

УДК 517.97

Г.А. Матвеев, Е.А. Трушкова

МОДЕЛЬ ДИНАМИЧЕСКОГО РАСПРЕДЕЛЕНИЯ РЕСУРСОВ¹

В статье описана математическая модель и метод динамического распределения ограниченных ресурсов для системы компьютерных приложений в совокупности, который учитывает потребности и приоритеты каждого из приложений при изменяющейся пользовательской нагрузке на них. Модель построена с использованием математической теории оптимального управления, при этом в качестве управления рассматривается количество ресурсов, выделяемых каждому компьютерному приложению.

Ключевые слова: распределение ресурсов, система компьютерных приложений, оптимальное управление.

G.A. Matveev, E.A. Trushkova

MODEL OF DYNAMIC RESOURCE SHARING

This article describes a mathematical model and method of resource sharing to the system of computer applications. Method takes into account the needs and priorities of each application under the influence of changing user load on applications. The model was constructed using the mathematical theory of optimal control. As control is considered the amount of resources allocated to each application.

Keywords: resource sharing, the system computer applications, optimal control.

Введение

Рассматривается задача автоматического управления ограниченными аппаратными ресурсами с учетом ценности выделенных компьютерному приложению ресурсов (например, количества виртуальных машин, обеспечивающих работу приложения; объема оперативной памяти и процессорного времени, предоставляемых виртуальным машинам или отдельным процессам) при текущей пользовательской нагрузке [1, 2]. Это позволит оптимально использовать ресурсы в процессе эксплуатации системы приложений, а именно, динамически добавлять или убирать количество любого ресурса во время работы с учетом потребностей и приоритетов каждого приложения системы и изменения пользовательской нагрузки на каждое приложение.

1. Постановка задачи

Построим математическую модель рассматриваемой задачи для системы, состоящей из n приложений, использующих один первичный ресурс и m различных категорий вторичных (зависимых от первичного) ресурсов.

Пусть $v^i(t)$, $i=1, \dots, n$ – количество первичного ресурса, обеспечивающего работу приложения i , в момент времени t . Обозначим через $r^{ij}(t)$, $i=1, \dots, n$, $j=1, \dots, m$ – количество ресурса j , выделенного приложению i в момент времени t . Ограничения на использование ресурсов записываются в виде:

$$v(t) \in V(t), \quad r^i(t) \in R^i(t, v^i(t)), \quad i=1, \dots, n, \quad (1)$$

где $V(t)$, $R^i(t, v^i)$ – некоторые компактные множества.

Предполагается, что каждое компьютерное приложение имеет индивидуальный набор k^i характеристик, которые однозначно описывают его текущее состояние с точки зрения пользователя (например, время отклика приложения, количество сетевых ошибок). Характеристика зависит от количества первичного ресурса, обеспечивающего работу приложения, объема вторичных ресурсов, предоставленных приложению и оказываемой на приложение с течением времени неуправляемой пользовательской нагрузки $L^i(t)$, $i=1, \dots, n$. При этом характеристики могут быть составлены либо на испытательном стенде до запуска приложения в эксплуатацию, либо непосредственно в процессе работы приложения – в табличном виде. Обозначим через $p^{ik} = p^{ik}(v^i(t), r^{i1}(t), r^{i2}(t), \dots, r^{im}(t), L^i(t))$, $i=1, \dots, n$, $k=1, \dots, k^i$ – характеристику k приложения i , через $\bar{p}^{ik}(t)$ – целевой уровень соответствующей характеристики (желаемое

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 09-01-170).

значение характеристики, ниже которого она не должна опускаться во время эксплуатации компьютерного приложения). С помощью функций

$$\sigma^i(t) = \sum_{k=1}^{k^i} \alpha_{ik} p^{ik} \left(v^i(t), r^{i1}(t), \dots, r^{im}(t), L^i(t) \right) \quad (2)$$

можно получить некоторую суммарную оценку характеристик каждого из приложений в момент времени t . Здесь через α_{ik} обозначены весовые коэффициенты, выбираемые согласно значимости каждой характеристики приложения. Будем называть функции $\sigma^i(t)$ уровнями сервиса приложений.

Задача состоит в поддержании значения уровня сервиса (2) не ниже некоторого целевого уровня (поддержание каждой характеристики не ниже ее целевого уровня) в моменты времени $T = \{t_0, t_0 + h, \dots, t_0 + qh = t_1\}$. Данная задача равносильна минимизации отклонения уровня сервиса

$$\delta^i = \sum_{k=1}^{k^i} \alpha_{ik} \sum_{t=0}^q \max \left\{ 0, \tilde{p}^{ik}(t_0 + th) - p^{ik} \left(v^i(t_0 + th), r^{i1}(t_0 + th), \dots, r^{im}(t_0 + th), L^i(t_0 + th) \right) \right\}$$

для системы приложений в совокупности с учетом приоритета каждого из приложений. Обозначим через δ^{i+} максимум δ^i при всех допустимых значениях ресурсов (минимум δ^i , очевидно, равен нулю).

Замечание. Для задачи поддержания характеристики не выше ее целевого уровня на T отклонение можно записать в виде

$$\sum_{t=0}^q \max \left\{ 0, -\tilde{p}^{ik}(t_0 + th) + p^{ik} \left(v^i(t_0 + th), r^{i1}(t_0 + th), \dots, r^{im}(t_0 + th), L^i(t_0 + th) \right) \right\};$$

для задачи поддержания положительной характеристики как можно ближе к нулю на T отклонение можно записать так:

$$\sum_{t=0}^q p^{ik} \left(v^i(t_0 + th), r^{i1}(t_0 + th), \dots, r^{im}(t_0 + th), L^i(t_0 + th) \right).$$

Тогда исходную задачу динамического распределения ресурсов можно поставить в виде дискретной задачи оптимального управления следующего вида:

$$\begin{aligned} y^i(t+h) &= y^i(t) + \sum_{k=1}^{k^i} \alpha_{ik} \max \left\{ 0, \tilde{p}^{ik}(t) - p^{ik} \left(v^i(t), r^{i1}(t), \dots, r^{im}(t), L^i(t) \right) \right\}, \\ t \in T &= [t_0, t_1], \quad y^i(t_0) = 0, \quad i = 1, \dots, n, \\ v(t) &\in V(t), \quad r^i(t) \in R^i(t, v^i(t)), \quad i = 1, \dots, n, \\ F(y(t_1)) &= \sum_{i=1}^n \beta_i \frac{y^i(t_1)}{\delta^{i+}} \rightarrow \min, \end{aligned} \quad (3)$$

где $y^i(t)$ – сумма отклонений уровня сервиса приложения i к моменту времени t (т. е. в моменты времени $t_0, t_0 + h, \dots, t$), через β_i обозначены весовые коэффициенты, выбираемые согласно приоритету каждого компьютерного приложения (большее значение весового коэффициента соответствует более высокому приоритету соответствующего приложения). Здесь роль управлений играют функции $v^i(t)$, $r^{ij}(t)$, $i = 1, \dots, n$, $j = 1, \dots, m$.

Рассмотрим один из самых подходящих для дальнейшей практической реализации вариантов поставленной задачи (3), а именно, предположим, что на рассматриваемом промежутке времени управления постоянны, т. е. $v^i(t) = v^i$, $r^{ij}(t) = r^{ij}$, $i = 1, \dots, n$, $j = 1, \dots, m$. Тогда исходная задача может быть сформулирована в виде последовательности задач поиска минимума функции $n \cdot (m+1)$ переменных $F(y(t_{1s})) = G(v_s^i, r_s^{ij})$ при условиях (1), т. е.

$$F(y(t_{1s})) = G(v_s^i, r_s^{ij}) \rightarrow \min_{v, r}, \quad v(t) \in V(t), \quad r^i(t) \in R^i(t, v^i(t)), \quad i = 1, \dots, n,$$

на множестве конечных временных отрезков $T^0 = [t_{00}, t_{10}]$, где $t_{10} = t_{00} + q^0 h^0$, $T^1 = [t_{01}, t_{11}]$, где $t_{11} = t_{01} + q^1 h^1$ и т. д.

Спецификой рассматриваемой задачи является зависимость вида динамической системы (3) от неизвестных функций $L^i(t)$, $i = 1, \dots, n$. При этом предполагается, что известны значения этих функций в некоторые предшествующие рассматриваемому отрезку времени моменты $t_{0s} - l^s h^s$, $t_{0s} - (l^s - 1)h^s$, ...,

$t_{0s} - h^s$. Этот факт позволяет осуществлять прогноз изменения функций $L'(t)$ на рассматриваемом отрезке времени (например, с помощью численных методов аппроксимации функций одной переменной).

Полученное таким образом управление (количество выделяемых ресурсов) будет носить кусочно-постоянный характер:

$$v'(t) = v_s^j, \quad r^{ij}(t) = r_s^{ij}, \quad t \in T^s = [t_{0s}, t_{1s}], \quad s = 0, 1, \dots,$$

что позволит легко реализовать его на практике в процессе эксплуатации системы приложений или использовать его в качестве начального управления в некоторой итерационной процедуре улучшения управления.

2. Алгоритм решения задачи (на примере двух компьютерных приложений)

Рассмотрим алгоритм решения задачи динамического управления ресурсами на примере системы двух приложений – картографического сервиса «MapServer» и вычислительного сервиса «X-Com» [1].

Ресурсы, предоставленные приложениям: $v'(t)$ – количество виртуальных машин (ВМ), обеспечивающих работу приложения, r^{i1} – оперативная память ВМ (RAM, Мбайт), r^{i2} – доля физического процессора, предоставляемая ВМ (CPU, %). Ограничения на предоставление ресурсов записываются в виде:

$$v'(t) \geq 1, \quad n \leq \sum_{i=1}^n v'(t) \leq v^+, \quad r^{j-} v'(t) \leq r^{ij}(t) \leq r^{j+} v'(t),$$

$$\sum_{i=1}^n r^{ij}(t) \leq r^{j+} v^+, \quad r^{j-} \leq r^{ij}(t) \leq r^{j+} v^+ - (n-1)r^{j-}, \quad i = 1, \dots, n, \quad j = 1, \dots, m,$$

где $n = 2$, $m = 2$, $r^{1-} = 128$, $r^{1+} = 512$, $r^{2-} = 10$, $r^{2+} = 100$, $v^+ = 3$.

Для построения уровня сервиса первого приложения на дискретном наборе моментов времени $\{5, 6, 7, 8, 9\}$ используются функции $p^{11}(v^1, r^{11}, r^{12}, L^1)$ – среднее время отклика приложения, $p^{12}(v^1, r^{11}, r^{12}, L^1)$ – процент сетевых ошибок и $L^1(t)$ – пользовательская нагрузка на приложение, а также величина желаемого времени отклика $\bar{p}^{11} = 3000$ миллисекунд. Функции $p^{11}(v^1, r^{11}, r^{12}, L^1)$ и $p^{12}(v^1, r^{11}, r^{12}, L^1)$ заданы таблично своим профилем производительности, функция $L^1(t)$ – значениями в моменты времени $\{0, 1, 2, 3, 4\}$. По этим данным были построены аппроксимации функциями $\bar{p}^{11}$, $\bar{p}^{12}$, $\bar{L}^1$ по МНК в виде

$$\bar{p}^{1k}(v^1, r^{11}, r^{12}, L^1) = \max \left\{ 0, \sum_{i=1}^{70} a_{ki}(v^1)(r^{11})^{s_{k1i}}(r^{12})^{s_{k2i}}(L^1)^{s_{k3i}} \right\},$$

где a_{ki} – найденные согласно МНК постоянные действительные числа, а целые показатели степеней s_{kli} удовлетворяют условиям $-2 \leq s_{kli} \leq 2$, $-1 \leq \sum_{i=1}^3 s_{kli} \leq 2$;

$$\bar{L}^1(t) = b_0(L^1(0), \dots, L^1(4)) + b_1(L^1(0), \dots, L^1(4))t,$$

где b_0 , b_1 – найденные согласно МНК постоянные действительные числа.

Теперь отклонение уровня сервиса построим следующим образом:

$$\delta^1 = \sum_{t=0}^4 \max \left\{ 0, -\bar{p}^{11} + \bar{p}^{11}(v^1, r^{11}, r^{12}, L^1(5+t)) \right\} + \sum_{t=0}^4 \bar{p}^{12}(v^1, r^{11}, r^{12}, L^1(5+t)).$$

Можно подсчитать и максимальное отклонение уровня сервиса $\delta^{1+} = 15070$.

Для построения уровня сервиса второго приложения используется функция $p^{21}(v^2, r^{21}, r^{22})$ – скорость расчета задач, находящихся в очереди, а также момент желаемого окончания расчетов θ и величина начальной желаемой скорости расчета задач $\bar{p}^{21}(0) = 18$ задач/сек, что позволяет рассчитать текущую желаемую скорость расчета задач $\bar{p}^{21}(5) = \bar{p}^{21}(0)(\theta - 5)$ задач/сек.

Функция $p^{21}(v^2, r^{21}, r^{22})$ задана таблично своим профилем производительности. По этим данным была построена аппроксимация функцией $\bar{p}^{21}$ в виде

$$\bar{p}^{21}(v^2, r^{21}, r^{22}) = 0.92451 - 0.00024r^{21} + 0.09777r^{22} - 0.345475v^2 + 0.0002r^{21}v^2.$$

Теперь отклонение уровня сервиса построим следующим образом:

$$\delta^2 = \sum_{t=0}^4 \max \left\{ 0, \bar{p}^{21}(5) - \bar{p}^{21}(v^2(5), r^{21}(5), r^{22}(5)) \right\}.$$

Можно подсчитать и максимальное отклонение уровня сервиса $\delta^{2+} = 90$.

С учетом вышеизложенного, функцию $G(v^j, r^{ij})$ запишем в виде:

$$G(v^1, v^2, r^{11}, r^{12}, r^{21}, r^{22}) = \sum_{i=1}^2 \beta_i \frac{\delta^i(v^j, r^{i1}, r^{i2})}{\delta^{i+}} \rightarrow \min, \quad (4)$$

где $\beta_1 = \beta_2 = 1$ (это означает равные приоритеты обоих приложений).

Написана программа, моделирующая работу системы двух указанных приложений при задаваемой случайным образом пользовательской нагрузке с применением описанного метода выбора ресурсов. Ниже приводится пошаговое описание работы программы.

Шаг 1. Полагается $s = 1$, $t_{0s} = 5$ мин.

Шаг 2. Снимаются показания пользовательской нагрузки $L^1(t)$ в моменты $t_{0s} - 5$, $t_{0s} - 4$, $t_{0s} - 3$, $t_{0s} - 2$, $t_{0s} - 1$. Производится расчет прогноза изменения пользовательской нагрузки по МНК в виде $\bar{L}^1(t) = b_0 + b_1 t$.

Шаг 3. Решается задача условной минимизации (4)

$$G(v^*, r^*) = \min_{(v, r) \in W} G(v^1, v^2, r^{11}, r^{12}, r^{21}, r^{22}),$$

$$W = \left\{ (v, r) \left| \begin{array}{l} v^j \geq 1, 2 \leq v^1 + v^2 \leq v^+, r^{j-} \leq r^{jj}(t) \leq r^{j+} v^+ - r^{j-}, \\ r^{1j} + r^{2j} \leq r^{j+} v^+, r^{j-} v^j(t) \leq r^{jj}(t) \leq r^{j+} v^j(t), i, j = 1, 2 \end{array} \right. \right\}$$

методом покоординатного спуска с постоянным шагом, в качестве начального приближения выбирается случайным образом набор точек в допустимой области.

Шаг 4. Перераспределяются ресурсы приложениям согласно найденным значениям ресурсов (v^*, r^*) .

Пересчитывается целевой уровень второго приложения по формуле

$$\bar{p}^{21} = \frac{\bar{p}^{21}(t_{0s})(\theta - t_{0s}) - 5 \bar{p}^{21}(v^{2*}, r^{21*}, r^{22*})}{\theta - t_{0s} - 5}. \text{ Полагается } s = s + 1, t_{0s} = t_{0s} + 5. \text{ Осуществляется переход к шагу 2.}$$

3. Вычислительные эксперименты

Для проведения тестовых расчетов был сгенерирован случайный набор гладких пользовательских нагрузок. Две из них показаны на рис. 1 (первый вариант пользовательской нагрузки представлен сплошной линией, второй – пунктирной).

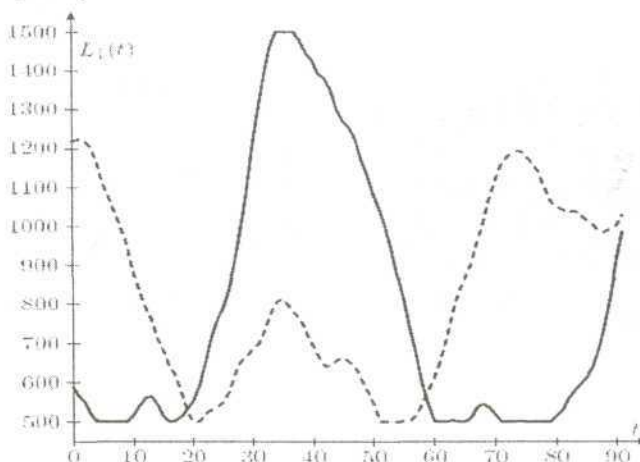


Рис. 1

Результаты работы программы при $v^+ = 3$ (не более трех ВМ на оба приложения в сумме) и увеличивающемся приоритете второго приложения (в зависимости от роста целевого уровня сервиса – желаемой скорости расчета задач) для каждого из вариантов прогноза представлены на рис. 2, 3 (рис. 2 соответствует первому варианту пользовательской нагрузки, рис. 3 – второму). Изменяющийся во времени целевой уровень второго приложения имеет смысл ограниченного конечного времени расчета. Из рисунков

видно, что при различных пользовательских нагрузках алгоритм дает устойчиво хорошее динамическое перераспределение ресурсов с учетом поддержания изменяющихся характеристик на целевом уровне.

Заключение

Представленная математическая модель задачи распределения ресурсов для системы компьютерных приложений построена с использованием теории оптимального управления, при этом в качестве управления рассматривается количество ресурсов, выделяемых каждому компьютерному приложению. Предложенный метод решения задачи учитывает потребности и приоритеты каждого из приложений при изменяющейся пользовательской нагрузке на приложения, что позволяет существенно повысить эффективность совместного использования ограниченных ресурсов системы.

Литература

1. Московский А.А., Первин А.Ю., Walker В. Оптимальное управление ресурсами виртуальных инструментов на вычислительном кластере // Параллельные вычисления и задачи управления: тр. IV Междунар. конф. / ИПУ им. В.А. Трапезникова РАН. — Москва, 2008. — С. 968-978.
2. Матвеев Г.А., Первин А.Ю., Трушкова Е.А. Динамическое распределение ресурсов для приложений // Материалы XVI междунар. конф. по вычислительной механике и современным прикладным программным системам. — М.: Изд-во МАИ-ПРИНТ, 2009. — С. 528-531.

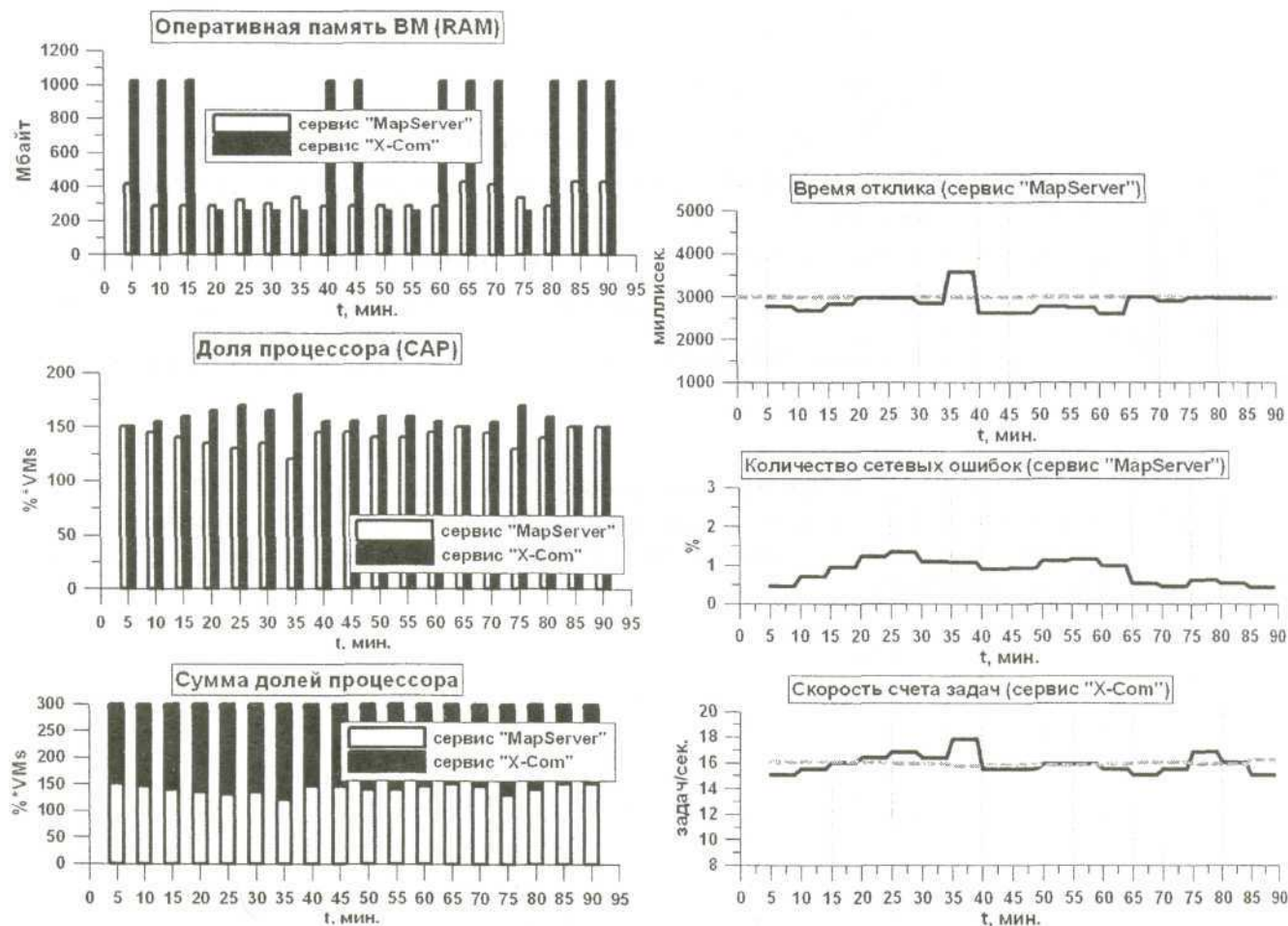


Рис. 2

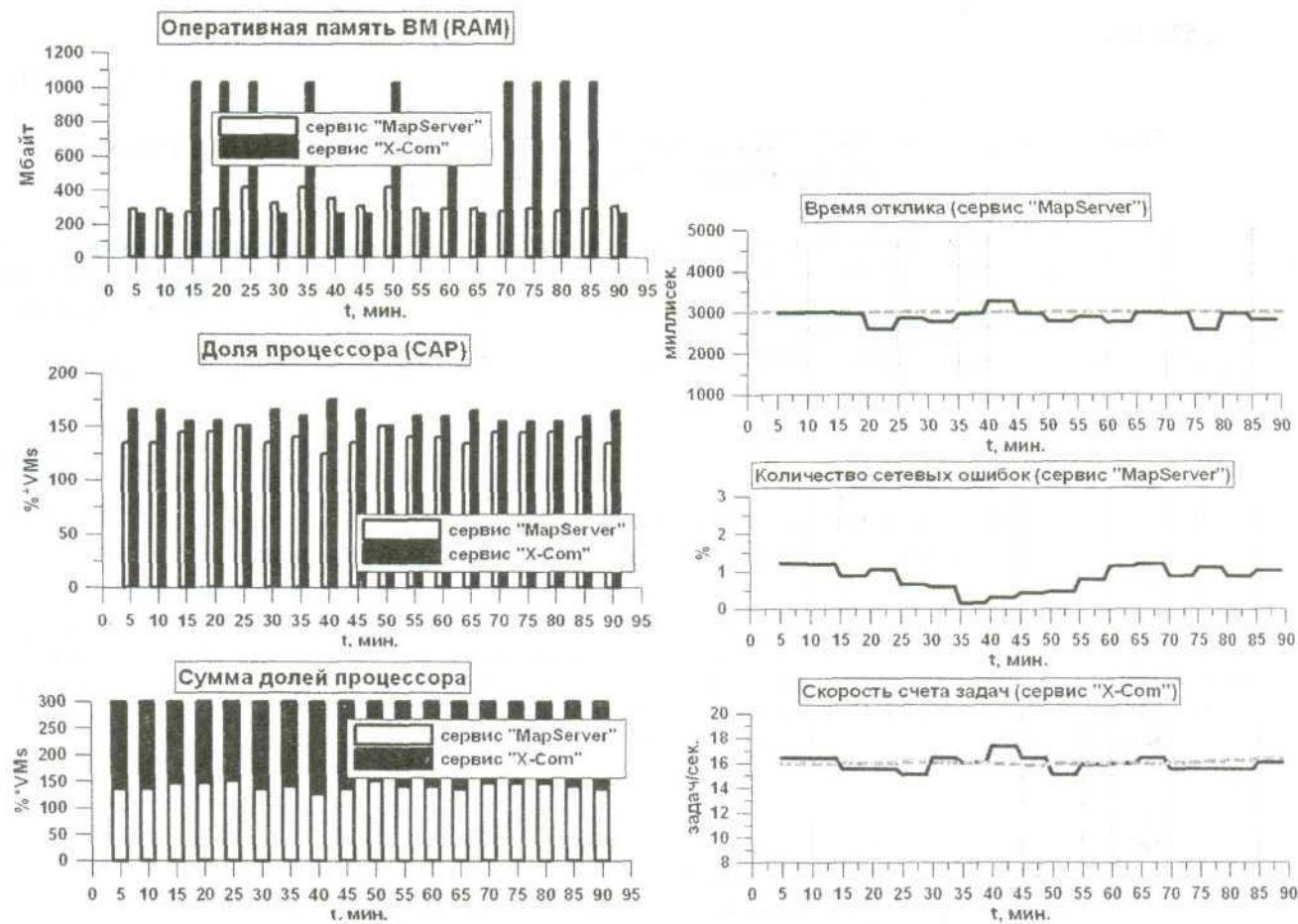


Рис. 3

Матвеев Герман Анатольевич, ведущий инженер-исследователь Исследовательского центра мультипроцессорных систем Института программных систем им. А.К. Айламазяна РАН, тел. (48535) 98095, e-mail: gera@prime.botik.ru

Трушкова Екатерина Александровна, кандидат физико-математических наук, старший научный сотрудник Исследовательского центра системного анализа Института программных систем им. А.К. Айламазяна РАН, тел. (48535) 98094, e-mail: katerina@trushkova.pereslavl.ru

Matveev German Anatolievich, Leading research engineer, Research Center for Multiprocessor Systems (RCMS), The Ailamazyan Program Systems Institute of RAS.

Trushkova Ekaterina Aleksandrovna, Candidate of Physics and Mathematics, Senior Researcher, System Analysis Research Center (SARC), The Ailamazyan Program Systems Institute of RAS.