Труды Карельского научного центра РАН
№ 8. 2017. С. 76–85
DOI: 10.17076/mat663

УДК 004.01:006.72 (470.22)

СТОХАСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА С ГИСТЕРЕЗИСНЫМ УПРАВЛЕНИЕМ СКОРОСТЬЮ ОБСЛУЖИВАНИЯ

А. С. Румянцев[1], К. А. Калинина[1], Т. Е. Морозова[2]

[1] *Институт прикладных математических исследований Карельского научного центра РАН, Петрозаводск*
[2] *Петрозаводский государственный университет*

Предлагается стохастическая модель многосерверной системы массового обслуживания с одновременным занятием и одновременным освобождением заявкой случайного числа серверов и пороговым (гистерезисным) управлением скоростью обслуживающих устройств. Для предложенной модели определены характеристики качества обслуживания и производительности системы. Представлены результаты численного эксперимента, иллюстрирующего зависимость характеристик модели от порога переключения. Предложенная модель позволяет оценить возможности экономии энергии на вычислительном кластере при контроле качества обслуживания без вмешательства в работу реальной системы.

Ключевые слова: стохастическое моделирование; гистерезисное управление; вычислительный кластер; энергоэффективность.

A. S. Rumyantsev, K. A. Kalinina, T. E. Morozova. STOCHASTIC MODELING OF A HIGH-PERFORMANCE CLUSTER WITH HYSTERETIC CONTROL OF SERVICE RATE

A stochastic model of a multiserver queueing system with simultaneous service of a customer by a random number of servers and threshold-based (hysteretic) control of the service rate is presented. The performance and quality-of-service measures of the model are defined. Numerical results of experiments studying the dependence of performance/energy measures on the service rate switching threshold are presented. The model allows studying the energy efficiency of a high-performance cluster under quality-of-service and performance constraints.

Keywords: stochastic modeling; hysteretic control; high-performance computing cluster; energy efficiency.

Введение

Высокопроизводительные вычислительные кластеры широко применяются для ускорения расчетов в науке и промышленности. Высокое энергопотребление кластеров и неравномерность нагрузки дают потенциал для применения методов экономии энергии. Общие для всех многосерверных систем механизмы управления энергопотреблением доступны для непосредственного применения на уровне системного и пользовательского про-

граммного обеспечения вычислительного кластера. Среди таких механизмов наибольший потенциал экономии обеспечивают состояния пониженного энергопотребления ACPI (сон, глубокий сон, выключение), однако применение данных механизмов негативно отражается на качестве обслуживания. Это связано с тем, что уход системы в состояние пониженного энергопотребления и возвращение в рабочий режим требуют значительных временных затрат. Поэтому оптимизация управления механизмами ACPI на многосерверных системах, как правило, выполняется относительно компромиссного критерия, включающего качество обслуживания и энергопотребление, в аддитивном либо мультипликативном виде [7, 8, 10].

Важным отличием высокопроизводительного кластера от классической многосерверной системы (например, серверной фермы) является возможность запуска пользовательской программы одновременно на всех вычислительных узлах (серверах), что может служить препятствием для выключения части серверов с помощью ACPI.

Один из базовых механизмов экономии, оказывающих наименьшее влияние на качество обслуживания, состоит в управлении скоростью серверов вычислительного кластера (например, с помощью подсистемы cpufreq операционных систем Linux). Как правило, подобные механизмы основаны на переключении состояний методом порогового (в частности, гистерезисного) управления, например, с использованием специальных приложений (таких, как регулятор ondemand из подсистемы cpufreq ядра операционной системы Linux). Такое переключение происходит практически мгновенно, при этом снижение частоты и питающего напряжения оказывает существенное влияние на энергопотребление системы [9]. Кроме того, в отличие от механизма ACPI управление скоростью обслуживания оставляет доступными все серверы, не препятствуя запуску на них пользовательских программ.

На практике подбор параметров управления зачастую производится экспериментально, поскольку аналитические результаты доступны лишь для упрощенных моделей, например, для односерверных систем [1, 13]. При этом важно не только учитывать максимально возможную экономию, но и контролировать качество обслуживания. Подчеркнем, что поток заявок на обслуживание в системе имеет случайный характер [6]. Таким образом, возникает задача разработки стохастической модели вычислительного кластера с управлением скоростью обслуживания для подбора оптимальных параметров конфигурации.

Стохастические модели многосерверных систем с одновременным занятием и одновременным освобождением заявкой случайного числа серверов рассматривались в ряде работ. Так, в работе [4] рассмотрена двухсерверная система с одновременным обслуживанием серверами одной заявки либо обслуживанием по отдельности двух последовательных заявок. В работе [12] представлена модель многосерверной системы, в которой заявке требуется случайное число процессоров на одно и то же случайное время, и с помощью матрично-аналитического метода проведен численный анализ основных характеристик системы. В работе [2] для двух серверов найдено стационарное распределение состояний системы. В работе [17] предложена модель процесса нагрузки вычислительного кластера, основанная на модификации рекурсии Кифера – Вольфовица. В работе [3] доказан критерий стационарности двухсерверной системы, а в работе [14] матрично-аналитическим методом доказан критерий стационарности модели вычислительного кластера с любым числом серверов, входным потоком марковского типа и экспоненциальным распределением времени обслуживания. В то же время общим базовым предположением указанных моделей является постоянная скорость обслуживания на серверах системы.

В данной работе предлагается новая стохастическая модель вычислительного кластера с управлением скоростью серверов гистерезисного типа, обобщающая ранее предложенную модель процесса нагрузки [17].

Структура работы следующая. В первом разделе предложена стохастическая модель системы массового обслуживания с гистерезисным управлением скоростью обслуживающих устройств, предложены рекуррентные соотношения для расчета динамики состояния системы, предложен способ построения вектора нагрузки в моменты наступления базовых событий. Во втором разделе представлены результаты численного эксперимента по анализу характеристик системы в зависимости от параметров управления.

Стохастическая модель вычислительного кластера с управлением скоростью обслуживания

Рассматривается система обслуживания, имеющая c серверов, которые одновременно могут работать на одной из двух скоростей, каждый при этом выполняя r_1 или r_2 ($\geqslant r_1$)

единиц работы за единицу времени. На вход системы поступает поток заявок, пусть t_j есть время поступления заявки с номером $j \geqslant 1$ (где $t_1 = 0$). Заявка j обладает двумя характеристиками: объемом работы $S_j \in \mathbb{R}_+$ и требуемым числом серверов $N_j \in \{1, \ldots, c\}$, которые занимаются и освобождаются заявкой одновременно. Если свободных серверов для обслуживания заявки недостаточно либо в системе ожидают обслуживания другие заявки, то пришедшая заявка поступает в общую очередь. Поступление ожидающих заявок на обслуживание происходит в порядке прихода в систему (дисциплина обслуживания First Come First Served, FCFS). Поступившая на обслуживание заявка проводит на серверах время до исчерпания объема работы S_j, при этом в процессе ее обслуживания скорость серверов может меняться. Это означает, что реальное время обслуживания заявки j заранее неизвестно, но, очевидно, лежит в интервале $[S_j/r_2, S_j/r_1]$.

Предполагается, что в системе реализован двухпороговый механизм управления скоростью серверов. Все серверы системы мгновенно переключаются на высокую скорость r_2, если сумма оставшихся объемов работ заявок, находящихся в системе, включая вновь пришедшую заявку (суммарная работа), превышает наперед заданный порог k_2. Если же суммарная работа в системе опустится ниже порога $k_1 \leqslant k_2$, то все серверы системы мгновенно снижают скорость до r_1. Отметим, что включение высокой скорости r_2 происходит лишь в моменты прихода заявок.

Управление скоростью обслуживающих устройств может оказывать влияние не только на энергопотребление системы, но и на метрики качества обслуживания, такие как среднее число заявок в системе/очереди, среднее время ожидания/пребывания заявки. Для нахождения оптимальной конфигурации системы необходимо определить как целевую функцию (связанную с энергопотреблением системы), так и ограничения, накладываемые на качество обслуживания. Целью дальнейшего анализа является построение математической модели системы и ее верификация методом имитационного моделирования.

Основные рекуррентные соотношения

Для дальнейшего анализа необходимо ввести ряд обозначений. Прежде всего заметим, что в системе возможны три типа *базовых событий*: приход заявки в систему, уход заявки из системы, снижение суммарной работы до уровня k_1. Важно подчеркнуть, что, в отличие от классической многопроцессорной системы без управления скоростью обслуживания, в момент ухода заявки может произойти как уменьшение, так и увеличение числа заявок, находящихся на обслуживании (что связано с дополнительным параметром заявки — числом требуемых серверов), причем начать обслуживание могут несколько заявок одновременно. Таким образом, для исследования динамики системы важны все три указанных типа базовых событий. В то же время превышение суммарной работой уровня k_2 возможно лишь в момент прихода очередной заявки, поэтому множество моментов переключения на высокую скорость есть подмножество моментов прихода.

Обозначим T_i — i-й момент наступления базового события в системе. В следующих обозначениях нижний индекс i означает, что объект рассматривается в момент времени T_i. Обозначим в момент времени T_i

- M_i — упорядоченное (в порядке поступления) множество номеров заявок, находящихся в системе;
- $B_i(j),\ j \in M_i$ — оставшаяся (незавершенная) работа заявки j, находящейся в системе;
- R_i — скорость серверов.

Будем называть $\{M_i, \{B_i(j), j \in M_i\}, R_i\}$ *состоянием системы* в момент времени T_i. Следующие (необходимые для анализа) величины являются производными величинами от состояния системы в момент времени T_i:

- $\mathcal{M}_i = \max\limits_{k \geqslant 1}\{j_1 < \cdots < j_k \in M_i : \sum_{t=1}^{k} N_{j_t} \leqslant c\} \subseteq M_i$ — множество номеров заявок, находящихся на обслуживании;
- $W_i = \sum_{j \in M_i} B_i(j)$ — суммарная работа;
- $\nu_i = |M_i|$ — число заявок в системе;
- $Q_i = |M_i \backslash \mathcal{M}_i|$ — число заявок в очереди.

Необходимо подчеркнуть, что определение множества $\mathcal{M}_i$ номеров заявок, находящихся на обслуживании, в общем случае *зависит от дисциплины обслуживания*. Изменяя правило включения заявок из M_i в множество $\mathcal{M}_i$, можно изменять дисциплину обслуживания системы без изменения модели.

Определим базис стохастической рекурсии. Будем полагать, что первым событием, наступающим в момент времени $T_1 = t_1 = 0$, явля-

ется приход первой заявки. Тогда

$$M_1 = \{1\},\ B_1(1) = S_1,$$
$$R_1 = r_1 I\{S_1 \leqslant k_2\} + r_2 I\{S_1 > k_2\}$$
$$\mathcal{M}_1 = \{1\}, W_1 = S_1,\ \nu_1 = 1,\ Q_1 = 0,$$

где $I\{\cdot\}$ есть индикатор события.

Изменение состояния системы при переходе от события i к событию $i+1$ определим с помощью рекуррентных соотношений. Момент времени T_{i+1} наступления события с номером $i+1$ можно определить как момент наступления ближайшего из трех событий: ближайшего прихода заявки, ближайшего потенциального ухода заявки и ближайшего потенциального момента включения пониженной скорости (полагаемого равным бесконечности, если суммарная работа ниже уровня k_1) соответственно, т. е.

$$T_{i+1} = \min\left\{ t_{A(T_i)+1}, T_i + \min_{j\in\mathcal{M}_i} \frac{B_i(j)}{R_i}, \right.$$
$$\left. T_i + \frac{W_i - k_1}{I\{W_i > k_1\} R_i} \right\}, \qquad (1)$$

где $A(t) = j$, $t_j \leqslant t < t_{j+1}$, $j \geqslant 1$ есть число приходов в $[0, t]$. Можно заметить, что $A(T_i) = j$, если $T_i = t_j$ для некоторого j. Подчеркнем, что указанные моменты наступления событий являются потенциальными, поскольку скорость обслуживания до этих моментов может измениться, тем самым изменив время наступления событий. Обозначим $\tau_i = T_{i+1} - T_i$ интервал времени между событиями, $i \geqslant 1$.

Для удобства введем следующие индикаторы: I_i^A — индикатор того, что момент T_i есть приход заявки, т. е. $I_i^A = I\{T_i = t_{A(T_i)}\}$; I_i^D — индикатор ухода заявки в момент T_i (если минимум в (1) реализуется на втором элементе) и I_i^S — индикатор включения пониженной скорости (из-за снижения суммарной работы до уровня k_1). Отметим, что $I_i^A + I_i^D + I_i^S = 1$. Можно считать, что указанные индикаторы являются *метками*, связанными с моментами событий T_i, что позволяет единообразно определить изменения состояния системы в моменты наступления базовых событий.

Множество номеров заявок в системе M_{i+1} в момент T_{i+1} наступления $i+1$-го события может измениться в связи с уходом либо приходом очередной заявки следующим образом:

$$M_{i+1} = M_i \cup \{A(T_{i+1}) : I_{i+1}^A = 1\} \setminus$$
$$\setminus \{d_{i+1} : I_{i+1}^D = 1\}, \qquad (2)$$

где d_{i+1} есть номер потенциально уходящей заявки в момент T_{i+1}, определяемый следующим образом:

$$d_{i+1} = \arg\min_{j\in\mathcal{M}_i} \frac{B_i(j)}{R_i}. \qquad (3)$$

Если же индикатор соответствующего события (приход/уход заявки) равен 0, то соответствующее одноэлементное множество считаем пустым.

Следующее соотношение определяет величину незавершенной работы заявки j, находящейся в системе в момент T_{i+1}:

$$B_{i+1}(j) = \begin{cases} B_i(j) - \tau_i R_i, & j \in \mathcal{M}_i \cap M_{i+1} \\ S_j, & j \in M_{i+1} \setminus \mathcal{M}_i. \end{cases} \qquad (4)$$

Таким образом, для заявки j, находившейся на обслуживании в момент T_i, работа $B_i(j)$ уменьшается к моменту T_{i+1} пропорционально скорости R_i. Для заявок, ожидающих в очереди, как и для вновь пришедшей заявки (если был приход заявки), работа равна первоначальной (зафиксированной в момент прихода заявки). Уходящая в момент T_{i+1} заявка исключается из дальнейшего рассмотрения. Заметим, что

$$M_{i+1} = (M_{i+1} \setminus \mathcal{M}_i) \cup (\mathcal{M}_i \cap M_{i+1}),$$

т. е. $B_{i+1}(j)$ в (4) полностью определен для всех $j \in M_{i+1}$.

Скорость обслуживания на всех процессорах системы будет определяться следующим соотношением:

$$R_{i+1} = r_1 I\{W_{i+1} \leqslant k_1\} + r_2 I\{W_{i+1} > k_2\}$$
$$+ R_i I\{k_1 < W_{i+1} \leqslant k_2\}. \qquad (5)$$

Напомним, что величина W_{i+1} может быть определена через состояние системы в момент T_i с использованием рекурсии (4). Отметим, что в частном случае $k_1 = k_2$ скорость обслуживания становится производной величиной от состояния системы, поскольку зависит лишь от текущего состояния системы (не от предыдущего значения скорости).

Характеристики системы

Определим некоторые характеристики *качества обслуживания* предложенной модели. Для заявки j момент поступления на обслуживание можно определить следующим образом:

$$t_j^S = \min\{T_i \geqslant t_j : j \in \mathcal{M}_i\}. \qquad (6)$$

Подчеркнем, что момент t_j^S совпадает либо с моментом прихода t_j заявки j, либо с моментом ухода заявки, находившейся в системе в момент t_j, уход которой освободит ожидаемые заявкой j ресурсы.

При этом время ухода самой заявки j можно определить как

$$t_j^D := \{T_i > t_j^S : j = d_i, I_i^D = 1\}. \qquad (7)$$

С помощью (6), (7) можно определить такие важные характеристики качества обслуживания заявки $j \geqslant 1$, как:

- $t_j^S - t_j$ — время ожидания заявки;
- $t_j^D - t_j^S$ — время обслуживания заявки;
- $t_j^D - t_j$ — время отклика системы;
- $\frac{t_j^D - t_j}{t_j^D - t_j^S}$ — *замедление* заявки;
- $\max\left[\frac{t_j^D - t_j}{\max(t_j^D - t_j^S, T_0)}, 1\right]$ — усеченное сверху замедление заявки (для фиксированного $T_0 > 0$);
- $\max\left[\frac{t_j^D - t_j}{N_j \max(t_j^D - t_j^S, T_0)}, 1\right]$ — усеченное сверху *нормированное замедление заявки*.

Отметим, что замедление заявки (отношение общего времени пребывания заявки в системе ко времени ее обслуживания) является характеристикой, специфичной для вычислительных кластеров [5]. Усеченное сверху, как и нормированное по количеству процессоров замедление являются вариантами характеристики замедления заявки, исключающими выбросы исходной характеристики при малых значениях времени обслуживания заявки [5, 16].

Определим теперь характеристики *производительности системы*. Важной характеристикой, связанной с дисциплиной обслуживания, является потерянная работа. При непустой очереди данная характеристика отражает простой серверов, вызванный их нехваткой для заявки, ожидающей в начале очереди (когда требуемое число серверов превышает число свободных). Определим число свободных серверов в момент T_i:

$$\psi_i = c - \sum_{j \in \mathcal{M}_i} N_j, \quad i \geqslant 1. \qquad (8)$$

Тогда потерянная работа за интервал $[T_i, T_{i+1})$ равна

$$\psi_i \tau_i R_i I\{Q_i > 0\}, \quad i \geqslant 1. \qquad (9)$$

Таким образом, потерянная работа является суммарной работой, которая могла быть выполнена на простаивающих серверах, при условии, что очередь системы непуста.

Еще одна важная характеристика производительности — уровень загрузки системы. Будем считать, что уровень загрузки равен 1, если в системе есть очередь. При этом простаивающая мощность, если она имеется, хотя и свободна, но не выполняет полезной работы. Если же очередь пуста, то уровень загрузки является отношением числа занятых к общему числу серверов. Таким образом, уровень загрузки в момент T_i равен

$$\frac{c - \psi_i I\{Q_i = 0\}}{c}, \quad i \geqslant 1. \qquad (10)$$

Наконец, рассмотрим *энергопотребление системы*. Если в момент T_i в системе находится только одна заявка и эта заявка покидает систему, то момент T_i является моментом опустошения системы. Обозначим $I_i^0 = I\{I_i^D = 1, \nu_i = 0\}$ индикатор опустошения системы в момент T_i. Очевидно, тогда в момент T_{i+1} возможен только приход заявки, а интервал $[T_i, T_{i+1})$ является интервалом простоя системы. В ином случае (т. е. при $\nu_i > 0$) система потратит за время τ_i энергию, соответствующую скорости R_i. Обозначим $e(R) = e_1 I\{R = r_1\} + e_2 I\{R = r_2\}$ функцию стоимости активного режима, где e_j есть стоимость единицы времени работы на скорости $r_j, j = 1, 2$. Тогда энергопотребление за интервал $[T_i, T_{i+1})$ будет равно

$$E_i := \tau_i \left[e_0 I_i^0 + e(R_i)(1 - I_i^0)\right], \qquad (11)$$

где e_0 есть стоимость единицы времени работы в режиме простоя.

Предложенные рекуррентные соотношения (2)–(5) для вычисления последовательных состояний системы позволяют формулировать задачи оптимизации с ограничениями как на производительность системы, так и на качество обслуживания заявок. В качестве примера приведем постановку задачи минимизации среднего энергопотребления при выполнении фиксированного числа заявок N_0, с учетом допустимого увеличения средней суммарной работы (*деградации* качества обслуживания). Обозначим среднее энергопотребление в системе с одним порогом переключения $k_1 = k_2 = k$

$$E(k) = \frac{\sum_{i=1}^{N(k)} E_i}{T_{N(k)}}, \qquad (12)$$

где $N(k) \leqslant 3N_0$ есть число базовых событий, наступивших в системе при обслуживании N_0 заявок. При этом $E(0)$ есть среднее энергопотребление в системе без управления, работающей на высокой скорости r_2. Обозначим среднюю суммарную работу в моменты прихода заявок

$$W(k) = \frac{\sum_{i=1}^{N(k)} W_i}{N(k)}. \qquad (13)$$

Тогда задача минимизации энергопотребления при ограничении на качество обслуживания

может быть сформулирована следующим образом:

$$E(k) \to \min_{k \geqslant 0},$$
$$W(k) \leqslant (1+\varepsilon)W(0),$$

где $\varepsilon \geqslant 0$ — неотрицательная константа, характеризующая деградацию качества обслуживания. Во втором разделе данной статьи представлены результаты численного эксперимента по решению данной задачи оптимизации.

Вектор нагрузки на серверы

Важной характеристикой системы является нагрузка (незавершенная работа на серверах) в момент T_i. Для исследования процесса нагрузки многопроцессорной системы применяется вектор Кифера – Вольфовица [11]. Процесс нагрузки представляет собой оставшуюся работу на серверах в момент непосредственно перед приходом очередной заявки. Отметим, что диспетчеризация заявки на серверы при построении вектора происходит сразу после прихода, поскольку заявка направляется на наименее занятый сервер. Как правило, серверы являются идентичными, и порядок серверов внутри вектора соответствует возрастанию оставшейся работы. Модификация рекурсии Кифера – Вольфовица для построения процесса нагрузки вычислительного кластера в моменты прихода заявок предложена в работе [17]. В данном разделе предложен метод построения вектора нагрузки на основе состояния системы в каждый момент времени T_i.

Рассмотрим множество неотрицательных упорядоченных векторов длины c:

$$Z = \{x \in \mathbb{R}_+^c : x_1 \leqslant \cdots \leqslant x_c\}.$$

Определим отображение $\sigma : Z \times \{1, \ldots, c\} \times \mathbb{R}_+ \to Z$, осуществляющее *планирование заявки* (диспетчеризацию заявки на серверы). Отметим, что это отображение зависит от дисциплины обслуживания. Пусть $W \in Z$ — вектор нагрузки (незавершенной работы) на серверах. Предположим, что необходимо запланировать обслуживание заявки с объемом работы $b \in \mathbb{R}_+$ и требуемым числом серверов $n \in \{1, \ldots, c\}$. В предположениях модели (уход на обслуживание в порядке поступления, дисциплина FCFS)

$$\sigma := \sigma[W, n, b] = \\ = \mathcal{R}(\underbrace{W_n + b, \ldots, W_n + b}_{n}, W_{n+1}, \ldots, W_c), \quad (14)$$

где отображение $\mathcal{R}(\cdot)$ упорядочивает компоненты вектора в порядке возрастания. Таким образом, планирование заявки с объемом работы b и требуемым числом серверов n на обслуживание осуществляется на n наименее занятых серверов в системе, при этом заявке необходимо ожидать освобождения наиболее занятого из них (ожидать выполнения работы объема W_n).

Для получения вектора нагрузки в момент T_i необходимо последовательно применить планирование ко всем заявкам $j_1, \ldots, j_{|M_i|} \in M_i$, находящимся в системе. Результатом будет вектор нагрузки $\mathcal{W}_i(|M_i|)$, получаемый с помощью следующего соотношения:

$$\mathcal{W}_i(k) = \sigma[\mathcal{W}_i(k-1), B_i(j_k), N_{j_k}], \quad (15)$$

где $k = 1, \ldots, |M_i|$, а базис индукции представлен нулевым вектором $\mathcal{W}_i(0) := \mathbf{0} \in Z$. Иными словами, (15) означает, что планирование заявки $j_k \in M_i$ осуществляется с учетом текущей нагрузки в результате диспетчеризации заявок $j_1, \ldots, j_{k-1}$, а также оставшегося времени обслуживания $B_i(j_k)$ и числа требуемых серверов N_{j_k}. При этом заявке j_k необходимо дождаться выполнения величины работы $[\mathcal{W}_i(k-1)]_{N_{j_k}}$ (т. е. N_{j_k}-й компоненты вектора нагрузки $\mathcal{W}_i(k-1)$) наиболее занятого из требующихся ей серверов.

Важно отметить связь предложенной модели с ранее исследованной моделью вычислительного кластера без управления скоростью обслуживания. В работе [15] для исследования критерия стационарности модели вычислительного кластера применялся матрично-аналитический метод. При этом состояние системы представляло собой двухкомпонентный марковский процесс $\{X(t), Y(t), t \geqslant 0\}$, где $X(t)$ есть число заявок в системе в момент t, а $Y(t)$ есть число серверов, которые требуются $\min\{X(t), c\}$ заявкам с наименьшими номерами среди находящихся в системе. Для данного процесса был предложен критерий стационарности в следующем виде:

$$\lambda \mathsf{E}S \sum_{i=1}^{c} \frac{1}{i} \sum_{j=i}^{c} p_j^{*i} \sum_{t=c-j+1}^{s} p_t < 1, \quad (16)$$

где

$$p_t = \mathsf{P}\{N = t\}, \quad t = 1, \ldots, c,$$

а p_j^{*i} есть j-я компонента i-кратной дискретной свертки вектора $p := (p_1, \ldots, p_c)$ с самим собой, т. е. $p_j^{*i} = \mathsf{P}(\hat{N}_1 + \cdots + \hat{N}_i = j)$, где $\hat{N}_i$ есть независимые копии с.в. N. (Здесь S — типичный объем работы заявки, а N — типичное число требуемых серверов.) Для получения критерия стационарности исследовалась интенсивность перехода процесса между состояниями в моменты прихода и ухода заявок.

Отметим, что предложенная модель позволяет получить значения процесса $\{X(t), Y(t)\}$ в указанные базовые моменты T_i. Действительно, предположим, что $k_1 = k_2 = 0$ (или $k_1 = k_2 = \infty$), что означает, что система всегда работает на скорости r_2 (или r_1). Тогда

$$X(T_i) = |M_i|,$$
$$Y(T_i) = (N_{j_1}, \ldots, N_{j_{\min(c,|M_i|)}}),$$
$$j_1, \ldots, j_{\min(c,|M_i|)} \in M_i, \;\; i \geqslant 1.$$

Это означает, что для исследования стационарности модели, работающей на постоянной скорости, можно применять ранее полученный критерий стационарности (16) (для пуассоновского входного потока и экспоненциального времени обслуживания).

Таким образом, предложенная модель позволяет исследовать метрики качества обслуживания вычислительного кластера с управлением скоростью обслуживающих устройств для определения оптимальной конфигурации, минимизирующей энергопотребление при контроле за качеством обслуживания.

Заметим, что для минимизации используемой памяти при проведении численных экспериментов состояние системы можно хранить в виде следующего множества:

$$\big\{T_i; I_i^A, I_i^D, I_i^S; R_i; \qquad (17)$$
$$\{A(T_{i+1}) : I_{i+1}^A = 1\} \cup \{\delta_{i+1} : I_{i+1}^D = 1\}\big\}, i \geqslant 1.$$

Таким образом, хранится только время наступления события, тип события и, при необходимости, номер приходящей/уходящей заявки. Такая запись позволяет по индукции восстановить состояние системы в каждый момент времени T_i.

Численный эксперимент

Для иллюстрации возможности анализа характеристик системы с помощью разработанной модели проведены численные эксперименты. В экспериментах рассматривается система, состоящая из $c = 10$ серверов, на вход которой поступает пуассоновский поток заявок интенсивности $\lambda = 1$. Размер работы распределен экспоненциально с $\mathsf{E}S = 1$. Количество серверов, требующихся заявке, имеет равномерное распределение $p_t = 1/c$, $t = 1, \ldots, c$. Высокая и низкая скорость обслуживания выбирались следующим образом: в первом эксперименте критерий стационарности (16) был выполнен как для системы, работающей только на скорости r_1, так и для системы, работающей на скорости r_2. Во втором эксперименте критерий (16) был нарушен для системы, работающей на скорости r_1. Таким образом, использовались следующие пары значений: для первого эксперимента $r_1 = 0.9, r_2 = 1.5$; для второго эксперимента $r_1 = 0.6, r_2 = 1.5$. В каждом эксперименте фиксировалось значение $k_1 = k_2 = k \in \{1, \ldots, 50\}$, затем генерировалось 100 траекторий процесса обслуживания, в каждой траектории проходило обслуживание 20 000 заявок. Стоимость энергопотребления в режиме простоя, на низкой и высокой скорости была выбрана следующим образом: $e_0 = 1, e_1 = 2, e_2 = 4$. Полученное в результате имитационного моделирования значение среднего энергопотребления (12) и средней суммарной работы (13) усреднялось по всем траекториям для данного k. Расчеты производились на кластере ЦКП КарНЦ РАН «Центр высокопроизводительной обработки данных». Результаты экспериментов представлены графически в форме зависимости $E(k)$ и $W(k)$ от $k = 1, \ldots, 50$. Приведем для сравнения базовые значения энергопотребления и качества обслуживания в системе без управления, работающей на скорости r_2 (напомним, при этом $k = 0$): $E(0) = 2.822888$, $W(0) = 1.636116$; и на скорости r_1 ($k = \infty$): $E(\infty) = 1.892117$, $W(\infty) = 6.528911$.

На рисунке 1 представлен результат первого численного эксперимента. Видно, что с ростом значения порога переключения скорости k средние значения энергопотребления и качества обслуживания стремятся к $E(\infty)$ и $W(\infty)$ соответственно.

На рисунке 2 представлен результат второго численного эксперимента. Видно, что с ростом значения порога переключения скорости k среднее энергопотребление стремится к постоянной величине из интервала $(E(0), E(\infty))$. В то же время в связи с нестационарностью системы, работающей на низкой скорости, средняя суммарная работа линейно растет.

Заключение

Предложена модель вычислительного кластера на основе рекуррентных соотношений, описывающих последовательные изменения состояния системы. Эта модель позволяет формализовать широкий круг систем с управлением скоростью обслуживания. Небольшие изменения ключевых соотношений модели позволяют охватить системы с различными дисциплинами обслуживания. Кроме того, модель может быть адаптирована для исследования эффекта от применения режимов пониженного энергопотребления ACPI. В дальнейшем планируется исследовать монотонность характеристик системы относительно управляющих последовательностей и ключевых параметров.

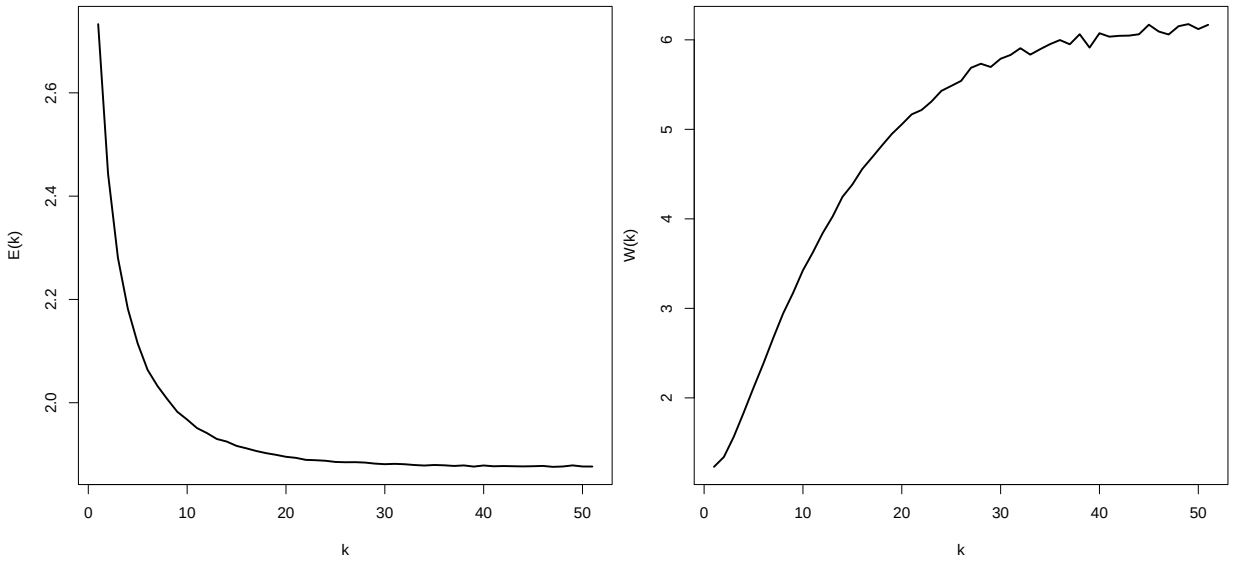

Рис. 1. Среднее энергопотребление (левая часть), средняя суммарная работа (правая часть) в 10-серверной системе. Критерий стационарности выполнен для системы при $r_1 = 0.9$ и $r_2 = 1.5$

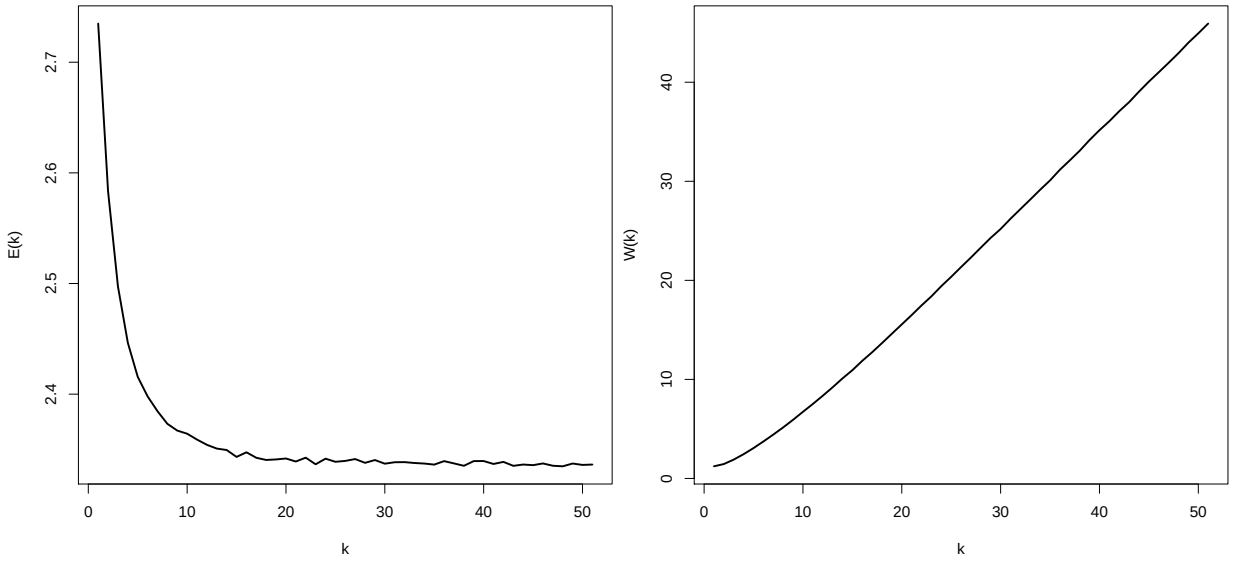

Рис. 2. Среднее энергопотребление (левая часть), средняя суммарная работа (правая часть) в 10-серверной системе. Критерий стационарности нарушен при скорости $r_1 = 0.6$ и выполнен при скорости $r_2 = 1.5$

Работа поддержана грантом Президента РФ МК-1641.2017.1, грантами РФФИ 15–07–02341, 15–07–02354, 15–07–02360, 16–07–00622, 15–29–07974.

Литература

1. *Bekker R., Borst S. C., Boxma O. J., Kella O.* Queues with workload-dependent arrival and service rates // Queueing Systems. 2004. Vol. 46. P. 537–556. doi: 10.1023/B:QUES.0000027998.95375.ee

2. *Brill P. H., Green L.* Queues in which customers receive simultaneous service from a random number of servers: a system point approach // Management Science. 1984. Vol. 30, no. 1. P. 51–68. doi: 10.1287/mnsc.30.1.51

3. *Chakravarthy S. R., Karatza H. D.* Two-server parallel system with pure space sharing and Markovian arrivals // Computers & Operations Research. 2013. Vol. 40, no. 1. P. 510–519. doi: 10.1016/j.cor.2012.08.002

4. *Evans R. V.* Queuing when Jobs Require Several Services which Need Not be Sequenced // Management Science. 1964. Vol. 10, no. 2. P. 298–315. doi: 10.1287/mnsc.10.2.298

5. *Feitelson D. G.* Metrics for parallel job scheduling and their convergence // Lecture Notes in Computer Science. Job Scheduling Strategies for Parallel Processing. 2001. Vol. 2221. P. 188–205. doi: 10.1007/3-540-45540-X_11

6. *Feitelson D. G.* Workload modeling for computer systems performance evaluation. Cambridge University Press, 2015. doi: 10.1017/CBO9781139939690

7. *Gandhi A., Harchol-Balter M., Das R., Lefurgy C.* Optimal power allocation in server farms // ACM SIGMETRICS Performance Evaluation Review. 2009. Vol. 37. P. 157–168. doi: 10.1145/1555349.1555368

8. *Gebrehiwot M. E., Aalto S. A., Lassila P.* Optimal sleep-state control of energy-aware M/G/1 queues // Proceed. of the 8th Int. Conf. on Performance Evaluation Methodologies and Tools. 2014. P. 82–89. doi: 10.4108/icst.valuetools.2014.258149

9. *Horvath T., Skadron K.* Multi-mode energy management for multi-tier server clusters // Proceed. of the 17th Int. Conf. on Parallel Architectures and Compilation Techniques. 2008. P. 270–279. doi: 10.1145/1454115.1454153

10. *Hyytiä E., Righter R., Aalto S.* Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure // Performance Evaluation. 2014. Vol. 75–76. P. 17–35. doi: 10.1016/j.peva.2014.01.002

11. *Kiefer J., Wolfowitz J.* On the theory of queues with many servers // Transactions of the American Mathematical Society. 1955. P. 1–18. doi: 10.1090/S0002-9947-1955-0066587-3

12. *Kim S. S.* M/M/s Queueing System Where Customers Demand Multiple Server Use, Ph.D. Dissertation, Southern Methodist University, 1979.

13. *Morozov E., Rumyantsev A.* A State-Dependent Control for Green Computing // Lecture Notes in Electrical Engineering. Information Sciences and Systems. 2015. Vol. 363. P. 57–67. doi: 10.1007/978-3-319-22635-4_5

14. *Morozov E., Rumyantsev A.* Stability Analysis of a MAP/M/s Cluster Model by Matrix-Analytic Method // Lecture Notes in Computer Science. Computer Performance Engineering: 13th European Workshop. 2016. Vol. 9951. P. 63–76. doi: 10.1007/978-3-319-46433-6_5

15. *Rumyantsev A., Morozov E.* Stability criterion of a multiserver model with simultaneous service // Annals of Operations Research. 2017 (First Online: 2015). Vol. 252, no. 1. P. 29–39. doi: 10.1007/s10479-015-1917-2

16. *Zotkin D., Keleher P. G.* Job-length estimation and performance in backfilling schedulers // Proceed. of the Eighth Int. Symposium on High Performance Distributed Computing. 1999. P. 236—243. doi: 10.1109/HPDC.1999.805303

17. *Морозов Е. В., Румянцев А. С.* Модели многосерверных систем для анализа вычислительного кластера // Труды Карельского научного центра РАН. 2011. № 5. С. 75–85.

Поступила в редакцию 09.06.2017

References

1. *Bekker R., Borst S. C., Boxma O. J., Kella O.* Queues with workload-dependent arrival and service rates. *Queueing Systems.* 2004. Vol. 46. P. 537–556. doi: 10.1023/B:QUES.0000027998.95375.ee

2. *Brill P. H., Green L.* Queues in which customers receive simultaneous service from a random number of servers: a system point approach. *Management Science.* 1984. Vol. 30, no. 1. P. 51–68. doi: 10.1287/mnsc.30.1.51

3. *Chakravarthy S. R., Karatza H. D.* Two-server parallel system with pure space sharing and Markovian arrivals. *Computers & Operations Research.* 2013. Vol. 40, no. 1. P. 510–519. doi: 10.1016/j.cor.2012.08.002

4. *Evans R. V.* Queuing when Jobs Require Several Services which Need Not be Sequenced. *Management Science.* 1964. Vol. 10, no. 2. P. 298–315. doi: 10.1287/mnsc.10.2.298

5. *Feitelson D. G.* Metrics for parallel job scheduling and their convergence. *Lecture Notes in Computer Science. Job Scheduling Strategies for Parallel Processing.* 2001. Vol. 2221. P. 188–205. doi: 10.1007/3-540-45540-X_11

6. *Feitelson D. G.* Workload modeling for computer systems performance evaluation. Cambridge University Press, 2015. doi: 10.1017/CBO9781139939690

7. *Gandhi A., Harchol-Balter M., Das R., Lefurgy C.* Optimal power allocation in server farms. *ACM SIGMETRICS Performance Evaluation Review.* 2009. Vol. 37. P. 157–168. doi: 10.1145/1555349.1555368

8. *Gebrehiwot M. E., Aalto S. A., Lassila P.* Optimal sleep-state control of energy-aware M/G/1 queues. *Proceed. of the 8th Int. Conf. on Performance Evaluation Methodologies and Tools.* 2014. P. 82–89. doi: 10.4108/icst.valuetools.2014.258149

9. *Horvath T., Skadron K.* Multi-mode energy management for multi-tier server clusters. *Proceed. of the 17th Int. Conf. on Parallel Architectures and Compilation Techniques.* 2008. P. 270–279. doi: 10.1145/1454115.1454153

10. *Hyytiä E., Righter R., Aalto S.* Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure. *Performance Evaluation.* 2014. Vol. 75–76. P. 17–35. doi: 10.1016/j.peva.2014.01.002

11. *Kiefer J., Wolfowitz J.* On the theory of queues with many servers. *Transactions of the American Mathematical Society.* 1955. P. 1–18. doi: 10.1090/S0002-9947-1955-0066587-3

12. *Kim S. S.* M/M/s Queueing System Where Customers Demand Multiple Server Use, Ph.D. Dissertation, Southern Methodist University, 1979.

13. *Morozov E., Rumyantsev A.* A State-Dependent Control for Green Computing. *Lecture Notes in Electrical Engineering. Information Sciences and Systems.* 2015. Vol. 363. P. 57–67. doi: 10.1007/978-3-319-22635-4_5

14. *Morozov E., Rumyantsev A.* Stability Analysis of a MAP/M/s Cluster Model by Matrix-Analytic Method. *Lecture Notes in Computer Science. Computer Performance Engineering: 13th European Workshop.* 2016. Vol. 9951. P. 63–76. doi: 10.1007/978-3-319-46433-6_5

15. *Rumyantsev A., Morozov E.* Stability criterion of a multiserver model with simultaneous service. *Annals of Operations Research.* 2017 (First Online: 2015). Vol. 252, no. 1. P. 29–39. doi: 10.1007/s10479-015-1917-2

16. *Zotkin D., Keleher P. G.* Job-length estimation and performance in backfilling schedulers. *Proceed. of the Eighth Int. Symposium on High Performance Distributed Computing.* 1999. P. 236—243. doi: 10.1109/HPDC.1999.805303

17. *Morozov E. V., Rumyantsev A. S.* Modeli mnogoservernykh sistem dlya analiza vychislitel'nogo klastera [Multi-server models to analyze high performance cluster]. *Trudy KarNTs RAN* [*Trans. KarRC RAS*]. 2011. No. 5. P. 75–85.

Received June 9, 2017

СВЕДЕНИЯ ОБ АВТОРАХ:

Румянцев Александр Сергеевич
научный сотрудник
Институт прикладных математических исследований Карельского научного центра РАН
ул. Пушкинская, 11, Петрозаводск,
Республика Карелия, Россия, 185910
эл. почта: ar0@krc.karelia.ru
тел.: (8142) 763370

Калинина Ксения Алексеевна
аспирант
Институт прикладных математических исследований Карельского научного центра РАН
ул. Пушкинская, 11, Петрозаводск,
Республика Карелия, Россия, 185910
эл. почта: kalininaksenia90@gmail.com
тел.: (8142) 763370

Морозова Таисия Евсеевна
студентка
Петрозаводский государственный университет
пр. Ленина, 33, Петрозаводск, Республика Карелия, Россия, 185910
эл. почта: tiamorozova@mail.ru
тел.: (8142) 719606

CONTRIBUTORS:

Rumyantsev, Alexander
Institute of Applied Mathematical Research,
Karelian Research Centre, Russian Academy of Sciences
11 Pushkinskaya St., 185910 Petrozavodsk,
Karelia, Russia
e-mail: ar0@krc.karelia.ru
tel.: (8142) 763370

Kalinina, Ksenia
Institute of Applied Mathematical Research,
Karelian Research Centre, Russian Academy of Sciences
11 Pushkinskaya St., 185910 Petrozavodsk,
Karelia, Russia
e-mail: kalininaksenia90@gmail.com
tel.: (8142) 763370

Morozova, Taisia
Petrozavodsk State University
33 Lenin Pr., 185910 Petrozavodsk, Karelia, Russia
e-mail: tiamorozova@mail.ru
tel.: (8142) 719606