составляет около 0,5 нс, через блок метки ускоренного переноса – 0,1 нс, а время переключения триггера – до 0,5 нс.

Время суммирования двух 16-разрядных операндов составляет 5 нс.

Время доступа ROM на основе таблиц, близкое к 10 нс [20].

Программируемые ПЗУ семейства XC 4000 решают функции комбинационного устройства, поэтому обладают высокой скоростью.

При использовании таблиц для суммирования двух операндов, реализованных на базе ПЗУ, общая емкость ПЗУ представляется в виде выражения $V_{ПЗУ} = \left(\sum_{i=1}^r p_i \log_2 p_i\right)_{bit}$. Быстродействие сумматоров определяется максимальным модулем p_i .

При нежестком допущении приведем сравнительную оценку по быстродействию предлагаемого алгоритма с известными.

Быстродействие предлагаемого алгоритма t_A определяется как $t_A = t_A(LUT_{(1,2)}) + t_2 +$

+ $t_A(LUT_3) = 10 \text{ нс} + 5 \text{ нс} + 10 \text{ нс} = 25$.

Быстродействие известных алгоритмов в основном определяется операцией расширения, которая базируется на алгоритме Гарнера и включает r операций вычитания и n операций умножения. Тогда быстродействие одной итерации известных алгоритмов t'_A определяется как $t'_A = nt_a + nt_y$. Для нашего случая $n = 3$, $t_a = 5 \text{ нс}$ и $t_y = 0,5 \text{ нс}$ (в качестве умножителей на константу используются LUT -таблицы), тогда $t'_A = 3 \cdot 5 + 3 \cdot 1,5 = 15 + 4,5 = 19,5 \text{ нс}$.

При делении чисел разрядностью 6–10 бит необходимо (рис. 1 и 2) 6–8 итераций, тогда быстродействие известных алгоритмов $t'_A \approx (6 - 8) \cdot 19,5 \text{ нс} = (117 - 156) \text{ нс}$.

Соотношение $\frac{t'_A}{t_A} \approx (5 - 6)$ в пользу предложенного алгоритма арифметического деления над полем Галуа. При увеличении количества модулей p , соотношение $\frac{t'_A}{t_A}$ растет, и уже при $n > 5$ выигрыш в скорости вычислений представленного метода выше, чем известного.

Предложенный метод модулярного деления на основе использования индексов легко реализуем на специальных наборах оснований СОК и может дать значительное преимущество не только в тех приложениях, в которых основная доля вычислений приходится на точное умножение, возведение в степень в сочетании со сложением и вычитанием, но и в тех приложениях, в которых часто появляется необходимость в делении, сравнении и определении знака числа.

Рассмотренный алгоритм над полем Галуа $GF(p)$ обеспечивает эффективность общего деления в системе остаточных классов. Полученные новые результаты эффективного выполнения модулярного деления в диапазоне 6–10 бит являются развитием теории математических основ разработки и проектирования точных и безошибочных теоретико-числовых арифметико-логических устройств процессоров, функционирующих в СОК.

Исследования выполнены при финансовой поддержке РФФИ в рамках научного проекта № 13-07-00478-а и ФЦП № 14.В37.21.1128.

СПИСОК ЛИТЕРАТУРЫ

1. Claudio E.D., Piazza F., Orlandi G. Fast combinatorial RNS processors for DSP applications // IEEE Trans. Comput. 1995. Vol. 44. No. 5. Pp. 624–633.
2. Waser S., Flynn M.J. Introduction to Arithmetic for Digital System Designers. Holt, Rinehart Winston. New York, 1982.
3. Taylor F.J. Residue arithmetic: A tutorial with examples // IEEE Comput. 1984. Vol. 17. No. 5. Pp. 50–62.
4. Червяков Н.И., Лавриненко И.Н., Лавриненко С.В., Мезенцева О.С. Методы и алгоритмы округления, масштабирования и деления чисел в модулярной арифметике // 50 лет модулярной арифметике. Юбилейная междунар. науч.-техн. конф. Сб. трудов. М., 2000. С. 291–310.
5. Chren W.A. A new residue number system division algorithm // Computers Math. Applic. 1990. Vol. 19. No. 7. Pp. 13–29.
6. Banerji D.K., Cheung T.Y., Ganesan V. A high-speed division method in residue arithmetic // IEEE Symp. Comput. Arithmetic. 1981. No. 5. Pp. 158–164.
7. Hits M.A., Kaltofen E. Integer division in residue number systems // IEEE Trans. Comput. 1995. Vol. 44. No. 8. Pp. 983–989.
8. Kinoshita E., Kosako H., Kojima Y. General division in the symmetric residue number system // IEEE Trans. Comput. 1973. Vol. 22. Pp. 134–142.
9. Lin M.L., Leiss E., McInnis B. Division and sign detection algorithm for residue number systems // Computers Math. Applic. 1984. Vol. 10. No. 4/5. Pp. 331–342.
10. Radhakrishnan D., Yuan Y. Novel approaches to the design of VLSI RNS multipliers // IEEE Trans. Circuits and System. 1992. Vol. 39. No. 1. Pp. 52–57.
11. Hung C., Parhami B. Fast RNS division algorithms for fixed divisors with application to RSA encryption // Information Processing Letters. 1994. No. 51. Pp. 163–169.
12. Hung C., Parhami B. An approximate sign detection method for residue numbers and its application to RNS division // Computers Math. Applic. 1994. Vol. 27. No. 4. Pp. 23–35.
13. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М.: Сов. радио, 1968. 440 с.
14. Содерстренд М.А., Верниа К. Недорогой быстродействующий умножитель по модулю и его применение в арифметических устройствах на основе системы счисления в остаточных

классах. ТИИЭР, 1980. № 4. Т. 68.

15. **Julien G.A.** Residue number scaling and other operations using ROM arrays // *IEEE Trans. on Comput.* 1978. Vol. 27. No. 4. Pp. 325–336.

16. **Beyer W.** *CRC-Standard Mathematical Tables and Formulae*. Edition 91-101. CRC Press, 1991.

17. **Talameh S., Siy P.** Arithmetic Division in RNS Using Galois Field $GF(p)$ // *Computers and Mathematics with Applications*. 2000. Vol. 39. Pp. 227–238.

18. **Червяков Н.И.** Методы, алгоритмы и

техническая реализация основных проблемных операций, выполняемых в системе остаточных классов // *Инфокоммуникационные технологии*. 2011. № 4. С. 4–12.

19. **Червяков Н.И., Ляхов П.А.** Метод определения знака числа в системе остаточных классов на основе приближенных вычислений // *Нейрокомпьютеры: разработка, применение*. 2012. № 12. С. 56–64.

20. **Alia G., Martinelli E.** Neuron: a Rom based RNS oligital neuron // *Neural Networks*. 2005. Pp. 174–189.

REFERENCES

1. **Claudio E.D., Piazza F., Orlandi G.** Fast combinatorial RNS processors for DSP applications. *IEEE Trans. Comput.* 1995, Vol. 44, No. 5, Pp. 624–633.

2. **Waser S., Flynn M.J.** *Introduction to Arithmetic for Digital System Designers*. Holt, Rinehart Winston. New York, 1982.

3. **Taylor F.J.** Residue arithmetic: A tutorial with examples, *IEEE Comput.*, 1984, Vol. 17, No. 5, Pp. 50–62.

4. **Chervyakov N.I., Lavrinenko S.V., Mezentseva O.S.** Metody i algoritmy okrugleniya, masshtabirovaniya i deleniya chisel v modulyarnoy arifmetike, *50 let modulyarnoy arifmetike. Yubileynaya mezhdunarodnaya nauchno-tekhnicheskaya konferentsiya*, Moscow, 2000, Pp. 291–310. (rus)

5. **Chren W.A.** A new residue number system division algorithm, *Computers Math. Applic.* 1990, Vol. 19, No. 7, Pp. 13–29.

6. **Banerji D.K., Cheung T.Y., Ganesan V.** A high-speed division method in residue arithmetic, *IEEE Symp. Comput. Arithmetic*, 1981, No. 5, Pp. 158–164.

7. **Hits M.A., Kaltofen E.** Integer division in residue number systems, *IEEE Trans. Comput.*, 1995, Vol. 44, No. 8, Pp. 983–989.

8. **Kinoshita E., Kosako H., Kojima Y.** General division in the symmetric residue number system, *IEEE Trans. Comput.*, 1973, Vol. 22, Pp. 134–142.

9. **Lin M.L., Leiss E., McInnis B.** Division and sign detection algorithm for residue number systems, *Computers Math. Applic.* 1984, Vol. 10, No. 4/5, Pp. 331–342.

10. **Radhakrishnan D., Yuan Y.** Novel approaches to the design of VLSI RNS multipliers, *IEEE Trans. Circuits and System*, 1992, Vol. 39, No. 1, Pp. 52–57.

11. **Hung C., Parhami B.** Fast RNS division

algorithms for fixed divisors with application to RSA encryption, *Information Processing Letters*, 1994, No. 51, Pp. 163–169.

12. **Hung C., Parhami B.** An approximate sign detection method for residue numbers and its application to RNS division, *Computers Math. Applic.*, 1994, Vol. 27, No. 4, Pp. 23–35.

13. **Akushskiy I. Ya., Yuditskiy D.I.** *Mashinnaya arifmetika v ostatochnykh klassakh*. Moscow: Sovetskoye radio Publ., 1968. 440 p. (rus)

14. **Soderstrend M.A., Vernia K.** Nedorogoy bystrodeystvuyushchiy umnozhitel po modulyu i yego primeneniye v arifmeticheskikh ustroystvakh na osnove sistemy schisleniya v ostatochnykhklassakh, *TIIEP*, 1980, No. 4, Vol. 68. (rus)

15. **Julien G.A.** Residue number scaling and other operations using ROM arrays, *IEEE Trans. on Comput.*, 1978, Vol. 27, No. 4, Pp. 325–336. (rus)

16. **Beyer W.** *CRC-Standard Mathematical Tables and Formulae*. Edition 91-101, CRC Press, 1991.

17. **Talameh S., Siy P.** Arithmetic Division in RNS Using Galois Field $GF(p)$, *Computers and Mathematics with Applications*, 2000, Vol. 39, Pp. 227–238.

18. **Chervyakov N.I.** Metody, algoritmy i tekhnicheskaya realizatsiya osnovnykh problemnykh operatsiy, vpolnyayemykh v sisteme ostatochnykh klassov, *Infokommunikatsionnyye tekhnologii*, 2011, No. 4, Pp. 4–12. (rus)

19. **Chervyakov N.I., Lyakhov P.A.** Metod opredeleniya znaka chisla v sisteme ostatochnykh klassov na osnove priblizhennykh vychisleniy, *Neyrokompyutery: razrabotka, primeneniye*, 2012, No. 12, Pp. 56–64. (rus)

20. **Alia G., Martinelli E.** Neuron: a Rom Based RNS Oligital Neuron, *Neural Networks*, 2005, Pp. 174–189.

ЧЕРВЯКОВ Николай Иванович — заведующий кафедрой прикладной математики и математического моделирования Института математики и естественных наук Северо-Кавказского федерального университета, профессор, доктор технических наук.

355029, Россия, г. Ставрополь, пр. Кулакова, д. 2.

E-mail: k-fmf-primath@stavsru

SHERVYAKOV, Nikolay I. *North-Caucasus Federal University.*

355029, Kulakov Ave. 2, Stavropol, Russia.

E-mail: k-fmf-primath@stavsru

БАБЕНКО Михаил Григорьевич — доцент кафедры высшей алгебры и геометрии Северо-Кавказского федерального университета, кандидат физико-математических наук.

355029, Россия, г. Ставрополь, пр. Кулакова, д. 2.

E-mail: whbear@yandex.ru

BABENKO, Mikhail Gr. *North-Caucasus Federal University.*

355029, Kulakov Ave. 2, Stavropol, Russia.

E-mail: whbear@yandex.ru

ЛЯХОВ Павел Алексеевич — доцент кафедры прикладной математики и математического моделирования Института математики и естественных наук Северо-Кавказского федерального университета, кандидат физико-математических наук.

355029, Россия, г. Ставрополь, пр. Кулакова, д. 2.

E-mail: ljahov@mail.ru

LYAKHOV, Pavel A. *North-Caucasus Federal University.*

355029, Kulakov Ave. 2, Stavropol, Russia.

E-mail: ljahov@mail.ru

ЛАВРИНЕНКО Ирина Николаевна — доцент кафедры высшей алгебры и геометрии Института математики и естественных наук Северо-Кавказского федерального университета, кандидат физико-математических наук.

355029, Россия, г. Ставрополь, пр. Кулакова, д. 2.

E-mail: Algebra223@yandex.ru

LAVRINENKO, Irina N. *North-Caucasus Federal University.*

355029, Kulakov Ave. 2, Stavropol, Russia.

E-mail: Algebra223@yandex.ru

ЛЯГИН Алексей Михайлович — доцент кафедры прикладной математики и математического моделирования Института математики и естественных наук Северо-Кавказского федерального университета, кандидат технических наук.

355029, Россия, г. Ставрополь, пр. Кулакова, д. 2.

E-mail: kfmf-primath@stavsru

LYAGIN, Aleksey M. *North-Caucasus Federal University.*

355029, Kulakov Ave. 2, Stavropol, Russia.

E-mail: kfmf-primath@stavsru



Конференция «Технологии Microsoft в теории и практике программирования»

Начиная с 2004 года Институт информационных технологий и управления (ранее — факультет технической кибернетики) Санкт-Петербургского государственного политехнического университета проводит конференцию студентов, аспирантов и молодых ученых Северо-Запада «Технологии Microsoft в теории и практике программирования». Цели конференции — подготовка участников к будущей работе в профессиональных программистских коллективах, создающих программный продукт высокого качества, поддержка изучения современных информационных технологий и инструментальных средств в соответствии с мировыми стандартами и действующими международными сертификационными требованиями, а также выявление талантливых молодых специалистов в области разработки и использования программных систем.

Особенность конференции в том, что участник должен проявить свои знания и умения не столько в области программирования различных математических головоломок, сколько в области разработки и использования программных продуктов и систем в условиях, максимально приближенных к реальным процессам проектирования и разработки современных систем различной степени сложности.

Конференция проводится в два этапа: на первом отбираются и ревьюируются работы участников, присылающих короткие статьи — заявки на участие. На их основе отбираются приглашенные докладчики на второй этап. Присланные статьи публикуются в сборнике материалов конференции. Второй этап проходит в виде докладов на секциях. Кроме того, предусмотрены пленарные доклады и мастер-классы по новациям в мире инструментов и технологий программирования.

Все участники секционных презентаций награждаются дипломами первой, второй и третьей степени. Решение о премировании участников конкурса принимает конкурсная комиссия, в которую входят руководители секций и представители организаций спонсоров.

В 2014 году спонсорами конференции являются всемирно известные компании Microsoft и EMC.

В данном журнале публикуется три статьи победителей — участников конференции 2014 года. Публикуемые работы отличает инновационный характер и ориентация на реализацию полученных результатов, широкое применение технологий MS при создании программного обеспечения и документации.

Директор ИИТУ, профессор,
доктор технических наук
М.В. Окрепилов

УДК 004.054

Б.В. Тютин, А.О. Веселов, В.П. Котляров

ТЕСТИРОВАНИЕ НА ОСНОВЕ КЛЮЧЕВЫХ СЛОВ С ИСПОЛЬЗОВАНИЕМ ДИАГРАММ ПОСЛЕДОВАТЕЛЬНОСТИ СОБЫТИЙ

B.V. Tiutin, A.O. Veselov, V.P. Kotlyarov

SEQUENCE DIAGRAMS APPLICATION FOR KEYWORD-DRIVEN TESTING

Рассмотрен подход к тестированию на основе ключевых слов, базирующийся на тестовых сценариях в формате диаграмм Message Sequence Charts. Изучены основные особенности и преимущества данного подхода. Приведено краткое описание существующего программного решения по автоматизации тестирования, в основе которого заложены представленные идеи.

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ; ТЕСТИРОВАНИЕ НА ОСНОВЕ КЛЮЧЕВЫХ СЛОВ; ФОРМАЛЬНЫЕ СПЕЦИФИКАЦИИ; MESSAGE SEQUENCE CHARTS.

The given paper focuses on the application of sequence diagrams in Message Sequence Chart format for keyword-driven testing. We have studied a keyword-driven approach and presented main advantages of sequence diagrams. The paper comes up with a brief description how to implement testing automation framework based on presented techniques.

TESTING AUTOMATION; KEYWORD-DRIVEN TESTING; FORMAL SPECIFICATION; MESSAGE SEQUENCE CHARTS.

Основным требованием к промышленным технологиям производства программного продукта (ПП) является возможность построения единой автоматизированной технологической цепочки, обеспечивающей в том числе контроль качества разрабатываемых продуктов. Поэтому в текущих условиях наибольшую эффективность обеспечивают процессы, базирующиеся на инструментальных средствах, функциональность которых может быть легко адаптирована или расширена к условиям бесшовной интеграции в рамках технологической цепочки.

Настоящая статья сфокусирована на особенности реализации эффективной технологической цепочки тестирования.

Реализация инструментов тестирования на основе известного подхода, управляемого данными (Data-Driven Testing – DDT) [1], позволяет решить проблемы автоматизации, снижения сложности и обеспечения понимания широким кругом специалистов. Среди разнообразия существующих подходов к управлению тестированием на основе

данных, наиболее перспективно тестирование на основе ключевых слов (Keyword-Driven Testing – KDT) [2]. Этот подход упрощает разработку и поддержку тестового набора путем отделения логики тестовых процедур от их реализации.

Разработка гибких и в то же время интегрированных в единую технологическую цепочку инструментов тестирования является актуальной задачей, решение которой позволяет снизить затраты на поиск дефектов в программном обеспечении. В данной статье рассматривается подход к автоматизации тестирования, объединяющий дружественную для пользователя нотацию MSC (Message Sequence Charts) [3] и метод тестирования на основе ключевых слов.

Подход, основанный на ключевых словах

Тестирование, основанное на ключевых словах, позволяет создавать и выполнять структурированные сценарии, в которых данные отделены от потока управления.

Сценарий теста состоит из списка ключевых слов и их параметров. Каждое ключевое

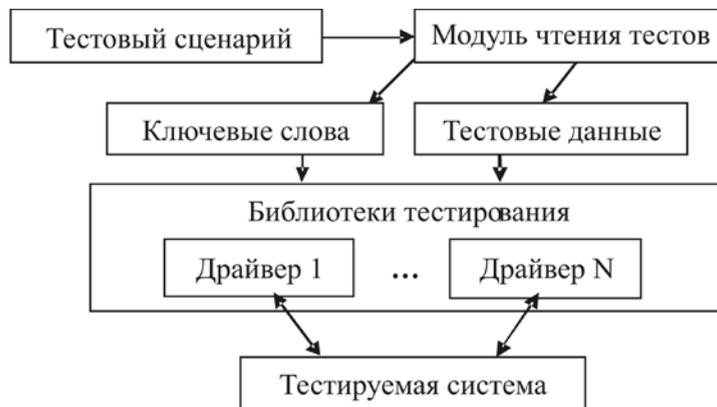


Рис. 1. Схема тестирования на основе ключевых слов

ческое слово представляет собой заданный перечень воздействий, выполняемых над тестируемой системой. Сам сценарий имеет табличный формат и может быть отредактирован как в обычном текстовом редакторе, так и при помощи специальных программных инструментов. Рис. 1 иллюстрирует концепцию подхода к тестированию на основе ключевых слов в случае автоматизированного тестирования.

Каждое ключевое слово имеет однозначное определение, которое может быть задано в виде кода для автоматизированного тестирования или в виде текстовой процедуры для ручного тестирования. В первом случае при помощи специального модуля осуществляется чтение и разбор тестового сценария. Найденные ключевые слова и их аргументы интерпретируются, после чего происходит выполнение соответствующих им действий. Программный код, реализующий данные действия, компонуется в библиотеки тестирования, также называемые драйверами. Именно они осуществляют взаимодействие с тестируемой системой и проводят регистрацию событий.

При таком подходе возможно создание определений ключевых слов отдельно и независимо от создания тестовых сценариев. Таким образом, создание тестов разделено на две независимых активности: разработка тестовых сценариев и создание тестовых действий – элементов, из которых строятся сценарии (ключевых слов).

При использовании подхода, основан-

ного на ключевых словах, затраты на создание и сопровождение тестового набора в среднем снижаются на 25 %. Это обусловлено тем, что изменения требований, в основном, требуют не изменения сценариев, а только уточнения определений ключевых слов. Тестирование становится масштабируемым, что означает возможность распределенного выполнения набора тестов среди имеющихся вычислительных ресурсов [4]. Модификация тестовых сценариев упрощается тем, что тесты оперируют высокоуровневыми абстракциями и переиспользуют существующие ключевые слова, что в свою очередь снижает требования к компетенции тестировщика в детальном знании объекта тестирования.

Преимущества MSC для тестирования

MSC описывают асинхронное взаимодействие нескольких процессов. Проще говоря, они представляют собой описание набора инстанций, которые взаимодействуют друг с другом при помощи обмена сигналами [3].

На практике оказывается, что использование диаграмм последовательности событий (MSC) является очень удобным для представления сценариев, для описания конкретных спецификаций телекоммуникационных протоколов или для протоколов обмена данными между различными компонентами разрабатываемой системы. Текстовое представление MSC легко может быть разобрано и проанализировано. Все это по-

зволяет использовать MSC для описания тестовых сценариев. В настоящей работе замечено, что концепция сигнала MSC очень близка к концепции ключевого слова. Оба понятия содержат информацию о тестовом действии и наборе данных для него, имеют контекстно-независимую семантику и могут быть однозначно интерпретированы согласно установленной процедуре тестирования.

MSC как формат теста, основанного на ключевых словах, имеет следующие преимущества. Он может быть представлен в графическом виде [5] и легко воспринимается человеком. Так как формат стандартизирован, возможно переиспользование сторонних инструментов и библиотек для разбора MSC. Некоторые существующие инструменты моделирования и разработки позволяют выдавать данные в формате MSC как для документирования поведения системы, так и для генерации тестов.

Сравнение возможностей существующих технологий позволило выявить ключевые аспекты функциональности, требуемые для автоматизации тестирования, а также существующие пробелы (см. табл.). Были рассмотрены системы и фреймворки тестирования Fit [6], Robot [7], JUnit [8], UniTesK [9], TAT [10], HP Quick Test Professional (QTP) [11], IBM Rational Functional Tester (RFT) [12].

Базовым элементом рассмотренных систем тестирования является хотя бы частичная реализация подходов DDT и KDT. Это позволяет создавать параметризованные тесты, в которых данные отделены от логики тестирования. Вторым по важности аспектом является возможность создания в рамках применяемого формата тестов

структурированных сценариев. Это позволяет разделить процессы по созданию и расширению тестового набора, задействовав в них различных специалистов.

Редактирование тестов в графическом представлении является менее трудоёмким процессом, в котором может быть задействован менее квалифицированный пользователь. Возможность графического представления тестовых сценариев имеется у всех технологий, за исключением JUnit и QTP, однако возможность их редактирования доступна только у TAT и RFT.

Важным аспектом языка тестовых сценариев является возможность реализовывать сложную, нелинейную логику, описывающую альтернативные ветки поведения, циклы, параллельные взаимодействия.

С точки зрения автоматизации тестирования рассмотренные инструменты предоставляют примерно равные возможности. Однако стоит отдельно отметить интеграцию JUnit в процесс разработки кода программного продукта, нашедшую широкое применение в Test Driven Development.

Использование web-технологий делает процесс формирования и визуализации результатов тестирования независимым от среды исполнения, что может использоваться для реализации облачного тестирования.

Концепция KDT на базе MSC

Основными элементами MSC диаграмм являются сигналы, действия (MSC actions) и нелинейные конструкции. Все они могут использоваться для реализации концепции тестирования, основанной на ключевых словах.

Сравнение каркасов построения систем тестирования

	Fit	Robot	JUnit	TAT	UniTesK	QTP	RFT
DDT	+	+/-	+/-	+/-	+/-	+/-	+
KDT	-	+	-	+/-	+/-	+/-	-
Параметризация тестов	-	+	-	+	+	+/-	+
Нелинейные сценарии	-	+	+	+	+	+	+
Параллельные сценарии	-	-	+/-	+	+/-	-	+/-
Отображение результатов тестирования в интерактивной форме	-	+/-	+	+	+/-	+	+

Обмен сигналами может интерпретироваться как последовательность ключевых слов, следующих одно за другим. Данные передаются как параметры сигналов. В контексте тестирования все инстанции MSC диаграммы разделяются на две группы: принадлежащие к тестовому окружению и принадлежащие к тестируемой системе. Таким образом, все сигналы от окружения к системе интерпретируются как воздействия на систему, в то время как сигналы в обратном направлении интерпретируются как ответы системы на воздействия от окружения. В тестировании, основанном на ключевых словах, различия в направлении сигналов стираются, т. к. каждый сигнал превращается в команду — ключевое слово. Таким образом, концепцию посылки и приема сигналов MSC становится необходимо адаптировать для ключевых слов таким же образом, чтобы сделать тесты читаемыми.

Для этого предложено рассматривать сигналы от окружения к системе как ключевые слова, непосредственно отвечающие за действия над системой, а сигналы от системы к окружению — как ключевые слова, отвечающие за проверку ответов системы, либо ее состояний.

Действия (actions) в MSC используются для описания внутренних событий инстанций, таких как вычисление некоторых значений, вызов функций, добавление комментария. Как элемент тестового сценария, действия нередко используются для повышения читаемости теста за счет указания в них комментариев, кодирующих соответствующую активность. При ручном тестировании в действиях могут содержаться дополнительные инструкции или находиться ссылки на необходимую документацию или раздел требований. В автоматическом тестировании действия могут использоваться для вставки целевого кода в тест, в т. ч. и для передачи в тест ключевых слов.

Таким образом, оказывается возможным использование MSC для тестирования, основанного на ключевых словах, и использование компонент диаграмм для того, чтобы сделать тесты легко воспринимаемыми. Роль каждого элемента должна быть ясно определена в том случае, когда

от тестовых сценариев требуется восприятие человеком.

Нелинейные выражения MSC позволяют задавать различные нелинейные сценарии тестов, в частности возможно использование циклов (loop expression), опциональных (opt expression) или альтернативных (alt expression) поведений. В сочетании с условными конструкциями (MSC condition), нелинейные выражения позволяют использовать if-else конструкции и циклы как в обычных языках программирования. Переменные, используемые в действиях, параметрах сигналов и в выражениях условных конструкций, могут быть определены при помощи специальной конструкции text.

Создаваемые тестовые сценарии за счет предложенных в работе идей по интерпретации элементов MSC позволяют описывать взаимодействие распределенных компонент системы. Помимо этого, от существующих решений, представленных в работах по использованию MSC для тестирования, разработанный подход отличается тем, что представленные на диаграмме недетерминированные варианты поведения могут определять не только взаимоисключающие, но и равновероятные варианты.

Следует отметить, что MSC как стандартизованная нотация позволяет с легкостью создавать на ее основе тесты, основанные на ключевых словах. Система тестирования по ключевым словам, в свою очередь, может автоматически запускать тесты, используя различные нижележащие драйверы и библиотеки.

В рамках данной работы была создана автоматизация системы тестирования, включающая систему трансляции тестовых сценариев с MSC представления в исполняемое представление, основанное на ключевых словах.

Реализация системы тестирования

Основная идея, реализованная в системе тестирования, заключается в вычлениении ключевых слов из MSC и их дальнейшем запуске при помощи драйверов и библиотек. Но вместо прямой интерпретации в команды в системе реализована



Рис. 2. Схема автоматизации тестирования на основе MSC

трансляция в целевой код. Такой подход нашел своё отражение в ряде работе по использованию MSC [13]. Полученный код теста на целевом языке представляет собой созданный по определенным правилам автомат, реализующий набор состояний и систему переходов, отражающих сценарий теста. Такая архитектура позволяет реализовать нелинейные конструкции MSC. Основные компоненты созданного программного комплекса проиллюстрированы на рис. 2.

В тестовых сценариях реализована возможность использования переменных,

которые могут применяться для задания тестовых данных и управления тестированием. Каждая инстанция MSC, относящаяся к тестовому окружению, выполняется в отдельном потоке и независима от остальных. Это требует наличия двух типов переменных: локальных (доступность которых ограничена рамками данной инстанции) и глобальных (которые доступны из любой инстанции). Для реализации безопасного доступа к глобальным переменным вводится дополнительный поток, отвечающий за хранение и доступ к глобальным переменным [14]. Данный вспомогательный

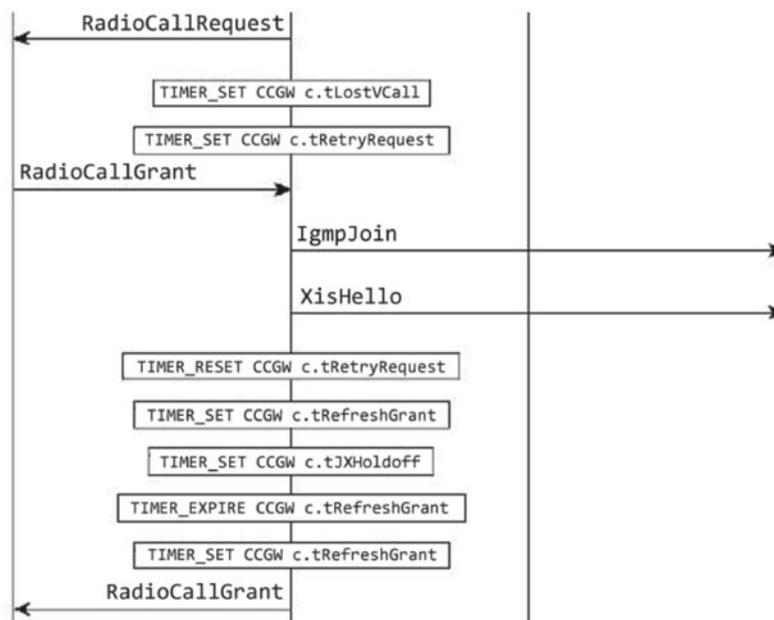


Рис. 3. Пример тестового сценария

поток также отвечает за принятие решений при выборе ветки альтернативы, когда решение зависит от нескольких инстанций сразу.

На рис. 3 приведен пример фрагмента тестового сценария в формате MSC для организации распределённого тестирования. Данный тест применялся в проекте по тестированию компонента встроенной телекоммуникационной системы, реализующей канал агрегации и передачи сигнальной информации в сети. Сценарий проверял взаимодействие модуля-медиатора с поставщиками информации.

На диаграмме представлены четыре взаимодействующих инстанции, одна из которых является тестируемой системой. Три оставшиеся эмулируют рабочее окружение, с которым тестируемая система обменивается сигналами (например, RadioCallGrant). Для иллюстрации ожидаемого изменения внутреннего состояния тестируемой системы использованы actions, содержащие псевдокод внутренних вызовов (например, TIMER_SET).

Представленное исследование ещё не завершено. В его рамках предложен подход реализации тестирования на основе ключевых слов с представлением тестовых сценариев в виде диаграмм последовательности событий и сформулированы правила интерпретации элементов MSC, обеспечивающие реализацию KDT.

На базе предложенных идей разработана архитектура системы распределённого тестирования. Текущая реализация фреймворка включает в себя транслятор из MSC в C, библиотеки ядра и драйверов и генератор отчетов.

Проведена апробация разработанных методов и программного комплекса. С применением результатов исследования было организовано тестирование в четырех проектах по созданию программного обеспечения в телекоммуникационной области. В итоге было зафиксировано ускорение этапов создания тестового набора, а также сокращение количества тестов, покрывающих требования без снижения количества найденных дефектов.

СПИСОК ЛИТЕРАТУРЫ

1. Beck K. Test-Driven Development by Example. SPb.: Piter, 2003.
2. Fought D.R. Keyword-Driven Testing. Sticky Minds. Software Quality Engineering [электронный ресурс] / URL: <http://www.stickyminds.com/article/keyword-driven-testing> (дата обращения 05.11.2004).
3. ITU-T Recommendation Z.120: Message sequence chart (MSC). Geneva, Switzerland, Oct. 1996.
4. Тютин Б.В., Веселов А.О., Котляров В.П. Масштабирование выполнения тестового набора при автоматизированном тестировании // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2013. № 3(174). С. 118–122.
5. Damm W., Harel D. LSCs: Breathing life into message sequence charts // Formal Methods in System Design. 2001. No. 19(1).
6. Mugridge R., Cunningham W. Fit for Developing Software. London: Prentice Hall, 2005. 384 p.
7. Robot framework [электронный ресурс] / URL: <http://robotframework.org> (дата обращения 25.05.2014).
8. JUnit framework [электронный ресурс] / URL: <http://junit.org> (дата обращения 24.05.2014).
9. UniTesK technology [электронный ресурс] / URL: <http://www.unitesk.ru> (дата обращения 24.05.2014).
10. Тютин Б.В., Никифоров И.В., Котляров В.П. Построение системы автоматизации статической и динамической проверки требований к программному продукту // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2012. № 4(152). С. 119–123.
11. HP Quicktest Professional [электронный ресурс] / URL: <http://www8.hp.com/ru/ru/software-solutions/software.html?compURI=1172957> (дата обращения 18.10.2012).
12. IBM Rational Functional Tester [электронный ресурс] / URL: <http://www-01.ibm.com/software/awdtools/tester/functional> (дата обращения 25.05.2014).
13. Hu W., Sun X. Test Case Generation Based on MSC TTCN-3 // Proc. of the Internat. Conf. on Information Engineering and Applications. London: Springer-Verlag, 2013. 888 p.
14. Kaner C., Bach J., Pettichord B. Lessons Learned in Software Testing: A Context-Driven Approach. NY: John Wiley & Sons, Inc., 2001. 320 p.

REFERENCES

1. **Beck K.** *Test-Driven Development by Example*. St. Petersburg: Piter Publ., 2003.
2. **Faught D.R.** *Keyword-Driven Testing. Sticky Minds. Software Quality Engineering*. Available: <http://www.stickyminds.com/article/keyword-driven-testing> (Accessed 05.11.2004).
3. *ITU-T Recommendation Z.120: Message sequence chart (MSC)*. Geneva, Switzerland, October 1996.
4. **Tyutin B.V., Veselov A.O., Kotlyarov V.P.** *Masshtabirovaniye vypolneniya testovogo nabora pri avtomatizirovannom testirovanii, Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye*. St. Petersburg: SPbGPU Publ., 2013, No. 3(174), Pp. 118–122. (rus)
5. **Damm W., Harel D.** LSCs: Breathing life into message sequence charts. *Formal Methods in System Design*, 2001, No. 19(1).
6. **Mugridge R., Cunningham W.** *Fit for Developing Software*. London: Prentice Hall, 2005, 384 p.
7. *Robot framework*. Available: <http://robotframework.org> (Accessed 25.05.2014).
8. *JUnit framework*. Available: <http://junit.org> (Accessed 24.05.2014).
9. *UniTesK technology*. Available: <http://www.unitesk.ru> (Accessed 24.05.2014).
10. **Tyutin B.V., Nikiforov I.V., Kotlyarov V.P.** *Postroyeniye sistemy avtomatizatsii staticheskoy i dinamicheskoy proverki trebovaniy k programmnomu produktu, Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye*. St. Petersburg: SPbGPU Publ., 2012, No. 4(152), Pp. 119–123. (rus)
11. *HP Quicktest Professional*. Available: <http://www8.hp.com/ru/ru/software-solutions/software.html?compURI=1172957> (Accessed 18.10.2012).
12. *IBM Rational Functional Tester*. Available: <http://www-01.ibm.com/software/awdtools/tester/functional> (Accessed 25.05.2014).
13. **Hu W., Sun X.** *Test Case Generation Based on MSC TTCN-3, Proceedings of the International Conference on Information Engineering and Applications*. London: Springer-Verlag, 2013, 888 p.
14. **Kaner C., Bach J., Pettichord B.** *Lessons Learned in Software Testing: A Context-Driven Approach*. NY: John Wiley & Sons, Inc., 2001, 320 p.

ТЮТИН Борис Викторович — аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: b.tyutin@gmail.com

TYUTIN, Boris V. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: b.tyutin@gmail.com

ВЕСЕЛОВ Алексей Олегович — научный сотрудник кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: veselov.alexey@gmail.com

VESELOV, Aleksey O. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: veselov.alexey@gmail.com

КОТЛЯРОВ Всеволод Павлович — профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: vpk@ics2.ecd.spbstu.ru

KOTLYAROV, Vsevolod P. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: vpk@ics2.ecd.spbstu.ru

УДК 004.415

П.Д. Дробинцев, И.В. Никифоров, Н.В. Воинов, В.П. Котляров

ПОДХОД К ТЕСТИРОВАНИЮ ПАРАЛЛЕЛЬНЫХ СИСТЕМ НА ОСНОВЕ UCM-СПЕЦИФИКАЦИЙ

P.D. Drobintsev, I.V. Nikiforov, N.V. Voinov, V.P. Kotlyarov

APPROACH TO CONCURRENT SYSTEMS TESTING BASED ON UCM SPECIFICATION

Изучен подход к тестированию параллельных программных систем, основанный на использовании спецификаций на языке UCM. Рассмотрены UCM-конструкции, позволяющие задавать многопоточное поведение в системе. Описаны эквивалентные им конструкции на языке базовых протоколов, с которым работает средство генерации тестовых сценариев. Представлен подход к автоматическому созданию тестов для многопоточных систем в виде MSC-диаграмм. Представленный подход позволяет на основе анализа независимости параллельных потоков автоматически получать корректные наборы тестов для тестирования параллельных распределенных систем.

ФОРМАЛЬНАЯ МОДЕЛЬ; СИНХРОНИЗАЦИЯ ПРОЦЕССОВ; ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ.

The current paper presents an approach to testing of concurrent program systems based on UCM specifications usage. We have studied UCM constructions specifying multithreading system behavior. We have also described equivalent constructions using the language of basic protocols that is operated by test scenario generator. The approach to automated creation of test scenarios in MSC notation for multithreading systems has been shown. The described approach based on the analysis of concurrent threads allows us to obtain automatically verified test suites for testing concurrent distributed systems.

FORMAL MODEL; THREAD SYNCHRONIZATION; PARALLEL COMPUTING.

В настоящее время в силу активного развития компьютерных технологий и повсеместной информатизации общества огромное количество задач решается с помощью программно-аппаратных комплексов, производящих распределенные вычисления. Распределенные вычисления — это способ решения трудоемких вычислительных задач с использованием нескольких компьютеров, чаще всего объединенных в параллельную вычислительную систему [1]. При этом параллельная вычислительная система представляет собой физическую компьютерную или программную систему, реализующую тем или иным способом параллельную обработку данных на многих вычислительных узлах [2]. Сложность подобных программных систем и высокие требования к их качеству делают все более актуальной задачу их тестирования.

Для многих приложений использование параллельных процессов решает задачу эффективности, т. е. получения результатов вычислений в рамках заданных временных ограничений. Однако тестирование программной системы на основе использования параллельных процессов сопряжено с проблемой взрыва состояний, порождаемого необходимостью проверки состояний всех процессов системы во время прогона теста [8].

В настоящей статье описывается подход, позволяющий решить данную проблему для определенного класса систем, требования на которые представлены в виде формальной модели на языке спецификаций высокого уровня UCM (Use Case Maps) [3].

Моделирование параллельного исполнения в UCM

Язык UCM позволяет создавать модели

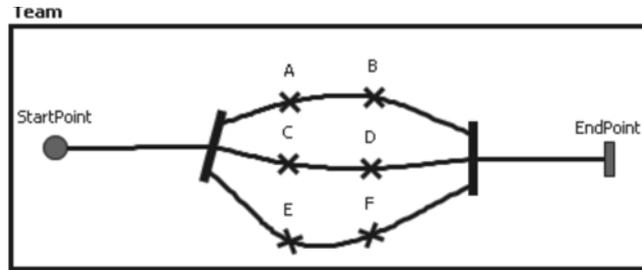


Рис. 1. Порождение и синхронизация процессов

систем, использующих параллельные потоки. Для порождения и последующей синхронизации потоков используются специальные элементы: AndFork – для порождения потоков и AndJoin – для их синхронизации. Следует отметить, что при создании модели с параллельным поведением необходимо соблюдать ограничения на создание подобных конструкций, которые позволят получить синтаксически корректную модель [4]. На рис. 1 приведена диаграмма, демонстрирующая применение элементов AndFork и AndJoin для создания и синхронизации трех параллельных потоков.

Особенностью разработки многопоточных UCM-моделей является интерпретация компонентов, являющихся элементами языка UCM, процессами, а параллельных ветвей, заданных элементами AndFork и AndJoin, – потоками внутри процесса. Все процессы в системе параллельны друг другу и существуют с начала исполнения сценариев, в то время как потоки внутри компонентов ограничены по времени жизни и не могут превышать время жизни процессородителя.

Создавать параллельные потоки на пути позволяет элемент AndFork. Каждая исходящая ветка соответствует параллельному

потоку. В соответствии с семантикой UCM в отличие от элемента OrFork, позволяющего определять альтернативное поведение системы, на исходящих ветвях AndFork нельзя задавать условия ветвления, т. е. при достижении элемента AndFork все исходящие ветки могут начать свое параллельное исполнение.

На основе формального представления UCM-модели может быть проведена автоматическая генерация модели в нотации базовых протоколов [5]. Подобный транслятор реализован и описан в рамках работ над созданием интегрированной технологии для анализа и верификации спецификаций [6].

Рассмотрим некоторые детали работы транслятора. При работе с параллельным поведением каждая исходящая ветка формализуется базовым протоколом, представляющим собой тройку Хоара, вида $\forall x(\alpha \rightarrow \langle u \rangle \beta)$, где x – список (типизированных) параметров, α и β – формулы базового языка, u – процесс протокола (вполне определенное поведение) [5]. При этом в каждом из базовых протоколов генерируется уникальный для ветки поток управления (рис. 2).

На диаграмме рис. 1 происходит гене-

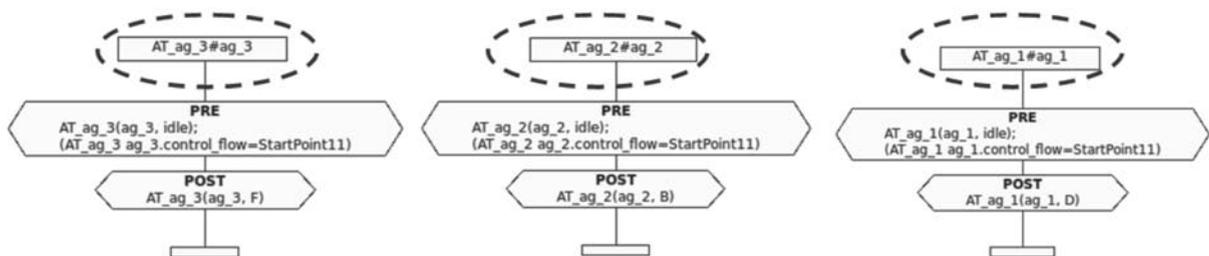


Рис. 2. Формализация параллельного поведения в виде базовых протоколов

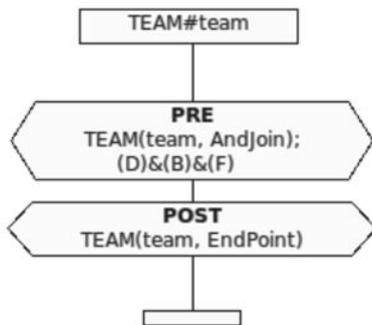


Рис. 3. Протокол синхронизации

рация трех потоков с тремя уникальными агентами. На рис. 2 уникальные агенты для потоков отмечены пунктирными овалами.

Конкретный поток управления используется во всех протоколах соответствующей ветви параллельного исполнения, что позволяет разделить множество протоколов по различным ветвям описания параллельного поведения.

Стоит отметить, что агенты, отвечающие за исполнение потока управления, содержатся в родительском потоке. Эта информация отображается в файле описания окружения формальной модели.

Элемент AndJoin в свою очередь позволяет синхронизовать входящие в него параллельные ветви. Для синхронизации необходимо, чтобы все входящие потоки завершили свое исполнение, после чего общий поток, определяемый как поток родитель для синхронизируемых потоков, продолжает свое исполнение. Элемент формализуется одним базовым протоколом (рис. 3).

На рис. 3 видно, что в базовом протоколе присутствует условие синхронизации — $V \& D \& E$, которое описывает необходимость завершения всех потоков до точки синхронизации и одновременно является условием применимости данного протокола.

Использование описанных правил преобразования элементов, задающих параллельное поведение на UCM-диаграмме, позволяет реализовать переход от UCM-диаграмм, описывающих многопоточное поведение, к формальной модели на языке базовых протоколов, сохраняя при этом семантику исходной диаграммы, что в свою

очередь предоставляет возможность проведения ее формальной верификации. Поведенческие сценарии или трассы, полученные на инструментальной системе VRS [5] в процессе символьной верификации поведенческой модели приложения, могут использоваться для его тестирования.

Подход к генерации тестовых сценариев для параллельных процессов

Одной из основных проблем при тестировании параллельных систем является проблема взрыва состояний, вызванная интерливингом между взаимодействующими параллельными потоками. Количество возможных состояний параллельных потоков, подлежащих проверке при тестировании, растет экспоненциально в зависимости от количества потоков и состояний каждого потока. Тестирование подобных систем чрезвычайно затруднительно в силу некоторых обстоятельств.

Во-первых, при тестировании возникает необходимость полного перебора всех возможных комбинаций состояний, в которых может оказаться тестируемая система, и соответственно путей, приводящих ее в это состояние.

Во-вторых, сам по себе перевод системы в необходимое состояние при наличии нескольких параллельных потоков является нетривиальной задачей, т. к. исполнение одного потока может оказывать влияние на исполнение другого, что приводит к необходимости управления каждым потоком в отдельности и анализа их взаимного влияния.

В случае если известно, что параллельные потоки независимы, то задача генерации тестовой трассы решается достаточно просто. Необходимо запомнить состояние модели перед элементом AndFork, выделить в тестируемой системе параллельные потоки (от AndFork до AndJoin) и вставить их в тестовый сценарий последовательно: один за другим, восстанавливая запомненное состояние перед началом каждой нити. В итоге после верификации получим корректный тестовый сценарий без применения интерливинга. Если параллельные потоки зависимы, то предварительно в ре-

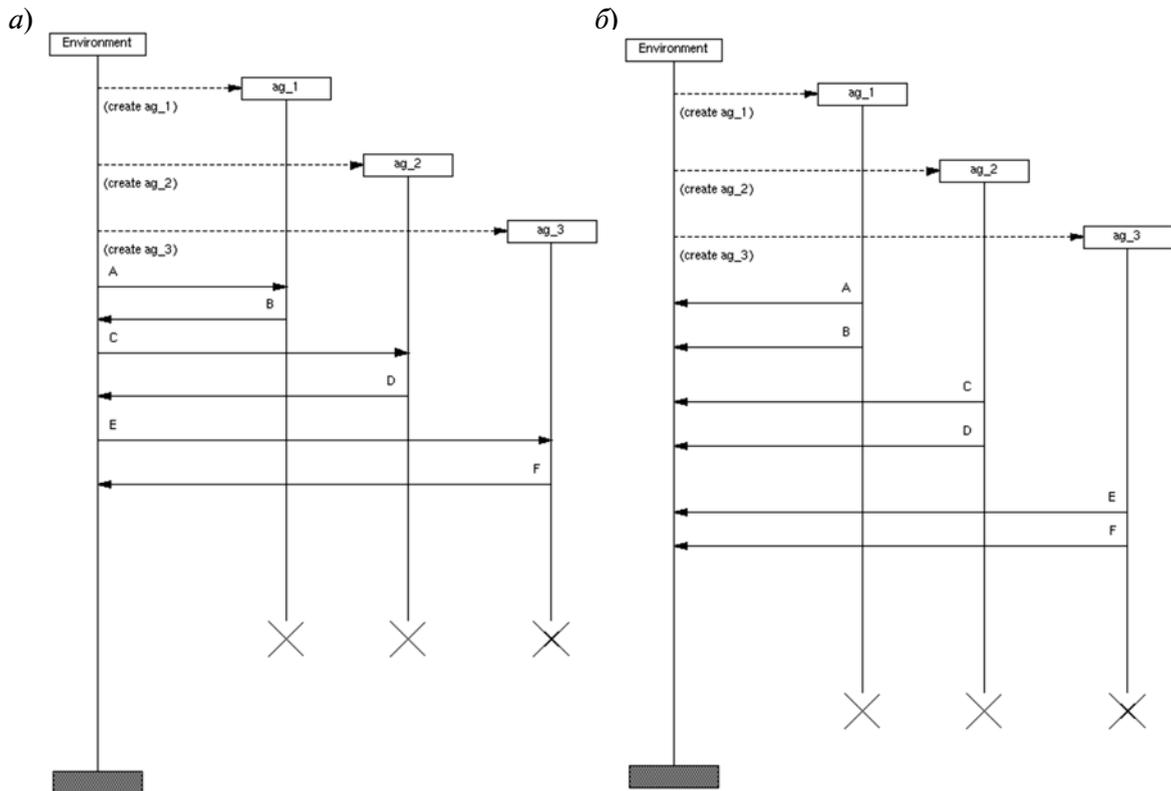


Рис. 4. Тесты для параллельных потоков

в результате статического анализа необходимо найти точки взаимозависимости и обеспечить синхронизацию зависимых взаимодействий, за счет чего скорректированные параллельные потоки станут независимыми от изменения последовательности воздействий на систему, что позволит свести задачу генерации тестовых трасс параллельных потоков к предыдущей.

Следует отметить, что даже при условии независимости параллельных потоков для проведения полноценного тестирования недостаточно использования одного теста. Это связано с тем, что нередко в системах с параллельными потоками управление каждым потоком с точным предсказанием времени его работы невозможно, и таким образом возникает необходимость применения нелинейных тестов. Проиллюстрируем данную проблему на примере.

Предположим, что потоки системы, изображенной на рис. 1, независимы, а элементы responsibility описывают прием и посылку пар сигналов (A; B), (C; D) и (D; E)

соответственно. Тогда тест, основанный на предположении о независимости потоков, может выглядеть следующим образом (рис. 4 а).

Для описания теста использован язык MSC (Message Sequence Charts) [6]. Из рис. 4 а видно, что единственная правильная последовательность поведения системы – это последовательность посылок и приемов сигналов A, B, C, D, E, F, и в данном случае она будет корректна, т. к. тест выступает инициатором обмена сообщениями с системой. Однако если предположить, что responsibility описывают прием пар сигналов (A; B), (C; D) и (E; F) от тестируемой системы, то полученный тест на рис. 4 б будет некорректным, т. е. будет завершаться с ошибкой даже в случае корректного поведения системы. Проблема в данном случае связана с тем, что при тестировании реального программного обеспечения невозможно без специального управления со стороны теста гарантировать, что пара сигналов (A; B) придет первой (то же утверждение

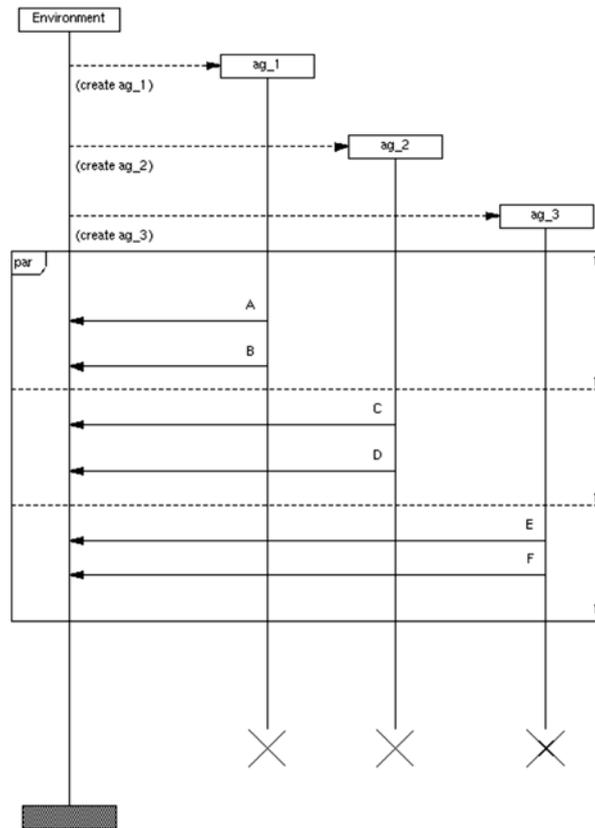


Рис. 5. Использование нелинейных конструкций языка MSC

касается остальных пар сигналов). Данная проблема носит название *гонки сигналов*.

Для решения данной проблемы может использоваться оператор *par*, являющийся синтаксической конструкцией языка MSC и позволяющий описывать взаимодействие параллельных процессов. На рис. 5 приведен корректный тест, полученный добавлением оператора *par*.

Из рисунка видно, что оператор состоит из трех блоков, каждый из которых описывает взаимодействие окружения с одним из параллельных потоков. Подобная запись определяет последовательность прихода сигналов в рамках одного конкретного потока и дает возможность полного перебора (без явного указания всех возможных вариантов, что существенно сокращает за-

пись теста) между параллельными процессами. При этом не происходит генерации тестов для описания каждого конкретного перебора и контроль правильности работы системы осуществляется в рамках одного теста с указанным оператором. Следует отметить, что добавление оператора *par* является простой процедурой и может быть полностью автоматизировано на основе информации о порождении и синхронизации потоков.

Таким образом, представленный подход позволяет на основе анализа независимости параллельных потоков автоматически получать корректные наборы тестов для тестирования параллельных распределенных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Таненбаум Э.С., ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.

2. Almasi G.S., Gottlieb A. Highly Parallel Computing. Benjamin-Cummings publishers, Redwood City, CA, 1989.

3. ITU-T Recommendation Z.151. User requirements notation (URN), 10.2012 [электронный ресурс].

4. Баранов С.Н., Дробинцев П.Д., Летичевский А.А., Котляров В.П. Верификационная технология базовых протоколов для разработки и проектирования программного обеспечения // Программные продукты и системы. 2005. № 1(69). С. 25–28.

5. Ануреев И.С., Баранов С.Н., Белоглазов Д.М., Бодин Е.В., Дробинцев П.Д., Колчин А.В., Котляров В.П., Летичевский А.А., Летичевский А.А. мл., Непомнящий В.А., Никифоров И.В., Потиевко С.В., Прийма Л.В., Тютин Б.В. Средства поддержки интегрированной технологии для анализа и верификации спецификаций телекоммуникационных приложений // Труды

СПИИ РАН. 2013. № 3(26). С. 349–384.

6. ITU Recommendation Z.120. Message Sequence Charts (MSC), 11.1999 [электронный ресурс].

7. Никифоров И.В., Котляров В.П., Дробинцев П.Д. Ограничения на многопоточные конструкции и временные задержки языка UCM // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2013. № 3(174). С. 148–153.

8. Бурдонов И.Б., Грошев С.Н., Демаков А.В., Камкин А.С., Косачев А.С., Сортов А.А. Параллельное тестирование больших автоматных моделей // Информационные технологии. Вестник Нижегородского университета им. Н.И. Лобачевского. 2011. № 3(2). С. 187–193.

REFERENCES

1. Tanenbaum A.S., van Steen M. *Raspredeleynnyye sistemy. Printsipy i paradigmy [Distributed systems. Principles and paradigms]*. St. Petersburg: Piter Publ., 2003, 877 p. (rus)

2. Almasi G.S., Gottlieb A. *Highly Parallel Computing*. Benjamin-Cummings publishers, Redwood City, CA, 1989.

3. ITU-T Recommendation Z.151. User requirements notation (URN), 10.2012.

4. Baranov S.N., Drobintsev P.D., Letichevskiy A.A., Kotlyarov V.P. Verifikatsionnaya tekhnologiya bazovykh protokolov dlya razrabotki i proyektirovaniya programmogo obespecheniya, *Programmnyye produkty i sistemy*, 2005, No. 1(69), Pp. 25–28. (rus)

5. Anureyev I.S., Baranov S.N., Beloglazov

D.M., Bodin Ye.V., Drobintsev P.D., Kolchin A.V., Kotlyarov V.P., Letichevskiy A.A., Letichevskiy A.A. ml., Nepomnyashchiy V.A., Nikiforov I.V., Potiyenko S.V., Priyma L.V., Tyutin B.V. Sredstva podderzhki integrirovannoy tekhnologii dlya analiza i verifikatsii spetsifikatsiy telekommunikatsionnykh prilozheniy, *Trudy SPII RAN*, 2013, No. 3(26), Pp. 349–384. (rus)

6. ITU Recommendation Z.120. Message Sequence Charts (MSC). 11.1999.

7. Nikiforov I.V., Kotlyarov V.P., Drobintsev P.D. Ogranicheniya na mnogopotochnyye konstruksii i vremennyye zaderzhki yazyka UCM, *Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie*. St. Petersburg: SPbGPU Publ., 2013, No. 3(174), Pp. 148–153. (rus)

ДРОБИНЦЕВ Павел Дмитриевич — доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: drob@ics2.ecd.spbstu.ru

DROBINTSEV, Pavel D. St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: drob@ics2.ecd.spbstu.ru

НИКИФОРОВ Игорь Валерьевич — ассистент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: igor.nikiforovv@gmail.com

NIKIFOROV, Igor V. St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: igor.nikiforovv@gmail.com

ВОИНОВ Никита Владимирович – доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: voinov@ics2.ecd.spbstu.ru

VOINOV, Nikita V. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: voinov@ics2.ecd.spbstu.ru

КОТЛЯРОВ Всеволод Павлович – профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: vpk@spbstu.ru

KOTLYAROV, Vsevolod P. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: vpk@spbstu.ru

УДК 004.65

Д.И. Осенняя, И.И. Закирова, П.Д. Дробинцев

РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ ПОДБОРА КОНФИГУРАЦИЙ ИНФОРМАЦИОННЫХ ЦЕНТРОВ

D.I. Osennaya, I.I. Zakirova, P.D. Drobintsev

DEVELOPING SOFTWARE FOR DATA CENTER CONFIGURATION

Изучена разработка программного продукта для автоматизированного подбора конфигураций информационных центров. Рассмотрены разработанная методика для подбора конфигураций информационных центров и подход к анализу характеристик полученной системы.

ИНФОРМАЦИОННЫЙ ЦЕНТР; EMC; ECLIPSE; EMF; GMF.

The paper describes an approach to the development of software products aimed at automating configuration of data centers. Also, the paper presents methods to select configurations for information centers and an approach to analyse their properties.

DATA CENTER; EMC; ECLIPSE; EMF; GMF.

Хранение информации – важнейшая составляющая информационных технологий. Объем информации, производимой компаниями и частными лицами, растет с экспоненциальной скоростью. И вместе с этим растет и сложность систем, обеспечивающих хранение таких объемов данных.

Успех в бизнесе зависит от быстрого и надежного доступа к соответствующей информации. Значимость информации для бизнеса постоянно увеличивается, вследствие чего усиливается потребность в защите и управлении данными. Информацию необходимо эффективно хранить, защищать, оптимизировать и управлять ею.

Для централизованной обработки информации всего предприятия организации содержат информационные центры. Информационные центры хранят огромное количество данных, необходимых для выполнения различных задач, и управляют этими данными [1].

Деятельность многих организаций предъявляет чрезвычайно высокие требования к доступности и производительности приложений, работающих с данными. Таковыми являются, например, финансовые организации, для которых важны как производительность (поскольку задержки в

функционировании основных систем могут стоить миллионы долларов), так и надежность (поскольку потеря важной финансовой информации может повлечь за собой еще более тяжелые последствия) [7].

Корпорация EMC [3] помогает предприятиям и поставщикам услуг преобразовать свои бизнес-процессы и предоставлять информационные технологии как услугу. В основе этой трансформации лежит облачная инфраструктура. Благодаря инновационным продуктам и услугам EMC ускоряет переход к облачным технологиям, позволяя ИТ-подразделениям более гибко, надежно и экономично хранить и защищать информацию, а также управлять ею.

Каждому поставщику ИТ-услуг требуется определенная конфигурация информационного центра, которая отвечает его запросам, удовлетворяет специфике поставленной задачи, а также обеспечивает оптимальное соотношение производительности, надежности и совокупной стоимости [8]. Поэтому процесс подбора конфигурации каждой системы трудоемок, требует индивидуального подхода и становится особенно актуальной задачей с ростом объема информации, подлежащей хранению.

Часто за основу информационного цен-

тра берется приблизительная конфигурация системы, которая в дальнейшем расширяется с помощью рекомендаций систем мониторинга. Эти системы определяют, какие элементы в информационном центре не справляются с нагрузкой и где требуется добавление дополнительных ресурсов.

Существует большое количество таких систем мониторинга. В арсенале компании EMC есть продукт ViPR SRM, позволяющий визуализировать взаимосвязи в информационном центре, анализировать тенденции нагрузок и конфигурацию системы, получать графики зависимостей приложений от информационного центра.

Компания IBM предлагает продукт Tivoli Storage Productivity Centre, обеспечивающий схожие сервисы: мониторинг информационного центра, формирование отчетов и управление емкостью, сетями, системами и сервисами репликации.

Данный подход нельзя назвать эффективным, т. к. конфигурация информационного центра, взятая за основу, часто бывает очень далека от оптимальной. Поэтому первоначальную конфигурацию системы необходимо выбирать, опираясь на результаты предварительного моделирования системы, в ходе которого рассчитываются ожидаемые характеристики надежности и производительности.

В настоящей статье изучаются методы для предварительного моделирования системы, которые дают возможность проводить ее анализ без непосредственной сборки системы, что позволяет существенно сократить затраты на ее построение.

Моделирование информационных систем

В качестве основы для моделирования информационных центров используется предоставление бизнес-процесса предприятия в разрезе информационных потоков и, в конечном счете, приложений, которые ими оперируют. Однако в данном подходе наблюдается ряд сложностей, основной из которых является разрыв между представлением менеджмента о бизнес-процессах, выражающего требования к информационному центру в терминах приложений и количества пользователей, и ИТ специалиста,

мыслящего категориями объемов памяти хранилищ, количества операций ввода/вывода в секунду и типов элементов, из которых строится система. Так, например, менеджер может указать в качестве требования необходимость размещения на системе хранения приложения Microsoft Exchange с общим количеством пользователей 50 человек и средней интенсивностью входящих и исходящих писем равной 20. Но данное требование абсолютно непонятно ИТ инженеру, которого в конечном итоге интересует количество оперативной памяти или частота и количество процессоров в составе серверов. Решение данной проблемы лежит в области совместной работы менеджмента, представляющего интересы бизнеса, и инженеров, являющихся специалистами в области построения информационных центров.

Для решения проблемы предлагается разработать программный продукт, позволяющий подбирать конфигурацию информационного центра в зависимости от требований заказчика к степени защищенности данных, с которыми оперируют его приложения, быстродействию приложений, а также к стоимости. Кроме того, приложение должно позволять рассчитывать различные характеристики будущей системы, определять ее «узкие места» и отказоустойчивость при выходе из строя различных элементов.

Предлагается собирать данные о приложениях в формате xls таблиц, которыми удобно оперировать менеджеру при составлении требований и строить графическое представление модели информационного центра, на основе которого обеспечить коммуникации менеджмента и инженеров. В результате подобный программный продукт позволит визуализировать предложенное решение для информационного центра и проводить ручное редактирование как набора составных элементов, так и их свойств.

Методика подбора конфигураций информационных центров

В процессе изучения предметной области была разработана методика подбора конфигураций информационных центров (рис. 1). Основой методики является созда-

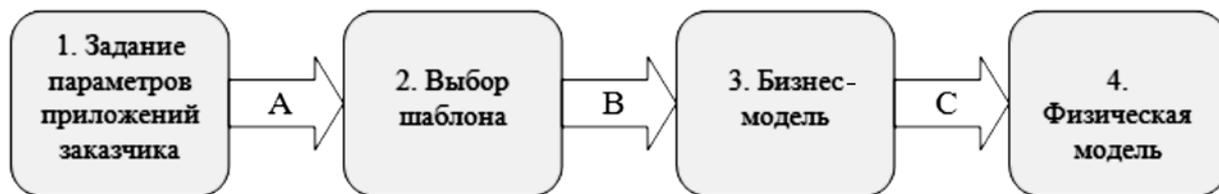


Рис. 1. Методика построения информационных центров

ние моделей различных уровней, которые могут использоваться как при формировании требований бизнеса к будущей системе, так и при разработке детального представления информационного центра на уровне физических элементов (серверов, коммутаторов и т. д.).

Рассмотрим более детально процесс подбора конфигураций информационных центров по предлагаемой методике.

На этапе 1 происходит инициализация исходных данных. В диалоговом окне пользователь указывает xls-файлы, которые содержат характеристики и требования приложений. Основными характеристиками является количество операций ввода-вывода в секунду (IOPS), средний размер одного блока информации, процент операций чтения и необходимый объем памяти. Требования приложений определяют уровень защиты от сбоев серверов и массивов данных. Таблицы составляются менеджерами, исходя из знания и понимания бизнес-процессов. После окончания этапа загруженная информация отображается в диалоговом окне и может быть при необходимости отредактирована.

На этапе 2 с помощью набора правил А, на основе загруженных на предыдущем

шаге данных, происходит выбор списка шаблонов, т. е. часто используемых конфигураций бизнес-моделей, которые подходят в данном случае. Бизнес-модель (БМ) – это графическое представление структуры информационного центра в виде логических сущностей (таких как кластер серверов, уровень виртуализации, система хранения данных [1]) и связей между ними. Здесь пользователю предоставляется выбор: отдать предпочтение системе с большей надежностью, но и с большей стоимостью, или же более дешевой и менее надежной системе.

Набор правил А, который применяется в прототипе программного продукта и позволяет задавать три уровня защиты от сбоев и использовать пять шаблонов, может быть представлен в виде таблицы (см. табл.).

На этапе 3 элементы шаблона с помощью набора правил В, заполняются свойствами на основе загруженных на первом этапе данных, и полученная модель визуализируется. На этом этапе пользователю предоставляется возможность редактирования модели для более тонкой настройки.

Одно из правил набора В, которое позволяет рассчитать количество серверов на каждом хосте, представлено ниже:

Выбор шаблонов на основе требований приложений

Уровень защиты от сбоев серверов	Уровень защиты от сбоев массивов данных	Шаблоны, отвечающие требованиям
Высокий	*(любой)	Gold, Diamond
*(любой)	Высокий	Gold, Diamond
Средний	Средний	Silver, Gold, Diamond
Средний	Низкий	Bronze, Silver, Gold, Diamond
Низкий	Средний	Bronze, Silver, Gold, Diamond
Низкий	Низкий	Wood, Bronze, Silver, Gold, Diamond

$$N_{\text{Server}} = \frac{\sum_{i=1}^k (\text{size } W[i] \cdot W\%[i] + \text{size } R[i] \cdot R\%[i]) \cdot \text{IOPS}[i]}{p},$$

где k – количество приложений в системе; p – пропускная способность портов хоста; $\text{size } W$, $\text{size } R$ – размеры блоков информации на запись и чтение соответственно; $W\%$, $R\%$ – процент операций записи и чтения.

На этапе 4 с использованием набора правил преобразования С, происходит преобразование схемы БМ в схему ФМ и ее визуализация. Физическая модель (ФМ) – подробное графическое представление конфигурации системы с помощью реальных устройств (таких как сервер, switch, VPLEX, RPA [2]). Как и для БМ, на этом

этапе существует возможность редактирования схемы. Также появляется возможность моделирования отказов элементов системы и определения «узких мест».

Описание одного из правил набора С приведено ниже. Элемент БМ Replication преобразуется в набор элементов ФМ – Recover Point Appliances (RPA) – устройств, обеспечивающих репликацию данных. Их количество рассчитывается исходя из пропускной способности элемента БМ Storage, с которым связан данный элемент Replication:

$$N_{\text{RPA}} = \left[\frac{\text{Storage.CommonTrafficCapacity_GbPerSec}}{P_{\text{portRPA}}} \right] + 1,$$

где $P_{\text{portRPA}} = \text{TrafficCapacity_GbitPerSec}$ – пропускная способность порта RPA.

В дальнейшем на основе ФМ проводится анализ системы на соответствие ее характеристик требованиям заказчика. В ходе работы рассмотрено две области характеристик информационных центров: надежность и производительность [8].

Надежность в данном случае характеризуется следующими параметрами:

RPO (Recovery Point Objective, Директивный срок восстановления);

RTO (Recovery Time Objective, Директивное время восстановления);

СМВО (Среднее время между отказами), СВВ (Среднее время восстановления) [1].

Параметры RPO и RTO являются очень удобными для задания требований к системе, т. к. они понятны заказчику и в то же время очень многое определяют для исполнителя.

RPO – это момент времени, к которому система должна быть восстановлена после простоя. Он характеризует тот объем данных, который предприятие может позволить себе потерять. Чем больше RPO, тем выше устойчивость бизнеса заказчика к информационным потерям. Основным определяющим фактором RPO является период

создания резервной копии.

RTO – промежуток времени, за который система, приложения и функции восстанавливаются после сбоя. Этот параметр определяет время, которое организация может позволить себя находиться в состоянии простоя и при этом избежать в дальнейшем серьезных последствий.

Рассчитанные в автоматическом режиме характеристики производительности позволяют показать, насколько оптимальна подобранная конфигурация информационного центра и какова рабочая эффективность всех компонент.

В рамках анализа производительности системы оцениваются такие характеристики, как время отклика, утилизация, емкость.

Время отклика – это обратное время запроса на операцию ввода/вывода, т. е. общее время от прибытия заявки до отправки из системы. Среднее время отклика зависит как от времени обслуживания и времени, затрачиваемого на обработку запроса, так и от загрузки системы. Несмотря на то что основным критерием работоспособности системы является доступность информации, время отклика может быть настолько большим, что работу системы нельзя будет назвать приемлемой.

Оценка уровня утилизации применима как к отдельным компонентам системы, так и к группам однотипных элементов и показывает, на сколько процентов используются доступные ресурсы устройств. Элементы с высоким уровнем утилизации могут стать причиной снижения производительности системы, поэтому они должны быть выявлены и, по возможности, либо продублированы, либо оснащены дополнительными ресурсами. Очень низкий уровень утилизации может указывать на то, что устройства «простаивают» и, возможно, для поставленных требований может быть разработана более оптимальная конфигурация системы.

Емкость характеризует сумму доступных ресурсов. Например, это может быть количество доступных портов на коммутаторе, или объем дискового пространства в хранилище. Управление емкостью лежит в основе обеспечения требований заказчика на работу приложений и уровень обслуживания.

Входными данными для расчета описанных характеристик информационного центра является ФМ системы, где для каж-

дого элемента известен набор характеристик, таких как средняя наработка до отказа, среднее время восстановления, скорость обработки информации и т. д. Также необходимо задать условия работоспособности системы в целом.

Описанная методика была использована при создании продукта DataCenter, предназначенного для построения, конфигурации и анализа информационных центров. Разработка продукта производилась в среде Eclipse с помощью инструментов Eclipse Modelling Framework (EMF) [4] и Graphical Modelling Framework (GMF) [5]. Для поддержки работы продукта с моделями различных уровней на первом этапе создаются EMF [4] метамодели классов и их связей друг с другом для БМ и ФМ, и по ним генерируется Java-код этих классов. На втором этапе, с помощью инструмента GMF, на основе сгенерированного кода классов моделей генерируется код для графических редакторов. Далее, оба полученных редактора объединяются в одно RCP (Rich Client Platform) [6] приложение, в которое по мере необходимости разработчик вручную вносит требуемые изменения.

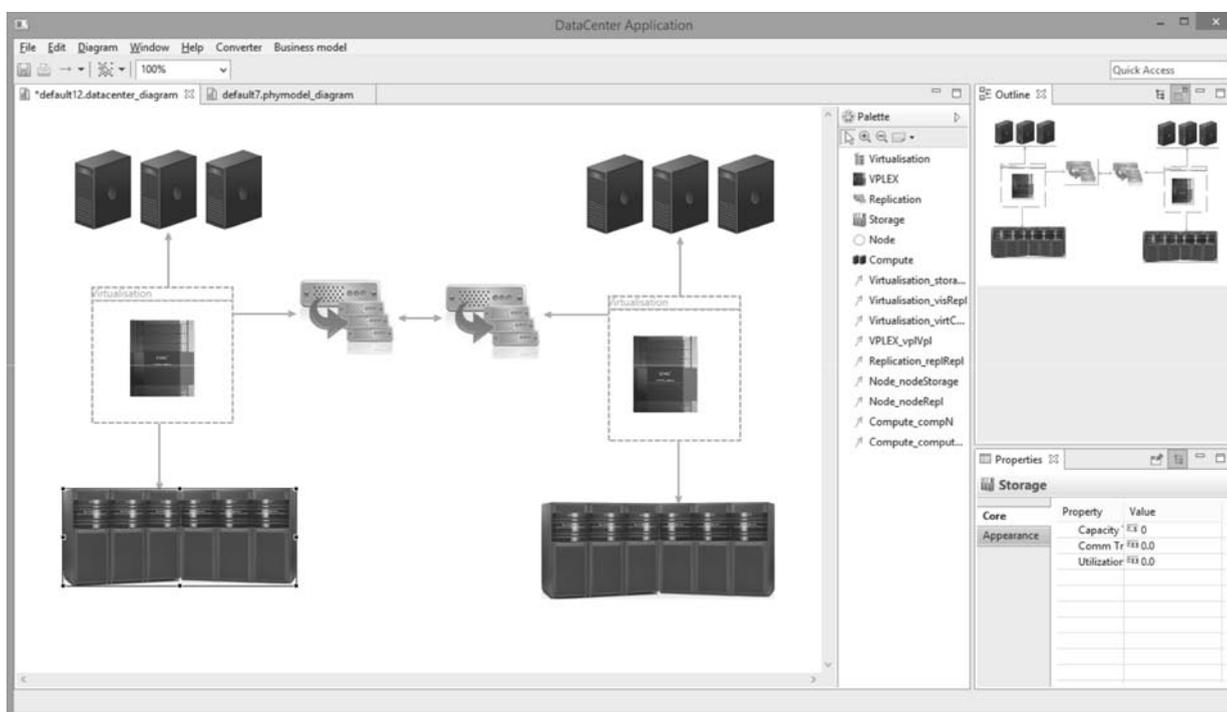


Рис. 2. Приложение DataCenter. Бизнес-модель

К настоящему моменту разработан прототип программного продукта, позволяющий:

1) создавать, редактировать, сохранять (в формате xml) БМ и ФМ;

2) читать информацию о приложениях из xls-файлов, и на их основе выбирать список подходящих шаблонов БМ;

3) заполнять свойствами и визуализировать выбранный шаблон;

4) генерировать на основе БМ ФМ с использованием временных правил преобразования;

5) проводить расчет и анализ характеристик надежности и производительности создаваемого информационного центра.

На рис. 2 изображено главное окно приложения DataCenter. Слева область редактора бизнес-модели, в нем визуализируется конфигурация информационного центра с двумя кластерами серверов, двумя уровнями виртуализации, содержащими по одному VPLEX [1], метро репликацией и двумя системами хранения данных. Правее модели находится палитра, а в правом нижнем углу параметры выделенного элемента (в данный момент это кластер массивов хра-

нения данных [1]).

Результаты. Четырехэтапная методика, описанная выше, лежит в основе проекта DataCenter и на данной стадии разработки показывает неплохие результаты для решения задачи конфигурирования информационных центров. Используя даже тестовые алгоритмы, она позволяет получить прототип результирующей модели. Несмотря на стремление к автоматизации процесса, на каждом этапе пользователю доступно ручное редактирование параметров системы, в будущем это должно позволить моделировать самые разнообразные индивидуальные решения. Усовершенствуя правила преобразования и расширяя набор встроенных шаблонов, можно добиться более точной работы программы.

Разрабатываемый программный продукт позволяет автоматизировать процесс подбора оптимальной конфигурации информационных центров для конкретных заказчиков, а также оценивать характеристики и отказоустойчивость будущей системы что существенно снижает затраты на построение информационных центров.

СПИСОК ЛИТЕРАТУРЫ

1. От хранения данных к управлению информацией. EMC. СПб.: Питер, 2010. 544 с.
2. **Dharma R., Jiang L.** Networking for Storage Virtualization and EMC RecoverPoint. EMC, 2013. 102 с.
3. EMC – ведущий поставщик проверенных ИТ-решений, технологий для облачных вычислений и работы с большими данными [электронный ресурс] / URL: <http://russia.emc.com/index.htm> (дата обращения 10.11.2013).
4. Eclipse Modelling Framework Project (EMF): [электронный ресурс] / URL: <http://www.eclipse.org/modeling/emf/> (дата обращения 11.11.2013).
5. Model Driven Architecture approach to

- domain of graphical editors [электронный ресурс] / URL: <http://eclipse.org/gmf-tooling/> (дата обращения 11.11.2013).
6. Rich Client Platform: [электронный ресурс] / URL: http://wiki.eclipse.org/index.php/Rich_Client_Platform (дата обращения 11.11.2013).
7. **Горбунов-Посадов М.М., Корягин А.Н.** Современные системы хранения данных старшего класса. М.: Препринты ИПМ им. М.В. Келдыша, 2008.
8. **Лобанов А.К.** Методы построения систем хранения данных [электронный ресурс] / URL: <http://citforum.ru/hardware/data/db/> (дата обращения 24.05.2014).

REFERENCES

1. *От khraneniya dannyh k upravleniyu informatsiyey.* EMC, St. Petersburg: Piter Publ., 2010, 544 p. (rus)
2. **Dharma R., Jiang L.** *Networking for Storage Virtualization and EMC RecoverPoint.* EMC, 2013, 102 p.
3. *EMC – vedushchiy postavshchik proverennykh IT-resheniy, tekhnologiy dlya oblachnykh vychisleniy*

- i raboty s bolshimi dannymi* Available: <http://russia.emc.com/index.htm> (Accessed 10.11.2013). (rus)
4. *Eclipse Modelling Framework Project (EMF).* Available: <http://www.eclipse.org/modeling/emf/> (Accessed 11.11.2013).
5. *Model Driven Architecture approach to domain of graphical editors.* Available: <http://eclipse.org/gmf-tooling/> (Accessed 11.11.2013).

6. *Rich Client Platform*. Available: http://wiki.eclipse.org/index.php/Rich_Client_Platform (Accessed 11.11.2013).

7. **Gorbunov-Posadov M.M., Koryagin A.N.** *Sovremennyye sistemy khraneniya dannykh starshego*

klassa. Moscow: Preprinty IPM im. M.V. Keldysha Publ., 2008. (rus)

8. **Lobanov A.K.** *Metody postroyeniye sistem khraneniya dannykh*. Available: <http://citforum.ru/hardware/data/db/> (Accessed 24.05.2014). (rus)

ОСЕННЯЯ Дарья Игоревна – студентка кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: osen1313@gmail.com

OSENNYA, Daria I. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: osen1313@gmail.com

ЗАКИРОВА Ирина Игоревна – студентка кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: from.mumla@gmail.com

ZAKIROVA, Irina I. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: from.mumla@gmail.com

ДРОБИНЦЕВ Павел Дмитриевич – доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: drob@ics2.ecd.spbstu.ru

DROBINTSEV, Pavel D. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: drob@ics2.ecd.spbstu.ru

ПРИГЛАШАЕМ НА КОНФЕРЕНЦИЮ



КОСТРОМА, КГТУ, 14-15 НОЯБРЯ 2014 ИНСТРУМЕНТЫ И МЕТОДЫ АНАЛИЗА ПРОГРАММ

Приглашаем студентов старших курсов и преподавателей принять участие в Международной научно - практической конференции «Tools & Methods of Program Analysis» («Инструменты и методы анализа программ, ТМРА-2014»), посвящённой одному из наиболее актуальных и важных направлений программной инженерии - анализу качества программного обеспечения.

Конференция пройдёт 14-15 ноября 2014 года в Костромском государственном технологическом университете (КГТУ) и соберёт студентов, аспирантов и молодых учёных, специализирующихся в области разработки программного обеспечения и заинтересованных в независимой экспертизе своих исследований, а также признанных учёных, готовых поделиться своими инновационными разработками. Организаторы ожидают интересных практических докладов и от ведущих фирм-разработчиков программного обеспечения.

К участию в программе конференции принимаются заявки в форме конкурсных докладов для отбора на рецензирование независимыми экспертами конференции «ТМРА-2014», в качестве которых выступают российские и зарубежные учёные в области разработки и анализа ПО.

Организаторы конференции:

Факультет автоматизированных систем и технологий, Костромской государственной технологической университет (ФАСТ КГТУ: fast.kstu.edu.ru)

Институт проблем информатики Российской академии наук (ИПИ РАН: www.ipiran.ru)

Санкт-Петербургский государственный политехнический университет (СПбГПУ: spbstu.ru), кафедра компьютерных систем и программных технологий института информационных технологий и управления СПбГПУ (КСПТ ИИТУ СПбГПУ: kspt.icc.spbstu.ru)

Компания «Инновационные Трейдинговые Системы» (ИТС: its-trading.ru)

Крайний срок подачи статей - 8 сентября 2014 года.

Регистрация на конференцию уже открыта! Количество мест ограничено.

По вопросам участия обращаться по телефону +7 495 640 24 60 или по электронной почте: program@tmpaconf.org

Подробности на официальном сайте конференции tmpaconf.org и в социальных сетях:

ВКонтакте



tmpaconf.org



Facebook



LinkedIn



Организаторы и партнеры:



НАУЧНОЕ ИЗДАНИЕ
«НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ
САНКТ-ПЕТЕРБУРГСКОГО
ГОСУДАРСТВЕННОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА.
ИНФОРМАТИКА. ТЕЛЕКОММУНИКАЦИИ. УПРАВЛЕНИЕ»
«ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY JOURNAL.
COMPUTER SCIENCE. TELECOMMUNICATIONS AND CONTROL SYSTEMS»

№ 3 (198) 2014

Учредитель – Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный политехнический университет»

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор).
Свидетельство о регистрации ПИ № ФС77-51457 от 19.10.2012 г.

Редакция журнала

д-р техн. наук, профессор *А.С. Коротков* – главный редактор
Е.А. Калинина – литературный редактор, корректор
Г.А. Пышкина – ответственный секретарь, выпускающий редактор

Телефон редакции (812)552-62-16, 297-18-21

E-mail: infocom@spbstu.ru

Компьютерная верстка *А.Н. Смирнов*

Директор Издательства Политехнического университета *А.В. Иванов*

Лицензия ЛР № 020593 от 07.08.97

Подписано в печать 27.06.2014. Формат 60×84 1/8. Бум. тип. № 1.
Печать офсетная. Усл. печ. л. 11,95. Уч.-изд. л. 11,95. Тираж 1000. Заказ

Санкт-Петербургский государственный политехнический университет
Издательство Политехнического университета
член Издательско-полиграфической ассоциации университетов России
Адрес университета и издательства: 195251, Санкт-Петербург, ул. Политехническая, д. 29.

УСЛОВИЯ ПУБЛИКАЦИИ СТАТЕЙ

в журнале «Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление»

1. ОБЩИЕ ПОЛОЖЕНИЯ

Журнал «Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление» является периодическим печатным научным рецензируемым изданием. Зарегистрировано Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ПИ № ФС77-51457 от 19 октября 2012 г. С 2008 года выпускается в составе сериального периодического издания «Научно-технические ведомости СПбГПУ» (ISSN 1994-2354).

Издание с 2002 года входит в Перечень ведущих научных рецензируемых журналов и изданий (перечень ВАК) и принимает для печати материалы научных исследований, а также статьи для опубликования основных результатов диссертаций на соискание ученой степени доктора наук и кандидата наук по следующим основным научным направлениям: **ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, РАДИОТЕХНИКА И СВЯЗЬ, ЭЛЕКТРОНИКА, ИЗМЕРИТЕЛЬНАЯ ТЕХНИКА, УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ И ЭКОНОМИЧЕСКИХ СИСТЕМАХ**. Научные направления журнала учитываются ВАК Минобрнауки РФ при защите докторских и кандидатских диссертаций в соответствии с Номенклатурой специальностей научных работников.

Сведения о публикациях представлены в РИНЦ, в Реферативном журнале ВИНИТИ РАН, в международной справочной системе «Ulrich's Periodical Directory».

Периодичность выхода журнала – 6 номеров в год.

2. ТРЕБОВАНИЯ К ПРЕДОСТАВЛЯЕМЫМ МАТЕРИАЛАМ

2.1. Оформление материалов

1. Рекомендуемый объем статей для авторов с ученой степенью доктора наук, званием профессора, соискателей ученой степени доктора наук (докторантов) 12–20 страниц формата А-4 с учетом графических вложений. Количество графических вложений (диаграмм, графиков, рисунков, таблиц, фотографий и т. п.) не должно превышать 4.

2. Рекомендуемый объем статей для преподавателей, авторов без ученой степени, соискателей ученой степени кандидата наук – 8–15 страниц формата А-4; аспирантов – 8 страниц формата А-4 с учетом графических вложений. Количество графических вложений (диаграмм, графиков, рисунков, таблиц, фотографий и т. п.) не должно превышать 3.

3. Авторы должны придерживаться следующей обобщенной структуры статьи: вводная часть (0,5–1 стр., актуальность, существующие проблемы); основная часть (постановка и описание задачи, изложение и суть основных результатов); заключительная часть (0,5–1 стр., предложения, выводы), список литературы (оформление по ГОСТ 7.05.-2008).

4. Число авторов статьи не должно превышать трех человек.

5. Набор текста осуществляется в редакторе **MS Word**, формул – в редакторе **MathType**. Таблицы набираются в том же формате, что и основной текст.

6. Шрифт – **TNR**, размер шрифта основного текста – 14, интервал – 1,5; таблицы большого размера могут быть набраны 12 кеглем. Параметры страницы: поля слева – 3 см, сверху, снизу – 2,5 см, справа – 2 см, текст размещается без переносов. Абзацный отступ – 1 см.

2.2. Предоставление материалов

Вместе с материалами статьи должны быть обязательно предоставлены:

- номер УДК в соответствии с классификатором (в заголовке статьи);
- аннотация на русском и английском языках;
- ключевые слова (5–7) на русском и английском языках;
- сведения об авторах на русском и английском языках: ФИО, место работы, должность, ученое звание, ученая степень, контактные телефоны, e-mail;
- аспиранты представляют документ отдела аспирантуры, заверенный печатью;
- акт экспертизы о возможности опубликования материалов в открытой печати.

С авторами статей заключается издательский лицензионный договор.

Предоставление всех материалов осуществляется в электронном виде через личный кабинет **ЭЛЕКТРОННОЙ РЕДАКЦИИ** по адресу <http://journals.spbstu.ru>

2.3. Рассмотрение материалов

Предоставленные материалы (п. 2.2) первоначально рассматриваются редакционной коллегией и передаются для рецензирования. После одобрения материалов, согласования различных вопросов с автором (при необходимости) редакционная коллегия сообщает автору решение об опубликовании статьи. В случае отказа в публикации статьи редакция направляет автору мотивированный отказ.

При отклонении материалов из-за нарушения сроков подачи, требований по оформлению или как не отвечающих тематике журнала материалы не публикуются и не возвращаются.

Редакционная коллегия не вступает в дискуссию с авторами отклоненных материалов.

Публикация материалов аспирантов очной бюджетной формы обучения осуществляется бесплатно в соответствии с очередностью.

При поступлении в редакцию значительного количества статей их прием в очередной номер может закончиться **ДОСРОЧНО**.

Более подробную информацию можно получить:

на сайте журнала <http://ntv.spbstu.ru>

по телефону редакции +7(812) 552-62-16 с 10⁰⁰ до 18⁰⁰ Галина Александровна

или по e-mail: infocom@spbstu.ru