

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

ПРИБОРОСТРОЕНИЕ

ИЗДАНИЕ САНКТ-ПЕТЕРБУРГСКОГО НАЦИОНАЛЬНОГО ИССЛЕДОВАТЕЛЬСКОГО УНИВЕРСИТЕТА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Журнал издается с января 1958 г.

ТОМ 57

АПРЕЛЬ 2014

№ 4

СПЕЦИАЛЬНЫЙ ВЫПУСК
к 100-летию С. А. МАЙОРОВА

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

*Под редакцией доктора технических наук, профессора Т. И. Алиева
и доктора технических наук, профессора А. А. Ожиганова*

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ.....	5
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ	
Батура В. А., Тропченко А. Ю. Эффективность алгоритмов маркирования цифровых изображений в частотной области на основе дискретного преобразования Адамара.....	7
Щеглов К. А., Щеглов А. Ю. Непротиворечивая модель мандатного контроля доступа.....	12
Маркина Т. А., Щеглов А. Ю. Метод защиты от атак типа drive-by-загрузка.....	15
Тропченко А. А. Методы повышения робастности распознавания в мультимодальных биометрических системах.....	20
Калинин И. В., Клименков С. В., Харитонов А. Е., Цопа Е. А. Преобразование естественного языка в формат RDF с помощью семантических анализаторов текстовой информации.....	23
Дергачев А. М., Дергачев А. А. Параметры качества обслуживания web-сервисов отрасли приборостроения.....	27
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ	
Алиев Т. И. Проектирование систем с приоритетами.....	30
Муравьева-Витковская Л. А. Метод расчета характеристик замкнутых детерминированных моделей мультисервисных компьютерных сетей.....	35
Соснин В. В. Время ожидания в неоднородных системах с очередями при обслуживании заявок в порядке поступления.....	39
Богатырев В. А., Богатырев А. В., Богатырев С. В. Оптимизация перераспределения нагрузки в кластерах при изменяющейся активности источников запросов.....	41
Богатырев В. А., Богатырев А. В., Богатырев С. В. Оценка надежности выполнения кластерами запросов реального времени.....	46

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

Платунов А. Е. Реконфигурируемые встраиваемые системы и системы на кристалле ...	49
Антонов А. А., Быковский С. В., Кустарев П. В. Монитор временных ограничений для систем на кристалле	53
Поляков В. И., Скорубский В. И. Использование многозначной логики при проектировании функциональных схем	57
Кормилицын А. Ю., Поляков В. И. Методы и средства мониторинга дыхания	61
Гедич А. А., Зыков А. Г., Лаздин А. В., Поляков В. И. Поиск процедур по графу переходов функциональной программы при верификации вычислительных процессов	64
Ожиганов А. А. Квазиабсолютный цифровой преобразователь угла на основе двух-дорожечной рекурсивной нелинейной кодовой шкалы	69
Ростовский К. М., Ожиганов А. А. Кодовые шкалы на основе инверсно-сопряженных двоичных последовательностей	74
SUMMARY (<i>перевод Ю. И. Копилевича</i>)	78

SPECIAL ISSUE
to 100th anniversary OF S. A. MAIOROV

INFORMATION TECHNOLOGIES AND SYSTEMS

*By Edition of T. I. Aliev, Doctor of Technical Science, Professor,
A. A. Ozhiganov, Doctor of Technical Science, Professor*

CONTENTS

PREFACE.....	5
INFORMATION TECHNOLOGIES	
Batura V. A., Tropchenko A. Yu. Efficiency of Frequency Domain Algorithm of Digital Image Watermarking Based on Discrete Hadamard Transform.....	7
Shcheglov K. A., Shcheglov A. Yu. A Consistent Model of Mandatory Access Control.....	12
Markina T. A., Shcheglov A. Yu. A Method of Protection against Drive-By Download Attacks.....	15
Tropchenko A. A. Ways for Improving Recognition Robustness in Multimodal Biometric Systems.....	20
Kalinin I. V., Klimenkov S. V., Kharitonova A. E., Tsopa E. A. Transformation of a Natural Language to RDF Format Using Semantic Analyzers of Textual Information ...	23
Dergachev A. M., Dergachev A. A. Service Quality Parameters of Electronic Industry Web Services.....	27
COMPUTER SYSTEMS AND NETWORKS	
Aliev T. I. Design of Systems with Priorities	30
Muravyeva-Vitkovskaya L. A. A Method for Calculation of Characteristics of Closed Deterministic Models of Multi-Service Computer Networks	35
Sosnin V. V. Waiting Time in FIFO-Based Multi-Class Queuing Systems	39
Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Optimization of Queries Redistribution in Clusters at Changing Activity of the Queries Sources	41
Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Estimation of Reliability of Execution of Real-Time Queries	46
SOFT HARDWARE OF INFORMATION-CONTROL SYSTEMS	
Platunov A. E. Reconfigurable Embedded Systems and System-on-Chip.....	49
Antonov A. A., Bykovsky S. V., Kustarev P. V. Transaction-Level Real-Time Constraints Monitor for System on Chip.....	53

Polyakov V. I., Skorubsky V. I. Application of Multivalued Logic in Functional Circuits Development.....	57
Kormilitsyn A. Yu., Polyakov V. I. Methods and Means for Monitoring of Respiration.....	61
Gedich A. A., Zykov A. G., Lazdin A. V., Polyakov V. I. Search for Procedure with the Use of Functional Program Transition Graph to Verify Computational Processes	64
Ozhiganov A. A. Quasi-Absolute Digitizer of Corner on the Base of Dual-Track Recursive Nonlinear Code Scale	69
Rostovsky K. M., Ozhiganov A. A. Code Scales on the Base of Inversely Conjugated Binary Sequences	74
SUMMARY	78

ПРЕДИСЛОВИЕ

В 2013 году Санкт-Петербургский национальный исследовательский университет информационных технологий, механики (НИУ ИТМО) стал одним из 15 победителей конкурса на предоставление государственной поддержки ведущим российским вузам в целях повышения их конкурентоспособности среди ведущих мировых научно-образовательных центров (проект 5-100). Один из основных показателей эффективности вуза в рамках этого конкурса – уровень научных исследований.

Кафедра вычислительной техники (ВТ), являясь одной из старейших и крупнейших в НИУ ИТМО, ведет подготовку специалистов высшей квалификации в области исследования, проектирования и разработки средств обработки и передачи данных. Высокий уровень подготовки обеспечивается участием студентов в научно-исследовательских и опытно-конструкторских работах.

В 2014 году исполняется 100 лет со дня рождения доктора технических наук, профессора Сергея Александровича Майорова, заведовавшего кафедрой ВТ с 1962 по 1986 год. Именно в этот период существенно увеличился объем научных исследований на кафедре, выросло число ее научных сотрудников и аспирантов, возникли новые направления в рамках научной школы кафедры, основу которой заложили ее первые руководители — профессора М. Ф. Маликов и С. А. Изенбек. В 1962 г. при кафедре была создана отраслевая лаборатория цифровых управляющих машин, развернулись работы по созданию методов проектирования цифровых вычислительных устройств, начались исследования по алгоритмизации процессов проектирования с применением методов аналитического и имитационного моделирования, были изготовлены цифровые вычислительные машины — полупроводниковая ЛИТМО-2 (1964 г.) и микропроцессорная ЛИТМО-3 (1979 г.). Итогом научно-исследовательских работ этого периода стали многочисленные публикации не только научного, но и учебно-методического характера.

В то же время стали формироваться новые направления научных исследований, по которым на кафедре до настоящего времени ведутся работы, связанные с разработкой моделей и методов исследования вычислительных систем и сетей различных классов, методов преобразования, хранения и обработки информации с использованием оптических принципов, программно-алгоритмических и аппаратных средств для цифровой обработки изображений. В последние годы активно ведутся работы в области встроенных информационно-управляющих систем, информационной безопасности и защиты информации, а также в области корпоративных и интеллектуальных информационных систем.

В предлагаемом Вашему вниманию сборнике представлены результаты выполняемых сотрудниками кафедры ВТ научно-исследовательских и опытно-конструкторских работ. Статьи помещены в три раздела: „Информационные технологии“, „Компьютерные системы и сети“, „Программно-аппаратные средства информационно-управляющих систем“, охватывающие основные направления научной школы кафедры ВТ НИУ ИТМО.

*Заведующий кафедрой
вычислительной техники НИУ ИТМО,
Заслуженный работник высшей школы РФ,
доктор технических наук, профессор Т. И. АЛИЕВ*

PREFACE

In 2013, St. Petersburg National Research University of Information Technologies, Mechanics and Optics (NRU ITMO) was one of 15 winners in contest for state grants to leading Russian institutes of higher education aimed at improvement of their competitiveness with world-leading scientific-educational centers (Project 5-100). In the frames of the contest, one of the most important efficiency indices of an institute of higher education was the level of scientific researches.

Department of Computer Technology is one of the oldest and largest departments of NRU ITMO; the department provides training of specialist in research, development, and design of means for data processing and data communication. The high training level is ensured by participation of the students in scientific researches and design projects.

In 2014, 100 years have passed since the birth of Sergey Maiorov, Doctor of technical sciences, professor. Sergey Maiorov was the Head of the Department from 1962 to 1986, the period when the department staff and number of post-graduate students increased significantly, several new lines of investigation eroded in the frames of scientific school of the department founded by professors M. F. Malikov and S. A. Izenbek, the first heads of the department. In 1962, the branch laboratory of digital control machines was created at the department, investigations in algorithmization of design processes with the use of analytical modeling and simulation methods was performed, digital computers were created — transistorized LITMO-2 (1964) and microprocessor-based LITMO-3 (1979). As a result of the activities carried out during the period, numerous publications not only of scientific but also of educational-methodical character were issued.

At the time in question, new avenues of investigation arose. The works in the areas related to development of models and methods for research of computing systems and networks of various classes, methods of transformation, storage, processing of information with the use of optical principles, algorithmic and hardware means for digital image processing have being held up to now. In recent years, active investigations are carried out in the fields of embedded information-control systems, information security and protection, as well as of corporative and intellectual information systems.

The present issue demonstrates results of scientific researches and design projects carried out by the Department of Computer Technology. The papers are assembled in three sections: “Information Technologies”, “Computer Systems and Networks”, “Soft Hardware of Information-Control System”, and covers the main lines of investigation carried out at scientific school of Department of Computer Technology of NRU ITMO.

*Head of the Department of Computer Technology, NRU ITMO,
Honored Worker of Higher Education School of RF,
Doctor of Technical Sciences,
Professor T. I. ALIEV*

В. А. БАТУРА, А. Ю. ТРОПЧЕНКО

ЭФФЕКТИВНОСТЬ АЛГОРИТМОВ МАРКИРОВАНИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ В ЧАСТОТНОЙ ОБЛАСТИ НА ОСНОВЕ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ АДАМАРА

Рассмотрены особенности применения дискретного преобразования Адамара для маркирования цифровых изображений. Исследована устойчивость алгоритма цифрового маркирования Elham к изменению размеров изображения, а также сжатие JPEG и JPEG2000.

Ключевые слова: стеганография, цифровое маркирование, преобразование Адамара, цифровое изображение, JPEG-сжатие.

Введение. В связи с широким распространением вычислительной техники актуальной стала задача защиты авторских прав на мультимедийную продукцию, в частности, неподвижные изображения. Эффективным средством решения этой задачи являются методы цифровой стеганографии [1].

В настоящей работе в качестве объекта защиты (контейнера) рассматриваются неподвижные (статические) цифровые изображения. Под скрываемым в объекте защиты сообщением подразумевается цифровой водяной знак (ЦВЗ), который не воспринимается глазом и в общем случае является некоторым двоичным кодом. Заметим, что для наглядности при извлечении ЦВЗ из маркированного изображения такой знак может представлять собой логотип [1]. На рис. 1 представлены контейнер (а), ЦВЗ — логотип (б) и стеганоконтейнер — изображение со встроенным ЦВЗ (в).

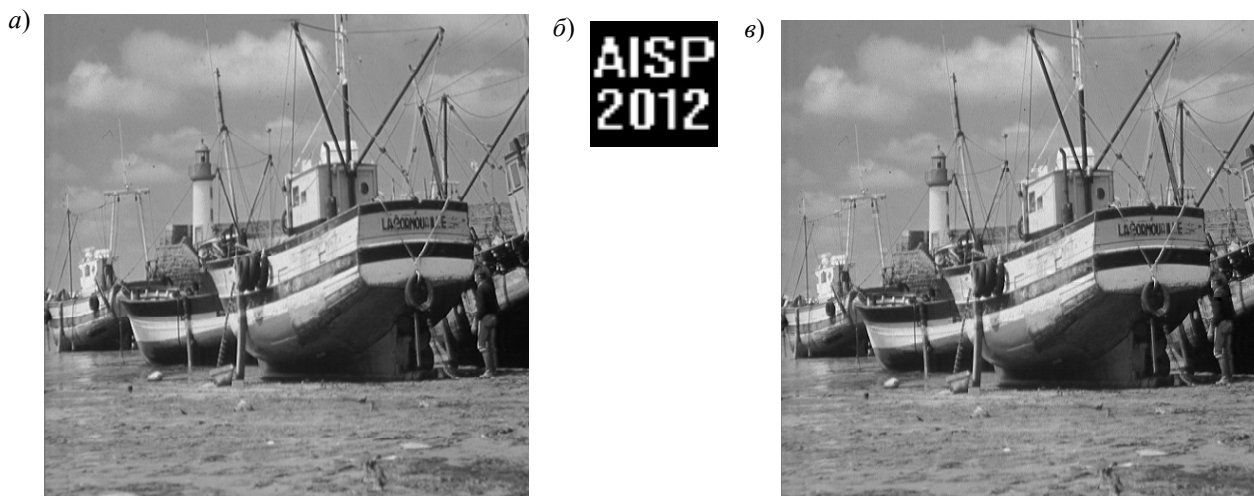


Рис. 1

Для защиты цифровых изображений разработано множество стеганографических алгоритмов, каждый из которых обладает разной устойчивостью к атакам различного типа. Для неподвижных изображений наиболее распространенное искажение — сжатие, примерами которого являются форматы сжатия с потерями JPEG и JPEG2000. В работе рассмотрены алгоритмы цифрового маркирования изображения для встраивания ЦВЗ в контейнер, что обеспечивается разложением контейнера на ряд частотных коэффициентов преобразования (как правило, среднечастотных или низкочастотных) с последующей их модификацией.

Существует несколько алгоритмов встраивания ЦВЗ с контейнером, приведем один из наиболее распространенных [2]:

$$c'_i = c_i + \alpha w_i, \quad (1)$$

где c_i — частотный коэффициент, подлежащий изменению; c'_i — измененный коэффициент, w_i — встраиваемый элемент водяного знака; α — коэффициент силы встраивания (весовой коэффициент).

Извлечение ЦВЗ осуществляется в соответствии с выражением:

$$w_i = (c'_i - c_i) / \alpha. \quad (2)$$

При большом значении α повышается устойчивость водяного знака, однако может сильно снизиться качество защищаемого изображения, а слишком маленькое значение α делает водяной знак крайне уязвимым к различным атакам, например, компрессии (сжатию) или зашумлению.

В ряде частотных алгоритмов применяются различные спектральные преобразования изображения, в том числе дискретное косинусное (ДКП) и дискретное вейвлет-преобразование (ДВП), что обусловлено их использованием в форматах JPEG и JPEG2000 соответственно. Однако алгоритмы цифрового маркирования, основанные на ДКП, могут быть неустойчивы к сжатию JPEG2000, то же относится к ДВП при сжатии по алгоритму JPEG [1]. В связи с этим актуальной является задача разработки алгоритмов цифрового маркирования, обладающих устойчивостью вне зависимости от метода сжатия. В исследованиях отмечена высокая эффективность преобразования Адамара в решении данной задачи [3].

В настоящей статье рассматриваются особенности алгоритмов маркирования, основанных на использовании двумерного преобразования Адамара.

Преобразование Адамара относится к классу ортогональных преобразований в диадных базисах [4], оно обладает малой вычислительной сложностью по сравнению с ДКП и ДВП.

Ядром этого преобразования является матрица Адамара, элементы которой принимают значения 1 и -1 , она описывает преобразование, связанное с разложением сигнала по семейству прямоугольных базисных функций [5].

Матрицы Адамара порядка $N=2^n$ (n — целое положительное число) формируются на основе операции кронекеровского умножения:

$$\mathbf{A}_{2N} = \mathbf{A}_N \otimes \mathbf{A}_2 = \begin{bmatrix} \mathbf{A}_N & \mathbf{A}_N \\ \mathbf{A}_N & -\mathbf{A}_N \end{bmatrix}, \quad (3)$$

где $\mathbf{A}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ — матрица наименьшего порядка.

Наибольшее значение принимает низкочастотный ДС-коэффициент, остальные коэффициенты — высокочастотные (АС). При этом полагается, что частота тем выше, чем чаще изменяется знак в строке. При этом АС-коэффициенты расположены в случайном порядке, что повышает надежность встраивания ЦВЗ [6].

Двумерное преобразование Адамара вычисляется по формуле:

$$\mathbf{F}_{KL} = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{mn} (-1)^{km+nl}, \quad (4)$$

где m, n — номера пикселей исходного изображения; k, l — коэффициенты преобразования ($k, l = 0, N-1$).

Преобразование Адамара обладает свойством делимости по переменным суммирования, для снижения вычислительных затрат двумерное преобразование реализуется строчно-столбцовым способом:

$$\mathbf{F}_N = \frac{1}{N} \mathbf{A}_N [\mathbf{X}_N \mathbf{A}_N], \quad (5)$$

где \mathbf{X}_N — исходное изображение, \mathbf{F}_N — преобразованное в набор коэффициентов изображение, N — размер изображения. Тем самым объем вычислений сокращается с N^4 до $2N^3$ базовых операций (действий под знаками двойной суммы).

Дальнейшего уменьшения вычислительных затрат можно достичь при использовании алгоритма быстрого преобразования Адамара, позволяющего сократить число базовых операций с N^2 до $(N \log_2 N)/2$ при выполнении каждого одномерного преобразования.

Алгоритмы цифрового маркирования. В последнее время разработан ряд алгоритмов [3, 6—10], основой которых являются разновидности преобразования Адамара. Некоторые из алгоритмов предназначены для стеганосистем закрытого типа, не требующих наличия исходного изображения-контейнера для извлечения ЦВЗ [6, 8]. Для повышения устойчивости к сжатию в некоторых алгоритмах используется адаптивная модуляция [7], а для определения местоположения частотных коэффициентов внедряемого ЦВЗ — вычислительный коэффициент энтропии блоков изображения [3, 9, 10].

Алгоритм Elham [10] предназначен для стегосистем закрытого типа, поскольку для извлечения встроенного ЦВЗ необходимо наличие исходного изображения. Алгоритм основан на использовании быстрого преобразования Адамара и вычислении коэффициентов энтропии блоков контейнера.

Для уменьшения объема встраиваемой информации ЦВЗ разбивается на блоки размером 8×8 и подвергается ДКП, тем самым производится операция, подобная сжатию JPEG. Полученные блоки при помощи зигзаг-преобразования преобразуются в векторы, которые затем объединяются.

Контейнер также разбивается на блоки размером 8×8 . Для каждого блока находится среднее значение коэффициентов энтропии. Блоки, среднее значение энтропии которых выше заданного, подвергаются преобразованию Адамара. В каждый преобразованный блок, с использованием коэффициента силы встраивания, по формуле (1) встраивается один коэффициент из вектора, полученного после преобразования Адамара. Операцию встраивания завершает обратное преобразование Адамара с последующим объединением модифицированных и немодифицированных блоков.

Для извлечения ЦВЗ требуется исходное изображение. Путем сегментации исходного изображения определяются координаты модифицированных блоков. Защищенное изображение также подвергается разбиению на блоки размером 8×8 . Согласно формуле (2) осуществляется извлечение каждого коэффициента ЦВЗ. Полученный вектор разбивается на фрагменты

заданной длины. За счет обратного зигзаг-преобразования формируются исходные блоки коэффициентов ЦВЗ, к каждому из которых применяется обратное ДКП. Полученные блоки формируют изображение встроенного ЦВЗ.

В работе [10] данный алгоритм был проверен на ряд атак, таких как среднечастотная фильтрация, зашумление, JPEG-сжатие, изменение размера и яркости. В настоящей работе исследуется его устойчивость к:

- JPEG-сжатию с высокими коэффициентами сжатия;
- сжатию JPEG2000;
- атаке изменения размера изображения.

Исходные параметры алгоритма следующие: размер контейнера 512×512; размер ЦВЗ 64×64; порог энтропии 4,5; сила встраивания 35; контейнер и ЦВЗ разбиваются на блоки размером 8×8; число модифицированных коэффициентов каждого блока ЦВЗ 15.

Для оценки качества защищенного изображения используем пиковое отношение сигнала к шуму (PSNR):

$$\text{PSNR} = 10 \log_{10} \frac{255^2 MN}{\sum_{m,n} (f(m,n) - \hat{f}(m,n))^2}, \quad (6)$$

где $f(m,n)$ и $\hat{f}(m,n)$ — исходное и защищенное изображения; m, n — номера пикселей; M и N — высота и ширина изображения.

В качестве меры качества извлеченного ЦВЗ используем коэффициент корреляции Пирсона:

$$K = \frac{\sum_m \sum_n (A(m,n) - \bar{A})(B(m,n) - \bar{B})}{\sqrt{\sum_m \sum_n ((A(m,n) - \bar{A})^2) \left(\sum_m \sum_n (B(m,n) - \bar{B})^2 \right)}}, \quad (7)$$

где $A(m,n)$, $B(m,n)$ — исходный и извлеченный ЦВЗ; \bar{A} , \bar{B} — среднее арифметическое пикселей исходного и извлеченного ЦВЗ.

Качество маркированного изображения считается удовлетворительным, если PSNR не менее 40 дБ, извлеченный ЦВЗ устойчив к определенной атаке, если коэффициент корреляции Пирсона не ниже 0,5.

В таблице представлены коэффициенты сжатия JPEG и JPEG2000 и соответствующие им коэффициенты корреляции. Устойчивость к изменению размера иллюстрирует рис. 2.

Коэффициент сжатия JPEG, %	Коэффициент корреляции	Коэффициент сжатия JPEG2000, %	Коэффициент корреляции
73,6532	0,9918	4,8175	0,9932
83,3284	0,9874	83,9365	0,9741
87,4279	0,9812	92,1363	0,9028
89,4857	0,9706	94,6910	0,7446
91,9054	0,9478	96,0526	0,6390
94,8762	0,8338	96,8634	0,5132
97,5797	0,3260	97,3967	0,3506

На рис. 2 приведен стеганоконтейнер, усеченный до размеров 256×256 (а) и 356×356 (б). Соответствующие извлеченные ЦВЗ представлены на рис. 2, в (коэффициент корреляции Пирсона 0,1853) и г (0,3295).

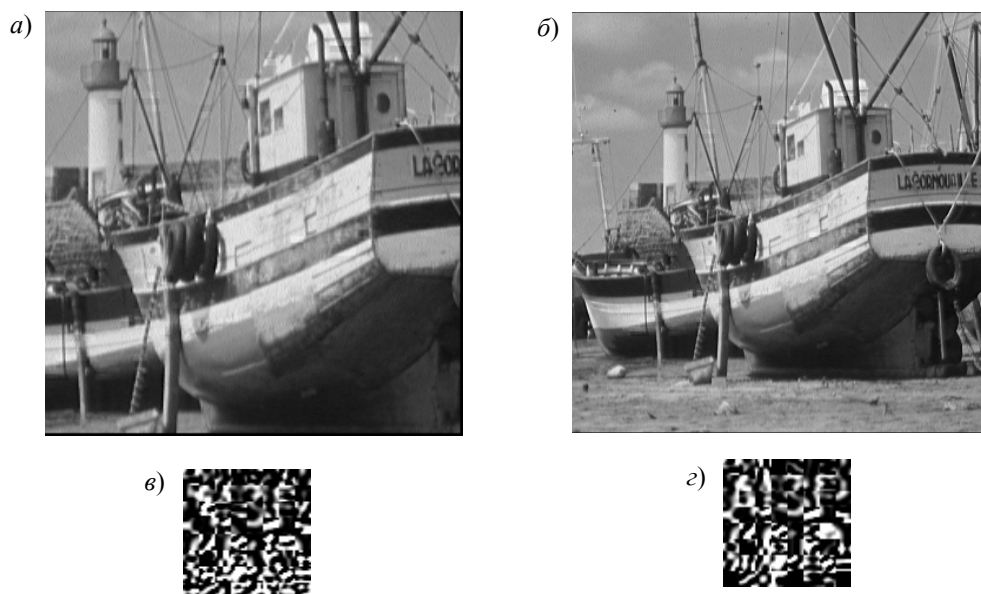


Рис. 2

На основании полученных данных можно сделать вывод о том, что алгоритм Elham устойчив к JPEG-сжатию. Однако его устойчивость к сжатию JPEG2000 не столь высока. Кроме того, алгоритм является уязвимым к атаке изменения размера изображения.

Заключение. В работе установлено, что использование преобразования Адамара позволяет повысить устойчивость ЦВЗ к JPEG-сжатию и обеспечивает удовлетворительную устойчивость к сжатию JPEG2000.

СПИСОК ЛИТЕРАТУРЫ

1. Грибунин В. Г., Оков И. Н., Туринцев В. И. Цифровая стеганография. М.: СОЛОН-Пресс, 2002. 272 с.
2. Cox I., Kilian J., Leighton T., Shamoon T. Secure spread spectrum watermarking for multimedia // IEEE Transact. on Image Processing. 1997. Vol. 6, N 12, P. 1673—1687.
3. Maity S. P., Kundu M. K. Perceptually adaptive spread transform image watermarking scheme using Hadamard transform // Information Sciences. 2011. Vol. 181. P. 450—465.
4. Тропченко А. Ю., Тропченко А. А. Цифровая обработка сигналов. Методы предварительной обработки: Учеб. пособие по дисциплине „Теоретическая информатика“. СПб: СПбГУ ИТМО, 2009. 100 с.
5. Прэнтт У. Цифровая обработка изображений. М.: Мир, 1982. Кн. 1. 312 с.
6. Ho A. T. S., Shen J., Chow A. K. K., Woon J. Robust digital image-image watermarking algorithm using fast Hadamard transform // Intern. Symp. on Circuits and Systems. 2003. Vol. 3. P. 826—829.
7. Maity S. P., Kundu M. K. DHT domain digital watermarking with low loss in image informations // Int. J. Electron. Commun. 2010. Vol. 64. P. 243—257.
8. Saryazdi S., Nezamabadi-pour H. A Blind Digital Watermark in Hadamard Domain // World Academy of Science, Engineering and Technology. 2007. Vol. 3.
9. Rajkumar F. // J. of Computer Applications. 2011. Vol. 12, N 9.
10. Fami E. Sh., Samavi Sh., Kaviani H. R., Radani Z. M. Adaptive Watermarking in Hadamard Transform Coefficients of Textured Image Blocks // 16th Intern. Symp. on Artificial Intelligence and Signal Processing (AISP). 2012.

- Владимир Александрович Батура* — *Сведения об авторах*
— аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: batu-vladimir@yandex.ru
- Александр Ювенальевич Тropicенко* — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: tau@d1.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.056.53

К. А. ЩЕГЛОВ, А. Ю. ЩЕГЛОВ

НЕПРОТИВОРЕЧИВАЯ МОДЕЛЬ МАНДАТНОГО КОНТРОЛЯ ДОСТУПА

Исследована модель мандатного контроля доступа, выявлены противоречия и недостатки, не позволяющие реализовать на ее основе безопасную систему. Разработаны и обоснованы модель целостности и доступности и „непротиворечивая модель мандатного контроля доступа“, применение которой позволяет предотвращать нарушение конфиденциальности информации, а также решать задачи обеспечения ее целостности и доступности в комплексе.

Ключевые слова: защита информации, контроль доступа, правила доступа, мандат, категории информации, уровни конфиденциальности.

Введение. Сегодня широкое практическое применение нашла модель мандатного контроля доступа, основанная на использовании меток безопасности (или мандатов) [1]. Эта предложенная Белла—ЛаПадулой [2] модель, согласно нормативному документу в области информационной безопасности [1], предназначена для использования в наиболее критичных приложениях с целью защиты обрабатываемой информации от нарушения конфиденциальности. Однако существует и альтернативная модель мандатного контроля доступа, предложенная Бибом [3], предназначенная для обеспечения целостности и доступности информации. Правила контроля доступа, определяемые этими моделями, полностью исключают друг друга. Вместе с тем задачи защиты конфиденциальности информации и обеспечения ее целостности и доступности системы должны решаться комплексно.

Альтернативные модели мандатного контроля доступа. Основу мандатного контроля доступа составляет возможность ранжирования (присвоения категории) обрабатываемой информации на основании какого-либо признака.

Практика секретного делопроизводства в компьютерной обработке информации, согласно модели Белла—ЛаПадулы [2], предполагает классификацию информации по уровням конфиденциальности.

Метки безопасности *объектов* отражают категорию конфиденциальности информации, которые могут быть сохранены в соответствующих объектах. Метки безопасности *субъектов* отображают полномочия или уровень допуска (по аналогии с формой допуска в секретном делопроизводстве) субъектов к информации различных уровней конфиденциальности.

Будем считать, что чем больше полномочия субъекта S и выше уровень конфиденциальности объекта O , тем меньше их порядковый номер в упорядоченных множествах:

$C = \{C_1, \dots, C_l\}$ и $O = \{O_1, \dots, O_l\}$ и тем меньшее значение метки безопасности M_i ($i = 1, \dots, l$) им присваивается, т.е. $M_1 < M_2 < M_3 < \dots < M_l$.

Таким образом, в качестве учетной информации каждому субъекту и объекту задаются метки безопасности из множества M . В общем случае метка присваивается группе равноправных (имеющих одинаковые полномочия) субъектов и группе объектов одного уровня конфиденциальности.

Обозначим метку безопасности субъекта (группы субъектов) и объекта (группы объектов) доступа как M_c и M_o .

Модель Белла—ЛаПадулы обеспечивает реализацию следующих правил, направленных на предотвращение понижения категории информации в процессе ее обработки:

1) субъект C имеет доступ к объекту O в режиме чтения, если выполняется условие: $M_c \leq M_o$;

2) субъект C имеет доступ к объекту O в режиме записи, если выполняется условие: $M_c = M_o$.

Иногда также рассматривается возможность записи и при условии: $M_c > M_o$.

Чем ниже уровень конфиденциальности информации, тем более широкие возможности по ее обработке предоставляются, например, могут использоваться неконтролируемые внешние накопители, принтеры или сетевые ресурсы. Таким образом, несанкционированное понижение категории обрабатываемой информации напрямую связано с изменением режима ее обработки и, как следствие, с возникновением „каналов“ ее хищения.

Согласно „модели целостности Биба“ [3], в систему включается иерархический признак „целостность“, отображаемый мандатом, или меткой безопасности. Вводятся уровни целостности, сопоставляемые с субъектами и объектами доступа, последним присваиваются метки безопасности. Соответствующим образом изменяются и правила контроля доступа:

1) субъект C имеет доступ к объекту O в режиме чтения, если выполняется условие: $M_c \geq M_o$;

2) субъект C имеет доступ к объекту O в режиме записи, если выполняется условие: $M_c \leq M_o$.

Практическое использование этой модели, инверсии модели Белла—ЛаПадулы, остается под большим вопросом. В частности, модель Биба критикуется специалистами за то, что она использует целостность как некую меру, в то время как целостность субъектов и объектов следует рассматривать как двоичный атрибут, который или есть, или нет. Кроме того, не понятен принцип классификации субъектов и объектов по параметру „целостность“.

Предлагаемая модель целостности и доступности. Непротиворечивая модель мандатного контроля доступа. Существует множество угроз, связанных с атаками со стороны приложений, наделяемых соответствующими функциями в результате прочтения ими вредоносного файла (не являющегося исполняемым), записанного на компьютер в процессе работы пользователя. К подобным приложениям, например, относятся офисные, которые могут приобрести вредоносные функции в результате прочтения системой документа, надленного макровирусом, всевозможные командные интерпретаторы, наделяемые дополнительным функционалом, в результате прочтения ими „активного“ содержимого, в частности, скриптов и ActiveX-компонентов [4] и др. Приложение, надленное вредоносными функциями, получает право записи (а также модификации и удаления) во все объекты, доступ к которым разрешен пользователю.

Построим модель мандатного контроля доступа. Будем считать, что вероятность записи (сохранения в процессе работы) вредоносного файла субъектом C_i ($i=1, \dots, l$) составляет $P_i(w)$ ($i=1, \dots, l$). Также будем считать, что чем меньше значение $P_i(w)$ для субъектов, соответственно объектов, в который разрешена запись субъекту, тем меньше их порядковый номер в упорядоченных множествах субъектов и объектов — $C = \{C_1, \dots, C_l\}$ и $O = \{O_1, \dots, O_l\}$, и тем

меньшее значение метки безопасности M_i ($i = 1, \dots, l$) им присваивается, т.е.: $M_1 < M_2 < M_3 < \dots < M_l$, при условии, что $P_1(w) \ll P_2(w) \ll \dots \ll P_l(w)$.

Принцип мандатного контроля доступа — информация различных категорий обрабатывается в различных режимах, с использованием разных устройств и объектов, что полностью соответствует заданному условию $P_1(w) \ll P_2(w) \ll \dots \ll P_l(w)$.

Между отношениями для меток безопасности M_i ($i = 1, \dots, l$): $M_1 < M_2 < M_3 < \dots < M_l$, назначаемых при обработке ранжированной информации, и для вероятностей записи вредоносного файла $P_i(w)$: $P_1(w) \ll P_2(w) \ll \dots \ll P_l(w)$, существует прямая связь.

Построим модель мандатного контроля доступа, позволяющую обеспечить целостность и доступность обрабатываемой информации. Эта модель регламентирует следующие правила доступа:

1) субъект C имеет доступ к объекту O в режиме чтения, если выполняется условие: $M_c \geq M_o$;

2) субъект C имеет доступ к объекту O в режиме записи, если выполняется условие: $M_c \leq M_o$.

Как видим, правила доступа для предложенной нами модели и модели Биба идентичны, однако различие этих моделей принципиально. Модель целостности Биба предполагает включение в систему иерархического признака „целостность“, отображаемого меткой безопасности, в то время как в представленной, так же как и в модели Белла—ЛаПадулы, вводится классификация (ранжирование) информации по уровням конфиденциальности — метки отражают категорию конфиденциальности информации.

Как следствие, именно эти модели можно рассматривать в качестве альтернативных решений, поскольку они основаны на использовании одного и того же ранжирующего признака, при этом предложенная модель является полной инверсией модели Белла—ЛаПадулы.

Исходя из сказанного можно сделать важнейший вывод о том, что безопасная обработка ранжированной по уровням конфиденциальности информации в общем случае достигается при реализации следующих непротиворечивых правил доступа:

1) субъект C имеет доступ к объекту O в режиме чтения и записи, если выполняется условие: $M_c = M_o$;

2) субъект C не имеет доступа к объекту O , если выполняется условие: $M_c \geq M_o$.

Именно эта модель, на наш взгляд, имеет все основания быть позиционирована как непротиворечивая модель мандатного контроля доступа, определяющая непротиворечивые правила доступа, поскольку ее применение позволяет решать задачи обеспечения конфиденциальности, целостности информации и доступности в комплексе, т.е. позволяет построить безопасную систему.

Заключение. В работе исследованы основные модели мандатного контроля доступа, в результате чего разработаны модель целостности и доступности и „непротиворечивая модель мандатного контроля доступа“, определяющая корректные правила доступа при обработке ранжированной по уровням конфиденциальности информации. Применение непротиворечивой модели позволяет решать задачи защиты конфиденциальности информации и обеспечения ее целостности и доступности в комплексе, что дает возможность построить безопасную систему.

Полученный результат имеет большое практическое значение, поскольку исследования продемонстрировали принципиальные недостатки применяемой модели контроля доступа, причем применяемой в наиболее критичных приложениях, что регламентируется соответствующим нормативным документом в области информационной безопасности [1].

СПИСОК ЛИТЕРАТУРЫ

1. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации. М., 1992.
2. Bell D. E., LaPadula L. J. Security Computer Systems: Unified Exposition and MULTICS Interpretation, Revision 1, US Air Force ESD-TR-306, MITRE Corporation MTR-2997. Bedford MA, March 1976.
3. Biba K. J Integrity Consideration for Security Computer System. The MITRE Corp., Report MTR N3153 Revision 1, Electronic System Division, U.S. Air Force Systems Command, Technical Report ESD TR 76 372. Belford, Massachusetts, April 1977.
4. Щеглов К. А., Щеглов А. Ю. Защита от атак со стороны приложений, наделяемых вредоносными функциями. Модели контроля доступа // Вопросы защиты информации. 2012. Вып. 4 (99). С. 31—36.

Сведения об авторах

- Константин Андреевич Щеглов** — студент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: schegl_70@mail.ru
- Андрей Юрьевич Щеглов** — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: info@npp-itb.spb.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.056

Т. А. МАРКИНА, А. Ю. ЩЕГЛОВ

МЕТОД ЗАЩИТЫ ОТ АТАК ТИПА DRIVE-BY-ЗАГРУЗКА

Рассмотрены атаки типа drive-by-загрузка, использующие скриптовые вредоносные программы. Предложен метод защиты от таких атак, заключающийся в запрете загрузки (установки) или запуска несанкционированных скриптов (программ), запрете запуска несанкционированных скриптов под видом санкционированных и запрете модификации санкционированных скриптов. Продемонстрирована реализация этого метода.

Ключевые слова: вредоносные программы, скрипт, защита информации, drive-by-атака, антивирусная программа.

Введение. Одним из наиболее распространенных методов заражения компьютеров вредоносными программами, по данным компании „Лаборатория Касперского“, являются drive-by-атаки [1]. Практически весь TOP 20 [1] детектируемых web-антивирусом объектов состоит из скриптовых вредоносных программ, которые принимают участие в таких атаках. Стоит отметить, что атаки типа drive-by используют только скриптовые вредоносные программы.

Общее количество сайтов, использующихся в данных атаках, к концу июня 2013 г., по данным лаборатории McAfee, превысило 74,7 млн, что соответствует 29 млн доменных имен. Наиболее критично то, что 96% вредоносных доменов используются для атак drive-by [2].

Описание атаки drive-by. Рассмотрим технологию атаки (рис. 1). Первое время злоумышленники, применявшие загрузки drive-by, создавали вредоносные сайты и, чтобы привлечь на них посетителей, использовали социальную инженерию. Такие web-страницы до сих пор остаются основным источником вредоносной сетевой активности. Однако в последнее время хакеры заражают вполне „законопослушные“ сайты, размещая на них скриптовые

эксплойты или код для переадресации запросов, что позволяет незаметно для пользователя запускать атаки через браузер. В ходе такой атаки вредоносная программа автоматически загружается на компьютер-жертву без уведомления. Используются три типа вредоносных программ: редиректоры, скриптовые загрузчики и эксплойты.

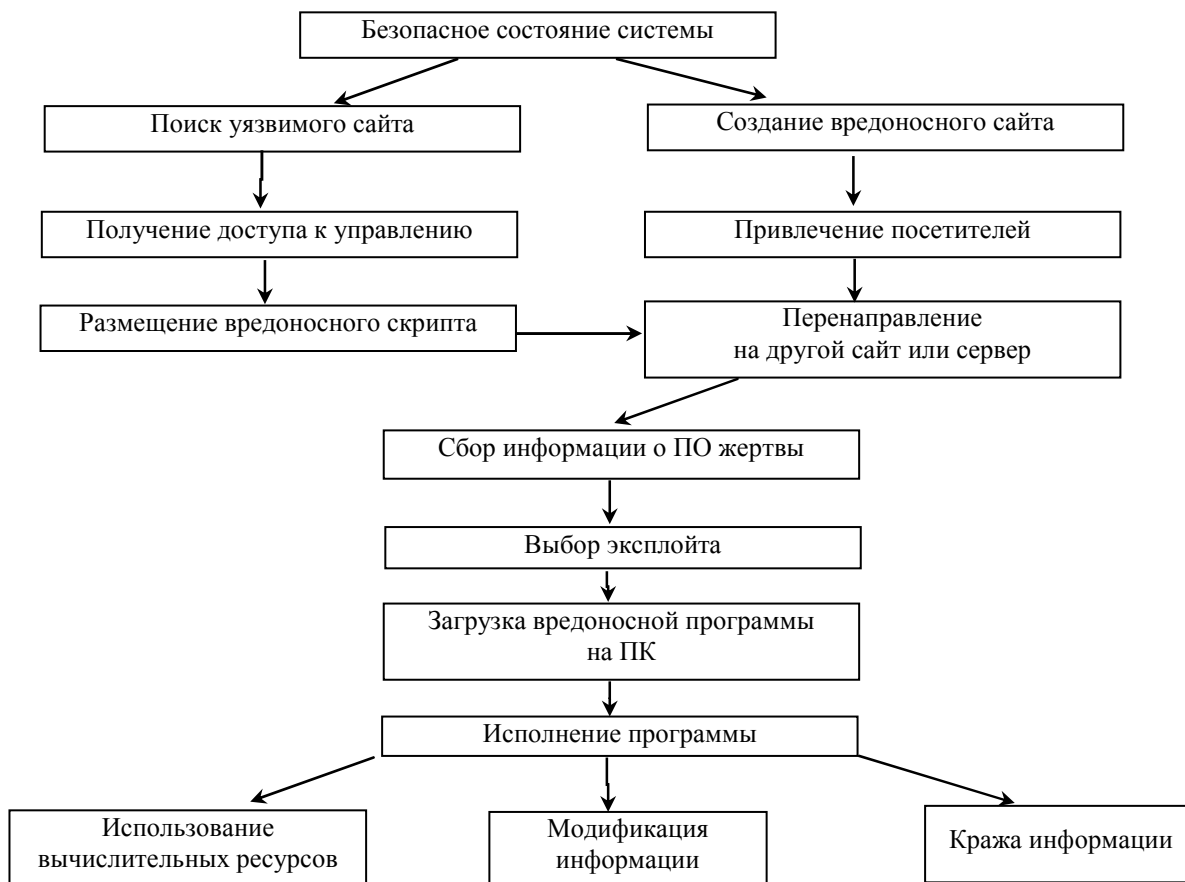


Рис. 1

Эксплойты, используемые при атаках drive-by, могут быть нацелены на незащищенные встраиваемые модули (плагины) web-браузера, уязвимости элементов управления ActiveX или средств защиты стороннего ПО [3]. Эксплойт выбирается на основании собранной информации о программах на компьютере жертвы (версия ОС, браузер и др.).

Сначала с зараженного сайта, в основном при помощи тэга `<iframe>`, происходит перенаправление пользователя на страницу, содержащую каскадные таблицы стилей (CSS) и вредоносный скриптовый загрузчик. Такой метод атаки затрудняет обнаружение вредоносных скриптов многими антивирусами и позволяет злоумышленникам избежать обнаружения загружаемых эксплойтов. Из рис. 1 видно, что прежде чем эксплойт будет загружен, может произойти любое количество переадресаций на другие сайты. В TOP 20 от компании „Лаборатория Касперского“ во втором квартале 2013 г. попали четыре редиректора: Trojan.Script.Iframer, Trojan.JS.FBook.q, Trojan.JS.Iframe.aeq и Trojan.JS.Redirector.xb.

Последние два при помощи тэга `<iframe>` перенаправляют пользователя на web-страницы злоумышленников. Первые два, помимо того что ими заражают легальные javascript-файлы, вызывают загрузку дополнительного скрипта JavaScript, это происходит, только если курсор движется в пределах рабочего окна (т.е. событие „mousemove“ объекта „window“). Подобная техника применяется для обхода некоторых „песочниц“ и эмуляторов.

Далее скриптовые загрузчики Trojan-Downloader.Script.Generic, Trojan-Downloader.JS.JScript.cb, Trojan-Downloader.Win32.Generic, Trojan-Downloader.HTML.Iframe.ahs с зараженных web-страниц запускают эксплойт.

Загрузчики для хранения основных данных скрипта используют html-тэги. Скрипты первой группы — поле „alt“ тэга , а второй — тэг <div>. Для исполнения Java-эксплоитов загрузчики используют уязвимости в браузерах или pdf-ридерах.

Средства защиты. Защититься от атаки типа drive-by можно, используя различные антивирусные программы, имеющие модуль, который позволяет обнаруживать, что сайт заражен; утилиту Rozzle; сервис для проверки сайта на наличие заражения.

Первый способ защиты основан на сигнатурном поиске на web-сайтах и на поведенческом анализе.

Программа Rozzle является совместной разработкой компании Microsoft и Венского технического университета, представляющей собой виртуальную машину JavaScript, которая имитирует различные установки, предоставляя вредоносной программе различные пути для атаки. Иными словами, программа выдает себя за уязвимый браузер и эксплоит, с помощью которого происходит загрузка вредоносной программы, „попадающей в ловушку“. Средства поведенческого анализа смогли определить только 2,5 % угроз такого типа, в то время как Rozzle — 17,5 % [4].

Сервис проверки подозрительных сайтов действует следующим образом: следует ввести адрес проверяемого сайта в специальное поле и нажать кнопку „Проверить“.

Этот способ, как и первый, основывается на использовании существующих баз сигнатур, скорость составления которых значительно отстает от требуемой. Рассмотренные подходы неэффективны. Даже если считать, что пропускается всего 1 % (хотя на практике пропускается как минимум 80 % [4]), это может составить примерно 138 зараженных сайтов в день.

Предлагаемый метод защиты от атак типа drive-by-загрузки основан на разграничительной политике, но права доступа задаются не для каждого объекта доступа. В отличие от классической модели ОС семейства Windows, предлагается задавать права для пары „субъект доступа—объект доступа“. Субъект доступа включает в себя всех пользователей системы и все процессы. Главная особенность предлагаемого метода заключается в том, что объект доступа задается при помощи масок и исходя из типов файлов — для этого в ОС семейства Windows следует использовать расширения файлов (например, *.js). Для пары „субъект—объект“ задается запрет исполнения, записи, модификации, в том числе путем переименования существующих легальных скриптов, и переименования в разрешенные к запуску объекты, включая изменение расширений. В результате разрешается запуск только санкционированно установленных скриптов и предотвращается любая попытка создания новых скриптов на компьютере.

Реализация метода. Метод реализован на основе механизмов комплексного средства защиты информации „Панцирь+“ для ОС Microsoft Windows. При помощи масок назначаются объекты файловой системы, которые считаются исполняемыми, запрещаются их модификация и создание новых подобных объектов, в том числе переименованием. При помощи масок назначаются объекты файловой системы с расширениями *.js, *.vbs, *.php и др., которые являются файлами, написанными на скриптовых языках программирования. Запрещается их изменение и создание новых подобных объектов, в том числе путем переименования.

Для практической реализации метода был использован механизм контроля доступа к файлам. Был создан субъект доступа „Любой“ (рис. 2), учитывающий все процессы в данной системе. Затем назначались объекты файловой системы, являющиеся скриптовыми исполняемыми файлами. Наиболее распространенными на данный момент являются следующие файлы-скрипты с указанными расширениями: *.vbs и *.vbe (VBScript) задается как „*.vb*“; *.js и *.jse (JavaScript) задается как „*.js *“; *.wsf; *.wsh задается как „*.ws*“; *.scpt (AppleScript); *.php и *.asp, *.cgi (рис. 3).

После создания всех нужных объектов доступа устанавливаются разграничения к ним. Добавляя новое правило для выбранного объекта доступа, следует запретить все режимы доступа к нему. При этом следует фиксировать любой отказ в доступе (рис. 4).

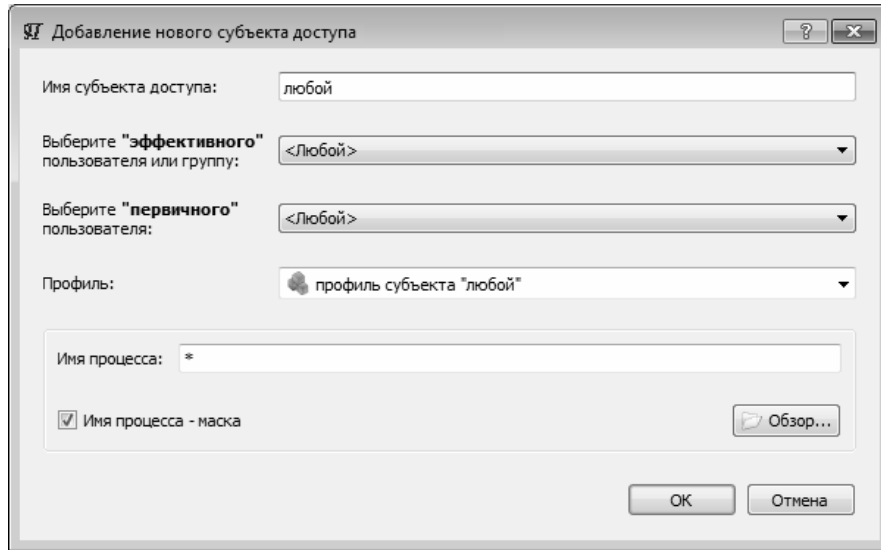


Рис. 2

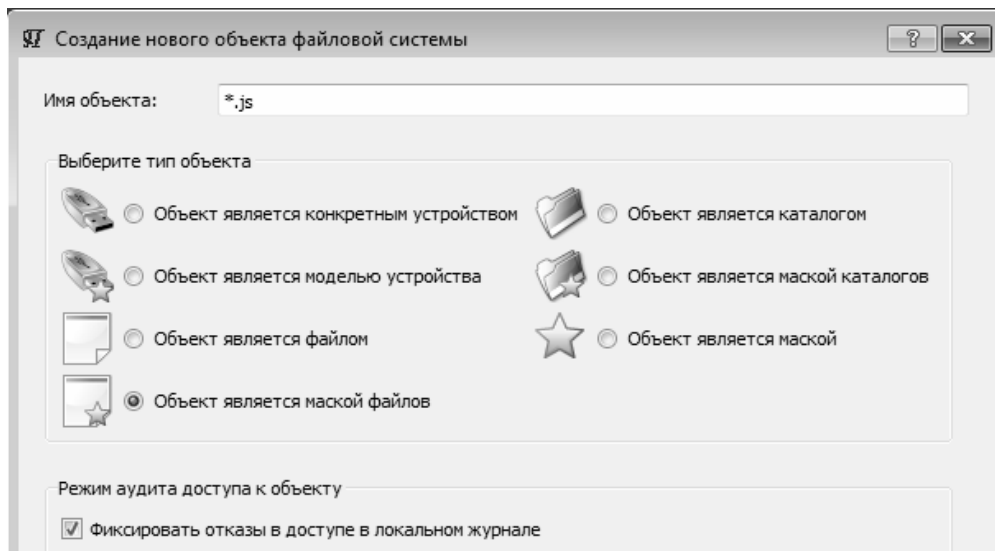


Рис. 3

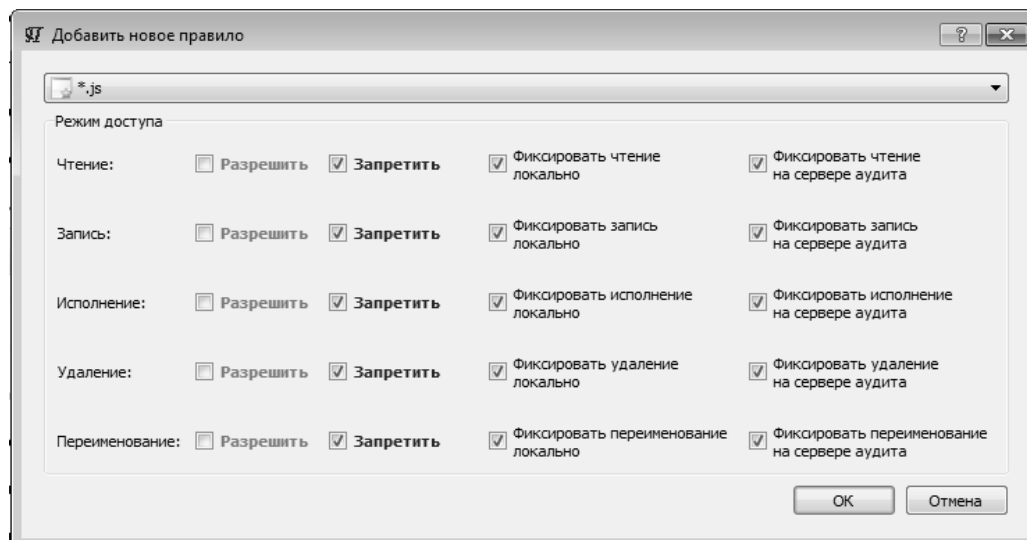


Рис. 4

Однако программы, которые присутствуют в системе и санкционированно исполняются, уязвимы. По-прежнему доступ для модификации, переименования и удаления остается открытым. Это, в свою очередь, является серьезной угрозой безопасности системы, так как критичные для пользователя программы могут быть удалены, испорчены или даже при помощи эксплойтов модифицированы таким образом, что станут выполнять вредоносные функции. Таким образом, следует обеспечить замкнутость среды. Для этого назначаются объекты файловой системы, необходимые для корректной работы операционной системы и требуемых программ, они находятся в системных каталогах %ProgramFiles% и %windir% (%systemroot%) и имеют расширения: *.exe, *.bat, *.com, *.config, *.dll, *.manifest, *.drv, *.fon, *.ttf, *.log, *.sys. После добавления объектов доступа создается правило для них, разрешающее чтение и исполнение, но запрещающее запись, удаление и переименование. При выполнении всех этих требований обеспечивается замкнутая среда исполнения. Теперь в системе возможен санкционированный запуск только необходимых программ, а вредоносное программное обеспечение, попавшее на компьютер тем или иным способом, не может быть запущено [5].

Для проверки эффективности предлагаемого метода следует посетить потенциально зараженные сайты. Предварительно необходимо настроить механизм аудита для всех созданных объектов на фиксацию всех запросов на запись, исполнение, удаление, переименование. После посещения зараженных страниц или страниц с дополнительной рекламой в журнале управления доступом к файловой системе появятся сообщения о запрете записи и исполнения скриптовых вредоносных файлов.

Как видим, метод позволяет полностью защититься от несанкционированной записи и исполнения скриптовых вредоносных программ.

Заключение. Исследования показали, что атаки типа drive-by приводят к внедрению вредоносной программы на компьютер-жертву. Соответственно и защищаться нужно от загрузки файлов. Предлагаемый метод позволяет полностью защититься от таких атак, поскольку при любом способе установки вредоносного ПО в систему (использовались уязвимость в браузере или ошибка в Adobe Acrobat Reader) данная атака будет предотвращена.

Особенностью предлагаемого метода является то, что не сохраняются и не запускаются легальные скрипты с web-сайтов. Большинство из этих скриптов, как правило, предназначено для автозаполнения форм, рекламного баннера, загрузки дополнительной страницы с рекламой. Такие ограничения позволяют дополнительно защитить пользователя от получения ненужной рекламы.

СПИСОК ЛИТЕРАТУРЫ

1. Масленников Д., Функ К. Развитие информационных угроз во втором квартале 2013 года [Электронный ресурс]: <https://www.securelist.com/ru/analysis/208050807/Razvitie_informatsionnykh_ugroz_vo_vtorom_kvartale_2013_goda#20>.
2. McAfee Threats Report: Second Quarter 2013 [Электронный ресурс]: <<http://www.mcafee.com/ru/resources/reports/tp-quarterly-threat-q2-2013.pdf>>.
3. Шубаева Т. А. Скриптовые вредоносные программы: способы внедрения и защита от них // Сб. тр. молодых ученых и сотрудников кафедры ВТ. 2012. Вып. 3. С. 78—80.
4. Kolbitsch C., Livshits B., Zorn B., Seifert Ch. Rozzle: De-Cloaking Internet Malware Microsoft Research [Электронный ресурс]: <<http://research.microsoft.com/pubs/152601/rozzle-tr-10-25-2011.pdf>>.
5. Шубаева Т. А., Щеглов А. Ю., Оголюк А. А. Защита от внедрения и запуска вредоносных программ // Вопросы защиты информации. 2011. Вып. 2 (93). С. 26—30.

Сведения об авторах

- Татьяна Анатольевна Маркина** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: tmark812@mail.ru
- Андрей Юрьевич Щеглов** — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: info@npp-itb.spb.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.627

А. А. ТРОПЧЕНКО

МЕТОДЫ ПОВЫШЕНИЯ РОБАСТНОСТИ РАСПОЗНАВАНИЯ В МУЛЬТИМОДАЛЬНЫХ БИОМЕТРИЧЕСКИХ СИСТЕМАХ

Рассмотрены методы повышения робастности (устойчивости) распознавания в мультимодальных биометрических системах идентификации личности.

Ключевые слова: распознавание личности, мультимодальные системы распознавания, объединение признаков, робастность.

Для аутентификации и идентификации человека применяется биометрика — область знаний, использующая индивидуальные биологические особенности. Биометрические данные включают множество признаков (модальностей): отпечаток пальца, изображение лица, речь, геометрия кисти руки и ушной раковины, сетчатка глаза, подпись, динамика нажатия клавиш, походка, физиологические сигналы (электрокардиограмма) и т.д. У использования каждого признака есть свои преимущества и ограничения с точки зрения точности, устойчивости и удобства работы. Например, использование сетчатки обеспечивает высокую точность и устойчивость распознавания, но требует дорогого оборудования и существенных затрат времени.

Идентификация в динамике гораздо более сложная задача, особенно когда велико число зарегистрированных в системе пользователей. Динамические идентификационные системы на основе анализа аудиосигналов достигают высокой производительности, когда высоко отношение сигнал—шум (SNR) распознаваемого отрезка (текста). Однако устойчивость быстро ухудшается, когда SNR набора тестов уменьшается [1]. Для исследования устойчивости (робастности) распознавания личности при различных уровнях искажений был проведен ряд экспериментов.

На рис. 1 продемонстрировано последовательное снижение качества тестового изображения для аудиовизуальной базы данных XM2VTS, уровень вносимых искажений определяет параметр QF (Quality Factor).



QF=50



QF=14



QF=2

Рис. 1

База данных XM2VTS содержит видеоролики, на которых 295 человек произносят тестовое предложение (“Joe took fathers green shoe bench out”) 4 раза с интервалами в месяц. Это предложение считается фонетически сбалансированным для английского языка.

Результаты экспериментов по распознаванию речи при различных уровнях зашумленности сигнала приведены на рис. 2 (P — вероятность правильного распознавания). Аудиосигнал был предварительно обработан для увеличения мощности в области более высоких частот с использованием фильтра $H(z) = 1/(1 - 0,97z^{-1})$. Затем сигнал был разделен на участки (фреймы) длиной 20 мс с перекрытием в 10 мс, что обеспечивает частоту аудиокадров 100 Гц. Далее из каждого фрейма извлекаются мел-частотные кепстральные коэффициенты (MFCC).

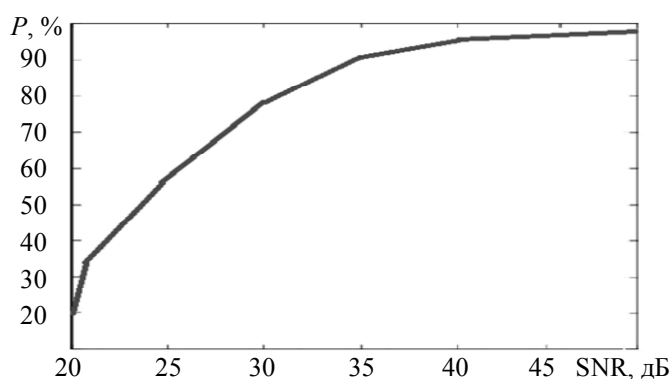


Рис. 2

Вероятность правильного распознавания 97,6 % была достигнута при SNR = 48 дБ. При снижении SNR до 21 дБ вероятность уменьшалась до 37 %. Для исследования робастности метода распознавания визуальных признаков использовались скрытые марковские модели [2]. На рис. 3 приведена зависимость точности распознавания от уровня искажений для лицевой модальности, показавшей значительно более высокую устойчивость к искажениям (при QF = 2 точность составила 48 %).

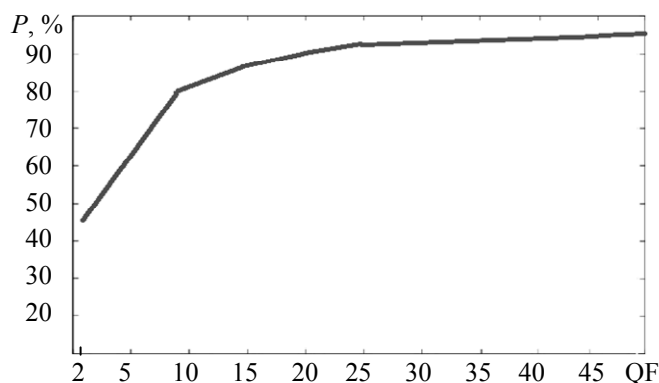


Рис. 3

Для объединения (фузирования) признаков различных модальностей необходимо в каждый момент времени располагать набором признаков от каждой модальности. Такие наборы должны быть выравнены по размеру, для этого использование разных модальностей, например звуковых и визуальных, должно быть синхронизировано с использованием интерполированных кадров до этапа слияния данных, что позволит использовать корреляции между ними [5]. Для оценки корреляции рассматривались задержки между последовательностями аудио- и видео-признаков в диапазоне 0—100 мс (с шагом 10 мс), причем точка с минимальным значением ошибки оказалась в значении 40 мс. Следует отметить, что в той же точке был достигнут максимум функций корреляций между аудио- и видеопризнаками [4].

Для дальнейшего повышения вероятности точного распознавания были проведены эксперименты по объединению модальностей. На рис. 4 приведены зависимости вероятности точного распознавания объединенной аудиовизуальной модальности (1) от уровней искажений SNR для звука (2) и QF для видео (3).

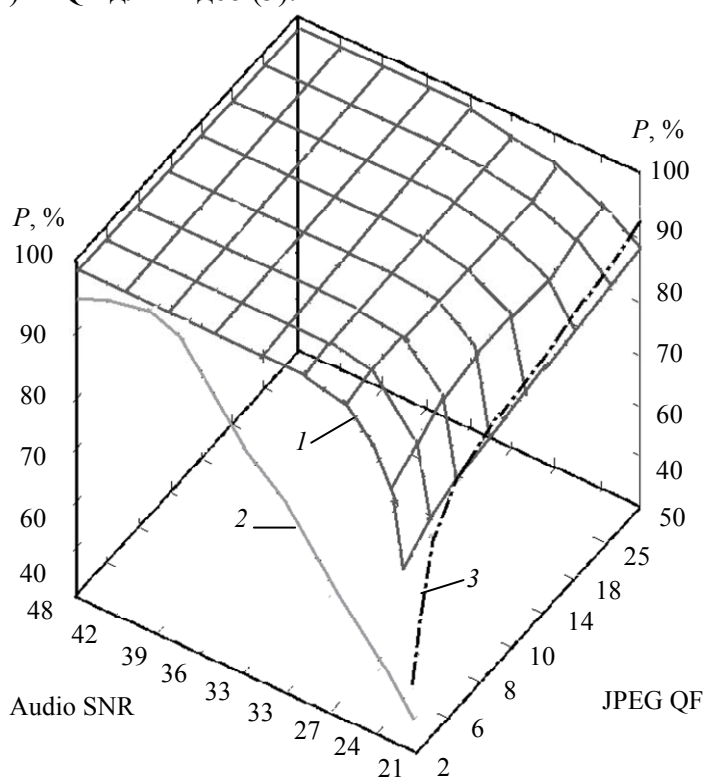


Рис. 4

В то время как использование аудиомодальности при максимальном уровне шумов обеспечило $P = 37\%$, а визуальной — 48% , использование объединенной аудиовизуальной модальности обеспечило $P = 70\%$ (см. таблицу).

Вероятность правильного распознавания аудиовизуальной модальности

QF	$P, \%$, при SNR, дБ									
	48	45	42	39	36	33	30	27	24	21
50	99,2	99,2	99,2	99,2	99,2	99,2	98,4	96,4	93,2	87,3
25	99,2	99,2	99,2	99,2	99,2	99,2	98,4	96,4	93,0	87,1
18	99,2	99,2	99,2	99,2	99,2	99,2	98,4	96,4	91,6	87,0
14	99,2	99,2	99,2	99,2	99,2	99,2	98,4	96,0	91,2	86,9
10	99,2	99,2	99,2	99,2	99,2	99,2	98,0	95,0	90,8	86,0
8	99,2	99,2	99,2	99,2	99,2	99,2	97,6	94,0	90,2	83,2
6	99,0	99,0	99,0	99,0	99,0	99,0	97,2	92,9	86,9	78,1
4	99,0	99,0	99,0	99,0	99,0	98,6	96,4	92,0	82,7	75,9
3	99,0	99,0	99,0	98,8	98,4	97,6	95,6	91,2	81,3	71,2
2	99,0	99,0	99,0	98,2	98,0	97,0	95,2	91,0	80,5	70,5

Результаты проведенных на двух аудиовизуальных базах данных экспериментов показали высокую устойчивость (робастность) мультимодальных биометрических систем идентификации личности [5, 6]. Однако требуется дальнейшее исследование влияния алгоритмов и уровня фузирования различных биометрических модальностей на робастность распознавания [7].

СПИСОК ЛИТЕРАТУРЫ

1. Городецкий В. И., Серебряков С. В. Методы и алгоритмы коллективного распознавания: обзор // Тр. СПИИРАН. 2006. Т. 1, вып. 3. С. 139—171.

2. Синицын И. Н., Новиков С. О., Ушмаев О. С. Развитие технологий интеграции биометрической информации // Системы и средства информатики. 2004. Вып. 14. С. 5—36.
3. Тропченко А. А., Тропченко А. Ю. Нейросетевые методы идентификации человека по изображению лица // Изв. вузов. Приборостроение. 2012. Т. 55, № 10. С. 31—36.
4. Ушмаев О. С. Методы мультибиометрической идентификации. М.: Изд-во ИПИ РАН, 2009. 114 с.
5. Dass S. C., Nandakumar K., Jain A. K. A principled approach to score level fusion in multimodal biometric systems // Audio- and Video-Based Biometric Person Authentication. 2005. P. 1049—1058.
6. Karam W., Bredin H., Greige H., Chollet G., Mokbel C. Talking-face identity verification, audiovisual forgery, and robustness issues // EURASIP J. Adv. Signal Process. 2009. Vol. 4. P. 1—15.
7. Ross A., Govindarajan R. Feature level fusion using hand and face biometrics // Proc. of the SPIE Conf. on Biometric Technology for Human Identification. Orlando, USA, 2005. P. 196—204.

Сведения об авторе

Андрей Александрович Тропченко — канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: zayka_98rus@mail.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.912

И. В. КАЛИНИН, С. В. КЛИМЕНКОВ, А. Е. ХАРИТОНОВА, Е. А. ЦОПА

ПРЕОБРАЗОВАНИЕ ЕСТЕСТВЕННОГО ЯЗЫКА В ФОРМАТ RDF С ПОМОЩЬЮ СЕМАНТИЧЕСКИХ АНАЛИЗАТОРОВ ТЕКСТОВОЙ ИНФОРМАЦИИ

Решена задача автоматического преобразования естественного языка (русского) в формат RDF средствами семантического анализа. Приведен алгоритм работы программных модулей, созданных для решения задачи.

Ключевые слова: текст, естественный язык, RDF, семантический анализ, тезаурус, AOT, Jena.

Введение. Консорциумом Всемирной паутины для машиночитаемого представления данных, в особенности — метаданных, была разработана модель RDF (Resource Description Framework) [1]. Одним из направлений развития сети Интернет является реализация механизмов машинной обработки информации [2].

В основе этих механизмов лежит работа с метаданными, однозначно идентифицирующими характеристики и содержание ресурсов Интернета. Обработка метаданных должна прийти на смену используемому в настоящий момент текстовому анализу документов [3].

Формирование RDF-описаний ресурсов обычно осуществляется вручную — авторами. Во многих случаях такой подход неэффективен: в частности, при необходимости формирования большого объема метаданных или при формировании содержимого ресурсов пользователями (интернет-энциклопедии, социальные сети). Таким образом, возникает потребность в автоматизации процесса формирования RDF-метаданных.

В настоящей статье рассмотрено решение задачи автоматизации преобразования текста на естественном языке в формат RDF с помощью технологий семантического анализа. Приведен пример разработки алгоритма анализа текстов технической документации на русском языке.

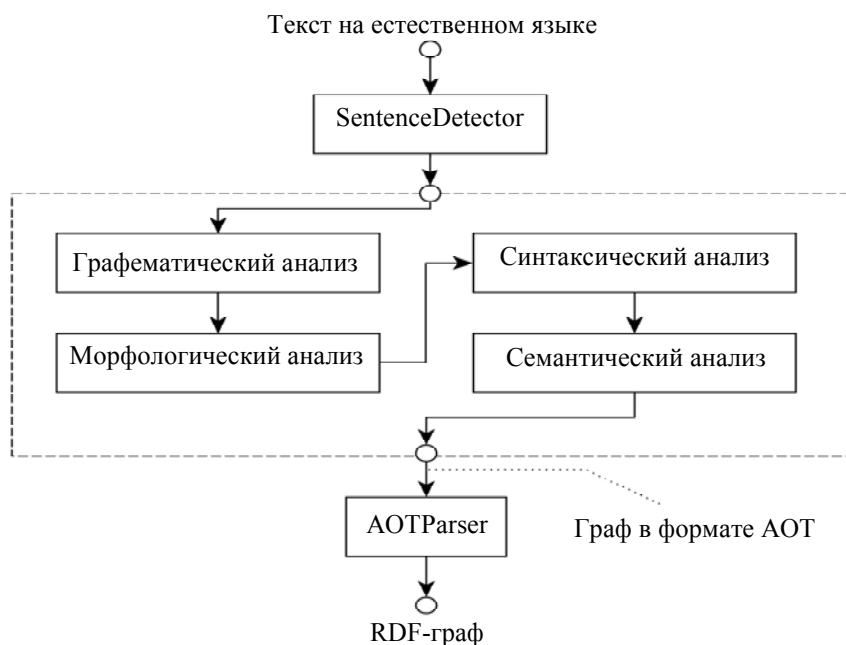
Алгоритм преобразования. Ввиду сложности поставленной задачи в алгоритме преобразования текста в формат RDF целесообразно выделить несколько последовательных этапов.

Так как минимальной единицей, с которой работают семантические анализаторы текста, является предложение [4, 5], на первом этапе необходимо разбить массив на отдельные предложения. После этого для построения модели необходимо определить семантические отношения между словами предложения. Далее, на основании полученной модели, можно синтезировать итоговый RDF-граф.

1. Разбор текста на предложения. Наибольшую трудность на этом этапе представляет однозначное определение роли того или иного знака препинания в предложении. Так, например, точка может обозначать сокращение слов, отделять целую часть числа от дробной (в десятичных дробях), быть частью многоточия (которое не всегда обозначает конец предложения) или, например, адреса электронной почты. Особенности обработки сокращений слов и знаков препинания (определения того, какие из знаков могут обозначать конец предложения) зависят от языка анализируемого текста. Важно учитывать, что „универсального“ алгоритма разбиения текста на предложения не существует, и любой алгоритм всегда будет давать некоторый процент некорректных срабатываний [6].

Авторами статьи предложен алгоритм разбиения текста, который для определения спорных случаев использует тезаурусы с набором терминов и сокращений, характерных для конкретной предметной области. При этом реализация алгоритма позволяет динамически переключать тезаурусы в случае изменения предметной области анализируемого текста.

На основании предложенного алгоритма авторами был разработан модуль SentenceDetector (см. рисунок), который [7] показал эффективность на тестовом наборе документов (сопроводительная документация к программным проектам) — корректно определены границы примерно 95 % предложений. Такая точность определения границ является приемлемой для решения поставленной задачи.



2. Семантический анализ предложения. После того как исходный текст на естественном языке успешно разбит на отдельные предложения, необходимо выполнить его семантический анализ. Цель семантического анализа — извлечение из текста информации посредством определения и формализации смысловых связей между словами в предложении.

В настоящей работе для семантического разбора предложения была адаптирована система автоматической обработки текста (АОТ), разработанная группой „Aot.ru“ и доступная по лицензии с открытым исходным кодом. Система АОТ предназначена для анализа предложений на естественном языке (в данный момент доступны русский, английский и немецкий словари [8—10]), она обрабатывает информацию по цепочке лингвистических процессоров [11]. Вход одного процессора является выходом другого.

АОТ разбирает заданное предложение по словам, переводит слова в начальную форму и определяет семантические связи между ними [12]. Для анализа текста используются подключаемые через унифицированный интерфейс словари и тезаурусы. На базе исходных кодов АОТ авторами настоящей статьи разработан модуль семантического анализа текста. Результатом работы модуля является семантический граф, представленный в собственном формате АОТ.

3. Преобразование выдачи АОТ в формат RDF. Формат, используемый системой АОТ для представления результатов семантического анализа, несовместим с какими-либо другими инструментами обработки текста. Поэтому необходимо приводить полученные на предыдущем этапе данные к некоему стандартному виду, которым, в рамках поставленной задачи, является формат RDF.

Эту задачу позволяет решить разработанный модуль AOTParser. Преобразование выходных данных АОТ в нем реализовано по собственному алгоритму, а генерация RDF-модели осуществляется с помощью библиотеки Apache Jena [13]. Полученная RDF-модель должна быть преобразована в формат, удобный для хранения и представления семантической информации. В качестве такого формата был выбран XML.

Тестирование разработанного программного модуля. На основе предложенного алгоритма преобразования текста был разработан программный модуль, блок-схема которого приведена на рисунке. Соответствие спецификации полученного в результате преобразования формата XML было проверено с помощью W3C Validation service [14]. Проведенное тестирование показало достаточно высокое качество преобразования — более 90 % загруженного текста было корректно преобразовано в RDF-модель.

Эксперименты выявили некоторые недостатки предложенного алгоритма преобразования текста, например, частичную потерю семантической информации (выполняемое преобразование не является эквивалентным), однако применительно к поставленным задачам (формирование метаданных), это несущественно, так как выполнение обратного преобразования не предполагается.

Заключение. В настоящей статье предложен алгоритм автоматического преобразования естественного языка в формат RDF и представлена программная реализация, включающая три модуля, осуществляющих преобразование в RDF-граф технических текстов на русском языке.

Тестирование разработанных модулей показало эффективность предложенного алгоритма и доказало возможность его дальнейшего применения при решении практических задач. Важным достоинством программной реализации является возможность ее использования для автоматического преобразования в формат RDF текстов другой тематики и на других языках без необходимости модификации алгоритма — с этой целью достаточно подключить к нему дополнительные тезаурусы.

СПИСОК ЛИТЕРАТУРЫ

1. RDF Primer. W3C Recommendation [Электронный ресурс]: <<http://www.w3.org/TR/rdf-primer/>>.
2. Бессмертный И. А. Семантическая паутина и искусственный интеллект // Научно-технический вестник информационных технологий, механики и оптики. 2009. № 6 (64).
3. Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American. May, 2001.

4. Сокирко А. Первичный семантический анализ [Электронный ресурс]: <<http://aot.ru/docs/seman.html>>.
5. LinkParser Grammar Tutorial [Электронный ресурс]: <<http://www.link.cs.cmu.edu/link/dict/index.html>>.
6. O'Neil J. Things with Words, Part Two: Sentence Boundary Detection [Электронный ресурс]: <<http://www.attivio.com/attivio/blog/263-doing-things-with-words-part-two-sentence-boundary-detection.html>>.
7. Тезаурусы ДИАЛИНГ [Электронный ресурс]: <<http://www.aot.ru/docs/thes.html>>.
8. Сокирко А. Технологии автоматической обработки текста [Электронный ресурс]: <<http://www.aot.ru/technology.html>>.
9. Сокирко А. Семантические отношения, используемые в модуле поверхностно-семантического анализа „Диалинг“ [Электронный ресурс]: <<http://www.aot.ru/docs/SemRels.htm>>.
10. Русский общесемантический словарь [Электронный ресурс]: <<http://www.aot.ru/docs/ross.html>>.
11. Рубинштейн М. Л. Описание английского общесемантического словаря [Электронный ресурс]: <<http://www.aot.ru/docs/aoss.html>>.
12. Немецкий общесемантический словарь [Электронный ресурс]: <<http://sourceforge.net/p/seman/svn/HEAD/tree/trunk/Dicts/>>.
13. Apache Jena RDF API Documentation [Электронный ресурс]: <<http://jena.apache.org/documentation/rdf/index.html>>.
14. World Wide Web Consortium. RDF Validator [Электронный ресурс]: <<http://www.w3.org/RDF/Validator/>>.

Сведения об авторах

- | | |
|--|--|
| Игорь Владимирович Калинин | — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: kalinin1989@mail.ru |
| Сергей Викторович Клименков | — Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; ассистент; E-mail: Serge.Klimenkov@tune-it.ru |
| Анастасия Евгеньевна Харитонова | — Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; ассистент; E-mail: Anastassia.Kharitonova@Elcom.SPb.ru |
| Евгений Алексеевич Цона | — Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; ассистент; E-mail: Evgenij.Tsopa@Elcom.SPb.ru |

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

А. М. ДЕРГАЧЕВ, А. А. ДЕРГАЧЕВ

ПАРАМЕТРЫ КАЧЕСТВА ОБСЛУЖИВАНИЯ WEB-СЕРВИСОВ ОТРАСЛИ ПРИБОРОСТРОЕНИЯ

Рассмотрены особенности сервис-ориентированного подхода к производству и поставке электронных компонентов. Определены параметры качества обслуживания web-сервисов отрасли приборостроения.

Ключевые слова: САПР, приборостроение, жизненный цикл, web-сервисы.

Введение. В полупроводниковой индустрии „виртуальные“ электронные компоненты разрабатываются и поставляются в виде кода на языках описания аппаратных средств Verilog и VHDL, в виде синтезированной принципиальной схемы (netlist), также называемой Firmware, либо в виде готовой топологии в формате GDSII [1]. Такие компоненты получили название „ядро полупроводникового интеллектуального продукта“ (Semiconductor Intellectual Property Core, SIP-компонент). Разработка SIP-компонентов различного функционального назначения, номенклатура и сложность которых постоянно увеличиваются, ведется множеством крупных и малых фирм. В настоящее время доступ к библиотекам SIP-компонентов предоставляют такие ведущие поставщики CAD/CAM/CAE-систем, как AutoCAD, SolidWorks, Synopsys, Mentor Graphics, Cadence, и разработчики SIP-компонентов „на продажу“: ARM, Dolphin, Verisilicon и др.

Разработка компонентов в электронном виде позволяет автоматизировать процесс их поставки и приобретения на программном уровне и свести к минимуму участие человека в поиске и выборе необходимого компонента. При организации доступа к таким библиотекам через Интернет возникает задача выбора поставщика из множества конкурирующих web-сервисов с учетом динамики изменения их количественного состава и показателей качества обслуживания. Возможным решением задачи могла бы стать программная реализация системы доступа к сервисам [2], позволяющая выполнять выбор поставщиков SIP-компонентов, учитывая показатели качества обслуживания. Для создания такой системы необходимо исследовать особенности взаимодействия web-сервисов.

Постановка задачи. Для проведения исследований была разработана формальная концепция взаимодействия web-сервисов [3], позволяющая организовать автоматизированный поиск, многокритериальный выбор из множества альтернатив и формализовать пооперационный доступ к сервисам. На основе формального представления необходимо сформулировать показатели качества обслуживания web-сервисов разработки SIP-компонентов для интеграции этих показателей непосредственно в запросы к сервисам с последующей программной реализацией методов и средств организации адаптивного доступа к web-сервисам [4].

Выбор и определение параметров качества обслуживания. Термины и их определения в сфере качества обслуживания именно web-сервисов (Quality of Web Service, QoWS) не определены национальными и международными стандартами. Поскольку CAD/CAM/CAE-системы различного назначения являются частью технически сложных проектно-производственных комплексов, для их стандартизации был использован ГОСТ Р 53480-2009 „Надежность в технике“ [5]. ГОСТ разработан ВНИИНАМШ с учетом основных нормативных положений международного стандарта МЭК 60050 (191):1990-12 „Надежность и качество услуг“ (IEC 60050 (191):1990-12 „Dependability and quality of service“, NEQ). ГОСТ допускает возможность изменения приводимых в нем определений, вводя в них производные признаки, раскрывая значения используемых терминов, указывая объекты, входящие в объем

определяемого понятия. В нашем случае такими объектами являются web-сервисы поддержки жизненного цикла электронных компонентов полупроводниковой индустрии. Учитывая рекомендации ГОСТ, а также исследования в области предпочтений пользователей по показателям качества обслуживания web-сервисов [6], можно определить следующие основные понятия, характеризующие качество обслуживания сервисов поддержки жизненного цикла электронных компонентов.

1. Время ответа (Q_1) — среднее время выполнения запроса к web-сервису (в секундах).
2. Готовность (Q_2) — вероятность того, что web-сервис в данный момент времени находится в работоспособном состоянии (отношение числа ответов сервиса к числу запросов).
3. Доступность (Q_3) — вероятность того, что операция web-сервиса доступна (отношение числа успешно выполненных запросов к общему числу запросов к сервису).
4. Стоимость (Q_4) — величина в денежном выражении, необходимая для вызова и выполнения проектно-производственной операции.
5. Репутация (Q_5) — оценка (от 1 до 10), сформированная пользователем (определяется на основе соотношения ожидаемых и реально получаемых результатов выполнения операции web-сервиса).

Для отражения параметров качества в формальном представлении можно определить отношение web-сервисов, аналогично отношению в базах данных, как набор экземпляров $I = \{(sid, op_1, \dots, op_n)\}$, где sid — уникальный идентификатор сервиса; op — операция сервиса, определенная как пара $op = (opid, \lambda(op))$, $opid$ — идентификатор операции и λ — функция, которая присваивает каждой сервисной операции набор значений параметров QoWS $Q = \bigcup_{i=1}^k Q_i$. Запись вида $\lambda_i(op) = Q_i$ определяет i -й параметр качества операции op , где i — индекс параметра QoWS.

Таким образом, параметры QoWS могут быть связаны с операциями для оценки различных экземпляров web-сервиса, а формальное представление, учитывающее функциональность, поведение и показатели качества обслуживания, позволяет осуществлять параметризованные запросы к сервисам.

Заключение. Определенный в работе набор параметров QoWS позволяет осуществлять выбор поставщика SIP-компонентов с соответствующими показателями, которые наиболее полно и объективно отражают качество обслуживания и могут использоваться в методах и алгоритмах формирования плана вызова web-сервисов с последующей реализацией на их основе исполнительного ядра системы — процессора запросов к сервисам.

СПИСОК ЛИТЕРАТУРЫ

1. Палташев Т., Игликов А., Алексеев М. Развитие индустрии полупроводниковых виртуальных компонентов // Компоненты и технологии. 2012. № 5. С. 44—50.
2. Марьин С. В., Ковальчук С. В. Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 21—28.
3. Дергачев А. М. Проблемы эффективного использования сетевых сервисов // Научно-технический вестник СПбГУ ИТМО. 2011. № 1 (71). С. 83—86.
4. Pernici B., Siadat S. H. Adaptation of Web Services Based on QoS Satisfaction // LNCS 6568 Services Science: Service-Oriented Computing. Berlin: Springer-Verlag, 2011. P. 65—75.
5. ГОСТ Р 53480-2009 Надежность в технике. М.: Стандартинформ, 2010. 33 с.
6. Feuerlicht G. Simple Metric for Assessing Quality of Service Design // LNCS 6568 Services Science: Service-Oriented Computing. Berlin: Springer-Verlag, 2011. P. 133—143.

Сведения об авторах

Андрей Михайлович Дергачев

— канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: dam600@mail.ru

Александр Андреевич Дергачев

— аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: born4fun@mail.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

Т. И. АЛИЕВ

ПРОЕКТИРОВАНИЕ СИСТЕМ С ПРИОРИТЕТАМИ

Рассматриваются особенности проектирования систем с приоритетами при наличии ограничений на среднее время пребывания запросов в них. В процессе проектирования определяются дисциплина обслуживания запросов в классе дисциплин со смешанными приоритетами и производительность устройства, обеспечивающие минимальную стоимость системы.

Ключевые слова: система с приоритетами, производительность, дисциплина обслуживания, смешанные приоритеты, время пребывания, стоимость системы.

Введение. Качество функционирования вычислительных систем и компьютерных сетей, задаваемое в виде ограничений на время реакции (задержки запросов), превышение которых недопустимо или крайне нежелательно, обеспечивается за счет применения приоритетных стратегий управления процессами обработки и передачи данных. Например, в информационно-управляющих системах, находящихся в контуре систем автоматического управления технологическим оборудованием или подвижными объектами, превышение ограничений на время реакции может привести к резкому снижению эффективности функционирования системы или вообще к выходу ее из строя. В маршрутизаторах и коммутаторах компьютерной сети при передаче мультимедийных пакетов применение приоритетных стратегий управления трафиком направлено на обеспечение допустимых задержек, значения которых приведены в рекомендациях ITU-T Y.1541 [1]. Поскольку указанные ограничения в таких системах могут составлять доли секунд и даже миллисекунд, то одна из особенностей таких систем заключается в отсутствии обмена с внешней памятью в процессе управления. Таким образом, внешняя память не влияет на эффективность функционирования системы в целом и, в частности, на время реакции, являющееся основной характеристикой функционирования системы. При исследовании таких систем применяются модели в виде системы массового обслуживания с одним обслуживающим устройством и неоднородным потоком запросов, обрабатываемых с заданной производительностью. При этом емкость накопителей предполагается неограниченной, что справедливо для реальных систем, в которых вероятность потери запросов из-за ограниченной емкости накопителей не превышает 10^{-3} [2]. В этом случае задача проектирования системы сводится к синтезу стратегии управления потоком поступающих в систему запросов, задаваемой в виде некоторой дисциплины обслуживания (ДО), и определению производительности системы, обеспечивающих заданные ограничения на время пребывания запросов в системе.

Постановка задачи проектирования. В качестве исходных данных для решения задачи проектирования системы с неоднородным потоком запросов и приоритетным обслуживанием используется следующая совокупность величин: число классов запросов H , поступаю-

щих в систему; интенсивности $\lambda_1, \dots, \lambda_H$ потоков запросов, которые будем полагать простейшими; средние ресурсоемкости $\theta_1, \dots, \theta_H$ обработки запросов, задаваемые в виде среднего числа команд (инструкций), выполняемых при обработке запроса соответствующего класса; коэффициенты вариации ν_1, \dots, ν_H ресурсоемкости обработки запросов; ограничения u_1^*, \dots, u_H^* на время пребывания в системе запросов:

$$u_k \leq u_k^* \quad (k = \overline{1, H}), \quad (1)$$

где u_k — среднее значение задержки (времени пребывания в системе) запросов класса k (k -запросов), зависящее от производительности V системы и ДО.

В качестве критерия эффективности рассмотрим стоимость системы: $S = S_1 + S_2$, где $S_1 = \gamma V^\chi$ — стоимость устройства (процессора), связанная зависимостью с его производительностью V через коэффициенты пропорциональности γ и нелинейности χ ; $S_2 = s_0 E$ — стоимость памяти, предназначенной для хранения поступающих в систему запросов (s_0 — стоимость единицы памяти, например, байта; E — емкость памяти). Емкость E определяется максимальным числом k -запросов \tilde{m}_k , которые могут одновременно находиться в системе:

$$E = \sum_{k=1}^H d_k \tilde{m}_k, \text{ где } d_k \text{ — объем памяти, занимаемый одним запросом класса } k. \text{ Максимальное}$$

число запросов \tilde{m}_k может быть представлено как $\tilde{m}_k = f_k m_k$, где $m_k = \lambda_k u_k$ — среднее число запросов в системе, f_k — коэффициент, зависящий от закона распределения числа запросов в системе и допустимой вероятности потерь запросов из-за ограниченной емкости памяти ($k = \overline{1, H}$). В частности, в случае геометрического закона и допустимой вероятности 10^{-3} коэффициент $f_k = 10$ для систем, загрузка которых 40 % и более.

Таким образом, стоимость системы составит:

$$S = \gamma V^\chi + s_0 \sum_{k=1}^H d_k f_k \lambda_k u_k. \quad (2)$$

Задача проектирования систем с приоритетами формулируется следующим образом: найти ДО и определить производительность системы, которые обеспечивают выполнение ограничений (1) при минимальной стоимости системы (2).

Проектирование систем с приоритетами реализуется в три этапа:

- 1) определение нижней границы производительности системы, начиная с которой можно искать ДО, обеспечивающую выполнение ограничений (1);
- 2) синтез ДО, обеспечивающей выполнение заданных ограничений при наименьшей производительности системы;
- 3) определение оптимальной производительности системы, обеспечивающей минимальную стоимость системы при выбранной ДО.

Нижняя граница производительности системы V_0 соответствует значению, начиная с которого может решаться задача выбора ДО. Другими словами, если $V < V_0$, то не может быть найдена ДО, обеспечивающая требуемое качество функционирования системы.

При отсутствии ограничений на время пребывания запросов в системе нижняя граница производительности V'_0 определяется из условия отсутствия перегрузки: $R < 1$, где

$R = \sum_{i=1}^H \rho_i$ — суммарная нагрузка системы, $\rho_i = \frac{\lambda_i \theta_i}{V}$ — нагрузка, создаваемая i -запросами,

откуда $V > \sum_{i=1}^H \lambda_i \theta_i$, $V'_0 = \sum_{i=1}^H \lambda_i \theta_i$.

Для систем с ограничениями (1) под нижней границей производительности V_0'' понимается значение, начиная с которого может существовать ДО, обеспечивающая выполнение заданных ограничений. Для определения V_0'' воспользуемся законом сохранения времени пребывания [2], который запишем в следующем виде:

$$\sum_{i=1}^H \rho_i u_i = \frac{R}{2(1-R)} \sum_{i=1}^H \lambda_i b_i^2 (1+v_i^2) + \sum_{i=1}^H \rho_i b_i.$$

Заменив в этом выражении среднее значение времени пребывания u_i на ограничение u_i^* и учитывая неравенство (1), получим:

$$\sum_{i=1}^H \rho_i u_i^* \geq \frac{R}{2(1-R)} \sum_{i=1}^H \lambda_i b_i^2 (1+v_i^2) + \sum_{i=1}^H \rho_i b_i. \quad (3)$$

Выражение (3) представляет собой необходимое условие существования ДО, обеспечивающей выполнение ограничений (1).

Заменив в (3) все величины, зависящие от производительности V , на соответствующие выражения: $b_i = \frac{\theta_i}{V}$; $\rho_i = \frac{\lambda_i \theta_i}{V}$; $R = \frac{1}{V} \sum_{i=1}^H \lambda_i \theta_i$ и решив квадратное неравенство относительно V , получим:

$$V > \frac{1}{2} \left(\sum_{i=1}^H \lambda_i \theta_i + \sum_{i=1}^H \lambda_i \theta_i^2 / \sum_{i=1}^H \lambda_i \theta_i u_i^* \right) + \left[\frac{1}{4} \left(\sum_{i=1}^H \lambda_i \theta_i + \sum_{i=1}^H \lambda_i \theta_i^2 / \sum_{i=1}^H \lambda_i \theta_i u_i^* \right)^2 - \sum_{i=1}^H \lambda_i \theta_i \left(2 \sum_{i=1}^H \lambda_i \theta_i^2 - \sum_{i=1}^H \lambda_i \theta_i^2 (1+v_i^2) \right) / 2 \sum_{i=1}^H \lambda_i \theta_i u_i^* \right]^{1/2}. \quad (4)$$

Обозначим через V_0'' выражение в правой части неравенства (4). Значение V_0'' позволяет учитывать ограничения u_i^* ($i = \overline{1, H}$) на время пребывания запросов по всем классам. С уменьшением u_1^*, \dots, u_H^* требуемое значение V_0'' растет, а при больших значениях u_1^*, \dots, u_H^* — стремится к V'_0 .

Значение V_0'' было получено из необходимого условия (3) существования ДО, обеспечивающей выполнение ограничений (1), и может рассматриваться как необходимое, но не достаточное для синтеза ДО. Кроме того, в (4) ограничения (1) учитываются по всем классам в совокупности, при этом не учитываются особенности каждого класса.

Значения производительности V_1''', \dots, V_H''' , учитывающие ограничения на времена пребывания запросов по каждому классу, могут быть получены на основе следующих рассуждений.

Минимальное время пребывания в системе для k -запросов может быть обеспечено за счет присвоения этому классу самого высокого абсолютного приоритета [2]. Это время, согласно (1), должно быть меньше u_k^* :

$$\frac{\lambda_k b_k^2 (1 + v_k^2)}{2(1 - \rho_k)} + b_k < u_k^* \quad (k = \overline{1, H}). \quad (5)$$

Заменяя в (5) параметры, зависящие от производительности, после некоторых преобразований получим:

$$V > \frac{1}{2} \left(\lambda_k \theta_k + \frac{\theta_k}{u_k^*} \right) + \left[\frac{1}{4} \left(\lambda_k \theta_k + \frac{\theta_k}{u_k^*} \right)^2 - \frac{\lambda_k \theta_k^2 (1 - v_k^2)}{2u_k^*} \right]^{1/2} \quad (k = \overline{1, H}). \quad (6)$$

Обозначим через V_k''' выражение, стоящее в правой части неравенства (6). Значение V_k''' представляет собой нижнюю границу производительности для k -запросов, учитывающую ограничение $u_k \leq u_k^*$ ($k = \overline{1, H}$).

Окончательно нижняя граница производительности системы V_0 при ограничениях (1) определяется как $V_0 = \max(V_0'', V_1''', \dots, V_H''')$.

Синтез дисциплины обслуживания. Очевидно, что требуемое качество функционирования системы, заданное в виде ограничений (1), может быть достигнуто при любой ДО только за счет производительности системы, при этом наилучшим решением следует считать ДО, обеспечивающую эти ограничения при производительности, близкой к нижней границе. Разработка высокоэффективных и хорошо формализованных алгоритмов синтеза приоритетных ДО, позволяющих получить однозначное оптимальное решение, представляет собой сложную задачу. В этом случае целесообразно использовать эвристические алгоритмы, предполагающие целенаправленный перебор множества ДО в классе ДО со смешанными приоритетами (СП) [3]. Дисциплина обслуживания СП задается матрицей приоритетов (МП) $Q = [q_{ij} \ (i, j = \overline{1, H})]$, где q_{ij} описывает приоритет i -запросов по отношению к j -запросам: 0 — нет приоритета, 1 — приоритет относительный (ОП) и 2 — приоритет абсолютный (АП). Тогда среднее время пребывания k -запросов [4]:

$$u_k = \frac{\sum_{i=1}^H (2 - q_{ki})(1 + q_{ki}) \lambda_i b_i^2 (1 + v_i^2)}{[2 - \sum_{i=1}^H q_{ik}(3 - q_{ik}) \rho_i][2 - \sum_{i=1}^H (1 - q_{ki})(2 - q_{ki}) \rho_i]} + \frac{2b_k}{2 - \sum_{i=1}^H q_{ik}(q_{ik} - 1) \rho_i}.$$

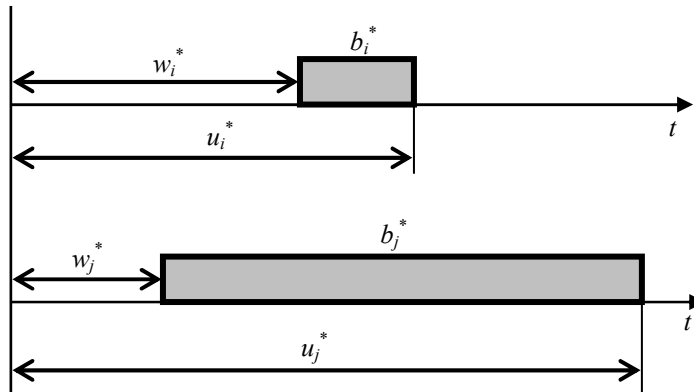
Задача синтеза ДО сводится к определению значений q_{ij} , при которых выполняются ограничения (1).

Аналитическое решение системы неравенств (1) не представляется возможным, поскольку число различных ДО СП даже при небольшом количестве классов запросов значительно. Так, в случае пяти классов ($H = 5$) число корректных ДО более 4,5 тыс, а при $H = 10$ — более 100 млн. Последовательный перебор всех возможных МП приводит к большим затратам времени, снизить которые можно, используя эвристические алгоритмы.

Алгоритм распределения приоритетов, основанный на целенаправленном переборе различных ДО, предполагает задание начального варианта назначения приоритетов. Классы запросов должны быть расположены в порядке убывания отношения θ_k / u_k^* , т.е. по правилу: $\theta_\alpha / u_\alpha^* > \theta_\beta / u_\beta^* > \dots > \theta_\omega / u_\omega^*$. Для последовательности номеров $\alpha, \beta, \dots, \omega$ формируется начальный вариант распределения приоритетов по правилу: классам запросов, расположенным правее в указанной последовательности, назначается приоритет не выше, чем классам, расположенным левее. В нашем случае наиболее высокий приоритет назначается запросам класса

α и самый низкий — запросам класса ω . В качестве начального варианта может быть выбрана ДО ОП или ДО АП.

Необходимость упорядочения классов запросов по указанному правилу иллюстрирует рисунок, из которого видно, что из-за большой ресурсоемкости обслуживания θ_j ($b_j = \theta_j / V$) j -запросов, по сравнению с i -запросами, целесообразно более высокий приоритет назначить j -запросам, несмотря на то что $u_j^* > u_i^*$. Это необходимо для обеспечения меньшего времени ожидания j -запросов: $w_j^* < w_i^*$.



Каждый последующий вариант назначения приоритетов формируется на основании показателя, определяющего необходимость изменения приоритета k -запросов. Таким показателем может служить относительное отклонение ζ_k времени пребывания u_k , полученного для рассматриваемого варианта назначения приоритетов, от заданного ограничения: $\zeta_k = (u_k^* - u_k) / u_k^*$ ($k = \overline{1, H}$). При этом в первую очередь необходимо повышать приоритет класса с минимальным значением ζ_k . Приоритет может изменяться путем изменения приоритета данного класса относительно других классов или других классов относительно данного. При этом необходимо стремиться к тому, чтобы значения ζ_k для всех классов были одинаковы.

Определение оптимальной производительности. На последнем этапе определяется оптимальное значение производительности, которое, обеспечивая при выбранной ДО выполнение заданных ограничений (1), позволяет минимизировать стоимость S системы (см. (2)).

Задача отыскания минимума функции S сводится к решению уравнения, полученного путем приравнивания нулю производной по V от функции S :

$$\gamma \chi V^{\chi-1} + s_0 \sum_{k=1}^H d_k f_k \lambda_k \frac{du_k}{dV} = 0,$$

где u_k определяется для выбранной на предыдущем этапе ДО СП. Это уравнение решается с учетом ограничений (1), к которым на данном этапе могут быть добавлены требования по производительности и надежности системы [4, 5]. Ограничения определяют область допустимых значений V , в которой ищется оптимальное значение производительности. В частности, решив систему неравенств $u_k(V) \leq u_k^*$, находим, что V должно определяться из условия: $V > \max \{V_1, \dots, V_H\}$, где $u_k(V_k) = u_k^*$ ($k = \overline{1, H}$).

Заключение. Предлагаемый подход к проектированию систем с приоритетами при наличии ограничений на среднее время пребывания в системе запросов разных классов позволяет решить задачу выбора наилучшей дисциплины обслуживания в классе дисциплин со смешанными приоритетами и обеспечить минимальную стоимость системы.

СПИСОК ЛИТЕРАТУРЫ

1. Рекомендация МСЭ-Т У.1541 (02/2006 г.). Требования к сетевым показателям качества для служб, основанных на протоколе IP. 2006.
2. Алиев Т. И. Основы моделирования дискретных систем. СПб: СПбГУ ИТМО, 2009. 363 с.
3. Алиев Т. И. Характеристики дисциплин обслуживания заявок с несколькими классами приоритетов // Изв. АН СССР. Техническая кибернетика. 1987. № 6. С. 188—191.
4. Алиев Т. И. Задачи синтеза систем с потерями // Изв. вузов. Приборостроение. 2012. Т. 55, № 10. С. 57—63.
5. Богатырев В. А., Богатырев С. В., Богатырев А. В. Функциональная надежность вычислительных систем с перераспределением запросов // Изв. вузов. Приборостроение. 2012. Т. 55, № 10. С. 53—56.

*Сведения об авторе***Тауфик Измайлович Алиев**

— д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; заведующий кафедрой
E-mail: aliev@dl.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.89: 002.53

Л. А. МУРАВЬЕВА-ВИТКОВСКАЯ

**МЕТОД РАСЧЕТА
ХАРАКТЕРИСТИК ЗАМКНУТЫХ ДЕТЕРМИНИРОВАННЫХ МОДЕЛЕЙ
МУЛЬТИСЕРВИСНЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ**

Рассматривается метод расчета характеристик мультисервисных компьютерных сетей, моделями которых являются замкнутые сети с детерминированным временем обслуживания заявок в узлах. Предположение о детерминированном обслуживании позволяет учитывать реальную статистику распределений фиксированных длин пакетов в мультисервисных компьютерных сетях.

Ключевые слова: мультисервисные компьютерные сети, детерминированное обслуживание, замкнутые сети массового обслуживания.

Введение. Для анализа процесса функционирования мультисервисных компьютерных сетей (КС) широко используются сетевые модели массового обслуживания, учитывающие наличие множества ресурсов. Замкнутые сети массового обслуживания (МО), содержащие постоянное число заявок, успешно применяются для моделирования работы мультисервисных КС [1].

Хорошо известны методы расчета характеристик замкнутых сетей МО при распределении по экспоненциальному закону временных интервалов обслуживания заявок в узлах сети. В настоящей статье предлагается метод расчета характеристик мультисервисных КС, моделями которых являются замкнутые сети с детерминированным временем обслуживания заявок в узлах. Предположение о детерминированном обслуживании позволяет учитывать реальную статистику распределений фиксированных длин пакетов [2].

Постановка задачи. В качестве моделей функционирования мультисервисных КС рассматриваются замкнутые сети МО, содержащие n узлов с произвольным количеством обслуживающих приборов, время обслуживания в которых детерминировано. Пакетам в моделях мультисервисных КС будут соответствовать заявки.

Для анализа процесса функционирования мультисервисных КС с кольцевой топологией, узлы которых связаны между собой последовательно, может использоваться циклическая модель с одноканальными узлами. Узлы модели отображают задержки, возникающие в узлах передачи данных, при этом выходная информация предыдущего узла является входной для последующего элемента структуры.

Циклическая модель с многоканальными узлами может применяться для анализа процесса функционирования локальных сетей, имеющих кольцевую топологию, центры коммутации в которых могут содержать один или несколько процессоров.

Для анализа работы мультисервисных КС, имеющих звездообразную структуру, возможно использовать модель центрального обслуживания [3]. В данной модели первый узел — центральный, отображающий работу главной ЭВМ; а узлы 2, ..., n — периферийные, соответствующие терминалам, посылающим сообщения через главную ЭВМ по дуплексным каналам; p_2, \dots, p_n — вероятность передач из центрального узла в периферийные. В модели выделяется дуга, являющаяся внешней по отношению к сети, на которой отмечается нулевой узел „0“. Относительно нулевого узла определяются временные характеристики функционирования сети [4].

Оценить качество функционирования мультисервисных КС можно, определив характеристики сетевых моделей, основной из которых является производительность λ_0 , измеряемая как интенсивность потока заявок, проходящих через нулевой узел.

Циклическая однородная модель с одноканальными узлами. Положим, что в n -узловой замкнутой сетевой модели циркулирует M заявок. Тогда коэффициент загрузки узла j ($j = 1, \dots, n$) определяется как

$$\rho_j = \begin{cases} b_j / b_{\max}, & \text{если } M b_{\max} \geq B; \\ M b_j / B, & \text{если } M b_{\max} < B, \end{cases} \quad (1)$$

где b_j — время обслуживания заявки в узле j ; $b_{\max} = \max\{b_1, \dots, b_n\}$; $B = \sum_{j=1}^n b_j$.

Справедливость выражения (1) вытекает из следующих рассуждений. Параметр B — время обслуживания одной заявки в сети, $M b_{\max}$ — время обслуживания всех заявок в максимально загруженном узле сети. Тогда условие $M b_{\max} \geq B$ означает, что время обслуживания всех заявок в максимально загруженном узле будет больше времени, затраченного на обслуживание одной заявки во всей сети, т.е. некоторая заявка вернется в максимально загруженный узел раньше, чем в нем завершится обслуживание всех остальных заявок. Таким образом, этот максимально загруженный узел никогда не будет простаивать, т.е. его коэффициент загрузки равен единице: $\rho_{\max} = 1$. Коэффициенты загрузки остальных узлов связаны с рассматриваемым коэффициентом известным соотношением $\rho_j / \rho_{\max} = \lambda_0 b_j / \lambda_0 b_{\max}$, откуда $\rho_j = b_j / b_{\max}$.

При $M b_{\max} < B$ коэффициент загрузки максимально загруженного узла меньше единицы. Тогда, рассмотрев интервал времени, равный B , можно заметить, что время, в течение которого узел j занят обслуживанием всех заявок, циркулирующих в сети, равно $M b_j$, откуда следует, что коэффициент загрузки определяется как $\rho_j = M b_j / B$, что и требовалось доказать.

Циклическая однородная модель с многоканальными узлами. Пусть, как и в предыдущем случае, в сети циркулирует M заявок. Число обслуживающих приборов K_j в узлах сети произвольно. Тогда коэффициент загрузки узла j ($j = 1, \dots, n$) определяется следующим образом:

$$\rho_j = \begin{cases} x_j / x_{\max}, & \text{если } M x_{\max} \geq B; \\ M x_j / B, & \text{если } M x_{\max} < B, \end{cases} \quad (2)$$

где $x_j = b_j / K_j$; $x_{\max} = \max \{x_1, \dots, x_n\}$.

Очевидно, что рассуждения, доказывающие справедливость формулы (1), сохраняют силу и в данном случае. Отличие заключается в способе определения значения времени обслуживания всех заявок в узле и коэффициентов загрузки. Для многоканального узла время обслуживания всех заявок в j -м узле определяется как $M b_j / K_j$, а коэффициент загрузки

$$\rho_j = \lambda_0 b_j / K_j. \quad (3)$$

Следовательно, если $\rho_{\max} = 1$, то $\rho_j / \rho_{\max} = x_j / x_{\max}$, откуда $\rho_j = x_j / x_{\max}$. При отсутствии узла с $\rho = 1$ параметр ρ_j можно определить как отношение времени обслуживания всех заявок в узле j ко времени обслуживания одной заявки в сети $\rho_j = M x_j / B$.

Однородная модель центрального обслуживания с одноканальными узлами. Заметим, что в этой модели коэффициенты передачи, в отличие от предыдущих моделей, совпадают со значениями вероятности передач: $\alpha_j = p_j$ ($j = 2, \dots, n$); $\alpha_1 = 1$. В сети, как и ранее, циркулирует M заявок.

Расчет такой модели осложняет случайный характер переходов из центрального узла в периферийные, описываемый в виде вероятностей передач p_2, \dots, p_n . Однако для модели центрального обслуживания (МЦО), в которой соблюдается локальный баланс для периферийных узлов, т.е. $\alpha_j b_j = \text{const}$ для $j = 2, \dots, n$, можно предложить достаточно простой метод приближенного расчета характеристик сети. Метод предполагает сведение исходной сети центрального обслуживания к двухузловой циклической, второй узел которой представляет собой многоканальную систему МО, с $n-1$ обслуживающими приборами, в которых время обслуживания определяется из условия эквивалентности экспоненциальных сетей МО с центральным обслуживанием и двухузловой с многоканальным (МК) вторым узлом [4]. Время обслуживания в приборах многоканального узла циклической сети будем полагать равным $\alpha_j b_j$ ($j = 2, \dots, n$). Тогда по формуле (2) определим коэффициент загрузки 1-го узла ρ_1 , производительность сети $\lambda_0^{\text{МК}} = \rho_1 / b_1$, затем для $\lambda_0^{\text{МЦО}} = \lambda_0^{\text{МК}}$ получим формулу для расчета коэффициентов загрузки узлов $j = 2, \dots, n$ в модели центрального обслуживания:

$$\rho_j = \alpha_j b_j \rho_1 / b_1. \quad (4)$$

Циклическая неоднородная модель с одноканальными узлами. Рассмотрим замкнутую сетевую модель с n одноканальными узлами, в которой циркулируют заявки H классов.

Число заявок в сети $M = \sum_{i=1}^H m_i$, где m_i — число заявок класса i . Время обслуживания заявок

класса i в j -м узле b_{ij} детерминировано, тогда $B_j = \sum_{i=1}^H m_i b_{ij}$ — время обслуживания всех заявок

в j -м узле ($j = 1, \dots, n$), а $B_{\max} = \max \{B_1, \dots, B_n\}$ — время обслуживания всех заявок в максимально загруженном узле. Время обслуживания заявки класса i в сети, т.е. время обслужи-

вания заявки класса i в сети $T_i = \sum_{j=1}^n b_{ij}$ ($i = 1, \dots, H$), а $T_{\max} = \max \{T_1, \dots, T_H\}$ — максимальное

время обслуживания в сети. Аналогично формуле (1) получим выражение для коэффициента загрузки узла j заявками класса i ($i = 1, \dots, H; j = 1, \dots, n$):

$$\rho_{ij} = \begin{cases} b_{ij} m_i / B_{\max}, & \text{если } B_{\max} \geq T_{\max}; \\ b_{ij} m_i / T_{\max}, & \text{если } B_{\max} < T_{\max}, \end{cases} \quad (5)$$

тогда коэффициенты загрузок узлов ρ_j ($j = 1, \dots, n$) можно определить как $\rho_j = \sum_{i=1}^H \rho_{ij}$.

Расчет характеристик замкнутых сетей МО. Характеристики замкнутых сетей МО, рассмотренных в работе, рассчитываются следующим образом. Определяются коэффициенты передач α_{ij} для всех узлов сети. Для циклических моделей $\alpha_{ij} = 1$, для модели центрального обслуживания $\alpha_{ij} = p_{ij}$ ($i = 1, \dots, H; j = 1, \dots, n$). Коэффициенты загрузок приборов для каждого узла сети определяются по формулам (1), (2), (4), (5) в зависимости от типа модели. Тогда производительность однородной замкнутой сети определяется как $\lambda_0 = \rho_j K_j / \alpha_j b_j$ ($j = 1, \dots, n$). Из формулы Литтла [5] находится среднее время пребывания заявки в замкну-

той сети $U = M / \lambda_0$. Для неоднородной замкнутой циклической модели $\lambda_0 = \sum_{i=1}^H \lambda_{0i}$, где

$\lambda_{0i} = \rho_{ij} / b_{ij}$ ($i = 1, \dots, H; j = 1, \dots, n$). Тогда среднее время пребывания заявки класса i в сети $U_i = m_i / \lambda_{0i}$, а среднее время пребывания заявок объединенного потока $U = M / \lambda_0$.

Примеры. Рассмотрим замкнутую циклическую модель с двумя узлами, число приборов и время обслуживания в которых соответственно: $K_1 = 3, b_1 = 2$ с; $K_2 = 5, b_2 = 4$ с. В сети циркулирует $M = 7$ заявок, тогда, согласно (2), коэффициенты загрузок $\rho_1 = 7/9$ и $\rho_2 = 14/15$. Производительность сети $\lambda_0 = 7/6$ с⁻¹, тогда среднее время пребывания заявки в ней $U = 6$ с.

Рассчитаем характеристики циклической неоднородной модели с тремя узлами, в которой циркулируют заявки двух классов. Число заявок каждого класса и время обслуживания заявок в узлах соответственно: $m_1=2; m_2=1; b_{11}=2$ с; $b_{21}=3$ с; $b_{12}=1$ с; $b_{22}=2,5$ с; $b_{13}=1,5$ с; $b_{23}=4,5$ с. Тогда, согласно (5), коэффициенты загрузок узлов заявками разных классов $\rho_{11} = 0,4; \rho_{21} = 0,3; \rho_{12} = 0,2; \rho_{22} = 0,25; \rho_{13} = 0,3; \rho_{23} = 0,45$, а коэффициенты загрузок узлов $\rho_1 = 0,7; \rho_2 = 0,45; \rho_3 = 0,75$ производительность сети $\lambda_0 = 3$ с⁻¹; среднее время пребывания заявок объединенного потока $U = 10$ с.

Заключение. Сравнительный анализ характеристик замкнутых сетей МО, рассчитанных в предположении об экспоненциальном и детерминированном характере обслуживания в узлах сети, показывает их существенное различие. В частности, результаты расчета характеристик обслуживания пакетов, полученные на сетевых моделях с детерминированным и экспоненциальным временем обслуживания в узлах, могут различаться на 30 % и более. Следует отметить, что различие в результатах существенно зависит от параметров модели. Наиболее сильно оно проявляется при больших нагрузках узлов и их сбалансированности.

СПИСОК ЛИТЕРАТУРЫ

1. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 600 с.
2. Шварц М. Сети ЭВМ. Анализ и проектирование. М.: Радио и связь, 1981. 336 с.
3. Алиев Т. И. Стохастические модели информационно-вычислительных систем // Современные технологии. Сб. науч. статей. СПб: СПбГУ ИТМО, 2003. С. 6—17.
4. Алиев Т. И., Никульский И. Е., Пятмаев В. О. Моделирование ядра мультисервисной сети с относительной приоритезацией неоднородного трафика // Научно-технический вестник СПбГУ ИТМО. 2009. Вып. 4 (62). С. 88—96.
5. Little J. D. C. A proof for the queuing formula $L = I * w$ // Operations Research. 1961. Vol. 9, N 3. P. 383—387.

Сведения об авторе

Людмила Александровна Муравьева-Витковская

— канд. техн. наук; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: muravyeva-vitkovskaya@yandex.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.89: 002.53

В. В. СОСНИН

**ВРЕМЯ ОЖИДАНИЯ
В НЕОДНОРОДНЫХ СИСТЕМАХ С ОЧЕРЕДЯМИ
ПРИ ОБСЛУЖИВАНИИ ЗАЯВОК В ПОРЯДКЕ ПОСТУПЛЕНИЯ**

Для системы массового обслуживания с заявками двух классов с помощью имитационного моделирования рассчитано среднее время ожидания в очереди при использовании бесприоритетной дисциплины обслуживания. Показаны условия, при которых различается среднее время ожидания заявок разных классов.

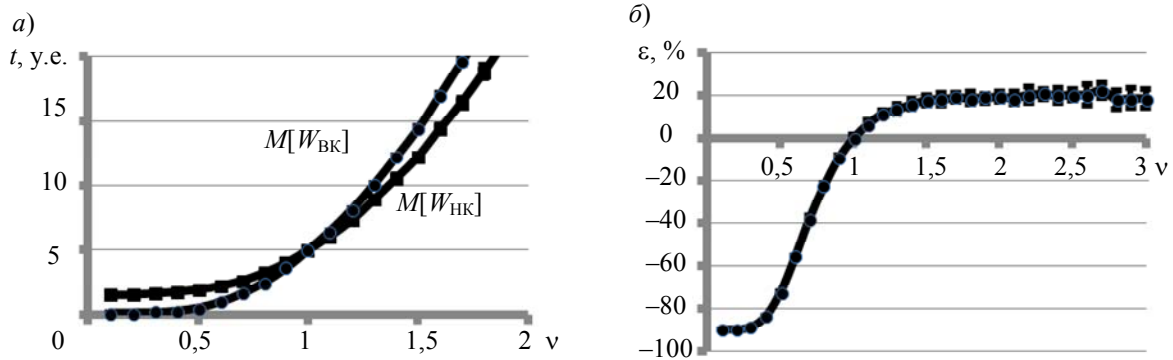
Ключевые слова: неоднородная система массового обслуживания, среднее время ожидания в очереди, бесприоритетная дисциплина обслуживания.

Введение. В теории массового обслуживания важное место занимает бесприоритетная дисциплина обслуживания (ДОБП). Традиционно считается [1], что при ДОБП качество обслуживания заявок разных классов одинаково (показателем качества считается среднее время ожидания заявки в очереди). В работе [2] показано, что в системе M/G/1 ДОБП ни один из классов не имеет преимуществ в качестве обслуживания, т.е. если значения времени ожидания в очереди заявок k классов суть случайные W_1, W_2, \dots, W_k , то их математические ожидания равны: $M[W_1] = M[W_2] = \dots = M[W_k]$. Однако цель настоящей работы — проверить, обладает ли этим свойством весь класс систем GI/GI/1 с ДОБП [3]. Для подтверждения корректности полученных автором результатов были проведены дополнительные исследования.

Поставленная задача решалась с помощью имитационного моделирования. Рассмотрим пример исследования системы массового обслуживания (СМО) GI/GI/1 с заявками двух классов (НК — низконагружающий, ВК — высоконагружающий класс), которые создают загрузки $\rho_{ВК} = 0,3$ и $\rho_{НК} = 0,03$. Время обслуживания заявок ВК и НК — случайная величина $B_{ВК}$ и $B_{НК}$ такая, что $M[B_{ВК}] = M[B_{НК}] = 10$ у.е. Время между приходом заявок ВК и НК — случайная величина $A_{ВК}$ и $A_{НК}$. Для моделирования $A_{ВК}$, $A_{НК}$, $B_{ВК}$ и $B_{НК}$ используется гамма-распределение, каждая из этих величин имеет фиксированное значение математического ожидания, а коэффициент вариации v изменяется от 0 до 3 с шагом 0,1 так, что $v = v[A_{ВК}] = v[A_{НК}] = v[B_{ВК}] = v[B_{НК}]$. При проведении имитационных экспериментов измеряются время ожидания в очереди заявок каждого из классов ($W_{ВК}$ и $W_{НК}$), а также относительное различие их средних значений:

$$\varepsilon = \frac{M[W_{ВК}] - M[W_{НК}]}{M[W_{НК}]} \cdot 100 \%$$

Результаты имитационного моделирования этой системы приведены на рисунке. Высота значков на кривых соответствует величине 99 %-ного доверительного интервала измеренных в соответствующей точке величин (по Стьюденту). На рисунке *a* видно, что средние задержки НК и ВК заметно различаются за пределами доверительного интервала. Относительное значение этого различия представлено на рисунке *б*: видно, что среднее время ожидания заявок ВК может быть на 90 % меньше или на 20 % больше среднего времени ожидания заявок НК.



Аналогичный характер имеет соотношение вариации времени ожидания, отличие состоит в том, что относительная разница их значений несколько меньше. Приведенный пример позволяет однозначно утверждать, что „свойство бесприоритетности“ ДОБП в СМО M/G/1 не может быть распространено на весь класс систем GI/G/1. Анализ экспериментальных данных показал, что заявки НК получают тем лучшее качество обслуживания, чем ближе параметры системы к следующим предельным значениям: $\nu[A_{NK}] \rightarrow 0$, $\nu[A_{VK}] \rightarrow \infty$, $\nu[B_{VK}] \rightarrow 0$, $\nu[B_{NK}] \rightarrow 0$, $\frac{\rho_{VK}}{\rho_{NK}} \rightarrow \infty$. Указанное свойство проверялось для $\nu[A_{VK}] \leq 5$. При $\nu[A_{VK}] > 5$ для получения результатов с приемлемым доверительным интервалом требуются существенные вычислительные мощности. Примем, что полученное свойство верно для всей области значений $0 < \nu[A_{VK}] < \infty$, тогда с помощью аналитических преобразований (см. подробный вывод в [3]) можно сформулировать следующее неравенство:

$$M[W_{NK}] > \rho M[W_{VK}]. \quad (1)$$

Этот результат подтверждается имитационными экспериментами с различными законами распределения (двухфазное Кокса, гамма, равномерное). Приведем пример возможного применения полученного результата. Рассмотрим процессы, протекающие на выходном порте некоторого маршрутизатора в компьютерной сети. В качестве потока заявок ВК рассматривается сетевой трафик, который проходит через порт. Современные маршрутизаторы позволяют в режиме реального времени получить данные о текущих характеристиках передачи пакетов, поэтому будем считать информацию о средних задержках пакетов потока ВК доступной и достоверной. В качестве потока заявок НК рассматривается сетевой трафик низкой интенсивности, который планируется дополнительно пустить через рассматриваемый порт. Тогда неравенство (1) позволяет оценить максимальное преимущество в качестве обслуживания, которое могут получить заявки НК. Например, пусть заявки ВК создают 60 %-ную нагрузку, а их среднее время ожидания в очереди равно 200 мс. Пусть требуется пустить в эту СМО еще один поток заявок, увеличивающий ее нагрузку на 3 %. Тогда формула (1) позволяет утверждать, что время ожидания в очереди заявок нового потока не может быть менее 123 мс.

Таким образом, в настоящей работе получены следующие результаты:

- 1) с помощью имитационных экспериментов показано, что при беспriorитетной дисциплине обслуживания заявки разных типов могут иметь разные средние значения времени ожидания в очереди, а также разные вариации этого времени;
- 2) получена численная оценка нижней границы для показателей ожидания низконагружающих потоков заявок.

СПИСОК ЛИТЕРАТУРЫ

1. Bolch G., Greiner S., Meer H., Triverdi K. Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. NY: John Wiley & Sons, 1998.
2. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 600 с.
3. Соснин В. В. Свойства беспriorитетной дисциплины обслуживания в системах вида GI/G/1 // Тр. 5-й Всерос. науч.-практ. конф. „Имитационное моделирование. Теория и практика“ (ИММОД 2011). СПб, 2011. Т. 2. С. 355—360.

Сведения об авторе

Владимир Валерьевич Соснин

— канд. техн. наук; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: vsosnin@mail.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 681.3

В. А. БОГАТЫРЕВ, А. В. БОГАТЫРЕВ, С. В. БОГАТЫРЕВ

ОПТИМИЗАЦИЯ ПЕРЕРАСПРЕДЕЛЕНИЯ НАГРУЗКИ В КЛАСТЕРАХ ПРИ ИЗМЕНЯЮЩЕЙСЯ АКТИВНОСТИ ИСТОЧНИКОВ ЗАПРОСОВ

Предложено решение задачи динамической оптимизации перераспределения запросов через сеть в общедоступный кластер, выполняемой с целью минимизации среднего времени пребывания запросов и позволяющей учитывать задержки отображения числа активных узлов, формирующих запросы.

Ключевые слова: отказоустойчивость, распределение запросов, кластер, оптимизация, адаптация.

Введение. Распределенные вычислительные системы должны характеризоваться минимальными задержками обслуживания при максимальных надежности, отказоустойчивости и производительности системы [1—3]. Эффективность распределенных компьютерных систем достигается при их адаптации к отказам, к изменениям параметров потоков запросов и состояний очередей узлов [3—5], в том числе в результате динамического перераспределения запросов между узлами компьютерной системы [4—9].

Адаптивное перераспределение запросов в реальном времени приводит, с одной стороны, к балансировке загрузки и росту эффективности системы (уменьшению среднего времени пребывания запросов), а с другой — к ее падению из-за дополнительных задержек, связанных с отображением состояний узлов. Это и определяет необходимость оптимизации процесса перераспределения запросов в вычислительных системах, включая объединение через распределенную инфраструктуру множества ресурсов, доступных из любой точки системы.

Постановка задачи. Рассмотрим распределенную компьютерную систему (рис. 1), состоящую из $n+1$ вычислительных узлов нижнего уровня и m серверов, объединяемых в общедоступный кластер на верхнем уровне. Кластер обеспечивает возможность адаптации системы к отказам и перегрузкам отдельных компьютерных узлов нижнего уровня. Перераспределение нагрузки между узлами нижнего уровня не предусматривается. Связь между компьютерами кластеров осуществляется через N резервированных коммутационных узлов.

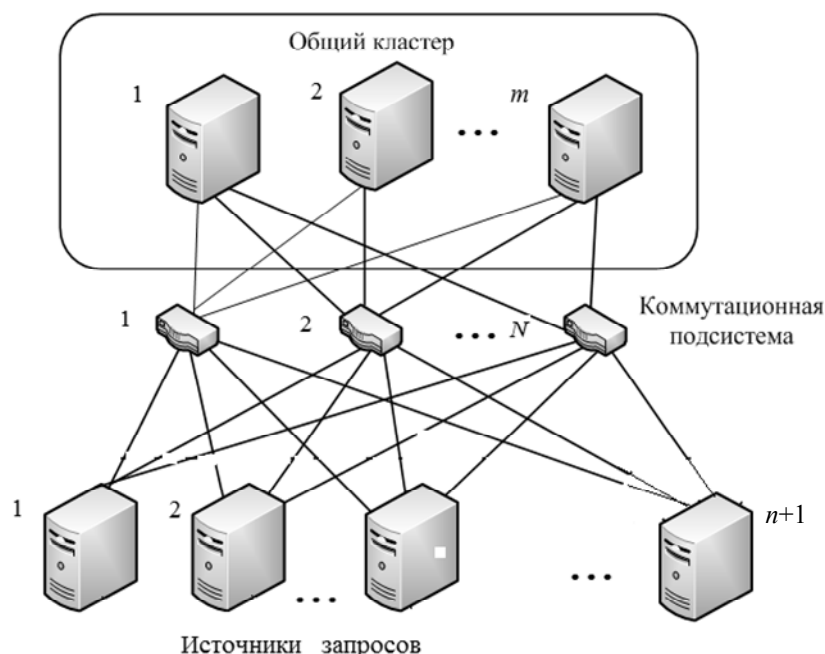


Рис. 1

Для решения задачи оптимизации необходимо найти долю запросов, перераспределяемых от каждого узла нижнего уровня через сеть в кластер, при которой достигается минимальное среднее время пребывания запросов в системе. Задача оптимизации процесса распределения запросов может решаться до начала функционирования системы с учетом априорных сведений о средних характеристиках потока запросов (статическая оптимизация) или динамически в процессе функционирования с учетом текущих измеряемых (наблюдаемых) состояний узлов и характеристик входного потока.

При оптимизации будем считать, что задано среднее время выполнения запросов в компьютерных узлах нижнего уровня, в коммутационных узлах и в серверах общедоступного кластера v_0, v_1, v_2 [6].

Выделим некоторый узел нижнего уровня, обслуживающий поток запросов. Будем считать, что каждый из n остальных компьютерных узлов нижнего уровня может находиться либо в активном состоянии, когда в нем формируется поток запросов с интенсивностью Λ , либо в пассивном (запросы не формируются). Будем считать известным среднее время нахождения узла нижнего уровня в активном t_1 и пассивном состоянии t_2 .

При статической оптимизации определяется доля перераспределяемых через сеть запросов вне зависимости от числа активных узлов нижнего уровня в момент распределения запросов, а при динамической эта доля определяется с учетом активности компьютерных узлов нижнего уровня.

При динамической оптимизации необходимо определить число активных узлов нижнего уровня в момент распределения запросов, что приведет к дополнительным временным издержкам.

Оптимизация распределения запросов

Вариант 1. Оптимизация по среднему значению активных компьютерных узлов (источников запросов).

Критерием оптимизации служит минимальное среднее время пребывания запросов при среднем числе активных узлов K [6]:

$$\min_g \left\{ g \left(\frac{v_0}{1-g\Lambda v_0} \right) + (1-g) \left(\frac{2v_1}{1-\frac{(1-g)(1+K)2\Lambda v_1}{N}} + \frac{v_2}{1-\frac{(1-g)(1+i)\Lambda v_2}{m}} \right) \right\}.$$

При оптимизации определяется доля запросов g , обслуживаемых в сформировавшем их узле, и их доля $1-g$, перераспределяемая в общедоступный кластер, с учетом ограничений:

$$g\Lambda v_0 < 1, \left[\frac{(1-g)(1+K)2\Lambda v_1}{N} \right] < 1, \left[\frac{(1-g)(1+i)\Lambda v_2}{m} \right] < 1, \tag{1}$$

где $K = \sum_{i=1}^n i C_n^i k^i (1-k)^{n-i}$, $k = t_1 / (t_1 + t_2)$, $C_n^i = \frac{n!}{i!(n-i)!}$.

Вариант 2. Оптимизация по математическому ожиданию среднего времени пребывания запросов с учетом вероятности активности i узлов нижнего уровня при $i=0, \dots, n$ в случае усредненной доли g перераспределяемых через сеть запросов [6] при ограничениях (1):

$$\min_g \left[\sum_{i=1}^n C_n^i k^i (1-k)^{n-i} \left[g \left(\frac{v_0}{1-g\Lambda v_0} \right) + (1-g) \left(\frac{2v_1}{1-\frac{(1-g)(1+i)2\Lambda v_1}{N}} + \frac{v_2}{1-\frac{(1-g)(1+i)\Lambda v_2}{m}} \right) \right] \right].$$

Вариант 3. Адаптивная оптимизация с определением в процессе функционирования системы доли перераспределяемых запросов исходя из числа активных узлов нижнего уровня в момент распределения запросов.

Доля перераспределяемых через сеть запросов g_i определяется для каждого значения i числа активных узлов нижнего уровня. Оптимизация проводится с учетом задержек, связанных с определением числа активных узлов по критерию:

$$\min_{g_i} \left[g_i \left(\frac{v_0}{1-g_i\Lambda v_0} \right) + (1-g_i) \left(\frac{2v_1}{1-\frac{(1-g_i)(1+i)2\Lambda v_1 + (\lambda i + \mu(n-i))v_3}{N}} + \frac{v_2}{1-\frac{(1-g_i)(1+i)\Lambda v_2}{m}} \right) \right],$$

при ограничениях:

$$g_i\Lambda v_0 < 1, \left[\frac{(1-g_i)(1+i)2\Lambda v_1 + (\lambda i + \mu(n-i))v_3}{N} \right] < 1, \left[\frac{(1-g_i)(1+i)\Lambda v_2}{m} \right] < 1,$$

где $\lambda=1/t_1$, $\mu=1/t_2$, v_3 — среднее время передачи пакета, сообщаемого об изменении активности узлов. Процедура определения изменений активности узлов может быть совмещена с процедурами доступа к сети [7—11].

Пример оптимизации. Проведем оптимизацию распределения запросов для $n=10$ шт., $N=1$ шт., $m=6$ шт.; $v_0=0,15$ с, $v_1=0,005$ с, $v_2=0,15$ с.

В результате расчетов (в системе MathCad 15) установлено, что для вариантов 1 и 2 оптимальны значения $g=0,556$ и $0,598$.

Для варианта 3 зависимость оптимальной доли g_i от числа активных узлов нижнего уровня i при $v_3=0,015$ представлена кривой 1 на рис. 2, кривые 2 и 3 соответствуют среднему времени пребывания запросов при оптимизации по вариантам 1 и 2; кривые 4 и 5 — среднему времени пребывания при адаптивной оптимизации при $v_3=0,015$ и $0,045$.

Зависимость среднего времени пребывания запросов от числа активных узлов нижнего уровня i при статической $T_1(i)$ [6] и динамической оптимизации $T_2(i)$ определяется как

$$T_1(i) = g \left(\frac{v_0}{1 - g\Lambda v_0} \right) + (1 - g) \left(\frac{2v_1}{1 - \frac{(1-g)(1+i)2\Lambda v_1}{N}} + \frac{v_2}{1 - \frac{(1-g)(1+i)\Lambda v_2}{m}} \right),$$

$$T_2(i) = g_i \left(\frac{v_0}{1 - g_i\Lambda v_0} \right) + (1 - g_i) \left(\frac{2v_1}{1 - \frac{(1-g_i)(1+i)2\Lambda v_1 + (\lambda i + \mu(n-i))v_3}{N}} + \frac{v_2}{1 - \frac{(1-g_i)(1+i)\Lambda v_2}{m}} \right).$$

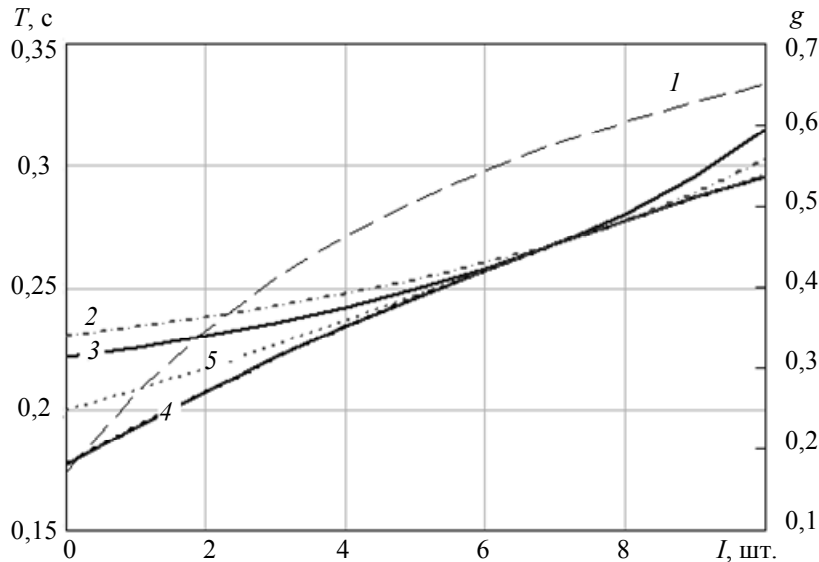


Рис. 2

Из рис. 2 видно, что при адаптивной оптимизации по мере роста суммарной интенсивности формируемого потока запросов возрастает и доля запросов, перераспределяемых через сеть в общедоступный кластер. Проведенные расчеты показали эффективность адаптивной оптимизации, причем эта эффективность может снижаться при возрастании временных издержек на определение числа активных узлов нижнего уровня. Граница этой эффективности может быть найдена из неравенства

$$\sum_{i=1}^n C_n^i k^i (1-k)^{n-i} T_2(i) \leq \sum_{i=1}^n C_n^i k^i (1-k)^{n-i} T_1(i).$$

Заключение. Поставлена и решена задача динамической оптимизации перераспределения запросов через сеть с учетом задержек определения числа активных узлов, формирующих запросы.

Показано существование границы эффективности динамической оптимизации при перераспределении запросов в зависимости от задержек, связанных с определением числа активных узлов.

СПИСОК ЛИТЕРАТУРЫ

1. Clark T. The New Data Center. New technologies are radically reshaping the data center. Brocade Bookshelf, San Jose, 2010.
2. Shooman M. Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design. John Wiley & Sons, Inc., 2002.
3. Алиев Т. И. Задачи синтеза систем с потерями // Изв. вузов. Приборостроение. 2012. Т. 55, № 10. С. 57—63.
4. Богатырев В. А. Мультипроцессорные системы с динамическим перераспределением запросов через общую магистраль // Изв. вузов. Приборостроение. 1985. № 3. С. 33—38.

5. Богатырев В. А. К повышению надежности мультимикропроцессорных систем // Изв. вузов. Приборостроение. 1982. № 8. С. 93—96.
6. Bogatyrev V. A., Bogatyrev S. V., Golubev I. Yu. Optimization and the process of task distribution between computer system cluster // Automatic Control and Computer Sciences. 2012. Vol. 46. N 3. С. 103—111.
7. Богатырев В. А. Счетно-эстафетный метод множественного доступа с динамическим отображением конфигурации локальных сетей магистральной топологии // Автоматика и вычислительная техника. 1992. № 2. С. 50—54.
8. Богатырев В. А. Маркерные методы доступа к распределенным ресурсам вычислительных систем // Автоматика и вычислительная техника. 2000. № 5. С. 71—80.
9. Богатырев В. А. Интервально-сигнальный метод динамического распределения запросов с балансировкой нагрузки системы // Автоматика и вычислительная техника. 2000. № 6. С. 73—81.
10. Богатырев В. А. Интервально-сигнальный приоритетный метод множественного доступа к магистрали // Электронное моделирование. 1990. № 3. С. 14—17.
11. Богатырев В. А. Децентрализованное кодовое управление динамическим распределением запросов к рассредоточенным ресурсам отказоустойчивых вычислительных систем // Информационные технологии. 2000. № 5. С. 13—17.

Сведения об авторах

- Владимир Анатольевич Богатырев** — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: Vladimir.bogatyrev@gmail.com
- Анатолий Владимирович Богатырев** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: Vladimir.bogatyrev@gmail.com
- Станислав Владимирович Богатырев** — Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра безопасности информационных систем; младший научный сотрудник;
E-mail: Vladimir.bogatyrev@gmail.com

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

В. А. БОГАТЫРЕВ, А. В. БОГАТЫРЕВ, С. В. БОГАТЫРЕВ

ОЦЕНКА НАДЕЖНОСТИ ВЫПОЛНЕНИЯ КЛАСТЕРАМИ ЗАПРОСОВ РЕАЛЬНОГО ВРЕМЕНИ

Предложен подход к оценке надежности вычислительных систем кластерной архитектуры с учетом требования своевременного выполнения критичных запросов.

Ключевые слова: надежность, реальное время, вычислительная система.

Введение. Управляющие вычислительные системы, особенно функционирующие в реальном масштабе времени, должны характеризоваться высокой отказоустойчивостью и надежностью и прежде всего — при выполнении критичных запросов.

Надежность выполнения системами кластерной архитектуры критичных запросов (задач) должна оцениваться с учетом: готовности системы в момент поступления запроса, безотказности во время пребывания запроса в системе, своевременности получения результатов в режиме реального времени.

Первая составляющая оценивается по коэффициенту готовности, вторая — по вероятности безотказной работы, а вторая и первая совместно — по коэффициенту оперативной готовности, третья составляющая может оцениваться по вероятности выполнения запроса с задержкой, меньшей предельно допустимой величины. Рассмотрим вычислительную систему кластерной архитектуры, komponуемую из n одинаковых вычислительных узлов, на которые с интенсивностью Λ поступает общий поток.

Оценка коэффициента готовности. Готовность системы кластерной архитектуры определяется вероятностью застать ее в момент поступления запроса в одном из работоспособных состояний. Рассматривая каждый из узлов кластера как систему массового обслуживания типа М/М/1 [1], находящуюся в стационарном режиме, определим минимальное число s исправных узлов кластера, обеспечивающих его работоспособность. Значение s вычислим как ближайшее целое, не меньшее Λv (v — среднее время выполнения запросов).

Для оценки коэффициента готовности кластера модель его отказов и восстановлений представим моделью размножения и гибели [1, 2]. Будем считать, что при работоспособности i узлов кластера суммарная интенсивность отказов системы $\lambda_i = i\lambda$, а восстановлений $\mu_i = \mu$ (λ и μ — интенсивность отказов и восстановлений одного узла). Модель размножения и гибели с ограниченным восстановлением позволяет определить ее коэффициент готовности как сумму значений вероятности работоспособного состояния r_i [2, 3]:

$$K = \sum_{i=s}^n r_i.$$

В случае обслуживания кластера с неограниченным восстановлением, когда $\mu_i = (n-i)\mu$, его коэффициент готовности определим как

$$K = \sum_{i=s}^n C_n^i k^i (1-k)^{n-i},$$

где $k = t_1 / (t_1 + t_2)$, $t_1 = 1/\lambda$ и $t_2 = 1/\mu$, $C_n^i = \frac{n!}{i!(n-i)!}$.

Полученная оценка значения коэффициента готовности может рассматриваться как верхнее приближение для систем с ограниченным восстановлением.

Оценка оперативной готовности. Будем считать, что каждый запрос распределяется в очередь одного из узлов, при отказе которого во время ожидания в очереди или его обслуживания запрос теряется, так как ограничение на время пребывания запросов в системе не позволяет повторно перераспределять их на выполнение в другой узел.

Рассматривая каждый из i работоспособных узлов кластера как систему массового обслуживания типа М/М/1, среднее время пребывания в нем запросов оценим как $T = v/(1-\Lambda v/i)$, а коэффициент оперативной готовности, т.е. вероятность поступления запроса при работоспособном состоянии кластера и безотказности выделенного узла, в течение времени пребывания запроса в системе, как

$$K = \sum_{i=s}^n r_i \exp(-\lambda T) = \sum_{i=s}^n r_i \exp\left(-\lambda \left(\frac{v}{1-\Lambda v/i}\right)\right).$$

Оценка вероятности своевременного выполнения запросов. Надежность функционирования системы P определим как вероятность $p(t_0)$ своевременного выполнения запроса с задержкой, меньшей предельно допустимой величины t_0 [4], при готовности системы в момент поступления запроса (коэффициент готовности) и ее безотказности во время ожидания и обслуживания запроса (коэффициент оперативной готовности):

$$P = \sum_{i=s}^n r_i \exp\left(-\lambda \left(\frac{v}{1-\Lambda v/i}\right)\right) p(t_0),$$

где $p(t_0) = 1 - \frac{\Lambda v}{i} \exp\left(-\left(\frac{1}{v} - \frac{\Lambda}{i}\right)t_0\right)$.

Рассмотрим решение задачи, требующей своевременного выполнения всех запросов, поступающих с интенсивностью Λ .

Надежность функционирования систем при своевременном выполнении запросов, поступающих во время решения задачи (с выделением на каждую задачу одного узла кластера), оценим как

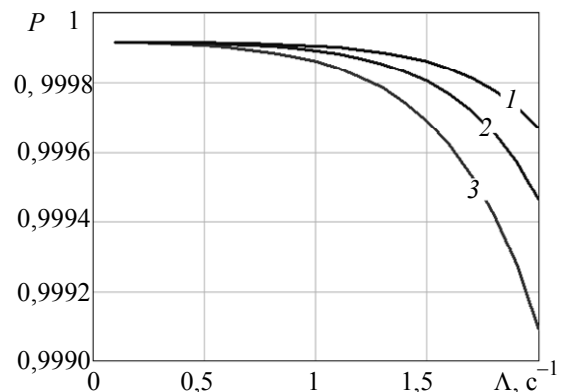
$$P = \sum_{i=s}^n r_i \exp(-\lambda t) \left[1 - \Lambda v \exp\left(-\left(\frac{1}{v} - \Lambda\right)t_0\right) \right]^{\Lambda t}.$$

Пример расчета. Зависимость вероятности своевременного выполнения критических запросов от интенсивности их поступления Λ представлена на рисунке кривыми 1—3 соответственно для $t_0 = 10, 11, 12$ с. Расчеты произведены при $n=5$, $v = 1$ с, $\lambda = 10^{-4}$ ч⁻¹, $\mu = 1$ ч⁻¹.

Математические модели надежности распределенных систем реального времени [5] могут быть детализированы с учетом влияния временных издержек на диспетчеризацию и множественный доступ к сетевым ресурсам [6—12].

Заключение. Таким образом, для кластерных систем предложен комплексный показатель надежности, учитывающий готовность вычислительной системы, ее безотказность и вероятность своевременного выполнения критических запросов.

Предложенная модель может быть использована при оценке надежности вычислительных и управляющих систем в случае выполнении ими критических запросов в реальном времени.



СПИСОК ЛИТЕРАТУРЫ

1. Алиев Т. И. Основы моделирования дискретных систем. СПб: СПбГУ ИТМО, 2009. 363 с.
2. Гуров С. В., Половко А. М. Основы теории надежности. СПб: БХВ-Петербург, 2006. 704 с.
3. Черкесов Г. Н. Надежность аппаратно-программных комплексов. СПб: Питер, 2005. 479 с.
4. Хинчин А. Я. Работы по математической теории массового обслуживания. М.: Гл. изд-во физ.-мат. лит-ры, 1963. 236 с.
5. Богатырев В. А., Богатырев А. В. Функциональная надежность систем реального времени // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 4. С. 150—151.
6. Богатырев В. А. Счетно-эстафетный метод множественного доступа с динамическим отображением конфигурации локальных сетей магистральной топологии // Автоматика и вычислительная техника. 1992. № 2. С. 50—54.
7. Богатырев В. А. Счетно-интервальный метод множественного доступа к общей магистрали // Электронное моделирование. 1991. № 2. С. 96—98.
8. Богатырев В. А. Счетно-эстафетный метод множественного доступа к магистрали // Электронное моделирование. 1991. № 3. С. 32—36.
9. Богатырев В. А. Беспriorитетный счетно-импульсный метод множественного доступа // Изв. вузов СССР. Приборостроение. 1990. № 12. С. 25—29.
10. Богатырев В. А. Повышение отказоустойчивости многомагистрального канала с помощью межмагистральной ретрансляции пакетов // Автоматика и вычислительная техника. 1999. № 2. С. 81—88.
11. Богатырев В. А. Отказоустойчивость и сохранение эффективности функционирования многомагистральных распределенных вычислительных систем // Информационные технологии. 1999. № 9. С. 44—48.
12. Bogatyrev V. A. Exchange of Duplicated Computing Complexes in Fault tolerant Systems // Automatic Control and Computer Sciences. 2011. Vol. 46, N 5. P. 268—276.

Сведения об авторах

- Владимир Анатольевич Богатырев** — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: Vladimir.bogatyrev@gmail.com
- Анатолий Владимирович Богатырев** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: Vladimir.bogatyrev@gmail.com
- Станислав Владимирович Богатырев** — Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра безопасности информационных систем; младший научный сотрудник;
E-mail: Vladimir.bogatyrev@gmail.com

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

УДК 004.2

А. Е. ПЛАТУНОВ

РЕКОНФИГУРИРУЕМЫЕ ВСТРАИВАЕМЫЕ СИСТЕМЫ И СИСТЕМЫ НА КРИСТАЛЛЕ

Рассмотрены возможности совершенствования технологии проектирования встраиваемых систем и систем на кристалле. Изучены особенности реконфигурируемой архитектуры специализированных вычислительных систем. Предложен перспективный подход к проектированию реконфигурируемых систем.

Ключевые слова: встраиваемая система, реконфигурируемая система, вычислительная архитектура, процесс проектирования, высокоуровневое проектирование.

Стремительное развитие встраиваемых вычислительных систем (ВС) и их разновидности — систем на кристалле — определяется возрастающей автоматизацией и компьютеризацией всех сторон жизни человека. Обеспечить кардинальное повышение надежности, безопасности, производительности, мобильности при одновременном сокращении сроков проектирования и стоимости позволит использование реконфигурируемой архитектуры.

Реконфигурируемые ВС в силу своей организации проблемно-ориентированы, что определяет их востребованность в сегменте встраиваемых систем и систем на кристалле. При этом они чрезвычайно разнообразны и, как следствие, сложны в проектировании.

Встраиваемая вычислительная система — любая ВС, не являющаяся системой общего назначения, т.е. не ПК и не сервер коллективного пользования. Система на кристалле — микросхема, представляющая собой совокупность взаимодействующих функциональных макроблоков ВС. Разнообразие этих систем породило большое число локальных методик проектирования. Однако системы этого типа — проблемно-ориентированные по сути и сложные по организации, что позволяет в едином ключе рассматривать методы их проектирования [1]. Основой для этого выступает проектная платформа как решение архитектурного уровня, которое определяет практически весь маршрут проектирования и реализации.

В методах и средствах проектирования систем все активнее используются принципы Hardware/Software (HW/SW) Codesign, при этом проектная платформа выступает одним из центральных понятий. Поддерживая в проектах архитектурную и микроархитектурную гибкость в широких пределах, такие методологии нуждаются в формализации и увеличении проектного пространства концептуальных решений. Распространенным технологиям, в основе которых лежат иерархии „простых“ программных платформ на фиксированной аппаратной платформе (процессоры общего назначения) или полностью заказное аппаратное проектирование (ASIC), необходима альтернатива. Сегодня на эту роль все активнее претендуют реконфигурируемые архитектуры.

В 1959 г. Дж. Эстрин ввел понятие „реконфигурируемая вычислительная машина“ как альтернативу программируемой машине фон Неймана. С этого момента работы в области реконфигурируемых архитектур (РА) ведутся постоянно. Понятие „реконфигурируемая вычислительная система“ (РВС) меняется от „программируемого кусочка кремния“ (S. Goldstein) до „системы, в которой после ее производства конфигурация подсистем или подсистемы может быть заменена или модифицирована для ее функционирования с определенной целью“ (L. Jozwiak). В работах [2—4] представлен спектр взглядов на реконфигурируемость в вычислительной технике, разнообразие РА, методы и технологии их проектирования.

В понятие РВС специалисты включают архитектуры от однородных вычислительных сред с различной гранулярностью элементов (например, FPGA) до вычислительных сетей разного масштаба (как пространственно-распределенные конфигурации, так и сети на кристалле). В пределе даже программируемые процессоры общего назначения могут определяться в качестве РВС, так как в них аппаратные блоки организуют различные тракты обработки в рамках выполнения конкретных команд [3]. Сегодня единая классификация РВС отсутствует, их сопоставляют по следующим основным характеристикам: роль и расположение ресурсов в системе; типы реконфигурируемых блоков; уровень гранулярности и степень однородности реконфигурируемых ресурсов; метод, время, степень реконфигурации.

Реконфигурируемость обеспечивает повышение производительности, функциональную гибкость и адаптивность, улучшение комплекса надежностных характеристик, сокращение энергопотребления и размеров. Другие достоинства РА требуют анализа жизненного цикла изделия и возможных подходов к его проектированию: удовлетворение нужд потребителей, быстрая разработка и вывод на рынок, повышение качества продукта, адаптация к новым стандартам, снижение стоимости разработки.

Практическое применение РА сдерживают проблемы архитектурного проектирования и отображения (mapping) прикладных алгоритмов на ресурсы РВС. Они должны решаться в комплексе с высокой степенью автоматизации для режимов статического и динамического реконфигурирования.

Разработками в области встраиваемых систем и систем на кристалле активно занимаются сотрудники научно-учебно-производственного кластера „Проектирование встраиваемых систем и СнК“ <<http://embedded.ifmo.ru>>, основу которого составляют кафедра вычислительной техники НИУ ИТМО и научно-производственная фирма „ЛМТ“ <<http://lmt.ifmo.ru>> [1]. Результатами работ являются специализированные вычислительные платформы и комплексы технических средств, на основе которых серийно выпускается множество систем и приборов различного назначения.

За последние 15 лет специалистами коллектива был создан и внедрен ряд систем, в которых активно используются принципы реконфигурируемости. Приведем самые интересные, на наш взгляд, проекты:

— ведущий процессорный модуль линейного пункта централизации — реконфигурируемый вычислитель МЗМ серийной системы железнодорожной автоматики КТС „ТРАКТ“ [5]: контроль и обработка сбоев, неисправностей, программных коллизий и их частичное исправление обеспечиваются на четырех уровнях гетерогенной реконфигурируемой архитектуры МЗМ;

— реконфигурируемые вычислительные платформы LIC5091 [6] и MiniLab [7, 8], используемые в серийных сложных аналитических приборах и инструментальных комплексах прототипирования систем на кристалле, обеспечивают до трех уровней пользовательского статического и динамического реконфигурирования;

— „Платформа автоматизированного проектирования реконфигурируемых аппаратных ускорителей“ [9] и „Инфраструктурные IP-ядра реконфигурируемых СнК“.

При создании ВС все большее значение приобретает этап высокоуровневого проектирования (HLD, High Level Design) [10], основные задачи которого создание концепции решения целевой задачи, формирование исходных спецификаций, выбор и согласование моделей вычислений, формирование архитектуры и микроархитектуры, формирование выходных спецификаций для этапа реализации. На основе опыта создания встраиваемых ВС, в том числе с РА, специалистами кластера предложены новые методики и технологии архитектурного проектирования, базирующиеся на системе архитектурных абстракций, объектно-событийных моделях вычислений и аспектной методологии проектирования [11].

Встраиваемые системы — специализированные микропроцессорные системы, непосредственно взаимодействующие с объектом контроля или управления. В отличие от традиционных микропроцессоров, ПЛИС и ASIC, система на кристалле рассматривается как ВС в интегральном исполнении, созданная на заказ по технологиям, доступным массовому разработчику (например, скомпилированная из готовых IP-ядер).

Объекты и процессы HLD представлены четырьмя группами абстракций:

- базовые элементы ВС (вычислительный механизм, вычислительная платформа, архитектурный агрегат);
- абстракции представления ВС на системном уровне (архитектура, архитектурная платформа, архитектурная модель, аспект);
- абстракции процесса проектирования (проектирование встраиваемых ВС, инфраструктура проекта, проектное пространство, аспектное пространство);
- понятия для анализа и оценки существующих архитектурных решений (вычислительный процесс, виртуальная машина, модель вычислений — МВ).

Модель вычислений — математическая модель, которая демонстрирует вычислительные возможности и правила использования субъекта вычислений (актора). Виртуальная вычислительная машина определяется как техническое решение, реализующее семантику МВ. Вычислительный механизм — техническое решение, реализующее субъект МВ. Вычислительная платформа как зафиксированный для повторного использования набор спецификаций предоставляет заданный уровень абстрагирования от конкретной реализации и выступает основным инструментом повторного использования на архитектурном уровне. Процесс проектирования встраиваемых систем рассматривается как организация вычислительного процесса в пространстве и времени с учетом исходных (техническое задание) и инфраструктурных ограничений. Аспектная модель проектирования основана на идее выделения сегментов проектного пространства, в рамках которых отражаются отдельные проблемы проекта. Разработчик формирует список концептуальных и локальных аспектов, характерных для всего проекта или отдельных его этапов.

Итак, в контексте высокоуровневого проектирования определим реконфигурируемость ВС как способность изменять МВ на этапе исполнения. Важно, что такой взгляд устраняет противопоставление, связанное с HW/SW-характером реализации реконфигурируемой части ВС.

РВС — сложный динамически изменяемый архитектурный объект, проектирование которого целесообразно выполнять с позиций унифицированного представления всех элементов вычислительного процесса, с возможно поздним их разделением на HW/SW-реализации. Задача отображения должна решаться в терминах вычислительных механизмов. Проектирование архитектуры и микроархитектуры реконфигурируемых встраиваемых вычислительных систем и систем на кристалле предлагается выполнять в представленной выше „системе вычислительных координат“. В комплексе это позволит от инженерной эмпирики переходить к целенаправленному поиску и отбору обоснованных концептуальных решений для всех уровней организации ВС.

Специалисты кластера „Проектирование встраиваемых систем и СнК“ развивают исследования в области систем со свойствами многоуровневого прикладного программирования/

конфигурирования, разрабатывают пространственно-распределенные архитектуры со сложными гетерогенными вычислительными ядрами. Архитектурные решения развиваются в комплексе с методологией проектирования и автоматизированной инструментальной инфраструктурой.

Использование принципа реконфигурируемости архитектуры позволяет получать вычислительные системы со многими полезными свойствами. Однако для этого необходимо обеспечить специалистов инструментами осознанного и контролируемого создания разнообразных архитектур и микроархитектур. Шаблонное проектирование аппаратно-программных систем должно отойти на второй план, будущее за технологиями проектирования проблемно-ориентированных ВС.

СПИСОК ЛИТЕРАТУРЫ

1. Платунов А. Е. Встраиваемые системы управления // Control Engineering Россия. 2013. Т. 43, № 1. С. 16—24.
2. Jozwiak L., Nedjah N. Modern architectures for embedded reconfigurable systems—A survey // J. of Circuits, Systems, and Computers. 2009. Vol. 18, N 2. P. 209—254.
3. Jozwiak L., Nedjah N., Figueroa M. Modern development methods and tools for embedded reconfigurable systems: A survey // Integration, the VLSI J. 2010. Vol. 43, N 1. P. 1—33.
4. Chattopadhyay A. Ingredients of Adaptability: A Survey of Reconfigurable Processors // VLSI Design. 2013. P. 18.
5. Гавриков В. О., Платунов А. Е., Никифоров Н. Л. Комплекс технических средств для систем железнодорожной автоматики // Автоматика, телемеханика и связь на железных дорогах. 1998. № 11. С. 5.
6. Голубок А. О., Платунов А. Е., Сапожников И. Д. Система управления сканирующим зондовым микроскопом // Научное приборостроение. 2003. Т. 13, № 3. С. 25—31.
7. Болгаров И. С., Маковецкая Н. А., Платунов А. Е., Постников Н. П. Проектирование приборных контроллеров // Изв. вузов. Приборостроение. 2012. Т. 55, № 10. С. 73—78.
8. Платунов А. Е., Постников Н. П. Перспективы формализации методов проектирования встроенных систем // Электронные компоненты. 2005. № 1. С. 24—29.
9. Румянцев А. С. Организация и инструментальные средства реконфигурируемых вычислительных систем // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 4. С. 79—84.
10. Platunov A., Kustarev P. Problems of Abstract Representation of Embedded Systems at High-level Stages Design // Networked Embedded and Control System Technologies: European and Russian R and D Cooperation. Proc. of the 1st Intern. Workshop. 2009. P. 100—107.
11. Platunov A., Nickolaenkov A., Penskoj A. Architectural Representation of Embedded Systems // Mediterranean Conf. on Embedded Computing. 2012. P. 80—83.

Сведения об авторе

Алексей Евгеньевич Платунов — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: platunov@lmt.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

А. А. АНТОНОВ, С. В. БЫКОВСКИЙ, П. В. КУСТАРЕВ

МОНИТОР ВРЕМЕННЫХ ОГРАНИЧЕНИЙ ДЛЯ СИСТЕМ НА КРИСТАЛЛЕ

Предложен метод формального описания внутренних временных ограничений вычислительных систем на уровне отдельных коммуникационных операций (транзакций), ориентированный на снижение сложности аппаратного исполнения соответствующих встроенных средств мониторинга и диагностики. На базе предложенного подхода спроектировано IP-ядро монитора временных ограничений для систем на кристалле с топологией „общая шина“, приведены результаты его экспериментальной реализации на ПЛИС.

Ключевые слова: *монитор ограничений, система на кристалле, реальное время, LTL, PSL, RTL, SoC, TLM.*

Введение. Соблюдение временных ограничений для процессов протекающих в сложных вычислительных системах, предназначенных для управления объектами и обработки данных в режиме реального времени, является ключевой задачей разработчика таких систем. Планирование вычислительного процесса на этапе проектирования не гарантирует фактического исполнения заданных временных ограничений во время работы. Причины нарушений могут быть как внутренними, так и внешними. Возможны ошибки в выборе архитектуры, в реализации функциональных блоков или неточное представление разработчика об окружении, в котором будет эксплуатироваться устройство. Поэтому возникает необходимость внедрения механизмов мониторинга, отслеживающих нарушения динамики вычислительного процесса.

Реализация этого подхода в системах на кристалле имеет свои особенности: возможность широкой интеграции механизмов мониторинга на аппаратном уровне существенно расширяет функционал, но видоизменяет и усложняет проектирование. Именно этой проблеме посвящена данная статья.

Обзор работ. Современные технологии проектирования предполагают разработку систем на кристалле одновременно на нескольких уровнях: системном, программном, на уровне коммуникационных процессов (Transaction Level) и регистровых пересылок (register transfer level, RTL).

В работе [1] выполнен обзор методов контроля временных ограничений на программном уровне для систем, построенных на базе программно-управляемых процессоров. Приводится описание таких методов, как Hybrid, ZM4, Trams. Но поскольку в современных системах на кристалле программно-управляемый процессор не всегда является центральным или даже обязательным компонентом системы, то программная или аппаратно-программная реализация мониторинга временных ограничений будет ограниченной по охвату системы или вообще невозможной.

„Доступность“ разработчику RTL-уровня — уникальное свойство технологии систем на кристалле, позволяющее на уровне регистровых пересылок наблюдать за поведением всех элементов системы. В настоящее время для формализации временных ограничений на RTL-уровне широко используется математический аппарат темпоральной логики (LTL, CTL). На базе этого аппарата разработаны язык PSL (Property Specification Language) и расширение SVA (SystemVerilog Assertion) для языка SystemVerilog. Отметим, что существуют и другие методы формального описания временных ограничений, например, изложенные в статье [2].

Для проверки ограничений, заданных с помощью указанных средств, используется два основных подхода: Symbolic Model Verification (SMV) и Bound Model Checking (BMC).

Из-за вычислительной сложности они неудобны для полноценной реализации в аппаратных мониторах, поэтому актуальна задача выделения такого подмножества средств проверки ограничений, которое допускает эффективный перенос и использование на аппаратной платформе.

В работах [3, 4] авторы описывают подход, позволяющий с помощью описания ограничений и свойств системы на языке PSL автоматически создавать (с помощью компилятора MBAC) аппаратные блоки контроля ограничений системы на RTL-уровне. Основным недостатком такого подхода — необходимость анализа большого объема данных, так как результаты наблюдения включают в себя маркеры изменения шин или даже отдельных электрических сигналов. В таких условиях трудно быстро локализовать проблемное место, помимо того необходим большой объем встроенной памяти системы.

В работе [5] предлагается использовать иерархический подход к формальной верификации вычислительных систем. Приведена идея проверки ограничений сначала на уровне транзакций, т.е. отдельных операций обмена данными между устройствами, для локализации места сбоя, а затем на RTL-уровне — для установления их причин. Однако в статье не описаны даже примерные варианты аппаратных мониторов транзакций.

Авторами настоящей статьи развивается и доводится до уровня конкретной реализации идея, предложенная в работе [5]. Предлагается метод формального описания временных ограничений, обеспечивающий максимально простую организацию блока проверки заданных требований. Приводится описание экспериментального аппаратного монитора транзакций, способного обнаруживать нарушения временных ограничений в реальном масштабе времени.

Метод описания временных ограничений систем на кристалле на уровне транзакций. Временные ограничения на события (ev), происходящие в системе, опишем с помощью только одного типа выражений

$$ev1 \xrightarrow{ev3} ev2, \quad (1)$$

которое следует читать так: событие $ev2$ должно произойти после наступления события $ev1$, но до $ev3$. Задав ограничения в такой форме, возможно однозначно определить правила, определяющие условия актуализации ограничения, его нарушения и выполнения.

Под актуализацией ограничения понимается внесение его в список наблюдения. Предполагается, что по умолчанию ограничения не контролируются до тех пор, пока не актуализирована определенная причинно-следственная связь. Только когда возникла такая причина, начинается ожидание следствия. В выражении (1):

- актуализация ограничения происходит в момент фиксации события $ev1$;
- нарушение ограничения происходит в момент фиксации $ev3$, если не наступило событие $ev2$;
- выполнение ограничения происходит в момент наступления события $ev2$, если до этого не произошло $ev3$.

В рамках рассматриваемого подхода множество событий предлагается разделить на два типа: транзакции обмена данными по шине и временные метки. Для формализации ограничений требуется составить списки: контролируемых событий-транзакций, возможных временных меток, связей (ограничений) между событиями.

Предположим, что для системы заданы ограничения:

$$ev1 \xrightarrow{ev3} ev2, \quad (2)$$

$$ev2 \xrightarrow{ev4=10 \text{ мс}} ev3, \quad (3)$$

согласно которым событие $ev2$ должно произойти после события $ev1$, а событие $ev3$ — в течение 10 мс после $ev2$. При этом не задаются какие-либо ограничения относительно времени

между событиями ev1 и ev2. Для данного примера списки формального описания представлены в таблице.

id	События-транзакции		
1	Write	Addr1	Data1
2	Read	Addr2	Data2
3	Write	Addr3	Data3

id	Список временных меток
4	10 мс

id	Список связей между событиями		
1	1	3	2
2	2	4	3

События-транзакции задаются шаблонами обмена по шине: адрес, слово данных, тип операции (чтение или запись). Если по шине возможно передавать данные в потоковом режиме, то можно фиксировать только первое слово в транзакции, используемое в качестве маркера, по которому монитор будет выделять определенный тип событий.

Для упрощения обработки и однозначной идентификации различных событий производится их сквозная нумерация (номера — id). Временные события являются порождаемыми. Отсчет временных промежутков запускается по факту фиксации событий-транзакций. В приведенном примере временное событие ev4 вводится в систему в момент прихода ev2.

Реализация монитора временных ограничений для систем на кристалле. Для описанной модели ограничений был разработан аппаратный монитор временных ограничений на уровне транзакций (рис. 1). Блок фиксации транзакций между устройствами в системе Bus Event Catcher анализирует входные сигналы шины, сопоставляет их с шаблоном контролируемых транзакций, хранящихся в памяти, и на выходе формирует значение id соответствующего события. Блок Event Handler принимает на вход id произошедших событий и помещает их в очередь Event FIFO. По необходимости блок может вводить в систему временное событие (метку), отдавая соответствующую команду блоку Time Event Generator. Блок Time Event Generator содержит набор таймеров, которые командой от Event Handler обновляются и потом декрементируются импульсами тактового сигнала. При достижении соответствующим таймером нуля генерируется временное событие time_ev_id.

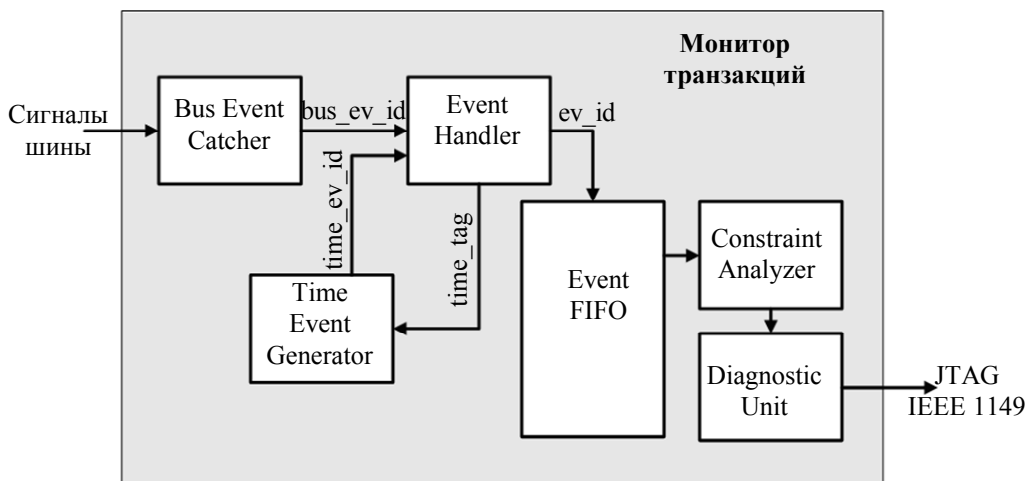


Рис. 1

Все возникающие в системе события-транзакции и временные события помещаются в порядке их возникновения в очередь Event FIFO для дальнейшей обработки (в очередь заносятся только id событий, это существенно упрощает дальнейшую обработку, так как для контроля временных ограничений не требуется сохранять и сверять время их поступления). Блок Constraint Analyzer включает память для списка всех ограничений и списка актуализированных ограничений. В процессе работы он выбирает по одному событию из очереди, анализирует список актуализированных ограничений, удаляет из него выполненные ограничения и

затем проверяет оставшиеся. Информация об обнаруженном нарушении ограничения помещается в диагностическую память блока Diagnostic Unit. В завершение анализируются все заданные ограничения и их часть, при необходимости, переносится в список актуализированных. Доступ к информации о нарушениях ограничений предоставляет контроллер диагностического интерфейса Diagnostic Unit. Конфигурация блоков монитора и съем результатов наблюдения за системой осуществляются с помощью интерфейса JTAG.

Разработанный аппаратный монитор прошел апробацию в составе экспериментальной системы на кристалле (рис. 2). Основные функции этой системы — настройка, сбор, обработка данных с „интеллектуальных“ датчиков (интерфейс I2C) и отправка результатов измерений потребителю (интерфейс CAN). В системе велось наблюдение за 11 событиями (типами транзакций). Прототип системы был реализован на ПЛИС Stratix IV GX EP4SGX230KF40C2N фирмы Altera. Функциональные параметры монитора: число используемых для реализации просмотрных таблиц ПЛИС — 1091, частота системного тактового сигнала, на котором работает разработанный монитор — 40 МГц, частота системной шины — 10 МГц, время обработки одного ограничения, которое определяет время принятия решения о нарушении наблюдаемого ограничения, равно $4T_{bus}$ (T_{bus} — время передачи слова данных по системной шине).

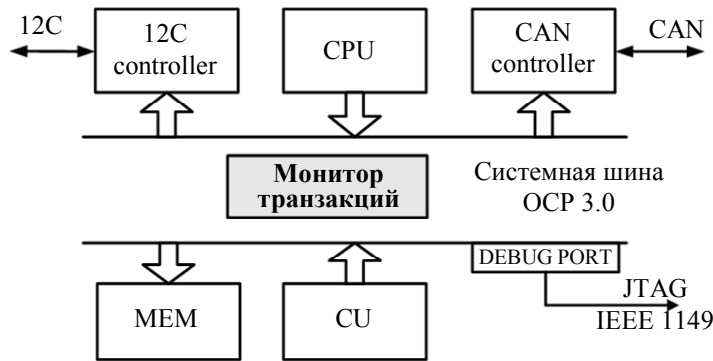


Рис. 2

Заключение. Авторами предложен метод формального описания временных ограничений для систем на кристалле на уровне транзакций. Метод обеспечивает эффективную аппаратную реализацию конечного аппаратного блока монитора.

На базе описанного метода разработан аппаратный монитор контроля временных ограничений для систем на кристалле с топологией „общая шина“. Монитор обеспечивает проверку системы в реальных условиях функционирования, с учетом всех факторов внешнего окружения; он характеризуется высокой скоростью процесса верификации требований (недостижимой при отладке модели системы в симуляторе) и возможностью контроля временных ограничений при эксплуатации системы.

СПИСОК ЛИТЕРАТУРЫ

1. Asadi N., Saadatmand M., Sjodin M. Run-Time Monitoring of Timing Constraints: A Survey of Methods and Tools // ICSEA 2013: The 8th Intern. Conf. on Software Engineering Advances. 2013. P. 391—401.
2. Aloysius K., Liu M., Liu G. Efficient Run-Time Monitoring of Timing Constraints // Proc. of the 3rd IEEE Real-Time Technology and Applications Symposium (RTAS '97). IEEE Computer Society. Washington, 1997. P. 252—257.
3. Boule M., Chenard J., Zilic Z. Adding Debug Enhancements to Assertion Checkers for Hardware Emulation and Silicon Debug // Computer Design. Intern. Conf. ICCD 2006. 2007. P. 294—299.
4. Boule M., Chenard J., Zilic Z. Assertion Checkers in Verification, Silicon Debug and In-Field Diagnosis // ISQED '07. The 8th Intern. Symp. on Quality Electronic Design. 2007. P. 613—620.

5. Kong Z., Bian J., Zhao Y., Deng S. Hierarchical Formal Verification Methodology based on Transactions // Workshop on RTL and High Level Testing. 2009. P. 1—4.

- Александр Александрович Антонов** — *Сведения об авторах*
— студент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: antonov.alex.alex@gmail.com
- Сергей Вячеславович Быковский** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: bsv.serg@gmail.com
- Павел Валерьевич Кустарев** — канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: kustarev@lmt.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.896

В. И. ПОЛЯКОВ, В. И. СКОРУБСКИЙ

ИСПОЛЬЗОВАНИЕ МНОГОЗНАЧНОЙ ЛОГИКИ ПРИ ПРОЕКТИРОВАНИИ ФУНКЦИОНАЛЬНЫХ СХЕМ

Рассмотрены приложения многозначной логики к задачам анализа цифровых схем (в том числе функционального и временного моделирования и тестирования). Рассматриваются особенности применения булевой алгебры к вычислениям над собственными подмножествами.

Ключевые слова: моделирование цифровых схем, временное моделирование, тестирование схем, трюичное тестирование.

Введение. Из многочисленных способов кодирования информации особый интерес представляет многозначная (более двух значений переменных) логика как один из опытов расширения границ осознания и формального описания логических связей реального мира. В работе [1] многозначность определена как способ отображения различных смысловых оттенков информации в рассуждениях. Таким образом, возникло направление многозначной логики, в котором работали математики, экономисты, философы, заинтересованные в повышении качества передачи информации в рассуждениях. Многозначные признаки информации — различные уровни дискретности (0, 1, 2) — в измерениях (мало, больше, среднее значение, много), ощущениях и представлениях требуют оценки, распознавания и принятия решений, несмотря на нечеткость значений. Таким образом, многозначная логика приобретает содержание в практических приложениях.

Временное моделирование цифровых схем. В САПР цифровых схем моделирование и тестирование работоспособности — обязательный этап, это единственный доступный при проектировании метод проверки качества синтеза схем с учетом реальных условий их работы.

Применяемые двузначные дискретные сигналы вследствие емкостной нагрузки могут быть представлены как трехзначные или четырехзначные на переходах, когда формируются значения на выходах логических элементов.

Работоспособность цифровых схем в ПЛИС в современных системах проектирования выявляется тестированием.

Языки HDL (Hardware Description Language) проектирования Verilog и VHDL [2] содержат соответствующие средства поддержки временного моделирования и опираются на четырехзначные логические вычисления.

В настоящей работе рассматривается подход к моделированию переходных процессов в цифровых схемах на основе четырехзначной логики.

Значения сигналов могут быть представлены четырьмя знаками $\{L, H, x/, x\}$, где $\{L, H\} = \{0, 1\}$ — постоянные, $x/$ — положительный фронт, $x\$ — отрицательный фронт.

Работу элементов с такими сигналами можно описать с помощью таблиц, заменяющих традиционные таблицы истинности (здесь ΔA — задержка):

A	\bar{A}
0	1
x/	x\
x\	x/
1	0

A&B	0	x/	x\	1
0	0	0	0	0
x/	0	x/	0	x/
x\	0	0	0	x\
1	0	x/	x\	1

A∨B	0	x/	x\	1
0	0	x/	x\	0
x/	x/	x/	1	x/
x\	x\	1	x\	x\
1	1	x/	x\	1

A⊕B	0	x/	x\	1
0	0	x/	x\	1
x/	0	x/	0	0
x\	x\	0	0	0
1	1	x\	x/	1

A	ΔA
0	0
x/	1
x\	0
1	1

На рис. 1 приведен пример вычислений. Из представленной комбинационной схемы видно, что на входе элемента 1 изменяется сигнал $x\ = H/0$, на выходе элемента 3 формируется устойчивое значение „0“. Набор значений на входе схемы выбирается с учетом задержек сигналов. Промежуточные значения формируются с задержками, и для расчета значений сигналов с учетом задержек на элементах и соединениях при моделировании выбирается такт.

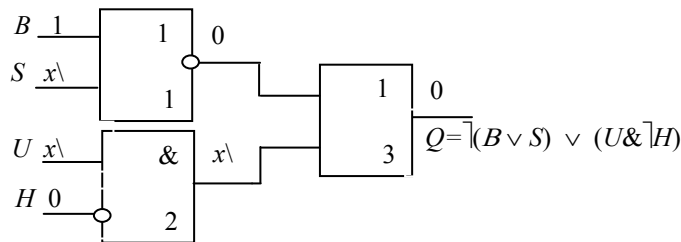


Рис. 1

Структурное трюичное тестирование цифровых схем. При изготовлении и отладке реальные цифровые схемы необходимо тестировать с целью выявления неисправностей, относящихся к „катастрофическим“ — обрыв соединений или короткие замыкания.

При этом следует сформировать смежные тесты, в результате исследования которых при изменении одного из входов изменяется выход схемы. Для известной структурной схемы можно представить это изменение как последовательное переключение входов и выходов элементов (путей) [3—5].

Обозначим значения входной переменной $\{0, 1, d\}$, где d — изменение переменной 0/1 или 1/0. В результатах контрольного теста обнаруживается неисправность, если значение d должно распространяться до контролируемого выхода схемы. Тестовая последовательность может быть выбрана при моделировании схемы с использованием табличных трюичных логических функций путем последовательного подбора значений входов при условии, что выход принимает значение d :

A	\bar{A}
d	d
0	1
1	0

A	B	A∨B	A&B	$\overline{(A \vee B)}$	A& \bar{B}
1	d	1	d	0	d
d	1	1	d	0	0
0	d	d	0	d	0
d	0	d	0	d	d

На рис. 2 приведен пример выбора одного из наборов теста $Q = d$.

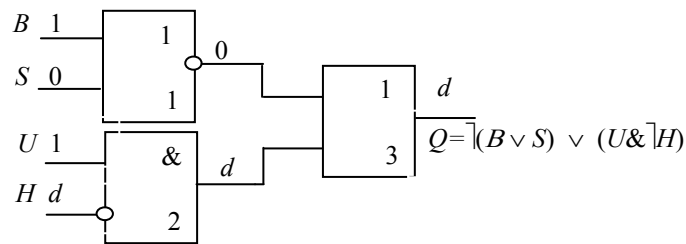


Рис. 2

Ниже представлена полная таблица наборов теста, полученная от выхода ко входам (сверху вниз).

	B	S	U	H		B	S	U	H
	1	0	1	d		d	0	0	0
$d \leftarrow$	0	$d \leftarrow$	0	1	$d \leftarrow$	d	0	1	0
	1	1	1	d		d	0	0	1
	1	1	d	1		0	d	0	1

Полный тест, контролирующий все пути схемы, может быть выбран по полученной таблице. Необходимо, чтобы подмножество тестовых наборов покрывало все входы схемы — $\vee (BSUH) = dddd$, например:

Тест	B	S	U	H
1	1	0	1	d
2	1	0	d	1
3	d	0	1	0
4	0	d	1	0

Тестовые последовательности поступают на входы схемы, тест будем считать минимальным, если найдена минимальная последовательность смежных тестов. Существование такого теста можно определить на графе смежности тестовых наборов (наборы смежны, если они различаются только по одной координате $d = \{0, 1\}$).

Тест	B	S	U	H
4	0	1	1	0
3, 4	0	0	1	0
1, 3	1	0	1	0
1, 2	1	0	1	1
2	1	0	0	1

Граф смежности тестовых наборов представлен на рис. 3.

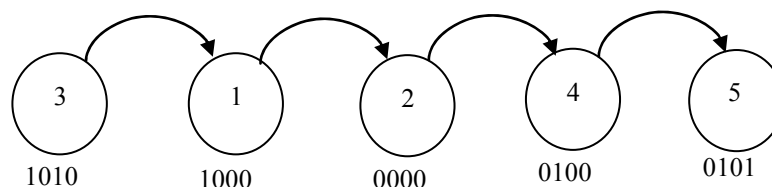


Рис. 3

Минимальный тест — это последовательность наборов $3 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$.

Булева алгебра подмножеств. Сформулируем в виде леммы известный факт из работы [6].

Лемма. Число собственных подмножеств множества M из N элементов, включая пустое множество, равно 2^N .

Упорядочим элементы множества M простым перечислением и представим его двоичным вектором $\mathbf{B} = (b_{N-1}, \dots, b_0)$ из N бит, где $b_i = 1$, если i -элемент принадлежит множеству M .

Таким образом, $\mathbf{B} = 2^N - 1$, и система из всех подмножеств M состоит из 2^N элементов, причем каждый элемент подмножества однозначно идентифицируется порядковым номером и каждое подмножество однозначно кодируется двоичным вектором (числом).

Лемма. Алгебра подмножеств с операциями $\&$ (поразрядная конъюнкция), \vee (поразрядная дизъюнкция) и $\bar{}$ (поразрядная инверсия — дополнение до \mathbf{B}) является булевой алгеброй, где $\mathbf{B} = 1$ и подмножество пустое.

Доказательством является выполнение законов булевой алгебры, что следует из справедливости этих законов для поразрядных логических операций с двоичными кодами для независимых двоичных разрядов.

Приведем пример вычисления (подмножества M). Вычислим логическое выражение, где B, S, U, H — подмножества из множества, содержащего 8 элементов, $B = 10000111$, $S = 10100100$, $U = 10000101$, $H = 00100011$:

$$\begin{aligned} Q &= \bar{}(B \vee S) \vee (U \& \bar{}H) = \bar{}(10100111) \vee (10000100) = \\ &= (01011000) \vee (10000100) = (11011100). \end{aligned}$$

Аналогичным образом может выполняться параллельное трехзначное векторное тестирование схем.

Заключение. Таким образом, многозначное кодирование логики можно использовать при решении практических задач анализа: в частности, при контроле переходных процессов в функциональных схемах и тестировании реальных схем на наличие неисправностей в виде обрывов и коротких замыканий.

Работа выполнена при финансовой поддержке РФФИ (грант 12-07-00376-а).

СПИСОК ЛИТЕРАТУРЫ

1. Карпенко А. С. Логика Лукасевича и простые числа. М.: Либроком, 2009. 256 с.
2. Harris D. M., Harris S. L. Digital Design and Computer Architecture. Elsevier Inc., 2012. 712 p.
3. Зыков А. Г., Немолочнов О. Ф., Поляков В. И. Построение комплексного покрытия последовательностных схем методом пересечения покрытий систем булевых функций // Научно-технический вестник СПб ГИТМО (ТУ). 2002. Вып. 6. С. 109—112.
4. Зыков А. Г., Немолочнов О. Ф., Поляков В. И. Методы верификации цифровых устройств // Тр. Междунар. науч.-техн. конф. „Интеллектуальные системы“ (IEEE AIS'04) и „Интеллектуальные САПР“ (CAD-2004). М.: Физматлит, 2004. Т. 2. С. 39—41.
5. Гатчин Ю. А., Зыков А. Г., Поляков В. И., Поляков И. В. Методологическое обеспечение синтеза тестов логических схем // „Информационные технологии в профессиональной деятельности и научной работе“: Сб. матер. Всерос. науч.-практ. конф. Йошкар-Ола: Марийский гос. тех. ун-т, 2011. Т. 1. С. 113—119.
6. Курош А. Г. Общая алгебра. М.: Физматлит, 1974. 162 с.

Сведения об авторах

Владимир Иванович Поляков

— канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: v_i_polyakov@mail.ru

Владимир Иванович Скорубский

— канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: vlis@km.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

А. Ю. КОРМИЛИЦЫН, В. И. ПОЛЯКОВ

МЕТОДЫ И СРЕДСТВА МОНИТОРИНГА ДЫХАНИЯ

Рассматриваются особенности применения средств компьютерной обработки информации в медицине. Представлены состав системы мониторинга и принципы диагностирования легких. Выделяются термальные методы как наиболее перспективные.

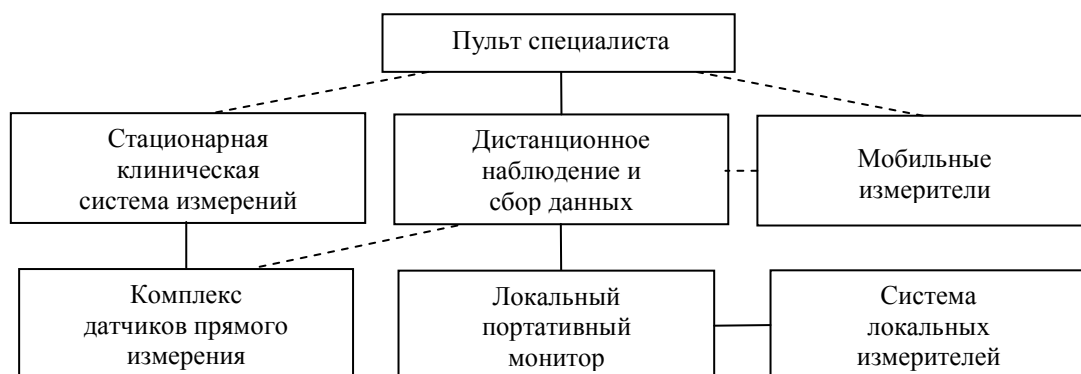
Ключевые слова: дистанционный мониторинг, диагностирование легких, спирометрия, параметры воздушного потока.

Введение. Современные средства компьютерной обработки информации активно используются в медицине. В настоящее время накоплен опыт обработки данных, автоматизации диагностики и существенно расширена электронная элементная база, что позволяет значительно повысить надежность и оперативность принятия решения (постановки диагноза).

Значительное внимание уделяется созданию систем дистанционного мониторинга на основе автономных портативных измерений, локальной предобработки и использования централизованных вычислительных средств под контролем специалиста.

Архитектура системы мониторинга. Комплекс средств мониторинга (см. рисунок) включает ряд устройств, выпускаемых фирмой ИНКАРТ (Санкт-Петербург). Такой комплекс может использоваться для контроля сердечной деятельности: при суточном мониторинге.

Подобным образом могут быть организованы наблюдения различных систем организма.



Процессор в комплексе средств мониторинга выполнен на основе микроконтроллера Cortex3m, позволяющего реализовать различные интерфейсы для поддержки периферии — USB, BlueTooth, универсальные цифровые порты. К ближайшей периферии относятся SD-память, звуковая карта, акселерометр, сенсорный экран; к дальней — 20-разрядный многоканальный АЦП и различные (до 20) датчики с аналоговыми выходами. Управление системами осуществляется в режиме реального времени, используются платформы Free RTOS или/и Android на уровне сбора и накопления данных

Измерение параметров дыхания. Одним из важных объектов анализа в медицинской диагностике является режим работы легких.

В клинической медицине стандартным способом оценки вентиляции легких является спирометрия [1], при этом измеряется объемная скорость воздушного потока. Датчики параметров воздушных потоков могут иметь разные физические принципы детектирования:

1) манометрический — напрямую измеряется давление воздушного потока при вдохе и выдохе;

2) электромагнитный — используются вихревые расходомеры, в которых жидкость, движущаяся в постоянном магнитном поле, создает ЭДС с частотой, прямо пропорциональной частоте вихреобразования;

3) ультразвуковой — вихри детектируются в нижней части потока. Из анализа амплитудно-модулированного ультразвукового сигнала определяется величина объемного расхода;

4) емкостной — регистрируется изменение емкости за счет деформации чувствительного элемента;

5) изгибных напряжений — пьезосенсор регистрирует совокупность тепловых и механических воздействий от вихревых потоков;

6) термальный — регистрируется динамика изменения температуры, т.е. вихревые колебания воздушного потока.

Большинство спирометрических приборов содержат датчики, в которых используются методы 1—5, они применяются как средства измерения при клинической аттестации, а также для проведения кратковременных медицинских проб. Однако эти приборы непригодны для длительного мониторинга состояния пациента.

К настоящему времени для возможности использования термальных методов хорошо изучены особенности возникновения вихревых потоков в дыхании и их влияние на измеряемые параметры, связанные с теплообменом.

Термальные методы диагностирования. Качество систем автоматической спирометрии определяется характеристиками датчиков параметров вихревых потоков, алгоритмами анализа сигналов и диагностики, производительностью и объемом памяти компьютеров.

В термальных методах измерений используются теплофизические расчеты [2]. В работе [1] показано, что потоки в верхних путях легких являются турбулентными (вихревыми). Свойства турбулентности для определения параметров потоков исследуются как прямыми клиническими методами измерений, так и оперативными, косвенными, в частности, с помощью акустических приборов.

Аускультация (анализ акустических явлений) легких производится в определенных точках на поверхности грудной клетки. При дыхании выявляются везикулярные и бронхиальные шумы.

Традиционно нормальные и патологические шумы представляют в виде диаграмм (спектрограмм). Нормальные звуки генерируются турбулентным потоком в воздушных путях, громкость (энергия колебаний) пропорциональна скорости потока. Аускультация отражает не только процесс генерации звука, но и процессы резонанса и поглощения между воздушными путями и датчиком звуков, что используется для диагностики по изменению спектра сигнала. Зарегистрированные акустические явления непосредственно в воздушном потоке отражают информацию о спектре и энергии процесса генерации звука. Спектр определяется ритмом работы сердца, а энергия — теплообменом воздушного потока.

Параметры турбулентных потоков, определяющие энергообменные процессы при дыхании или процессы массообмена, являются важнейшими показателями изменений состояния сердечно-сосудистой системы. С их помощью оцениваются отклонения характеристик процессов массо- и теплообмена от среднестатистических значений, принимаемых в качестве нормы в клинических исследованиях.

Внешнему мониторингу доступны интегральные массоэнергетические параметры воздушных потоков:

- температура воздушного потока на вдохе и выдохе;
- объемный (массовый) расход, или скорость воздушного потока;
- выделяемая в выдыхаемом потоке энергия, или мощность теплового потока.

Параметры связаны между собой следующим образом [2]: мощность воздушного потока P рассчитывается на основе измерения его тепловых параметров и складывается из мощ-

ности, определяемой теплосодержанием P_t и кинетической энергией потока P_k . Теплосодержание потока пропорционально объемному расходу, а кинетическая энергия пропорциональна степени объемного расхода.

Поскольку $P_t > P_k$, амплитуда пульсаций, связанная с воздействием на датчик импульсов отдельных вихрей, мала по сравнению с общей амплитудой сигнала. Для выделения этих пульсаций необходимо использовать схемотехнические и алгоритмические средства определения параметров вихревых компонентов сигнала.

Нами была рассмотрена теплофизическая модель энергетического обмена воздушных потоков вдоха и выдоха, в которой использованы свойства турбулентности [3]. На основе измерений объемного расхода воздуха, скорости потока и термодинамической температуры были вычислены важные для контроля параметры потоков в канале дыхания. Показано, что на основе этой модели могут быть построены датчики измерения и методы расчета параметров потоков, характеризующих объем легких и скорость воздушных потоков.

Заключение. Результаты исследований использованы при разработке датчика параметров вихревых потоков дыхания и алгоритмов расчета характеристик в портативных приборах функциональной диагностики.

Работа выполнена при финансовой поддержке РФФИ (грант 12-07-00376-а).

СПИСОК ЛИТЕРАТУРЫ

1. Уэст Д. Физиология дыхания. Основы. М.: Мир, 1988. 200 с.
2. Дульнев Г. Н., Парфенов В. Г., Сигалов А. В. Методы расчета теплового режима приборов. М.: Радио и связь, 1990. 312 с.
3. Кормилицын А. Ю., Ханков С. И., Скорубский В. И. Измерение параметров дыхания датчиком воздушных потоков // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 3. С. 122—129.

Сведения об авторах

- Александр Юрьевич Кормилицын** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: Alex_kkk@incart.ru
- Владимир Иванович Поляков** — канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: v_i_polyakov@mail.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

А. А. ГЕДИЧ, А. Г. ЗЫКОВ, А. В. ЛАЗДИН, В. И. ПОЛЯКОВ

ПОИСК ПРОЦЕДУР ПО ГРАФУ ПЕРЕХОДОВ ФУНКЦИОНАЛЬНОЙ ПРОГРАММЫ ПРИ ВЕРИФИКАЦИИ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Решена задача поиска процедур в программном продукте при верификации вычислительного процесса, реализованного в виде программы. Предложен алгоритм поиска начальных адресов процедур и их распространения в адресном пространстве программы.

Ключевые слова: верификация, граф переходов, функциональная программа, процедура.

Введение. Экспоненциальный рост сложности вычислительных систем и программного обеспечения требует повышенного внимания к верификации их как на этапах проектирования и разработки, так и при эксплуатации. Анализ вычислительных процессов, реализуемых в виде программ, может проводиться как по исполняемому коду, так и по графоаналитическим моделям. В настоящей работе рассматривается задача поиска процедур по графу переходов функциональной программы, восстановленному из исполнимого кода программы.

При решении задачи верификации или оценки структурной сложности программы возникает необходимость ее представления в виде графа переходов (ГП) [1]. В настоящей работе под программами понимаются функциональные программы (ФП), алгоритм формирования графа переходов для которых приведен в работе [2]. Процедура — часть кода ФП, предназначенного для многократного использования. Структура процедуры во многом совпадает со структурой программы: наличие линейных участков, ветвлений, циклов, вызовов других процедур. Следовательно, выделение процедур в отдельные функциональные блоки, которые могут быть представлены единственной вершиной графа переходов ФП, существенно упрощает анализ структуры программы. Тождественность структуры программ и процедур позволяет использовать методы оценки структурной сложности программ применительно к процедурам.

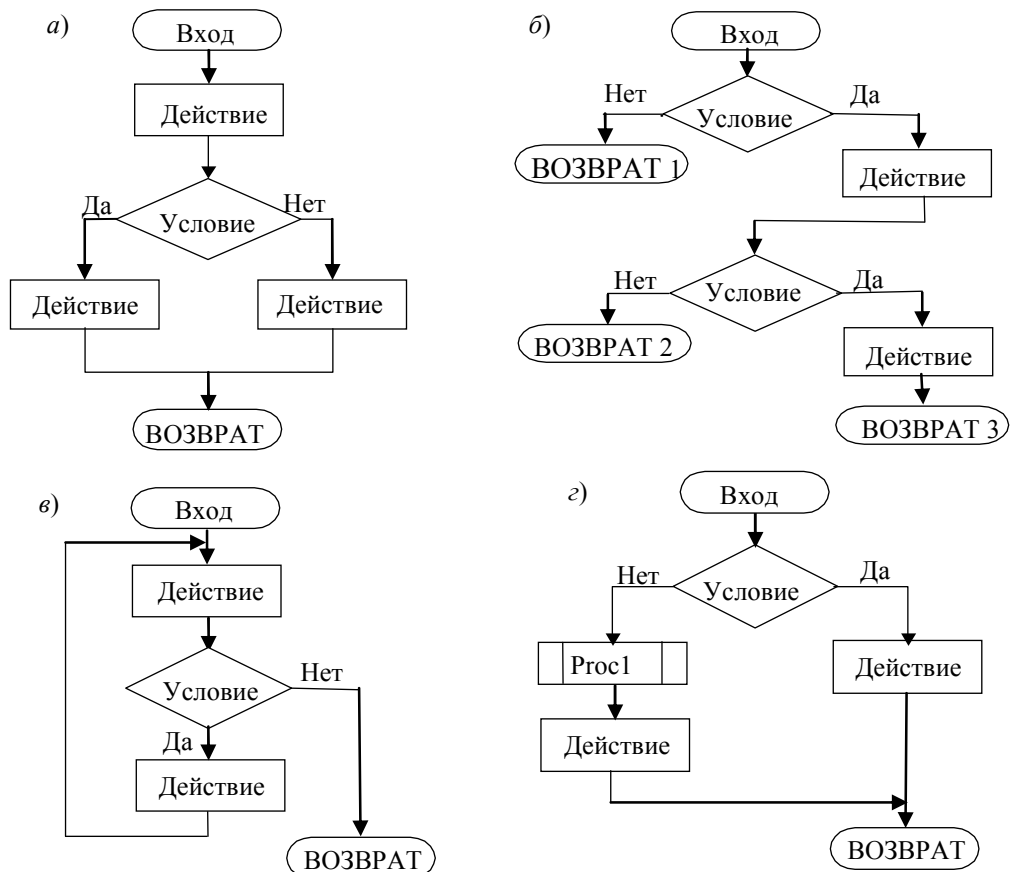
Анализ и постановка задачи. В общем случае структура процедуры заранее не известна, она зависит от назначения последней. Обращения к процедуре и возвраты из нее могут быть осуществлены посредством команд условного или безусловного переходов. При написании программ на языке Ассемблера допустимы переходы из тела процедуры к произвольным командам в теле программы (в том числе и к другим процедурам) и обращения к процедурам путем перехода (условного или безусловного) к любой помеченной команде в теле процедуры или выполнения команд вызова, так как имя процедуры рассматривается как метка, т.е. можно говорить о множестве точек входа в процедуру. Вышеперечисленное существенно затрудняет поиск произвольно оформленных процедур. Произвольное оформление процедур ухудшает структурированность программы и затрудняет выделение части кода в независимые блоки.

В отличие от Ассемблера большинство современных языков программирования (Си, Паскаль, Фортран) существенно ограничивают способы передачи управления к процедурам и выходам из них. Например, в языке Си имена меток локализуются в теле функции и к ним не допускаются обращения из других частей программы. Тем не менее, на практике структура процедур часто оптимизируется компиляторами, что ведет к появлению участков кода, называемых сопrogramмами, которые невозможно выделить в независимые блоки.

Необходимо заметить, что тело процедуры может содержать косвенные безусловные переходы, использующие таблицы адресов и в некоторых случаях — вспомогательные таблицы индексов. Так называемая switch-конструкция используется не только во многих языках высокого уровня, но и в программах на Ассемблере, она затрудняет локализацию последней команды процедуры.

В настоящей работе под процедурой будет пониматься группа команд, предназначенная для решения локальной задачи, имеющая единственную точку входа, передача управления которой может осуществляться только посредством выполнения команд вызова; выход из процедур осуществляется выполнением команд возврата. Другие способы входа и выхода из процедур запрещены. Наложённые ограничения позволяют существенно упростить поиск процедур, оформленных по правилам, принятым в языках высокого уровня, или с использованием директивы PROC в языке Ассемблера. Для процедур выполняется правило: ближайшему вызову соответствует ближний возврат, а дальнему — дальний. Команда возврата не обязательно должна быть расположена последней в последовательности машинных инструкций процедуры, последней также может быть команда передачи управления инструкции в теле процедуры или команда вызова другой процедуры, если вызываемая процедура является источником исключения или использует функции среды выполнения, завершающие работу программы. В таком случае компилятор оптимизирует эпилог процедуры. Если в коде программы находятся процедуры, не соответствующие наложенным ограничениям, они рассматриваются как часть головной программы. Для процедур, вход в которые был выполнен командой вызова, недопустимы переходы из тела процедуры.

Кроме того, необходимо учесть, что в теле процедуры может выполняться обращение к другим процедурам. Возможные структуры процедур приведены на рисунке (*а* — единственная команда возврата; *б* — несколько команд возврата; *в* — команда возврата не является последней инструкцией процедуры; *г* — вложенный вызов процедуры).



Для определения адресов перехода, используемых конструкцией `switch`, необходимо найти границы таблицы, которые чаще всего встраиваются в конец процедуры или, в некоторых случаях, в тело процедуры. Для доступа к таблице адресов переходов (ТАП) используется ассемблерная инструкция вида `JMP CS:[reg*ptr_size + table_label]`, где `reg` — регистр, содержащий индекс элемента, `ptr_size` — размер указателя (4 на 32-битной архитектуре), `table_label` — абсолютный адрес таблицы. Ближней границей (БГ) ТАП называется абсолютный адрес, близкий к `table_label`, дальней (ДГ) ТАП называется граница, противоположная БГ.

Стоит заметить, что в целях оптимизации компилятором может быть отрицательный индекс, т.е. ТАП будет увеличиваться в сторону меньших адресов. Таким образом, таблица может иметь два направления роста, задаваемых знаком индекса. Часто используется конструкция *N*-based `switch`, где *N* — число начальных индексов, запрещенных для использования. Это делает невозможным применение `table_label` в качестве начального адреса ТАП. В редких случаях компилятором может проводиться оптимизация, при которой `table_label` указывает на середину ТАП. Этот случай в работе не рассматривается.

Отметим, что, даже точно определив начальный адрес ТАП, можно точно определить конечный. Логично предположить, что конечным будет адрес последнего элемента таблицы, являющийся валидным. Однако считанное значение может быть валидным адресом, являясь на самом деле инструкцией. Не совсем верно предположение, что считываемый адрес указывает на тело анализируемой процедуры, чаще всего он указывает на часть процедуры, которая еще не получена, что делает невозможной данную проверку валидности адреса. Если адрес указывает на ту же секцию кода, в которой находится анализируемая процедура, то часто получаются неверные результаты.

В ТАП приводится неупорядоченная последовательность значений, возможно повторяющихся, представляющих адреса. Упорядочив значения адресов, можно получить зависимость, близкую к линейной.

Локализация границ ТАП производится следующим образом. Сначала считывается *N* значений в оба направления от `table_label`, где *N* — входной параметр поиска. Далее для полученных групп значений рассчитывается коэффициент корреляции Пирсона, отражающий линейность системы. Для расчета коэффициента требуются две координаты, в качестве второй вводится индекс элемента ТАП. Выбирается группа значений с наибольшим коэффициентом, а вторая отбрасывается.

Таким образом, находится направление увеличения ТАП. Далее происходит последовательное удаление значений из группы со стороны ДГ до тех пор, пока удаление этого значения приводит к снижению коэффициента линейности системы. После того как ДГ определена, аналогичным образом производится нахождение БГ ТАП.

Алгоритмы поиска процедур. Поиск процедур осуществляется в два этапа. Сначала в списке узлов ГП производится поиск всех обращений к процедурам, не совпадающие адреса вызовов помещаются в список начальных адресов процедур. Далее для каждого начального адреса выполняется поиск адреса последней команды процедуры с целью ее последующей локализации в рамках ГП ФП.

Алгоритм 1. Формирование списка начальных адресов процедур

1. Установить счетчик найденных процедур (СЧНП) в нуль. Установить указатель вершин на первый элемент в списке вершин графа.
2. Считать тип текущего узла.
3. Если это не команда вызова, перейти к п. 6.
4. Если адрес обращения к процедуре совпадает с одним из элементов списка адресов процедур, перейти к п. 6.

5. Инкрементировать СЧНП. Добавить новый элемент в список начальных адресов процедур (НАП).
6. Для текущего элемента в списке НАП установить тип вызова (дальний или ближний).
7. Если обработанный узел последний, перейти к п. 10.
8. Переместить указатель узла на следующий элемент списка узлов графа.
9. Перейти к п. 2.
10. Конец.

После завершения формирования списка начальных адресов процедур осуществляется поиск адреса последней команды процедуры.

Алгоритм 2. Выделение тела процедур

1. Установить счетчик обработанных процедур в единицу.
2. В списке вершин ГП найти узел i такой, что $AУ_i > НАП_{K+1}$ и $AУ_{i-1} < НАП_{K+1}$ ($AУ_i$ — адрес i -го узла).
3. Если найденный узел не соответствует командам возврата или переходов, то перейти к п. 7.
4. Если типы вызова и возврата не совпадают, то выдать сообщение об ошибке и перейти к п. 9, иначе — запомнить адрес узла в качестве потенциального адреса конца процедуры.
5. Если узел соответствует команде перехода и адрес перехода меньше адреса текущего рассматриваемого узла и меньше $НАП_{K+1}$, то запомнить адрес узла в качестве потенциального адреса конца процедуры, если это переход вперед, то запомнить адрес (если в ходе рассмотрения встречается несколько команд перехода вперед, запоминается самый старший адрес перехода).
6. Если в списке вершин ГП нет узлов, ссылающихся на адрес, который был выбран в качестве потенциального адреса конца процедуры, то выбрать следующий узел ГП и перейти к п. 3; иначе — считать его адресом завершения процедуры.
7. Если адрес конца процедуры меньше найденного адреса перехода вперед, выдать сообщение об ошибке и перейти к п. 9.
8. Увеличить счетчик обработанных процедур на единицу. Если остались необработанные элементы в списке НАП, перейти к п. 2.
9. Конец.

Предложенные алгоритмы реализованы на языке C++.

Заключение. В работе проанализирована проблема поиска процедур в программном продукте при верификации вычислительного процесса, реализованного в виде программы. Предложен алгоритм поиска начальных адресов процедур и их распространения в адресном пространстве программы. В развитие работы [3] представлены результаты исследований по поиску локальных переменных и аргументов процедур. Результаты работы использованы при модернизации САПР верификации вычислительных процессов, разрабатываемой на кафедре информатики и прикладной математики НИУ ИТМО [4, 5].

Работа выполнена при финансовой поддержке РФФИ (грант 12-07-00376-а).

СПИСОК ЛИТЕРАТУРЫ

1. *Луцаев В. В.* Обеспечение качества программных средств: Методы и стандарты. М.: СИНТЕГ, 2001. 380 с.
2. *Лаздин А. В., Немолочнов О. Ф.* Метод построения графа функциональной программы для решения задач верификации и тестирования // Научно-технический вестник СПб ГИТМО (ТУ). 2002. Вып. 6. С. 118—122.
3. *Гедич А. А., Зыков А. Г., Лаздин А. В.* Автоматический поиск локальных переменных и аргументов процедуры в исполняемом коде программы при верификации вычислительных процессов // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 5 (87). С. 117—122.

4. Зыков А.Г., Безруков А.В., Немолочнов О.Ф., Поляков В.И., Андронов А.В. Графоаналитические модели вычислительных процессов в САПР // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2011. № 4 (74). С. 116—120.
5. Зыков А. Г., Немолочнов О. Ф., Поляков В. И., Безруков А. В., Македонский А. А. Учебно-исследовательская САПР верификации вычислительных процессов // Тр. Конгресса по интеллектуальным системам и информационным технологиям „IS&IT'11“. М.: Физматлит, 2011. Т. 2. С. 109—112.

Сведения об авторах

- Андрей Алексеевич Гедич** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра информатики и прикладной математики;
E-mail: muzhedgehog@list.ru
- Анатолий Геннадьевич Зыков** — канд. техн. наук; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра информатики и прикладной математики; доцент;
E-mail: zykov_a_g@mail.ru
- Артур Вячеславович Лаздин** — канд. техн. наук; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра информатики и прикладной математики; доцент;
E-mail: lazdin@yandex.ru
- Владимир Иванович Поляков** — канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: v_i_polyakov@mail.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

А. А. ОЖИГАНОВ

КВАЗИАБСОЛЮТНЫЙ ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ УГЛА НА ОСНОВЕ ДВУХДОРОЖЕЧНОЙ РЕКУРСИВНОЙ НЕЛИНЕЙНОЙ КОДОВОЙ ШКАЛЫ

Предложена структура квазиабсолютного реверсивного цифрового преобразователя угла на основе двухдорожечной рекурсивной нелинейной кодовой шкалы.

Ключевые слова: нелинейная двоичная последовательность, рекурсивная кодовая шкала, считывающие элементы, цифровой преобразователь угла.

Цифровые преобразователи угла (ЦПУ), построенные по методу параллельного считывания, используются в устройствах вычислительной техники и системах управления. Их основной элемент — кодовая шкала (КШ). При построении ЦПУ используются КШ с числом информационных кодовых дорожек (КД) и считывающих элементов (СЭ), как правило, равным разрядности преобразователей.

В работе [1] предложены рекурсивные нелинейные кодовые шкалы (РНКШ) всего с двумя (информационной и служебной) КД и четырьмя СЭ. Именно на таких шкалах построены ЦПУ, рассматриваемые в настоящей статье. Разрешающая способность δ преобразователей на основе РНКШ равна разрешающей способности преобразователей с классическими КШ, маска которых выполнена в обыкновенном двоичном коде или коде Грея, т.е. $\delta=360^\circ/2^n$, где n — разрядность преобразователей. В таких ЦПУ для получения достоверной информации о положении кодируемого объекта первые $n-1$ сдвигов на один квант по шкале в одном направлении, являются подготовительными, далее устройства работают как классические преобразователи, построенные по методу считывания.

Для пояснения предлагаемого технического решения остановимся на принципах построения РНКШ. При синтезе кодирующей маски информационной дорожки РНКШ используется нелинейная последовательность двоичных символов a длиной $B=2^n$, удовлетворяющих рекуррентному соотношению

$$a_{n+j} = \bigoplus_{i=0}^{n-1} a_{i+j} h_i \oplus \prod_{i=1}^{n-1} a_{i+j}, \quad j = 0, 1, \dots, B - n - 1, \quad (1)$$

где знак \oplus означает суммирование по модулю два, а индексы при символах последовательности берутся по модулю B ; начальные значения символов $a_0 a_1 \dots a_{n-1}$ выбираются произвольно, h_i — коэффициенты, зависящие от вида примитивного полинома степени n с коэффициентами поля Галуа $GF(2)$, т. е.

$$h(x) = \sum_{i=0}^n h_i x^i, \quad (2)$$

где $h_0=h_n=1$, а $h_i=0,1$ при $0 < i < n$,

$$\prod_{i=1}^{n-1} a_{i+j} = \begin{cases} 1, & \text{если все } \bar{a}_{i+j} = 1, \\ 0 & \text{в других случаях.} \end{cases} \quad (3)$$

Первое слагаемое в (1) определяет правило получения М-последовательности, которая линейна по отношению к оператору \oplus . Второе слагаемое указывает на операцию умножения

значений $n-1$ кодовых символов, соответственно полученная последовательность символов нелинейная и содержит комбинацию из n последовательных нулей.

Правило получения символов последовательности $\mathbf{b}=101010\dots1010$ длиной $2B$, используемых для получения кодирующей маски служебной дорожки РНКШ для квазиабсолютного ЦПУ, определяется следующим образом:

$$10 \Leftrightarrow a_j, \quad (4)$$

где a_j — символы нелинейной последовательности, \Leftrightarrow — знак соответствия.

Первые информационный и служебный СЭ располагаются на одной линии считывания. Второй информационный СЭ смещается вдоль информационной дорожки шкалы против хода часовой стрелки относительно первого на n элементарных участков (квантов, δ) информационной дорожки. Вторым служебный СЭ смещается вдоль служебной дорожки шкалы против хода часовой стрелки относительно первого на $k = (2m + 1) / 2, m = 0, 1, 2, \dots$ элементарных участков ($\delta/2$) служебной дорожки. Информационные и служебные СЭ взаимодействуют соответственно с элементарными участками информационной и служебной дорожек шкалы.

Элементарные участки КД шкалы представляются одним символом двоичных последовательностей, причем единичным символам последовательностей соответствуют активные участки, а нулевым — пассивные.

На рис. 1 приведен пример трехразрядной двухдорожечной РНКШ.

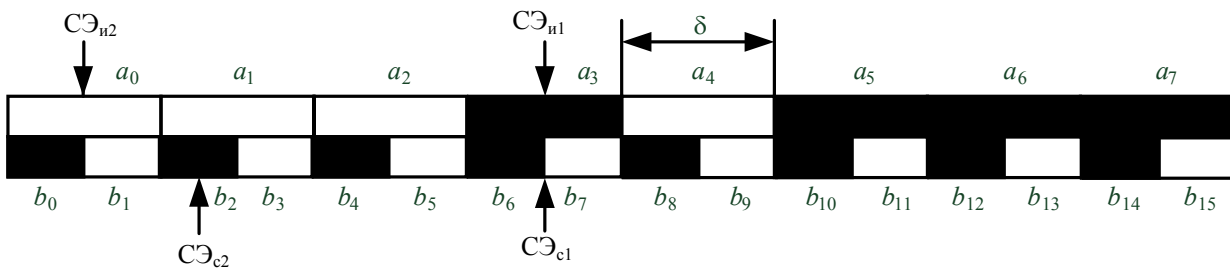


Рис. 1

Информационная дорожка шкалы выполнена в соответствии с нелинейной последовательностью $\mathbf{a} = a_0a_1a_2a_3a_4a_5a_6a_7 = 00010111$ длиной $B=2^n=2^3=8$, для построения которой использован примитивный полином $h(x) = x^3 + x + 1$, а символы a_{3+j} последовательности \mathbf{a} при начальных значениях $a_0=a_1=a_2=0$ удовлетворяют рекуррентному соотношению $a_{3+j}=a_{1+j} \oplus a_j \oplus \bar{a}_{1+j} \bar{a}_{2+j}$, $j=0, 1, \dots, 4$.

Служебная дорожка шкалы выполнена в соответствии с последовательностью $\mathbf{b} = b_0b_1\dots b_{14}b_{15} = 1010\dots1010$, полученной по правилу (4).

Первые информационный (СЭ_{и1}) и служебный (СЭ_{с1}) элементы расположены на одной линии считывания. Вторым информационный СЭ_{и2} смещен вдоль информационной дорожки шкалы против хода часовой стрелки относительно первого на 3 кванта, вторым служебный СЭ_{с2} смещен вдоль служебной дорожки шкалы против хода часовой стрелки относительно первого на $k=4,5$ элементарных участка служебной дорожки.

При вращении РНКШ против часовой стрелки информация с дорожки снимается посредством СЭ_{и1} и поступает на прямой вход трехразрядного реверсивного сдвигающего регистра, в котором после трех квантов перемещения шкалы будет записана кодовая комбинация 011, затем 111, 110 и т.д. (см. таблицу).

Последовательность кодовых комбинаций трехразрядного сдвигающего регистра

№ углового положения РНКШ	1-й разряд	2-й разряд	3-й разряд	Десятичный эквивалент кода
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	1	0	1	5
4	0	1	1	3
5	1	1	1	7
6	1	1	0	6
7	1	0	0	4

При изменении направления вращения шкалы информация с дорожки будет сниматься СЭ_{и2} и поступать на инверсный вход сдвигающего регистра. Например, если на момент изменения направления вращения шкалы в регистре была зафиксирована кодовая комбинация 000, то следующей будет 100, затем 110 и т.д. (см. таблицу).

Информация со служебных СЭ используется для выработки управляющих и тактовых импульсов, необходимых для функционирования реверсивного сдвигающего регистра.

На рис. 2 приведена функциональная электрическая схема *n*-разрядного квазиабсолютного ЦПУ на основе двухдорожечной рекурсивной нелинейной кодовой шкалы [2]. Здесь РНКШ, а также размещенные на ней СЭ выполнены в соответствии с принципами построения, рассмотренными выше. Схема включает в себя следующие элементы: пороговые элементы ПЭ₁,...,ПЭ₄, одновибраторы ОВ₁ и ОВ₂, два логических элемента „И“, четыре „ИЛИ“, два двоичных счетчика по модулю *n* (Сч₁ и Сч₂), два триггера (Т₁ и Т₂), *n*-разрядный реверсивный сдвигающий регистр (Рг), преобразователь кодов (ПК) и схема обнуления (Сх).

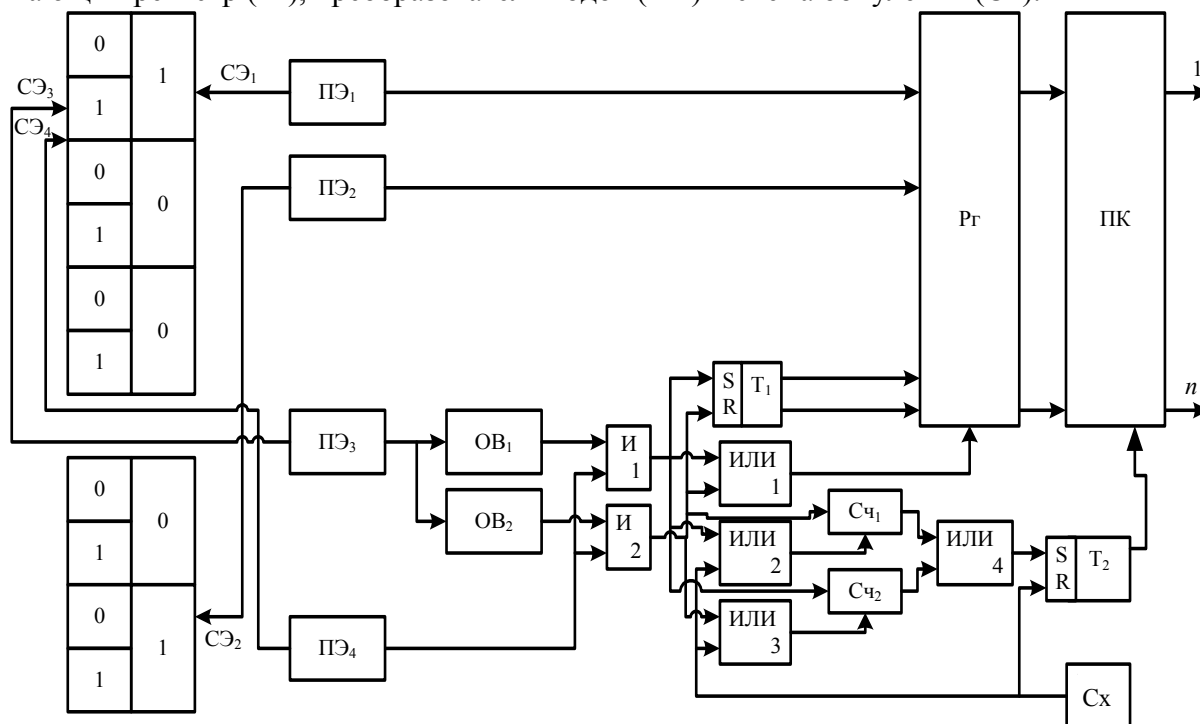


Рис. 2

Рассмотрим сначала работу тракта, используемого для выработки сигналов, определяющих направление перемещения шкалы. Тракт включает в себя служебную дорожку шкалы, СЭ₃ и СЭ₄, ПЭ₃ и ПЭ₄, ОВ₁ и ОВ₂ и логические элементы И₁ и И₂. Эпюры напряжений, снимаемые с элементов тракта, приведены на рис. 3.

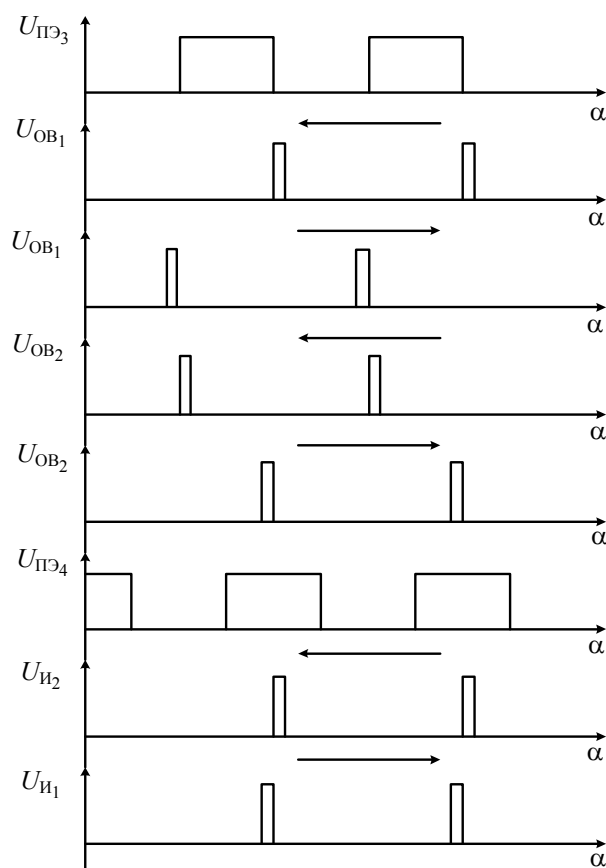


Рис. 3

При перемещении шкалы СЭ₃ и СЭ₄ могут взаимодействовать как с активными, так и с пассивными участками служебной дорожки. При взаимодействии СЭ с активными участками дорожки с ПЭ₃ и ПЭ₄ снимается сигнал, соответствующий уровню логической единицы U_1 , а при взаимодействии с пассивными — логического нуля U_0 (рис. 3, $U_{ПЭ3}$ и $U_{ПЭ4}$). В этом случае ОВ₁ и ОВ₂ вырабатывают короткие прямоугольные импульсы в моменты прихода с ПЭ₃ заднего и переднего фронтов сигнала соответственно. Импульсы с ОВ₁ и ОВ₂ соответствуют перемещению шкалы как по ходу, так и против хода часовой стрелки.

При перемещении шкалы против хода часовой стрелки сигнал U_1 с $U_{ПЭ4}$ „разрешает“ прохождение импульсов с ОВ₁ через И₁ ($U_{И1}$), а при перемещении по ходу стрелки с ОВ₂ через И₂ ($U_{И2}$).

Таким образом, импульсы с выходов И₁ и И₂ снимаются в момент прохождения СЭ₃ центров квантов информационной дорожки.

Рассмотрим работу преобразователя в целом. При включении напряжения питания импульсом с выхода схемы обнуления устанавливаются в нулевое состояние триггер разрешения (T_2) считывания информации и через элементы ИЛИ₂ и ИЛИ₃ — счетчики по модулю n Сч₁ и Сч₂.

При перемещении шкалы против хода часовой стрелки импульсы с выхода И₁ начинают поступать на счетный вход Сч₂, через ИЛИ₂ — на вход установки в нулевое состояние Сч₁, через ИЛИ₁ — на тактовый вход Рг и на установочный вход триггера управления (T_1).

Первым импульсом с выхода И₁ триггер управления устанавливается в единичное состояние, и сигнал с единичного выхода T_1 поступает на первый управляющий вход сдвигающего регистра, на второй управляющий вход которого поступает сигнал U_0 с нулевого выхода триггера. Одновременно сигналы U_1 и U_0 , снимаемые СЭ₁ с информационной дорожки, через ПЭ₁ поступают на прямой вход Рг. По импульсам, которые подаются на тактовый вход Рг, информационные сигналы последовательно справа налево заполняют ячейки его памяти.

После поступления n -го импульса с выхода I_1 на счетный вход $Sч_2$ на выходе $Sч_2$ формируется импульс, который через ИЛИ₄ установит T_2 в единичное состояние, и сигнал U_1 с единичного выхода будет подан на управляющий вход ПК. После этого с преобразователя может быть снята информация о положении кодируемого объекта.

После поступления n -го импульса на тактовый вход $Рг$ будет сформирован соответствующий код, причем шкала переместится на n квантов.

При дальнейшем перемещении шкалы в том же направлении каждый поступающий импульс сдвигает код на один разряд влево, при этом на прямой вход регистра поступают информационные сигналы.

При поступлении на тактовый вход регистра $B=2^n$ импульсов, начиная с n -го, в $Рг$ будет последовательно зафиксировано B различных состояний, что соответствует полному обороту шкалы. ПК после поступления на его управляющий вход разрешающего сигнала U_1 преобразует B различных n -разрядных кодовых комбинаций $Рг$ в выходной n -разрядный двоичный код.

При вращении шкалы по часовой стрелке импульсы с выхода I_2 поступают на счетный вход $Sч_1$, через ИЛИ₃ — на вход установки в нулевое состояние $Sч_2$, через ИЛИ₁ на тактовый вход $Рг$ и обнуляющий вход T_1 .

Первым импульсом с выхода I_2 триггер управления устанавливается в нулевое состояние, и сигнал U_1 с нулевого выхода T_1 поступает на второй управляющий вход $Рг$, на первый вход которого поступает сигнал U_0 с единичного выхода T_1 .

Одновременно при перемещении шкалы в том же направлении сигналы, снимаемые $SЭ_2$ с информационной дорожки через ПЭ₂, поступают на реверсивный вход $Рг$.

По импульсам, которые подаются на тактовый вход $Рг$, информационные сигналы слева направо заполняют ячейки памяти. После поступления n -го импульса на счетный вход $Sч_1$ на его выходе формируется импульс, который через ИЛИ₄ устанавливает T_2 в единичное состояние, и сигнал U_1 с единичного выхода поступает на управляющий вход ПК. После этого с ПК может быть осуществлен съем информации о положении кодируемого объекта.

После поступления n -го импульса на тактовый вход $Рг$ в нем будет зафиксирован соответствующий код, причем шкала переместится при этом на n квантов.

При дальнейшем перемещении шкалы по ходу часовой стрелки каждый поступающий импульс сдвигает код на один разряд вправо, на реверсивный вход регистра поступают информационные сигналы.

Рассмотрим реверсивный режим работы преобразователя. При сдвиге шкалы более чем на n квантов по ходу или против хода часовой стрелки T_2 устанавливается в единичное состояние, и сигнал U_1 с его единичного выхода поступает на управляющий вход ПК, а в регистр поступает достоверная информация о положении кодируемого объекта.

При изменении направления перемещения шкалы первым импульсом с выхода I_2 или I_1 происходит переключение T_1 . Сигналом U_1 с единичного или нулевого выхода T_1 элементы $SЭ_1$ или $SЭ_2$ подключаются соответственно к прямому или реверсивному входу $Рг$. Другими словами, сигналы с выходов T_1 используются для выбора режима работы $Рг$ при изменении направления перемещения шкалы.

Два дополнительных младших разряда ЦПУ могут быть получены со служебной КД шкалы посредством первого и второго служебных СЭ. Причем разряд $n+1$ снимается непосредственно с первого (после порогового элемента), а разряд $n+2$ (самый младший) с выхода дополнительного двухвходового сумматора по модулю два (на рис. 2 не показан), на вход которого (после пороговых элементов) поступают сигналы с первого и второго служебных СЭ.

Необходимо отметить, что квазиабсолютные ЦПУ с РНКШ не являются преобразователями, построенными по методу непосредственного считывания, а занимают промежуточное положение между ними и устройствами, построенными по методу последовательного счета.

Отметим также, что рассмотренные принципы построения квазиабсолютных ЦПУ могут быть использованы при проектировании цифровых преобразователей линейных перемещений.

Область применения таких преобразователей ограничена системами, в которых кратковременная потеря значения кода, например, после аварийного выключения источника питания, прохождения помехи или превышения допустимой скорости вращения вала, не является критической.

СПИСОК ЛИТЕРАТУРЫ

1. Ожиганов А. А., Прибыткин П. А. Использование нелинейных последовательностей при построении двухдорожечных кодовых шкал для преобразователей угловых перемещений // Изв. вузов. Приборостроение. 2010. Т. 53, № 7. С. 39—41.
2. А.С. № 1619398 (СССР). Преобразователь угол-код / И. В. Месъкин, Л. Н. Мальцев, Ю. А. Сторожук, А. А. Ожиганов // Б.И. 1991. № 1.

Сведения об авторе

Александр Аркадьевич Ожиганов — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: ojiganov@mail.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 621.3.085

К. М. РОСТОВСКИЙ, А. А. ОЖИГАНОВ

КОДОВЫЕ ШКАЛЫ НА ОСНОВЕ ИНВЕРСНО-СОПРЯЖЕННЫХ ДВОИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Предложен новый тип кодовых шкал на основе инверсно-сопряженных двоичных последовательностей.

Ключевые слова: цифровой преобразователь угла, кодовая шкала, считывающие элементы, инверсно-сопряженная двоичная последовательность.

В работе [1] рассмотрены классические кодовые шкалы (КШ) цифровых преобразователей угла (ЦПУ), кодовая маска (КМ) которых выполнена в обыкновенном двоичном коде или коде Грея. Трудоемкость изготовления таких КШ зависит в основном от сложности их КМ, которая, в свою очередь, определяется числом наносимых границ смены кодового рисунка T и с увеличением разрядности шкал возрастает.

Для шкал, КМ которых выполнена в обыкновенном двоичном коде (ОДК), число границ определяется как $T_{\text{ОДК}} = 2^{n+1} - 2$, а для шкал, КМ которых выполнена в коде Грея — как $T_{\text{Гр}} = 2^n$, где n — разрядность шкалы, число кодовых дорожек (КД) и считывающих элементов (СЭ) [2]. Разрешающая способность таких шкал $\delta = 360^\circ / 2^n$.

С помощью большинства методов считывания информации могут быть реализованы рекурсивные кодовые шкалы (РКШ) на основе нелинейных двоичных последовательностей (НП), которые имеют одну информационную КД с расположенными вдоль нее n СЭ с шагом,

равным одному элементарному участку (кванту) шкалы, и также обеспечивают $\delta=360^\circ/2^n$ [3]. Для РКШ $T_{\text{РКШ}} = 2^{n-1}$, где n — разрядность шкалы и число СЭ.

ЦПУ, построенные по методу абсолютного отсчета, могут быть реализованы при различных физических принципах съема информации в широком диапазоне информационной емкости. Сравнив КШ по числу наносимых границ смены КМ [1], можно сделать вывод о том, что РКШ являются наиболее технологичными. Однако разместить СЭ вдоль КД такой шкалы в основном можно только с шагом в один квант, что ограничивает габариты РКШ при заданной разрешающей способности.

В настоящей работе предлагается выполнять единственную (как и в РКШ) информационную КД шкалы в соответствии с символами инверсно-сопряженной двоичной последовательности (ИСП), а сами шкалы называть инверсно-сопряженными кодовыми шкалами (ИСКШ). При таком подходе можно уменьшить трудоемкость изготовления КШ за счет меньшего значения T , а также размещения СЭ вдоль информационной КД с постоянным, отличным от единичного, угловым шагом δ .

Инверсно-сопряженная двоичная последовательность имеет длину периода $N = 2^n$; имеет одинаковое число нулей и единиц, равное 2^{n-1} ; состоит из двух равных частей: вторая является абсолютной инверсией первой и сопряжена (неразрывно связана) с первой половиной, образуя совместно полную последовательность.

Авторами на основе комбинаторного алгоритма была разработана программа в математическом пакете MatLab, позволяющая генерировать ИСП с длиной периода 16, 32, 64 и 128 двоичных символов.

В табл. 1 приведены инверсно-сопряженные и нелинейные двоичные последовательности, равные по длине периода.

Таблица 1

N	ИСП	НП
$2^{n=4} = 16$	0010000011011111	0000100110101111
$2^{n=5} = 32$	00000110001000001111100111011111	00000100101100111110001101110101
$2^{n=6} = 64$	00011000000000001110010000110000 11100111111111110001101111001111	00000010000110001010011110100011 10010010110111011001101010111111
$2^{n=7} = 128$	00000000110001100010001101110100 111001100000000000110000000000 111111100111001101110010001011 000110011111111111001111111111	00000001000001100001010001111001 00010110011101010011111010000111 00010010011011010110111101100011 01001011101110011001010101111111

С помощью табл. 1 сравним ИСКШ и РКШ по числу T с использованием для построения информационной КД последовательностей (табл. 2, здесь $\Delta = \frac{(T_{\text{РКШ}} - T_{\text{ИСКШ}}) \cdot 100\%}{T_{\text{РКШ}}}$).

При этом значение T будет равно суммарному числу переходов в последовательностях из 0 в 1 и наоборот.

Таблица 2

N	$T_{\text{РКШ}}$	$T_{\text{ИСКШ}}$	$\Delta, \%$
$2^{n=4} = 16$	$2^{n-1} = 2^3 = 8$	6	25
$2^{n=5} = 32$	$2^{n-1} = 2^4 = 16$	10	37,5
$2^{n=6} = 64$	$2^{n-1} = 2^5 = 32$	18	43,75
$2^{n=7} = 128$	$2^{n-1} = 2^6 = 64$	38	40,625

Поясним принцип построения ИСКШ, для простоты ограничившись четырьмя разрядами преобразования (на рисунке приведена линейная развертка шкалы).



Информационная КД шкалы построена в соответствии с символами инверсно-сопряженной нелинейной двоичной последовательности 0010000011011111 с длиной периода 16. Последовательность состоит из двух равных частей, вторая является абсолютной инверсией первой половины. Последовательность должна быть нанесена на шкалу в виде пассивных (нули последовательности — светлые участки шкалы) и активных (единицы последовательности — темные) участков (квантов) информационной КД, например, по ходу часовой стрелки. Последовательность с $N=16$ определяет число квантов информационной КД шкалы, которое в данном примере равно 16. Отсюда $\delta=360^\circ/16=22,5^\circ$. В примере размещение СЭ₁, СЭ₂, СЭ₃ и СЭ₄ вдоль информационной КД осуществляется с шагом, равным величине двух квантов δ по ходу часовой стрелки.

Фиксируя с помощью СЭ₁, СЭ₂, СЭ₃ и СЭ₄ последовательно кодовую комбинацию, при перемещении ИСКШ циклически на один квант ($\delta=22,5^\circ$) информационной КД, например, против хода часовой стрелки, получаем 16 различных четырехразрядных кодовых комбинаций, которые соответствуют 16 угловым положениям шкалы (табл. 3).

Таблица 3

№ пол. ИСКШ	СЭ ₁	СЭ ₂	СЭ ₃	СЭ ₄	ОДК
0	0	1	0	0	4
1	0	0	0	0	0
2	1	0	0	1	9
3	0	0	0	1	1
4	0	0	1	0	2
5	0	0	1	1	3
6	0	1	0	1	5
7	0	1	1	1	7
8	1	0	1	1	11
9	1	1	1	1	15
10	0	1	1	0	6
11	1	1	1	0	14
12	1	1	0	1	13
13	1	1	0	0	12
14	1	0	1	0	10
15	1	0	0	0	8

По принципу, рассмотренному выше, могут быть построены пяти-, шести- и семиразрядные ИСКШ, при шаге размещения СЭ соответственно в два, десять и шесть квантов шкалы δ .

Снизить трудоемкость изготовления предложенных ИСКШ возможно при контактном, емкостном или электромагнитном методе съема информации.

Также отметим, что ИСКШ могут использоваться при разработке высокоразрядных псевдорегулярных кодовых шкал, принципы построения которых подробно рассмотрены в работах [4—9].

СПИСОК ЛИТЕРАТУРЫ

1. Домрачев В. Г., Мейко Б. С. Цифровые преобразователи угла: принципы построения, теория точности, методы контроля. М.: Энергоатомиздат, 1984. 328 с.
2. Ожиганов А. А. Анализ кодовых шкал преобразователей угла // Изв. вузов. Приборостроение. 1996. Т. 39, № 4. С. 32—35.
3. Азов А. К., Ожиганов А. А., Тарасюк М. В. Рекурсивные кодовые шкалы // Информационные технологии. 1998. № 6. С. 39—43.
4. Ожиганов А. А., Прибыткин П. А. Кодовые шкалы на основе композиции нелинейных рекуррентных последовательностей // Тр. Нижегородского гос. тех. ун-та им. Р. Е. Алексеева. 2010. № 4(83). С. 309—316.

5. Ожиганов А. А., Прибыткин П. А. Псевдорегулярные кодовые шкалы для цифровых преобразователей угла // Научно-технический вестник СПбГУ ИТМО. 2011. № 1(71). С. 67—72.
6. Пат. 2434323 РФ, МПК H03M 1/22, G06F 5/00. Рекурсивная кодовая шкала / В. А. Шубарев, А. А. Ожиганов, П. А. Прибыткин, В. В. Павлов. Заявл. 16.08.2010; опубл. 20.11.2011. Б.И. 32.
7. Пат. 2444126 РФ, МПК H03M 1/22. Рекурсивная кодовая шкала / В. А. Шубарев, А. А. Ожиганов, П. А. Прибыткин, В. В. Павлов. Заявл. 22.11.2010; опубл. 27.02.2012. Б.И. 6.
8. Пат. 2446557 РФ, МПК H03M 1/22. Рекурсивная кодовая шкала / В. А. Шубарев, А. А. Ожиганов, П. А. Прибыткин, В. В. Павлов. Заявл. 17.03.2011; опубл. 27.03.2012. Б.И. 9.
9. Пат. 2450437 РФ, МПК H03M 1/22. Рекурсивная кодовая шкала / В. А. Шубарев, А. А. Ожиганов, П. А. Прибыткин, В. В. Павлов. Заявл. 29.04.2011; опубл. 10.05.2012. Б.И. 13.

Сведения об авторах

- Кирилл Михайлович Ростовский** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра безопасности технических систем; E-mail: kmrost@ya.ru
- Александр Аркадьевич Ожиганов** — д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: ojiganov@mail.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

SUMMARY

P. 7—12.

EFFICIENCY OF FREQUENCY DOMAIN ALGORITHM OF DIGITAL IMAGE WATERMARKING BASED ON DISCRETE HADAMARD TRANSFORM

Peculiarities of discrete Hadamard transform application to digital image watermarking are considered. The tolerance of the digital marking algorithm Elham to image size and compression is studied for JPEG and JPEG2000 formats.

Keywords: steganography, digital marking, Hadamard transform, digital images, JPEG compression.

Data on authors

- Vladimir A. Batura* — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: batu-vladimir@yandex.ru
- Alexander Yu. Tropchenko* — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: tau@d1.ifmo.ru

P. 12—15.

A CONSISTENT MODEL OF MANDATORY ACCESS CONTROL

A much used confidential leveling mandatory access control model is studied. Inconsistency and principle disadvantages of the model are revealed which do not allow for creation of a safe system based on it. New integrity and accessibility model, as well as consistent mandatory access control model are developed and justified; application of the latter model is shown to prevent confidential information leakage and to solve the problems of information integrity and accessibility.

Keywords: information security, access control, access rights, mandate, information leveling, confidential levels.

Data on authors

- Konstantin A. Shcheglov* — Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: schegl_70@mail.ru
- Andrey Yu. Shcheglov* — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: info@npp-itb.spb.ru

P. 15—20.

A METHOD OF PROTECTION AGAINST DRIVE-BY DOWNLOAD ATTACKS

Most actual types of drive-by download attacks using harmful script programs are considered. Existing methods of protection against similar attacks and corresponding security features are analyzed. A new method of protection against malicious software is proposed. Implementation of this method is demonstrated.

Keywords: malware, script, information security, drive-by attack, antivirus software.

Data on authors

- Tatiana A. Markina* — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: tmark812@mail.ru
- Andrey Yu. Shcheglov* — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: info@npp-itb.spb.ru

P. 20—23.

WAYS FOR IMPROVING RECOGNITION ROBUSTNESS IN MULTIMODAL BIOMETRIC SYSTEMS

Ways and means for increasing robustness of person recognition by multimodal biometric systems are considered.

Keywords: person recognition, multimodal recognition system, modality fusion, robustness.

Data on author

- Andrey A. Tropchenko* — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: zayka_98rus@mail.ru

P. 23—26.

TRANSFORMATION OF A NATURAL LANGUAGE TO RDF FORMAT USING SEMANTIC ANALYZERS OF TEXTUAL INFORMATION

The problem of automated transformation of a text in a natural language (in Russian) into RDF format using semantic analysis methods is solved. Major stages of this transformation and software modules required for their implementation are considered in detail.

Keywords: text, natural language, RDF, semantic analysis, sentence tokenizer, thesaurus, automatic text processing, vocabulary, Jena.

Data on authors

- Igor V. Kalinin* — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: kalinin1989@mail.ru
- Sergey V. Klimenkov* — St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; Assistant Lecturer;
E-mail: Serge.Klimenkov@tune-it.ru
- Anastasia E. Kharitonova* — St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; Assistant Lecturer;
E-mail: Anastassia.Kharitonova@Elcom.SPb.ru
- Evgeny A. Tsopa* — St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; Assistant Lecturer;
E-mail: Evgenij.Tsopa@Elcom.SPb.ru

P. 27—29.

SERVICE QUALITY PARAMETERS OF ELECTRONIC INDUSTRY WEB SERVICES

Development of service-oriented approach in automated design system is considered. Principles of selection and determination of service quality parameters of web services to support electronic devices life cycle are formulated.

Keywords: CAD, electronic device, life cycle, web services.

Data on authors

Andrey M. Dergachev — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: dam600@mail.ru

Alexander A. Dergachev — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: born4fun@mail.ru

P. 30—35.

DESIGN OF SYSTEMS WITH PRIORITIES

The problems of design of queuing system with priorities under restrictions on the average residence time of the query are considered. During the design process, queuing discipline with mixed priorities and the facility performance are determined as to minimize the overall system cost.

Keywords: system with priorities, device performance, service discipline, mixed priorities, delay time, cost of system.

Data on author

Taufik I. Aliev — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; Head of the Department;
E-mail: aliev@d1.ifmo.ru

P. 35—39.

A METHOD FOR CALCULATION OF CHARACTERISTICS OF CLOSED DETERMINISTIC MODELS OF MULTI-SERVICE COMPUTER NETWORKS

A method for evaluation of characteristics of closed-type network models of multi-service computer networks with determined service time is proposed. The assumption of determined service time makes it possible to account for actual packet size distribution in multi-service computer networks.

Keywords: multi-service computer networks, deterministic service time, closed queuing networks.

Data on author

Ludmila A. Muravyeva-Vitkovskaya — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: muravyeva-vitkovskaya@yandex.ru

P. 39—41.

WAITING TIME IN FIFO-BASED MULTI-CLASS QUEUING SYSTEMS

Peculiarities of the FIFO (FCFS) rule are discussed as applied to multi-class queuing systems. Surprising results of simulations are obtained for GI2/GI2/1 for two classes of transactions: mean waiting time is different for the two classes. A quantitative estimation of the lower bound of mean waiting time for the transactions of a flow that has lesser utilization factor is proposed.

Keywords: queuing systems, FIFO, FCFS, mean waiting time.

Data on author

Vladimir V. Sosnin — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: vsosnin@mail.ru

P. 41—45.

OPTIMIZATION OF QUERIES REDISTRIBUTION IN CLUSTERS AT CHANGING ACTIVITY OF THE QUERIES SOURCES

The problem of dynamic optimization of redistribution of queries through a network in a public cluster is formulated and solved. Optimization is performed to minimize the average time of queries stay, and to account for delays in display of number of the active knots which generate the queries.

Keywords: fault tolerance, distribution of queries, cluster, optimization, adaptation.

Data on authors

Vladimir A. Bogatyrev — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: Vladimir.bogatyrev@gmail.com

Anatoly V. Bogatyrev — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: Vladimir.bogatyrev@gmail.com

Stanislav V. Bogatyrev — St. Petersburg State University of Aerospace Instrumentation, Department of Information Systems Security; Junior Scientist; E-mail: Vladimir.bogatyrev@gmail.com

P. 46—48.

ESTIMATION OF RELIABILITY OF EXECUTION OF REAL-TIME QUERIES

An approach to the problem of reliability estimation for computer system of cluster architecture is proposed. The approach enables an account for requirement of timely execution of critical queries.

Keywords: reliability, real time, computer system.

Data on authors

Vladimir A. Bogatyrev — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: Vladimir.bogatyrev@gmail.com

Anatoly V. Bogatyrev — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: Vladimir.bogatyrev@gmail.com

Stanislav V. Bogatyrev — St. Petersburg State University of Aerospace Instrumentation, Department of Information Systems Security; Junior Scientist; E-mail: Vladimir.bogatyrev@gmail.com

P. 49—52.

RECONFIGURABLE EMBEDDED SYSTEMS AND SYSTEM-ON-CHIP

The possibilities of improvement of technologies of embedded systems and system-on-chip design are considered. Peculiarities of reconfigurable architecture of specialized computer systems are analyzed. A promising approach to reconfigurable system design is proposed.

Keywords: embedded system, reconfigurable system, computer architecture, design process, high-level design.

Data on author

Alexey E. Platunov — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: platunov@lmt.ifmo.ru

P. 53—57.

TRANSACTION-LEVEL REAL-TIME CONSTRAINTS MONITOR FOR SYSTEM ON CHIP

A method of the formal description of internal timing restrictions for computing systems at transaction level is proposed. The method is focused on reduction of hardware complexity of embedded monitoring and diagnostics tools. Based on the proposed approach, the timing restrictions hardware monitor for systems on chip (SoC) with bus-topology is designed, and results of its experimental implementation are presented.

Keywords: SoC, real-time constraints, constraints monitor, RTL, TLM, SVA, PSL, LTL, CTL.

Data on authors

Alexander A. Antonov — Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: antonov.alex.alex@gmail.com

Sergey V. Bykovsky — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: bsv.serg@gmail.com

Pavel V. Kustarev — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: kustarev@lmt.ifmo.ru

P. 57—60.

APPLICATION OF MULTIVALUED LOGIC IN FUNCTIONAL CIRCUITS DEVELOPMENT

Applications of multivalued logic to the problems related to functional analysis of digital circuits including functional and temporal simulation and testing are reviewed. Peculiarities of the use of Boolean algebra in evaluation of proper subspace are considered.

Keywords: simulation of digital circuits, temporal modeling, circuit testing, ternary testing.

Data on authors

Vladimir I. Polyakov — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: v_i_polyakov@mail.ru

Vladimir I. Skorubsky — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology; E-mail: vlis@km.ru

P. 61—63.

METHODS AND MEANS FOR MONITORING OF RESPIRATION

Peculiarities of computer information processing application in medicine are considered. Structure of respiration monitoring system and the principles of lungs diagnostics are described. The focus is made on thermal methods supposed to be the most promising.

Keywords: remote monitoring, lungs diagnostics, spirometry, airflow parameters.

Data on authors

- Alexander Yu. Kormilitsyn* — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: Alex_kkk@incart.ru
- Vladimir I. Polyakov* — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: v_i_polyakov@mail.ru

P. 64—68.

SEARCH FOR PROCEDURE WITH THE USE OF FUNCTIONAL PROGRAM TRANSITION GRAPH TO VERIFY COMPUTATIONAL PROCESSES

The problem of search of a procedure in software product to verify computational processes is analyzed and solved. An algorithm of search for a procedure initial address and its propagation in the address space of the application is proposed.

Keywords: verification, transition graph, functional application, procedure.

Data on authors

- Andrey A. Gedich* — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Informatics and Applied Mathematics;
E-mail: muzhedgehog@list.ru
- Anatoly G. Zykov* — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Informatics and Applied Mathematics;
E-mail: zykov_a_g@mail.ru
- Arthur V. Lazdin* — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Informatics and Applied Mathematics;
E-mail: lazdin@yandex.ru
- Vladimir I. Polyakov* — Cand. Techn. Sci.; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: v_i_polyakov@mail.ru

P. 69—74.

QUASI-ABSOLUTE DIGITIZER OF CORNER ON THE BASE OF DUAL-TRACK RECURSIVE NONLINEAR CODE SCALE

A structure of quasi-absolute reversible digitizer of corner based on a dual-track recursive nonlinear code scale is proposed.

Keywords: nonlinear binary sequence, recursive code scale, reading-out elements, corner digitizer.

Data on author

- Alexander A. Ozhiganov* — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: ojiganov@mail.ifmo.ru

P. 74—77.

CODE SCALES ON THE BASE OF INVERSELY CONJUGATED BINARY SEQUENCES

A new type of code scale on the base of inversely conjugated binary sequences is proposed.

Keywords: digital angle converter, code scale, reading elements, inversely conjugated binary sequence.

Data on authors

- Kirill M. Rostovsky* — Post-Graduate Student; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Technical Systems Security;
E-mail: kmrost@ya.ru
- Alexander A. Ozhiganov* — Dr. Techn. Sci., Professor; St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Department of Computer Technology;
E-mail: ojiganov@mail.ifmo.ru