

Т. Н. ЧУРОВ, Д. А. НАСОНОВ, Б. Я. ШТЕЙНБЕРГ, А. В. БУХАНОВСКИЙ

## ОСОБЕННОСТИ ВЕРОЯТНОСТНОГО АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ И НАДЕЖНОСТИ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Предложена вероятностная модель функционирования среды облачных вычислений; рассмотрены способы оценки производительности и надежности предоставления сервисов с ее использованием.

**Ключевые слова:** облачные вычисления, сервисы, распределенная среда, производительность, время наработки на отказ.

**Введение.** Концепция облачных вычислений предполагает, что пользователю предоставляются вычислительные ресурсы и сервисы с гарантированным уровнем качества, априори обеспеченным провайдером. Мера качества определяется бизнес-моделью предоставления сервиса. В частности, в рамках модели SaaS (Software as a Service) она связана как с надежностью функционирования, так и с производительностью (временем предоставления сервиса). Особенность определения этих характеристик состоит в том, что среда облачных вычислений представляет собой распределенный программно-аппаратный комплекс, включающий в себя управляющую инфраструктуру среды облачных вычислений, вычислительные ресурсы, а также пакеты прикладных программ, на основе которых формируются сервисы. По сравнению с нормативами автоматизированных систем для традиционных видов [1], оценка надежности сред облачных вычислений требует учета таких факторов, как распределенность, иерархичность и многовариантность использования. Как следствие, применение аппарата оценки показателей надежности программного обеспечения [2—4] требует определенной модификации в соответствии с принципами работы среды облачных вычислений как системы массового обслуживания.

В настоящей работе рассматриваются вероятностная модель функционирования среды проблемно-ориентированных облачных вычислений, связанные с ней показатели производительности прикладных сервисов и среды в целом, а также интегральный подход к оценке временных характеристик надежности.

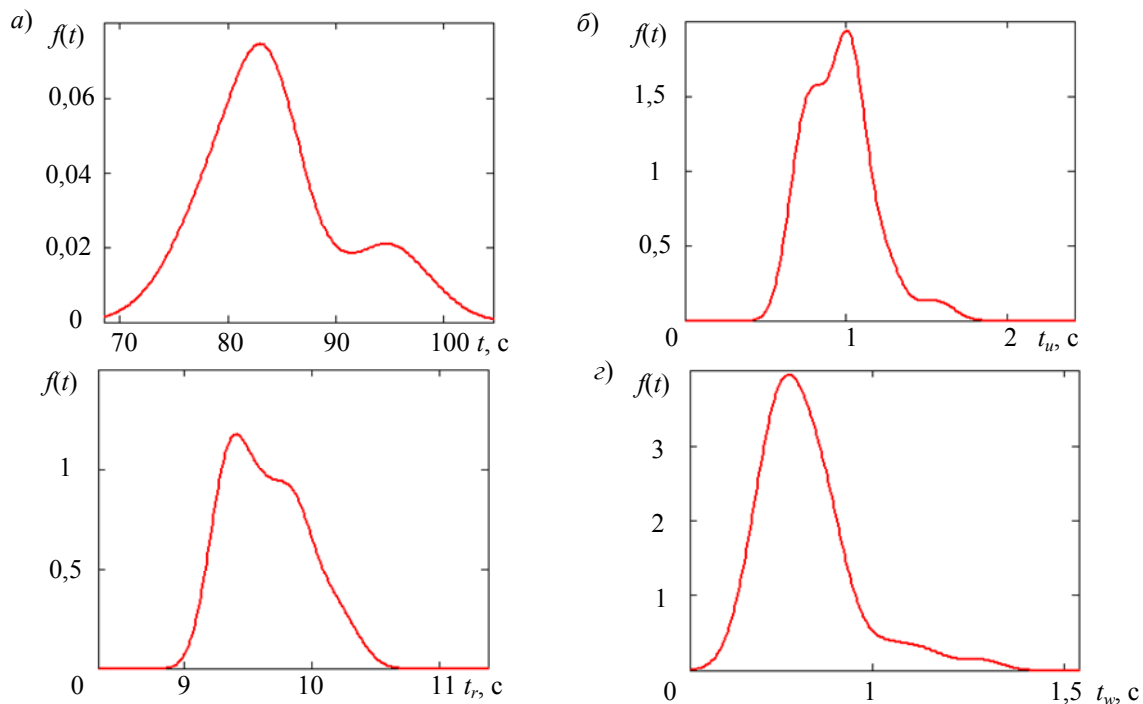
**Вероятностная модель среды облачных вычислений.** Основой для функционирования среды является облако вычислительных ресурсов  $\Omega = \{\Omega_i\}_{i=1}^n$ ; где  $\Omega_i$  — физические ресурсы — отдельные вычислительные системы (серверы приложений, хранилища данных и пр.). Каждый вычислительный ресурс описывается кортежем, определяющим основные его технические характеристики:

$$\Omega_i = \langle B, (q, p, \Pi)_B, (m, \mu), (d, D), \mathfrak{F}_t \rangle_i, \quad (1)$$

здесь  $B$  — элемент из множества классов вычислительных архитектур (SMP, MPP, GPGPU, FPGA, ...), определяющих свойства, отражающие целесообразность запуска тех или иных пакетов. Производительность каждого ресурса характеризуется триплетом  $(q, p, \Pi)_B$ , где  $q$  — количество ядер,  $p$  — количество процессоров,  $\Pi$  — количество узлов; марка  $B$  определяет их иерархию в конкретной архитектуре. Оперативная память описывается набором  $(m, \mu)$  объемных характеристик памяти на узле  $m$  и в целом в системе  $\mu$  (могут быть пересчитаны через  $p, q, \Pi$ ). Дисковая память характеризуется аналогичными по смыслу величинами  $(d, D)$ .

Многомерное вероятностное распределение  $\mathfrak{Z}_l(T)$ , где  $T = \{t_1, \dots, t_z\}$ , описывает составляющие случайной изменчивости временных характеристик накладных расходов работы с данной вычислительной системой как с физическим ресурсом  $\Omega_i$  в распределенной среде. В простейшем случае они могут включать в себя время  $t_w(\Xi)$  на передачу данных  $\Xi$  на физический ресурс  $\Omega_i$  и накладные расходы  $t_r(\Omega)$  системы управления вычислительным ресурсом  $\Omega_i$ . Такая высокоуровневая детализация позволяет использовать модель (1) как для „настоящих“ физических ресурсов (серверов и суперкомпьютеров), так и для виртуальных систем (например, Грид-инфраструктур), доступных в „облаке“. В случае виртуальных систем значение  $t_r$  может быть представлено в форме суммы нескольких составляющих накладных расходов, связанных с работой отдельных управляющих сервисов Грид.

В качестве иллюстрации на рисунке приведены ядерные оценки маргинальных распределений времени  $f(t)$  накладных расходов, возникающих при выполнении заданий в среде облачных вычислений HPC-NASIS II [5], использующей ресурсы Грид Национальной нанотехнологической сети [6]. На рисунке, *a* показана плотность распределения времени выполнения задания в Грид; видно, что при среднем времени выполнения около 85 с разброс может составлять от 70 до 110 с, что характеризуется латентностью самой Грид-среды. Из этого следует, что составляющая  $t_r(\Omega)$  может достигать 40 с. На рисунке, *б* представлена плотность распределения нахождения задания в очереди, *в* — суммарные издержки среды облачных вычислений на основе платформы CLAVIRE (CLOUD Applications VIRTUAL Environment). Видно, что их разброс невелик; в среднем они в четыре раза меньше, чем издержки, связанные с Грид. На рисунке, *г* приведена плотность распределения времени передачи данных; для рассмотренного примера оно незначительно. В целом из рисунка видно, что латентность Грид является существенной по сравнению с остальными накладными расходами среды облачных вычислений, что требует ее отдельного учета в распределении  $\mathfrak{Z}_l(T)$ .



При предоставлении пользователю сервиса в процессе его запуска в облаке происходит выделение (резервирование) виртуального ресурса  $\Omega_i^*$  на физическом ресурсе  $\Omega_i$ , при этом оставшийся ресурс  $\tilde{\Omega}_i = \Omega_i \setminus \Omega_i^*$  может быть задействован для другой задачи. Эта операция

позволяет рассматривать  $\tilde{\Omega}_i$  как физический ресурс, однако — с „ухудшенными“ (за счет выделения под заданную задачу) характеристиками по сравнению с  $\Omega_i$ . Следует отметить, что изменение касается не только числовых характеристик вычислителей (количество процессоров, узлов, ядер) и памяти, но и свойств распределения  $\mathfrak{Z}_i(T)$ : при увеличении количества независимых приложений объем накладных расходов может возрастать (хотя бы в силу загрузки управляющего узла).

**Вероятностная модель производительности сервисов в рамках SaaS.** Модель (1) описывает ресурсную составляющую облачной среды в рамках модели IaaS (Infrastructure as a Service). Для описания процесса запуска на ней прикладных пакетов следует ввести модель сервиса

$$\Sigma_j = \langle \Xi_j, S_j(\Xi_j, P, \Omega_i), \kappa_{ij}, T_{0j} \rangle, \quad (2)$$

который ассоциирован с каждым из физических ресурсов  $\Omega_i$  в (1). Здесь  $\Xi_j$  — характеристики (объем, структура) входных и выходных данных сервиса,  $S_j(\Xi_j, P, \Omega_i)$  — формализация „внутреннего“ параллельного ускорения (например, в виде параметрической модели [7]) на физической системе  $\Omega_i$  в зависимости от количества занимаемых вычислительных ресурсов  $P = (q, p, \Pi)$ ,  $\kappa_{ij}$  — относительная пиковая производительность системы  $\Omega_i$  для заданного сервиса  $\Sigma_j$ , а  $T_{0j}$  — абсолютное время исполнения сервиса на „эталонном“ (т.е. выбранном для сопоставления) ресурсе облачной среды в последовательном режиме. Следует отметить, что сервисы могут исполняться на тех системах, где они установлены, или развертываться на новых целевых системах (если они совместимы). На каждом ресурсе могут быть доступны все сервисы (или только несколько).

Время исполнения конкретного сервиса в рамках модели облачной среды (1)—(2) может быть представлено выражением:

$$T_{ij} = t_{ij} + \kappa_{ij} T_{0j} / S(\Xi_j, P, \Omega_i). \quad (3)$$

Здесь индексы  $i, j$  обозначают целевой ресурс и тип сервиса. Величина  $t$  в (3) является характеристикой накладных расходов на запуск сервиса, задаваемой распределением  $\mathfrak{Z}_i(T)$ , и может быть представлена следующим образом:

$$t = t_u(\Omega, \Xi) + t_r(\Omega) + t_w(\Xi), \quad (4)$$

где  $t_u$  — накладные расходы на подготовку задания к исполнению в распределенной среде  $\Omega$  с данными  $\Xi$ ,  $t_r(\Omega)$  — внутренние накладные расходы распределенной среды,  $t_w(\Xi)$  — совокупное время подготовки и передачи данных в распределенной среде.

При запуске каждого сервиса создается *сессия*  $\Sigma_{ij}^{(k)} = \Sigma_j(\Xi_j^{(k)}, S_j(\Xi_j^{(k)}, P^*, \Omega_i^*), \kappa_{ij})$ . Ее окончание включает в себя прикладной пакет  $j$  с заданными характеристиками исполнения, функционирующий на выделенном виртуальном ресурсе  $\Omega_i^*$ . Здесь  $P^*$  в общем случае соответствует оптимальному режиму исполнения сервиса (для максимума функции  $S_j(\Xi_{ij}, P^*, \Omega_i)$  при ограничении на предельное число вычислителей на ресурсе  $\tilde{\Omega}_i$ ).

Таким образом, временной срез „общего“ процесса функционирования облачной среды в рамках модели SaaS представляется в виде

$$\Sigma = \left\langle \left\{ \Sigma_{ij}^{(k)}(\Omega_i^*) \right\}_{\substack{i=1, N \\ j \in W}}^{k=1, M} \middle| \bigcup_l \Omega_l \setminus \bigcup_i \Omega_i^* \right\rangle. \quad (5)$$

Здесь  $k$  — индекс сквозной нумерации сессий пользователей,  $i$  — индекс сквозной нумерации виртуальных ресурсов, выделяемых под предоставление им сервисов, а  $j \in W$  — идентификаторы конкретных содержательных сервисов. При этом допускается одновременное наличие нескольких сессий одних и тех же пользователей с одинаковыми сервисами, различающимися, например, только параметрами запуска  $\Xi_{ij}^{(k)}$ .

Множество  $\bar{\Omega} = \bigcup_l \Omega_l \setminus \bigcup_i \Omega_i^*$  определяет доступное облако ресурсов, которое может быть представлено в виде

$$\bar{\Omega} = \bigcup_l \tilde{\Omega}_l = \bigcup_l \left( \Omega_l \setminus \left\{ \Omega_{al}^*, \Omega_{bl}^*, \dots \right\} \right),$$

т.е. с учетом того, что виртуальный ресурс выделяется на конкретной физической целевой системе. Таким образом, облако ресурсов может быть описано набором свободных физических систем  $\bar{\Omega}$  с меньшим количеством вычислителей и иными характеристиками доступной оперативной и дисковой памяти, по сравнению с исходной средой  $\Omega$ .

Модель (3)—(4) определяет размерные характеристики производительности сервиса. Однако для анализа эффективности применения среды облачных вычислений в рамках модели SaaS, по сравнению с традиционными моделями использования программного обеспечения, целесообразно ввести ряд безразмерных характеристик:

— технологическая эффективность  $\mathfrak{G}_1$  использования среды облачных вычислений в сопоставлении с выделенной вычислительной системы с предустановленным пакетом (учитываются накладные расходы на управление выделенной системой);

— пользовательская эффективность  $\mathfrak{G}_2$  использования среды облачных вычислений в сопоставлении с „чистым“ временем исполнения пакета на идеальной вычислительной системе (накладные расходы отсутствуют);

— абсолютная эффективность  $\mathfrak{G}_3$  технических решений среды облачных вычислений в сопоставлении с техническими решениями низкоуровневой вычислительной инфраструктуры (кластеры, Грид);

— относительная эффективность  $\mathfrak{G}_4$  технических решений среды облачных вычислений в сопоставлении с техническими решениями низкоуровневой вычислительной инфраструктуры для решения конкретной прикладной задачи, требующей передачи данных определенного объема.

Поскольку выражение (3) определяет случайную функцию в соответствии с  $\mathfrak{Z}_l(T)$ , определяемые через него безразмерные характеристики  $\mathfrak{G}_1 - \mathfrak{G}_4$  являются случайными величинами. В табл. 1 приведены вероятностные характеристики эффективности среды облачных вычислений НРС-NASIS II для примера, соответствующего рисунку.

Таблица 1

Характеристика	Моменты, с		Квантиль 10 %-ной обеспеченности, с		
	$m_x$	$\sigma_x$	$q_{10\%} - I_{95\%}$	$q_{10\%}$	$q_{10\%} + I_{95\%}$
$\mathfrak{G}_1$	88,0	0,9	88,9	89,4	89,8
$\mathfrak{G}_2$	72,3	4,5	76,2	78,1	80,0
$\mathfrak{G}_3$	159,4	64,5	235,8	275,6	315,4
$\mathfrak{G}_4$	58,3	9,7	67,6	72,0	76,3

Из таблицы следует, что оценка технологической эффективности  $\mathfrak{G}_1$  достаточно устойчива (около 88—89 %). Разброс пользовательской эффективности  $\mathfrak{G}_2$  существенно выше, а

среднее значение — ниже (72 %), что можно объяснить учетом накладных расходов среды ГридННС. Абсолютная эффективность технических решений  $\vartheta_3$  значительно превышает 100 % вследствие того, что накладные расходы платформы CLAVIRE, используемой в НРС-NASIS II, существенно меньше накладных расходов ГридННС (см. рисунок, в). Относительная эффективность технических решений  $\vartheta_4$  меньше, чем пользовательская эффективность  $\vartheta_2$ , в силу сходных затрат на передачу данных.

**Вероятностная модель надежности предоставления сервисов.** В соответствии с моделью (1)—(2), среда облачных вычислений может иметь два режима функционирования:

— ожидания, когда в среде не выполняется ни одной сессии; все ресурсы свободны,  $\bar{\Omega} = \Omega$ ;

— предоставления сервисов в соответствии с (5), когда  $\bar{\Omega} \subset \Omega$ : в этой ситуации может быть одновременно запущено  $k$  сессий с различным временем выполнения.

Каждая сессия задействует ресурс  $\Omega_i$ , который может включать несколько вычислителей (узлов, процессоров, ядер). Как следствие, вероятность отказа пропорционально увеличивается. Это позволяет использовать для оценки надежности *условное* время работы  $T_{0j}$  параллельной программы на ресурсе  $\Omega_i$ , которое равно времени работы этой программы на одном процессоре с данными характеристиками  $\Xi_{ij}$  (3). Совокупное время работы в режиме предоставления сервисов является случайной величиной, зависящей от количества пользователей, работающих в данный момент в среде, выбора сервисов, исходных данных и режимов работы. В предположении о достаточности вычислительных ресурсов среднее время работы в режиме предоставления сервисов может быть представлено в виде:

$$\tilde{T} = \sum_{l=1}^R C_l \bar{T}_{0l} f_l(P_l^*) (1 - \pi_l). \quad (6)$$

Здесь  $C_l$  — весовой коэффициент, отражающий потребительский интерес к использованию прикладного пакета  $l$ ,  $P_l^*$  — оптимальное количество вычислителей (ядер, процессоров, узлов),  $\bar{T}_{0l}$  — среднее время использования пакета (по разным наборам исходных данных),  $\pi_l$  — вероятность отказа конкретного сервиса. Функция  $f_l(P_l^*) \geq 1$  отражает дополнительные накладные расходы на поддержку параллельного выполнения пакета. Она может быть представлена в различных параметризациях, например, в форме закона Амдала. Однако применительно к задаче определения среднего времени наработки на отказ можно полагать  $f \equiv 1$  с ошибкой в безопасную сторону. Величина  $T_{0l}$  определяется путем тестового исполнения типовых задач. Величины  $C_l$  определяются экспертно специалистами предметной области, а  $\pi_l$  — путем тестирования соответствующих прикладных пакетов в процессе их разработки и опытной эксплуатации. Например, пусть на момент тестирования пакета имеется  $K$  ошибок. На первом этапе тестирования было произведено  $\tau_1$  запусков программного обеспечения и обнаружено  $k_1$  ошибок. После проведения мероприятий, направленных на устранение ошибок, в пакете остается  $K - \alpha k_1$  ошибок. Коэффициент  $\alpha$  связан с тем, что при внесении исправлений (устранении обнаруженных при тестировании ошибок) могут появиться новые ошибки. Соответственно после проведения  $m$ -го этапа тестирования, потребовавшего  $\tau_m$  запусков программы, в ней остается  $K - \alpha \sum_{j=1}^m k_j$  ошибок.

В предположении, что число ошибок  $c\lambda$  в программе распределено по закону Пуассона, где  $\lambda$  — частота появления ошибки, а  $c$  — коэффициент пропорциональности, вероятность безотказной работы  $p(0) = 1 - \pi$  и среднее время наработки на отказ  $ET$  имеют вид

$$p(0) = \exp\left(-\left(\hat{\gamma} - \hat{\beta} \sum_{i=1}^m k_i\right)t\right), ET = \frac{1}{1 - \exp\left(-\left(\hat{\gamma} - \hat{\beta} \sum_{i=1}^m k_i\right)\right)}, ET \approx \frac{1}{\hat{\gamma} - \hat{\beta} \sum_{i=1}^m k_i}, \quad (7)$$

где  $\gamma = cK, \beta = c\alpha$  — приведенные параметры, оцениваемые методом максимального правдоподобия. В общем случае оценки  $\hat{K}$  и  $\hat{K} - \hat{\alpha}k_i$  могут принимать нецелые значения, так как число ошибок до проведения тестирования — случайная величина, и в модели в качестве параметра присутствует математическое ожидание числа ошибок. Для оценки частоты возникновения ошибок по данным поэтапного тестирования используется метод наименьших квадратов.

Учитывая, что среда облачных вычислений используется разными пользователями, запускающими различные сервисы, можно определить среднее время безотказной работы среды в целом как среднюю длину серии успешных независимых запусков до первого отказа. Поскольку каждый запуск подчиняется распределению Бернулли, то длина всей серии будет распределена в соответствии с геометрическим законом  $f(n) = (1 - \pi_\Sigma)^n \pi_\Sigma$ . Среднее значение этого распределения  $\langle n \rangle = 1/\pi_\Sigma$ , таким образом:

$$T_\Sigma = \hat{T}/\pi_\Sigma. \quad (8)$$

Модель (8)—(9) является обобщенной по отношению к исходному разнообразию функциональных возможностей среды облачных вычислений и вариантов использования прикладных сервисов, в силу того что рассматривает последовательность запуска одинаковых „усредненных“ сервисов со средним временем выполнения  $T_\Sigma$ .

В качестве примера в табл. 2 приведены значения параметров выражения (6) для предметно-ориентированных сервисов среды HPC-NASIS II.

Таблица 2

Прикладной пакет	Параметры			
	$C_i, \%$	$T_i, \text{мин}$	$P_i^*$	$\pi_i, \%$
SEMP	10	600	1	1
PRIRODA	15	30	8	0,5
NDDO	10	5	8	3
DPIMC	0,1	300	8	1
MD- kMC	15	600	8	1,5
NTDMFT	5	300	8	1
JAGGREGATE	0,1	3	8	1
QDLaser	0,1	3	8	1
PLASMON	0,1	1	8	1
NANOIMAGE	10	10	24	1
NANOFLOW	4,4	60	24	5
NAEN	0,1	3	24	2
MAGNET	0,1	10	24	3
GAMESS	15	60	8	0
ORCA	15	60	8	0

В табл. 2 для компонентов GAMESS и ORCA принято  $\pi_i = 0$ , в силу того что эти компоненты являются внешними подключаемыми пакетами и не подвергались тестированию в ходе разработки. Используя данные из табл. 1 в (6)—(8), получим точечную оценку среднего

времени безотказной работы всего комплекса —  $T_{\Sigma} = 965$  ч. При этом выборочная изменчивость такой оценки достаточно велика. В частности, для верхней границы 90 % доверительного интервала  $\pi_{\Sigma}$ , в силу того что в (8) выполняется деление на относительно малую величину, получается следующая оценка снизу времени безотказной работы:  $T_{\Sigma}^{-90\%} = 643$  ч.

**Заключение.** Вероятностные модели производительности (1)—(5) и надежности (6)—(8) ориентированы на выполнение поверочных расчетов для оценки качества функционирования сервисов в ходе разработки и опытной эксплуатации проблемно-ориентированных сред облачных вычислений.

Работа выполнена при поддержке проектов Минобрнауки по реализации Постановлений № 218 и 220 Правительства Российской Федерации (2010—2012 гг.).

#### СПИСОК ЛИТЕРАТУРЫ

1. Надежность автоматизированных систем управления. ГОСТ 24.701 86.
2. Shootan M. L. Operational Testing and Software Reliability Estimation During Program Development, Record // IEEE Symp. on Computer Software Reliability. 1973. N 73. P. 51—76.
3. Благодатских В., Волик В., Посакалов К. Стандартизация разработки программных средств. М.: Финансы и статистика, 2005. 288 с.
4. Майерс Г. Искусство тестирования программ. М.: Финансы и статистика, 1982. 174 с.
5. Спельников Д. М., Гуськов А. А., Маслов В. Г., Бухановский А. В. Учебно-научный комплекс „Компьютерное моделирование в нанотехнологиях“ на основе Грид-среды // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 44—50.
6. [Электронный ресурс]: <www.ngrid.ru>.
7. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.

#### Сведения об авторах

- Тимофей Николаевич Чуров** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: tchurovtim@gmail.com
- Денис Александрович Насонов** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: denis.nasonov@gmail.com
- Борис Яковлевич Штейнберг** — д-р техн. наук; Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; зав. кафедрой;  
E-mail: borsteinb@mail.ru
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор;  
E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.