

УДК 004.383.3

**В.А. Романчук, В.В. Лукашенко**

Рязанский государственный университет им. С.А. Есенина, Рязань, Россия

## **МОДЕЛИРОВАНИЕ СТРУКТУРЫ НЕЙРОКОМПЬЮТЕРНОГО ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА**

Рассматривается модель вычислений глобального распределенного вычислительного кластера на базе нейрокомпьютеров, которая имеет программную реализацию в виде системы управления глобальным распределенным вычислительным кластером. Показанная обобщенная модель формирования вычислительной структуры кластера на базе нейропроцессоров, включает модель формирования вычислительной нейрокластерной структуры, модель передачи пакетов по сети, модель загрузки подзадач в вычислительную структуру кластера нейрокомпьютеров. Разработанные модели позволяют реализовывать эффективные кластерные параллельные и распределенные системы с заданными параметрами производительности. Высокий уровень распараллеливания обеспечивает возможность реализации «гибких», перестраиваемых систем различной структуры, т.е. формировать аппаратную базу под заданный алгоритм задачи. Существуют два принципиальных подхода к формированию вычислительной структуры кластера: статический и динамический, поскольку перед непосредственным распределением очередной подзадачи на очередной вычислительный модуль необходимо ввести его в состав вычислительной структуры, а затем направить на него саму задачу. Для реализации модели формирования структуры был выбран динамический подход, согласно которому очередной вычислительный узел резервируется динамически посредством опроса всех вычислительных машин после выполнения задачи предыдущим вычислительным модулем. Плюсом такого вычислительного кластера является сверхбыстрое выполнение задач, минусом – возможные ошибки в маршрутизации и в графе выполнения программ поступившей задачи. В разрабатываемой системе применен принцип динамического резервирования вычислительных машин. Предлагаемая модель передачи сообщений основана на принципе коммутации пакетов при передаче данных, однако следует заметить, что в рамках настоящего исследования рассматриваются и моделируются однонаправленные коммуникации.

**Ключевые слова:** вычислительная система, облачные вычисления, выбор структуры, граф потока управления, нейрокомпьютер.

**V.A. Romanchuk, V.V. Lukashenko**

Ryazan State University named after S.A. Yesenin, Ryazan, Russian Federation

## **MODELING OF THE STRUCTURE OF THE BRAIN-COMPUTER COMPUTING CLUSTER**

In this paper the model of computation of global distributed computing cluster based on neuro-computers, which is a software implementation of a management system globally distributed computing cluster. Shows a generalized model of cluster computing cluster structure based neuroprocessors includes model of computing neuro-clusters structure, packet transmission network model, model download sub-tasks in the computing cluster structure of neuro-computers. The developed models allow to realize

effective clustered parallel and distributed systems with the specified parameters of performance. The high level of parallelization allows the implementation of the "flexible", reconfigurable systems with different structures, that is to form the hardware basis for a given algorithm problem. There are two principal approaches to the formation of cluster computing structures: static and dynamic as before the direct distribution of the next sub-tasks for the next calculation module you must enter it in the computer structure, and then direct the task itself to it. To implement dynamic approach was chosen as the model of formation of the structure, according to which another compute node dynamically reserved by interviewing all the computers after the task the previous computing fashion-lem. The advantage of this is the ultra-fast computing cluster tasks. The downside of possible errors in routing and execution of programs box, received the task. The developed system uses the principle of dynamic provisioning of computing machines. Offered messaging model is based on the principle of packet switching for data transmission, but it should be noted that in this study are considered and modeled one-way communication.

**Keywords:** computer system, cloud computing, the choice of structure, control flow graph, neurocomputer.

**Введение.** В современной научной и производственной сфере достаточно актуальна задача эффективного использования систем обработки данных из-за недостаточности вычислительных ресурсов для решения ряда задач. Модель вычислений осуществляет связь между архитектурой и моделью программирования в вычислительных системах, а также отражает взаимодействие процессов в них. Но в настоящее время для классифицированных технологий нейрокомпьютерных кластерных вычислений универсальных моделей не построено, что объясняется сильной зависимостью модели вычислений от архитектуры распределенной системы.

В работе рассматривается модель вычислений глобального распределенного вычислительного кластера на базе нейрокомпьютеров, которая имеет программную реализацию в виде системы управления глобальным распределенным вычислительным кластером.

**Статический и динамический подходы к формированию вычислительной структуры кластера.** Существуют два принципиальных подхода к распространению задач внутри кластера [1]. В рассматриваемом случае это означает, что существуют два подхода к формированию вычислительной структуры кластера: статический и динамический, поскольку перед непосредственным распределением очередной подзадачи на очередной вычислительный модуль необходимо ввести его в состав вычислительной структуры, а затем направить на него саму задачу.

При статическом подходе к распространению подзадач внутри вычислительной структуры вычислительные машины резервируются заранее и остаются занятыми до конца вычислений, после чего запус-

кается процедура резервирования, и вычислительные машины возвращаются в режим ожидания. Достоинствами такого способа являются четкая маршрутизация и четкое следование по графу выполнения программы. К недостаткам можно отнести замедление выполнения задач на вычислительном кластере [2, 3].

При динамическом подходе очередной вычислительный узел резервируется динамически посредством опроса всех вычислительных машин после выполнения задачи предыдущим вычислительным модулем. Плюсом такого вычислительного кластера является сверхбыстрое выполнение задач, а минусом – возможные ошибки в маршрутизации и в графе выполнения программ поступившей задачи. В разрабатываемой системе применен принцип динамического резервирования вычислительных машин.

С математической точки зрения задача динамического резервирования определенных машин кластера состоит в нахождении такой функции  $g_i(t), i = \overline{1, N}$ , где  $t \in [0; +\infty)$  – произвольный момент времени, что

$$g_i(t) \longrightarrow G_i(t) = \{P_i(t), K_i(t), E_i(t), C_i(t), M_i(t), D_i(t), d_i(t), m_i(t), u_i(t)\}.$$

Функция  $P_i(t), i = \overline{1, N}$  показывает, сколько необходимо вычислительных машин кластера для его вычислительной работы в момент времени  $t \in [0; +\infty)$ .

Функция  $K_i(t), i = \overline{1, N}$  показывает, сколько необходимо коммутаторов для связи вычислительных машин в единую сеть в момент времени  $t \in [0; +\infty)$ . В данном случае в роли коммутатора также может выступать и вычислительная машина, не участвующая в данных вычислениях, но передающая данные далее по сети.

Функция  $E_i(t), i = \overline{1, N}$  показывает сетевой маршрут между вычислительными машинами, коммутаторами в момент времени  $t \in [0; +\infty)$ .

Функции  $C_i(t), M_i(t), D_i(t), i = \overline{1, N}$  определяют для каждого вычислительного узла количество вычислительных ядер – нейропроцессоров, объемы оперативной и дисковой памяти в мегабайтах в момент времени  $t \in [0; +\infty)$ .

Функции  $d_i(t)$  и  $m_i(t)$ , как и ранее, показывают объемы доступной оперативной и дисковой памяти, а функция  $u_i(t)$  характеризует загруженность вычислительных ядер.

Вычислительные ядра могут относиться как к отдельным процессорам, так и к нескольким многоядерным процессорам. С целью упрощения модели они рассматриваются как единое множество вычислительных ядер узла  $p_i: X_i = \{x_1, \dots, x_N\}, i = \overline{1, N}$ .

**Вычисление динамических параметров кластера.** Пусть  $X = \bigcup_{i=1}^J X_i$  – множество вычислительных ядер всех узлов кластера. Тогда рассмотрим задачу вычисления динамических параметров  $d_i(t)$ ,  $m_i(t)$ ,  $u_i(t)$ . Процессы, выполняющиеся на вычислительных ядрах, могут напрямую считывать и записывать информацию в оперативную и дисковую память данного узла, взаимодействуя с ними через системную шину или коммутатор. При необходимости передачи сообщения другому вычислительному узлу в рамках одного кластера задействуется сетевая подсистема, осуществляющая разбиение сообщения на пакеты и передачу их по высокопроизводительной коммуникационной сети. Сетевая подсистема каждого узла также содержит коммутатор, выполняющий функции маршрутизации собственных и транзитных пакетов.

Обозначим в качестве  $id(x, t)$  номер задачи, исполняющейся на ядре  $x \in X$  в момент времени  $t$ . Если же ядро  $x$  в момент времени  $t$  было свободно, то  $id(x, t) = 0$ . Пусть  $state(x, t) = \{free, communication, computation\}$  – состояния вычислительного ядра  $x \in X$  в момент времени  $t$  (*free* – свободно, *communication* – исполняет процесс на стадии коммуникации, *computation* – исполняет процесс на стадии вычислений).

Значения динамических параметров узла  $p_i$  в момент времени  $t$  могут быть вычислены по следующим формулам:

$$u_i(t) = \frac{|\{x \in X_i : id(x, t) > 0 \& state(x, t) = computation\}|}{C_i}, \quad (1)$$

$$m_i(t) = M_i - \sum_{x \in X_i, e=id(x, t) > 0} m r_e, \quad (2)$$

$$d_i(t) = D_i - \sum_{x \in X_i, e=id(x, t) > 0} d r_e, \quad (3)$$

где  $mr_e$  и  $dr_e$  – соответственно объемы оперативной и дисковой памяти, необходимой каждому процессу  $e$ -й параллельной подзадачи.

Также важно отметить, что индекс  $i = \overline{1, N}$  строго зависит от  $N$  – количества доступных к обработке подпрограмм:

$$PR = \langle MK_1, MK_2, \dots, MK_i, \dots, MK_N \rangle.$$

**Модель передачи сообщений.** Предлагаемая модель передачи сообщений основана на принципе коммутации пакетов при передаче данных. Пусть  $b$  – стандартный фиксированный размер одного пакета, тогда  $h < b$  – размер заголовочной части пакета, в котором содержатся стандартные инструкции по передаче пакета по сети: адреса получателя и отправителя, контрольные суммы, а  $t < b$  – хвостовая часть пакета, в которой содержится служебная информация об очередности данного программного кода в графе выполнения программы и очередной номер узла в вычислительной нейрокластерной структуре. Тогда для передачи сообщения размером  $m$  необходимо будет

$$\text{Pack} = \left\lfloor \frac{m}{b - h - t} \right\rfloor \text{ пакетов.}$$

Тогда модель передачи пакетов по сети распределенного вычислительного кластера можно представить в виде маршрута – множества, членами которого будут являться вычислительные узлы и коммутаторы, которые проходит пакет от вычислительного узла с адресом начала до узла с адресом конца, не нарушая контрольных сумм маршрута:

$$\alpha = \{n_1, \dots, n_i\}, n_i \in P_i \cup K_i, i = \overline{1, N}. \quad (4)$$

Основным параметром, характеризующим передачу пакетов по маршруту сети, а также являющимся наиболее важной составляющей общей производительности вычислительного кластера, является время передачи пакета от узла  $n_1$  до  $n_i$  –  $T_{\text{packtrans}}$ . Для этого сначала определим функцию задержки пакета при передаче от узла  $n_1$  до  $n_2$ , т.е. задержку при передаче пакета между крайними узлами  $\text{Delay}(E_{1,2})$ , где  $E_{1,2}$  – сетевая связь из множества  $E_i$  между вычислительными узлами  $n_1$  и  $n_2$  при условии, что эти узлы связаны напрямую. В данном случае коммутаторы являются узлами, передающими пакеты далее по сети. В вычислениях они не участвуют. Также некоторые вычислительные узлы могут быть заняты для вычислений в данный момент времени  $t$ ,

но свободны для передачи пакетов далее по сети – в таком случае эти вычислительные узлы выступают в роли коммутаторов.

Таким образом  $T_{\text{packtrans}}$  – время передачи пакета Pack по сетевому маршруту  $\alpha$  можно вычислить следующим образом:

$$T_{\text{packtrans}}(\alpha, \text{Pack}) = \sum_{x=1}^{i-1} \frac{b}{\text{Delay}(E_{x,x+1})} + \frac{b(\text{Pack}-1)}{\min_{x=0,i-1} \text{Delay}(E_{x,x+1})} + t_{\text{routing}} \text{Pack}(i), \quad (5)$$

где  $t_{\text{routing}}$  – среднее время маршрутизации пакета. Это время затрачивает каждое сетевое устройство любого узла. В данной формуле не учитываются задержки, связанные с сетевой конкуренцией пакетов, поскольку она исключается использованием вычислительных нейрокластерных структур и динамического распределения задач по узлам вычислительного кластера.

Вычислительная нагрузка кластера формируется из потока подзадач  $\text{Task} = \{\text{task}_1, \dots, \text{task}_i\}$ ,  $i = \overline{1, N}$ , которые формируются в результате работы  $A_z$  – алгоритма разбиения задач, поступивших на выполнение в вычислительный кластер, помещаемых им в очередь задач. Каждая задача является параллельной неинтерактивной программой, способной работать в пакетном режиме. Процессы этой задачи запускаются планировщиком, реализующим один из подходов к распространению подзадач внутри вычислительной структуры, в данной работе выбран динамический подход. Процессы запускаются одновременно по мере освобождения вычислительных машин от предыдущих задач, во время работы вычислительные ядра-нейропроцессоры обмениваются друг с другом сообщениями.

Для каждой задачи  $\text{task}_i$  необходимо математически определить требования к ресурсам кластера, такие как: количество необходимых нейропроцессоров  $x_i$ , объем оперативной  $mr_e$  и дисковой памяти  $dr_e$ , необходимой каждому процессу  $e$ -й параллельной подзадачи в килобайтах, временную оценку  $t_i$  – время, необходимое для исполнения каждого процесса в нейропроцессоре, и время выполнения подзадачи на нейропроцессоре целиком –  $T_{\text{взп}}$ .

$$\text{task}_i = (x_i, mr_e, dr_e, t_i, T_{\text{взп}}), \quad (6)$$

$$T_{\text{взп}} = t_1 \cup t_2 \dots \cup t_i, \quad i = \overline{1, N}. \quad (7)$$

С учетом выделенных планировщиком ресурсов  $T_{\text{взп}}$  можно вычислить по формуле

$$T_{\text{взп}} = \frac{t_i}{\min_{i \in N_j} S_i}, \quad (8)$$

где  $S_i$  – относительная производительность каждого нейропроцессора, которая определяет, во сколько раз ядра данного узла работают быстрее вычислительных ядер самого непроизводительного узла,  $N_j$  – множество номеров вычислительных узлов  $p_i$ , отданных подзадаче.

Структура каждой задачи  $\text{task}_i$  может быть описана с помощью SPMD-модели. SPMD (Single Program Multiple Data) – модель, подкласс MIMD моделей по классификации Флинна, описывает систему, где на всех процессорах MIMD-машины выполняется только одна единственная программа, и на каждом процессоре она обрабатывает разные блоки данных, именно такая модель присуща распределенным вычислительным кластерам [6]. Выполнение каждого процесса параллельной задачи представляет собой чередование коммутационной фазы и фазы вычислений.

Опишем коммуникационную фазу выполнения подзадачи. Коммуникационная фаза на каждой итерации может быть представлена в виде одного коммуникационного паттерна  $CP_i$  из множества  $CP_i = \{CP_1, \dots, CP_i, \dots, CP_N\}, i = \overline{1, N}$ :

$$CP_i = (\text{proc}_i, p_i(\text{proc}_i, \text{proc}_{i+1})), \quad (9)$$

где  $\text{proc}_i$  – множество процессов подзадачи, а  $p_i(\text{proc}_i, \text{proc}_{i+1})$  – функция, определяющая количество пакетов сообщения, передаваемого от процесса  $\text{proc}_i$  к процессу  $\text{proc}_{i+1}$ . Выполнение коммуникационной фазы для каждого процесса  $\text{proc}_i$  состоит из двух шагов: на первом шаге происходит неблокируемая рассылка сообщений всем свободным процессам, ожидающим приема пакетов, а на втором шаге происходит блокируемый прием пакетов от конкретных процессов, которым предназначаются рассылаемые неблокируемые данные. В литературе встречаются следующие типичные паттерны: Random, Pairs, Ring, One-to-all и All-to-all [4–8].

В рамках настоящего исследования рассматриваются и моделируются однонаправленные коммуникации. Моделирование коллективных коммуникаций достаточно сложно и не входит в задачи исследования. Опишем вычислительную фазу выполнения подзадачи. Основным параметром, характеризующим эту фазу, является время выполнения процесса  $\text{proc}_i$  на нейропроцессоре. Введем функцию  $j(\text{proc}_i) \rightarrow p_i$ , которая ставит в соответствие каждому процессу  $\text{proc}_i$  вычислительный модуль – нейропроцессор, на который он назначен алгоритмом планирования. В общем виде будем считать, что каждый процесс тратит одинаковое время на выполнение вычислительной части каждой итерации SPMD-модели задачи. Тогда время выполнения коммутационной фазы процессом  $\text{proc}_i$  можно вычислить по формуле:

$$T_{\text{выч}}(\text{proc}_i) = \frac{t_i}{m_i S_i(j(\text{proc}_i))}. \quad (10)$$

Следовательно, если взять  $T_{\text{ком}}(\text{proc}_i)$  – время выполнения коммуникационной фазы процессом  $\text{proc}_i$  задачи  $\text{task}_i$ , тогда реальное время выполнения задачи  $T_{\text{task}}$  может быть вычислено по формуле:

$$T_{\text{task}} = \max \left\{ \sum_{i=1}^{m_i} T_{\text{ком}}(\text{proc}_i) + \frac{t_i}{m_i S_i(j(\text{proc}_i))} \right\}. \quad (11)$$

Следует отметить, что величина  $T_{\text{ком}}(\text{proc}_i)$  зависит от сетевой конкуренции и от физического расположения нейропроцессов на вычислительных узлах, ее значение может быть вычислено непосредственно в процессе работы вычислительного кластера и его управляющей системы. Также следует отметить, что в рамках данной модели происходит пренебрежение временем разбиения сообщения на пакеты ввиду его незначительности [9–11].

**Выводы.** Показаны формализованное описание и обобщенная модель функционирования вычислительной системы на базе нейрокомпьютеров, включающая ее характеристики, возможные структуры при взаимодействии с памятью, модель передачи сообщений и модель распределения задач внутри облачного кластера. Модель нейрокомпьютерного кластера позволяет более эффективно использовать системы обработки данных на базе специализированных аппаратных ресурсов – нейропроцессоров.



### **Библиографический список**

1. Галушкин А.И. Нейронные сети: основы теории. – М.: Горячая линия-Телеком, 2010.
2. Топорков В.В. Модели распределенных вычислений. – М.: Физматлит, 2004.
3. Ручкин В.Н., Романчук В.А., Лукашенко В.В. Обобщенная модель вычислений кластера нейροкомпьютеров // Вестник Рязан. гос. ун-та им. С.А. Есенина. – 2015. – № 2(47). – С. 146–150.
4. Лукашенко В.В. Анализ основных вопросов классификаций распределенных вычислительных систем // Современная техника и технологии. – 2015. – № 4(44). – С. 65–69.
5. Романчук В.А., Ручкин В.Н. Оценка результатов моделирования вычислительных систем на базе нейροпроцессоров // Известия Тульск. гос. ун-та. Технические науки. – 2013. – Вып. 9–2. – С. 194–203.
6. Романчук В.А., Ручкин В.Н. Разработка алгоритмов определения вида структуры нейροпроцессорной системы на основе описания связей ее элементов // Информатика и прикладная математика: межвуз. сб. науч. тр. – 2011. – Вып. 17. – С. 106–109.
7. Романчук В.А., Лукашенко В.В. Модель описания функционирования нейροпроцессора как сложной параллельной системы // Автоматизация и управление в технических системах. – 2015. – № 2(14). – С. 13–24.
8. Романчук В.А. Разработка алгоритмов определения связей элементов вычислительной структуры на базе нейροпроцессоров // Информатика и прикладная математика: межвуз. сб. науч. тр. – Рязань: Изд-во Рязан. гос. ун-та им. С.А. Есенина, 2011. – Вып. 17. – С. 102–105.
9. Макаров Д.В., Романчук В.А. Облачные SaaS, IaaS, PaaS системы для искусственного интеллекта [Электронный ресурс] // Современная техника и технологии. – 2015. – № 5(45). – С. 75–78. – URL: <http://technology.snauka.ru/2015/05/6731> (дата обращения: 04.01.2017).
10. Бурцев В.С. Параллелизм вычислительных процессов и развитие архитектуры супер ЭВМ. – М.: Изд-во ИВВС РАН, 1997.
11. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.

## References

1. Galushkin A.I. Neironnye seti: osnovy teorii [Neuron networks: the theory base]. Moscow: Goriachaia liniia-Telekom, 2010.
2. Toporkov V.V. Modeli raspredelennykh vychislenii [Grid computer models]. Moscow: Fizmatlit, 2004.
3. Ruchkin V.N., Romanchuk V.A., Lukashenko V.V. Obobshchennaia model' vychislenii klastera neirokomp'iuterov [Generic models of the neural computers calculation clusters]. *Vestnik Riazanskogo gosudarstvennogo universiteta imeni S.A. Esenina*, 2015. no. 2(47), pp. 146-150.
4. Lukashenko V.V. Analiz osnovnykh voprosov klassifikatsii raspredelennykh vychislitel'nykh sistem [The analysis of the distributed system classification main questions]. *Sovremennaia tekhnika i tekhnologii*, 2015, no. 4(44), pp. 65-69.
5. Romanchuk V.A., Ruchkin V.N. Otsenka rezul'tatov modelirovaniia vychislitel'nykh sistem na baze neiroprotssessorov [Results assessment of computing system modeling on the base of neuroprocessor]. *Izvestiia Tul'skogo gosudarstvennogo universiteta. Tekhnicheskie nauki*, 2013, iss. 9-2, pp. 194-203.
6. Romanchuk V.A., Ruchkin V.N. Razrabotka algoritmov opredeleniia vida struktury neiroprotssessornoi sistemy na osnove opisaniia sviazei ee elementov [Algorithms development of neuroprocessor system structure type definition]. *Mezhvuzovskii sbornik nauchnykh trudov "Informatika i prikladnaia matematika"*, 2011, iss. 17, pp. 106-109.
7. Romanchuk V.A., Lukashenko V.V. Model' opisaniia funktsionirovaniia neiroprotssessora kak slozhnoi parallel'noi sistemy [The model of neuroprocessor functional description as complex parallel system]. *Avtomatizatsiia i upravlenie v tekhnicheskikh sistemakh*, 2015, no. 2(14), pp. 13-24.
8. Romanchuk V.A. Razrabotka algoritmov opredeleniia sviazei elementov vychislitel'noi struktury na baze neiroprotssessorov [Algorithm development of computing structure elements connection definitions on the base of neuroprocessor]. *Mezhvuzovskii sbornik nauchnykh trudov "Informatika i prikladnaia matematika"*. Riazanskii gosudarstvennyi universitet imeni S.A. Esenina, 2011, iss. 17, pp. 102-105.
9. Makarov D.V., Romanchuk V.A. Oblachnye SaaS, IaaS, PaaS sistemy dlia iskusstvennogo intellekta [Cloud-based systems SaaS, IaaS, PaaS for artificial intelligent]. *Sovremennaia tekhnika i tekhnologii*, 2015,

no. 5, available at: URL: <http://technology.snauka.ru/2015/05/6731> (accessed 04 January 2017).

10. Burtsev V.S. Parallelizm vychislitel'nykh protsessov i razvitie arkhitektury super elektronno-vychislitel'nykh mashin [Computing process concurrent and super computer architecture derivation]. Moscow: Institut vysokoproizvoditel'nykh vychislitel'nykh sistem Rossiiskoi akademii nauk, 1997.

11. Voevodin V.V., Voevodin V.I.V. Parallel'nye vychisleniia [Parallel computing]. Saint Petersburg: BKhV-Peterburg, 2002.

### **Сведения об авторах**

**Романчук Виталий Александрович** (Рязань, Россия) – кандидат технических наук, доцент кафедры информатики, вычислительной техники и методики преподавания информатики Рязанского государственного университета им. С.А. Есенина (390000, Рязань, ул. Свободы, 46, e-mail: [v.romanchuk@rsu.edu.ru](mailto:v.romanchuk@rsu.edu.ru)).

**Лукашенко Владислав Владиславович** (Рязань, Россия) – инженер отдела телекоммуникаций Рязанского государственного университета им. С.А. Есенина (390000, Рязань, ул. Свободы, 46, e-mail: [v.lukashenko@rsu.edu.ru](mailto:v.lukashenko@rsu.edu.ru)).

### **About the authors**

**Romanchuk Vitaliy Aleksandrovich** (Ryazan, Russian Federation) is a Ph.D. in Technical Sciences, Associate Professor of computer science, computer engineering and computer science teaching methodology at Ryazan State University named after S.A. Yesenin (390000, Ryazan, 46, Svobody Str., e-mail: [v.romanchuk@rsu.edu.ru](mailto:v.romanchuk@rsu.edu.ru)).

**Lukashenko Vladislav Vladislavovich** (Ryazan, Russian Federation) is an Engineer of telecommunications Department at Ryazan State University named after S.A. Yesenin (390000, Ryazan, 46, Svobody Str., e-mail: [v.lukashenko@rsu.edu.ru](mailto:v.lukashenko@rsu.edu.ru)).

Получено 16.02.2017