Задача прогнозирования нагрузки для повышения энергетической эффективности вычислительного кластера*

Е.Е. Ивашко, А.С. Румянцев, А.Л. Чухарев

Федеральное государственное бюджетное учреждение науки
Институт прикладных математических исследований
Карельского научного центра Российской академии наук

Исследуется задача прогнозирования нагрузки для построения системы повышения энергоэффективности вычислительного кластера. Предложены величины, отражающие качество восприятия системы пользователем, способ построения прогноза и метод оценки качества прогноза.

Введение

Многопроцессорные системы (МС) представляют собой активно развивающееся в настоящее время направление наращивания мощности вычислителей [1]. Среди МС следует выделить вычислительные кластеры (ВК) и системы распределенных вычислений (СРВ). Архитектура ВК такова, что множество процессорных ядер, оперативная память, дисковая подсистема, быстрая вычислительная сеть воспринимаются пользователем как единый аппаратный ресурс. ВК позволяют вести одновременный расчет заявки на множестве процессоров. СРВ (например, грид-системы) обладают менее тесно связанными вычислителями. Типичная заявка для СРВ состоит из группы относительно независимых заданий, каждое из которых выполняется на отдельном процессоре.

Для целей оптимального разделения ресурсов МС между конкурирующими заявками, в МС используются менеджеры очередей (МО). Это программные системы, которые на основе данных о текущей нагрузке в МС и некоторой дополнительной информации управляют выбором задач из очереди, вычислением, приостановкой, завершением задач с учетом относительных приоритетов. Для МО характерно два типа поведения: нацеленные на получение наиболее плотного расписания и нацеленные на снижение времени отклика системы. Использование МО первого типа приводит к росту загрузки системы, однако, с точки зрения пользователя такая система может не являться комфортной для работы. МО второго типа нацелены на повышение комфорта пользователя, в то же время это может приводить к появлению простоев оборудования и снижению загрузки (за счет эффекта обратной связи загрузка в такой системе может увеличиваться с повышением комфорта пользователя). Таким образом, целью владельца, как правило, является эксплуатация оборудования без простоя, поскольку эксплуатационные расходы кластера являются весьма высокими. Целью пользователя является минимизация времени отклика.

ВК обладают высоким энергопотреблением, в то же время, существенная часть ВК недогружена [2], что дает возможность временно переводить часть устройств в режим пониженного энергопотребления (РПЭ). Нахождение компромисса между загрузкой кластера, временем отклика системы и энергопотреблением системы — комплексная задача, подразумевающая оптимальную настройку параметров системы, выбор МО, взаимодействие с пользователями. В этой связи отметим систему Moab HPC Suite Enterprise Edition [3], которая, по заявлению производителя, частично решает проблемы снижения энергопотребления системы за счет более плотной упаковки расписания, прогнозирования рабочих циклов

*Исследования выполнены при поддержке РФФИ (проект 12-07-31147) и Фонда содействия малым формам предпринимательства в научно-технической сфере (г/к №10491р/16862 от 08.06.2012).

нагрузки и выбора для расчета задачи наиболее энергоэффективных (с максимальным соотношением числа флопсов на ватт) узлов. Однако, такие системы повышения энергоэффективности (СПЭ), как правило, работают со сложным набором правил (см. также [4]), т. е. требуют индивидуального подхода в каждом отдельном случае.

1. Вероятностное моделирование процесса нагрузки ВК

Большой поток вычислительных задач и заранее неизвестное время выполнения задачи в МС приводят к необходимости использования вероятностных моделей для описания функционирования МС. Параллельные вычисления в качестве основы для вероятностной модели впервые упомянуты в работе [10]. Отметим работы [5–7], исследующие возможности моделирования ВК, а также работу [8], в которой проведено сравнение имеющихся моделей ВК на основе данных реальных лог-файлов ВК. Наконец, отметим работу [9], которая содержит обзор имеющихся в литературе моделей массово-параллельных МС, а также анализ некоторой новой модели на основе марковских цепей.

При этом, как правило, рассматривается следующая общая постановка задачи. В МС в случайные моменты времени $t_i, i \geqslant 1$ поступает поток заявок. Заявке i требуется (случайное или детерминированное) число процессоров N_i для выполнения заданий (в дальнейшем используем термин процессор для разделяемых вычислителей, которыми могут быть процессорные ядра, графические ускорители и пр.). Если $N_i \equiv 1$, говорят о классической МС типа GI/G/m. Если заявке с номером i одновременно требуется случайное число $N_i \geqslant 1$ процессоров (т. е. задания должны начать выполнение одновременно), то системы можно разделить на МС с независимым освобождением процессоров [11, 12] и МС с одновременным освобождением процессоров [13, 14]. В системах второго типа (наиболее подходящих для описания ВК) времена обслуживания на всех N_i процессорах идентичны, т. е. используется одна и та же реализация времени обслуживания S_i, что существенно усложняет анализ [14]. Для таких систем известны лишь численные результаты [13], а при отсутствии буфера — также некоторые аналитические результаты [10, 15, 16].

В работах [17, 18] предложена и исследована новая модель ВК, представляющая собой модификацию модели Кифера–Вольфовица для вектора нагрузки W_i в момент прихода заявки i, которой требуется одновременно N_i процессоров на время S_i. Рекурсия для вектора нагрузки имеет вид

$$W_{i+1} = R(W_{i,N_i} + S_i - T_i, \ldots, W_{i,N_i} + S_i - T_i, W_{i,N_i+1} - T_i, \ldots, W_{i,m} - T_i)^+, \qquad (1)$$

где $R(\cdot)$ есть отображение упорядочивания координат вектора по возрастанию и $(\cdot)^+ = \max(\cdot, 0)$. Проверка адекватности модели проводилась с использованием численного моделирования на основе лог-файлов ВК. Отметим, что основную сложность при использовании модели (1) составляет построение модели управляющей последовательности $\{T_i, S_i, N_i\}$, которая обладает сложной структурой взаимосвязей между с. в., не являющимися н. о. р.

Напомним особенности функционирования m-процессорного ВК в режиме разделения доступа между пользователями из множества пользователей кластера $U = \{u_1, \ldots, u_n\}$ с использованием сессий удаленного доступа. Пусть $\tau_i(u), i \geqslant 1$ есть времена начала сессий доступа к ВК и $\sigma_i(u)$ есть продолжительность сессии с номером i для пользователя u. Тогда активные в момент времени t сессии пользователя u (т.е. такие, что $\tau_i(u) \leqslant t \leqslant \tau_i(u) + \sigma_i(u)$) порождают поток заявок, каждая из которых характеризуется временем прихода $t_j(u)$, требуемым числом процессоров $N_j(u)$ и временем обслуживания $S_j(u)$ (которое может быть заранее не известно, либо известно приближенно). Суперпозиция потоков заявок от всех активных пользователей в каждый момент времени представляет собой управляющий поток заявок в системе. Будем называть загрузкой ВК величину

$$\rho = \lambda \frac{E(SN)}{m},$$

где λ есть интенсивность входного потока заявок (в случае однородного потока $\lambda = \frac{1}{ET}$), а без индексов обозначены типичные элементы последовательностей случайных величин.

2. Прогнозирование нагрузки ВК

Поскольку модель процесса нагрузки хорошо описывает поведение ВК на небольших, не сильно нагруженных системах [19], то указанную модель можно использовать в качестве источника информации о поведении процесса нагрузки при заданной реализации управляющей последовательности заявок. В случае, когда используется особая дисциплина обслуживания, несколько очередей или имеются другие особенности системы, возможно использование МО в режиме симуляции [20].

2.1. Оценка качества обслуживания на ВК

В качестве метрик, отражающих качество обслуживания на вычислительном кластере, как правило, используют следующие величины [2]: время отклика системы $D_i + S_i$; замедление заявки $\frac{D_i+S_i}{S_i}$; загрузка системы ρ; допустимая загрузка системы $\rho_{\max}$, не приводящая к неограниченному росту очереди. Отметим, что интенсивность входного потока заявок λ также косвенно отражает качество обслуживания на кластере. Это связано с наличием эффекта обратной связи между потоком заявок и состоянием системы [2].

Пусть в результате функционирования СПЭ на ВК часть устройств переводится в РПЭ, время выхода из которых различается. В случае необходимости задействования таких ресурсов для обслуживания заявки i обозначим $V_i \geqslant 0$ время до полной готовности всех ресурсов к обслуживанию данной заявки. Отметим, что в первом приближении можно считать распределение V_i дискретным. Тогда время отклика системы будет составлять величину $D_i + V_i + S_i$, среднее замедление заявки составит $\frac{D_i+V_i+S_i}{S_i}$. При этом с ростом значений V_i повышается экономия энергии (т.к. устройства будут дольше находиться в РПЭ), но ухудшается качество обслуживания в системе с точки зрения пользователя.

Необходимо отметить, что в некоторых случаях может наблюдаться одновременный рост средней загрузки в системе и снижение среднего замедления, либо снижение средней загрузки и рост среднего замедления [22]. Поэтому целесообразно определить величину, отражающую качество восприятия (Quality of Experience) системы пользователем на заявке с номером i следующим образом:

$$R_i = \alpha(D_i + V_i + S_i) + \beta\frac{D_i + S_i + V_i}{S_i},$$

где α, β— некоторые параметры, $0 \leqslant \alpha, \beta \leqslant 1,\ \alpha + \beta = 1$. В пределе при $\alpha = 1$ метрика R_i соответствует времени отклика системы для заявки i, а при $\beta = 1$ соответствует замедлению заявки. Тогда качество обслуживания есть некоторая функция $F(R_1, \ldots, R_n)$. В простейшем случае можно полагать

$$F(R_1, \ldots, R_n) = \frac{1}{n}\sum_{i=1}^{n} R_i.$$

Для оценки изменения качества обслуживания в системе с введением системы повышения энергетической эффективности можно воспользоваться отношением

$$\frac{F(R_1, \ldots, R_n)}{F(\hat{R}_1, \ldots, \hat{R}_n)},$$

где $\hat{R}_i = \alpha(D_i + S_i) + \beta\frac{D_i+S_i}{S_i}$ есть качество восприятия на заявке i без использования системы повышения энергоэффективности. Тогда можно полагать, что указанная система

не причиняет ущерб качеству обслуживания, если

$$P\left(\frac{F(R_1,\ldots,R_n)}{F(\hat{R}_1,\ldots,\hat{R}_n)}\in[1,1+\varepsilon)\right)=1-\delta,$$

где $0\leqslant\varepsilon,\delta\ll 1$ — некоторые параметры.

2.2. Метод построения прогноза

Предположим, что в момент времени T в системе имеется свободный ресурс из K устройств. Если в системе нет очереди, то число свободных устройств до ближайшего прихода заявки не уменьшится, а в момент прихода заявки это значение может уменьшиться. Поэтому целесообразно принимать решение о переводе устройств в РПЭ до ближайшего прогнозируемого прихода. Если же в системе очередь заявок не пуста, то ближайшее уменьшение числа свободных устройств возможно либо в момент ближайшего прихода заявки (если $N_j<K$ и используется дисциплина обслуживания, отличная от FIFO), либо в момент ближайшего ухода заявки и начала обслуживания следующих по порядку очереди заявок. Таким образом, перевод устройств в РПЭ может производиться до минимального по времени момента из двух: ближайшего прогнозируемого прихода и ближайшего прогнозируемого ухода. Обозначим $\hat{t}$ момент времени, в который в соответствии с прогнозом ожидается уменьшение свободного ресурса устройств. Пусть $\hat{N}$ есть ожидаемый размер уменьшения ресурса в момент $\hat{t}$. Целесообразно производить построение прогноза в момент T окончания обслуживания очередной заявки (при этом могут начать обслуживание одновременно несколько заявок из очереди). Если в момент $T+0$ остался свободный ресурс, то можно планировать его перевод в РПЭ на время $\hat{t}-T$. Корректировка прогноза может производиться в следующие ключевые моменты: приход или уход очередной заявки, начало или окончание сессии пользователя.

Предположим, что каждое устройство имеет несколько режимов, C_0 (режим обычной работы), $C_1,\ldots,C_k$ (РПЭ, больший индекс соответствует большей экономии). Каждый РПЭ характеризуется величиной t_k времени, такого что на время, меньшее t_k, переход в режим C_k нецелесообразен (в частности, t_i связано с временем ухода и возвращения из состояния C_i). Тогда, на основании прогноза $\hat{t},\hat{N}$ необходимо построить вектор $\kappa=(\kappa_0,\ldots,\kappa_k)\in\mathbb{Z}_+^{k+1}$, такой что $\kappa_0+\cdots+\kappa_k=K$, в котором κ_i есть количество устройств, которое необходимо перевести в состояние C_i, а также вектор $\tilde{t}=(\tilde{t}_0,\ldots,\tilde{t}_k),0\leqslant\tilde{t}_i\leqslant\hat{t}$, состоящий из времен выхода из соответствующих состояний. Таким образом,

$$(\kappa,\tilde{t})=\varkappa(\hat{t},\hat{N},t_1,\ldots,t_k).$$

В простейшем случае можно определить функцию $\varkappa$, исходя из следующих соотношений. Если $\hat{t}<t_1$, то $\kappa=(K,0,\ldots,0),\tilde{t}=(\hat{t},0,\ldots,0)$. Если $t_i\leqslant\hat{t}\leqslant t_{i+1}$, то $\kappa_i=K-\hat{N},\kappa_{i-1}=\hat{N},\tilde{t}_i=\hat{t},\tilde{t}_{i-1}=\hat{t}-t_{i-1}$, а все остальные величины искомых векторов нулевые. В более общем случае, функция $\varkappa$ позволяет управлять качеством обслуживания в системе, поэтому более точное определение функции $\varkappa$ является предметом дальнейших исследований, в том числе с использованием численных экспериментов.

Опишем более подробно способ построения величин $\hat{t}$ и $\hat{N}$. Рассмотрим систему, в которой очередь пуста. Тогда в момент построения прогноза T пользователь $u\in U$ является активным, если найдется индекс i такой, что $\tau_i(u)\leqslant T\leqslant\tau_i(u)+\sigma_i(u)$, в противном случае пользователь не активен. Для активного пользователя u известно ближайшее к моменту T время постановки задачи в очередь, т.е. $\max\{t_j(u):t_j(u)<T\}$. Для не активного пользователя u известно время окончания ближайшей сессии, $\tau_i(u)+\sigma_i(u)$. Для каждого активного пользователя произведем реализацию $\hat{t}(u)-t_i(u)$ случайной величины (с.в.) с ф.р. $F_A(x)$, т.е. возможное время до прихода ближайшей заявки от данного пользователя, а также реализацию с.в. $\hat{\sigma}(u)$ с ф.р. $F_{ON}(x)$ длительности активной сессии. Если $\hat{t}(u)>t_i(u)+\hat{\sigma}(u)$, то

пользователь завершит сессию до постановки задачи, поэтому его необходимо рассмотреть повторно вместе с не активными. В противном случае величина $\hat{t}(u)$ представляет собой ближайшее время прихода задачи от данного пользователя. Одновременно необходимо произвести реализацию величины $\hat{N}(u)$ числа требуемых процессоров для заявки, пришедшей в момент времени $\hat{t}(u)$ от пользователя u (отметим также, что для особых дисциплин обслуживания также необходимо произвести выборку из распределения $F_B(x)$ прогнозируемого времени выполнения $\hat{S}(u)$ ближайшей заявки от пользователя u для расстановки относительных приоритетов в очереди). Далее для каждого не активного пользователя реализуем величину $\hat{\tau}(u) - \tau_i(u) - \sigma_i(u)$ с ф.р. $F_{OFF}(x)$ длительности периода неактивности пользователя, тем самым получаем момент начала новой сессии, а также аналогично случаю с активным пользователем находим время прихода ближайшей заявки данного пользователя $\hat{t}(u)$ реализацией с.в. $\hat{t}(u) - \hat{\tau}(u)$ с ф.р. $F_A(x)$ (отметим, что первый интервал прихода после начала новой сессии может иметь другое распределение, в стационарном случае ф.р. имеет вид $\frac{1}{E(\hat{t}(u)-\hat{\tau}(u))}\int_0^x F_A(y)\,dy$). Аналогично, реализуем $\hat{N}(u)$. При этом предполагается отсутствие сессий без постановки задачи в очередь. Тогда прогнозируемые значения $\hat{t}, \hat{N}$ могут быть вычислены как значения функционалов

$$\hat{t} = \varphi(\hat{t}_1(U), \ldots, \hat{t}_n(U)),$$
$$\hat{N} = \psi(\hat{N}_1(u), \ldots, \hat{N}_n(u))$$

где $\hat{t}_i(U)$ есть вектор ближайших прогнозируемых приходов заявок пользователей U после момента времени T, полученный в i-той реализации описанного выше процесса ($N_i(U)$ в очевидных обозначениях). Отметим, что точность прогноза зависит как от числа реализаций n, так и от вида функционалов $\varphi(\cdot), \psi(\cdot)$. В простейшем случае можно полагать $\phi(\cdot) = \psi(\cdot) = \min(\cdot)$. В то же время можно предложить также более гибкий и одновременно простой вид функционала, который позволяет управлять энергоэффективностью, а именно

$$\varphi(\hat{t}_1(U), \ldots, \hat{t}_n(U)) = \min_{1 \leqslant i \leqslant n} \hat{t}_i^{(j)}(U),$$

где $x^{(j)}$ есть j-я порядковая статистика вектора x. Меняя значение j, можно (при возрастании j) увеличивать энергоэффективность, одновременно повышая риск нарушения прогноза и снижая качество обслуживания. Функционал $\psi(\cdot)$ можно определить аналогично.

Отличие случая с непустой очередью заключается в том, что кроме вычисления прогнозируемых векторов $\hat{t}_i(U), 1 \leqslant i \leqslant n$ приходов ближайших заявок и размеров $\hat{N}_i(U)$, необходимо также прогнозировать ближайшее время завершения вычисления заявки на ВК (среди заявок, уже занимающих ресурсы ВК), а также рассматривать заявки, которые уже находятся в очереди, исследуя возможность их постановки на выполнение в момент ближайшего прогнозируемого ухода заявки. При этом необходимо учитывать возможность одновременного начала выполнения нескольких заявок на освободившемся ресурсе. Тогда ближайшее прогнозируемое время $\hat{t}$ уменьшения свободного ресурса необходимо вычислять в виде функционала вида

$$\hat{t} = \varphi'(\hat{t}_1(U), \ldots, \hat{t}_n(U), \hat{S}, N_{i_1}, \ldots, N_{i_s}),$$

где $\hat{t}_i(U)$ есть вектор ближайших прогнозируемых приходов заявок от пользователей U (определяемый аналогично случаю пустой очереди), $\hat{S}$ есть ближайшее прогнозируемое время ухода заявки с обслуживания, $\hat{\nu}$ есть величина свободного ресурса в момент ухода заявки (с учетом освободившихся процессоров), а $N_{i_1}, \ldots, N_{i_s}$ есть число процессоров, требуемых заявкам $i_1, \ldots, i_s$, находящимся в очереди в момент времени T. Прогнозируемое уменьшение ресурса $\hat{N}$ определяется аналогично в виде функционала

$$\hat{N} = \psi'(\hat{t}_1(U), \ldots, \hat{t}_n(U), \hat{S}, N_{i_1}, \ldots, N_{i_s}),$$

где в простейшем случае при реализации события {ближайший уход состоится раньше, чем ближайший приход} величина $\hat{N}$ определяется из соотношения

$$\hat{N} = \max_{b_k \in \{0,1\}, 1 \leqslant k \leqslant s} \left\{ \nu = \sum_{j=1}^{s} b_j N_{i_j} : \nu < \hat{\nu} \right\}.$$

2.3. Оценка качества прогноза нагрузки

Пусть в момент времени T появляется возможность перевести часть устройств в РПЭ. Тогда СПЭ должна с учетом прогноза $(\kappa, \tilde{t})$ рассчитать времена $\delta = (\delta_1, \ldots, \delta_m)$ простоя процессоров с номерами $i = 1, \ldots, m$. В случае более раннего, чем прогнозируемое, востребования ресурсов некоторых процессоров, СПЭ должна фиксировать максимальное время V_j до полной готовности всех ресурсов к принятию вновь пришедшей заявки с номером j, что отразится на качестве обслуживания в системе. Предположим, что заявка j поступает в систему в момент времени $t_j > T$. Пусть $b(j) = (b_1(j), \ldots, b_m(j))$, где $b_i(j) = 1$, если заявке j требуется устройство i и $b_i(j) = 0$ иначе, $i = 1, \ldots, m$. Тогда метрика ошибки прогноза нагрузки в момент прихода заявки j может быть некоторой функцией вида

$$Q(T, \delta, b(j), t_j, V_j).$$

Отметим, что целесообразно в качестве метрики для ошибки прогноза для узла i принять величину $\frac{(\delta_i - t_j + T)^+}{\delta_i}$, где $(\cdot)^+ = \max(0, \cdot)$. Тогда можно предложить в качестве метрики для ошибки прогноза использовать величину

$$Q(T, \delta, b(j), t_j, V_j) = C V_j \sum_{i=1}^{m} b_i(j) \frac{(\delta_i - t_j + T)^+}{\delta_i},$$

где в качестве константы C можно положить, например, $\frac{1}{|b(j)|}$. Тогда если прогноз нарушен незначительно ($t_j - T \approx \delta_i$ и $\delta_i \gg 0$), то $Q(T, \delta, b(j), t_j, V_j) \approx 0$, т.е. ошибка прогноза незначительна. Если же $t_j - T \approx 0$ и $\delta_i \gg 0$, то ошибка прогноза будет соответствовать накладным расходам от использования системы снижения энергопотребления, т.е. $Q(T, \delta, b(j), t_j, V_j) \approx V_j$. В то же время, если $V_j \approx 0$, то ошибку прогноза можно считать незначительной.

Для оценивания качества прогноза нагрузки, необходимо минимизировать ошибку прогноза на достаточно большом промежутке времени, что требует введения некоторой оценки, например, интегрального вида

$$Q = \int_{T_1}^{T_2} Q(T, \delta, b(j), t_j, V_j)\, dT.$$

2.4. Оценка энергетической эффективности

Наиболее популярной среди метрик оценки энергоэффективности ВК является PUE (Power Usage Efficiency) [21], которая отражает отношение энергии, потребляемой всем вычислительным комплексом (включая инфраструктуру) к энергии, потребляемой IT-компонентами комплекса (вычислителями, системами хранения, сетевой компонентой):

$$\text{PUE} = \frac{\text{Total facility power}}{\text{IT equipment power}}.$$

Необходимо отметить, что при внедрении СПЭ будет происходить рост PUE в связи с уменьшением суммарного и мгновенного значений энергопотребления. Таким образом, данный коэффициент будет некорректно отражать экономический эффект от внедрения указанной

системы. Поэтому для применения на ВК с СПЭ можно предложить метрику, отражающую коэффициент полезного действия системы повышения энергоэффективности комплекса в виде отношения суммарного энергопотребления за некоторый период времени с установленной системой к суммарному потреблению за этот же период без нее, т.е.

$$\frac{E(V,\rho)}{E_0}.$$

3. Постановка задачи и перспективы исследования

Суммируя вышеизложенное, можно поставить следующую задачу оптимизации: минимизация суммарного энергопотребления при сохранении качества обслуживания.

$$E(V,\rho) \to \min,$$

$$P\left(\frac{F(R_1,\ldots,R_n)}{F(\hat{R}_1,\ldots,\hat{R}_n)} \in [1,1+\varepsilon)\right) = 1-\delta.$$

В дальнейших исследованиях планируется провести ряд численных экспериментов для оценки качества прогноза нагрузки на основе данных лог-файлов ВК, в частности, ВК ЦКП КарНЦ РАН «Центр высокопроизводительной обработки данных» [23] с целью подбора оптимального вида функционалов, влияющих на качество прогноза, а также оптимальных значений параметров СПЭ.

Литература

1. Parkhurst J., Darringer J., Grundmann B. From single core to multi-core: preparing for a new exponential // Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design. ICCAD'06. New York, NY, USA: ACM, 2006. Pp. 67–72.

2. Feitelson D. G. Workload modeling for computer systems performance evaluation (web draft): URL: http://www.cs.huji.ac.il/ feit/wlmod/wlmod.pdf (дата обращения 06.12.2012)

3. Adaptive Computing: URL: http://www.adaptivecomputing.com (дата обращения 06.12.2012)

4. HP.com - HP Power Management Software: URL: http://h18004.www1.hp.com/products/servers/management/dynamic-power-capping/index.html (дата обращения: 17.08.2012).

5. Jann J., Pattnaik P., Franke H. et al. Modeling of Workload in MPPs // Job Scheduling Strategies for Parallel Processing, IPPS'97 Workshop, Geneva, Switzerland, April 5, 1997, Proceedings. Vol. 1291 of Lecture Notes in Computer Science. Springer, 1997. Pp. 95–116.

6. Feitelson D. G. Packing Schemes for Gang Scheduling // In Job Scheduling Strategies for Parallel Processing. Springer-Verlag, 1996. Pp. 89–110.

7. Gehring J., Preiss T. Scheduling a Metacomputer with Uncooperative Sub-schedulers // Proceedings of the Job Scheduling Strategies for Parallel Processing. IPPS/SPDP '99/JSSPP '99. London, UK, UK: Springer-Verlag, 1999. Pp. 179–201.

8. Talby D., Feitelson D. G., Raveh A. A Co-Plot analysis of logs and models of parallel workloads // ACM Transactions on Modeling and Computer Simulation (TOMACS). 2007. Vol. 17, no. 3.

9. Krampe A., Lepping J., Sieben W. A hybrid Markov chain model for workload on parallel computers // Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. HPDC '10. New York, NY, USA: ACM, 2010. Pp. 589–596.

10. Dijk N., Smeitink E. A non-exponential queueing system with batch servicing. Researchmemorandum, No. 13. Free University, Amsterdam. 1988.

11. Brill P., 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. Pp. 51–68.

12. Green L. A Queueing System in Which Customers Require a Random Number of Servers // Operations Research. 1980. Vol. 28, no. 6. Pp. 1335–1346.

13. Kim S. M/M/s Queueing System Where Customers Demand Multiple Server Use: Dr. Sci. dissertation / Southern Methodist University. 1979.

14. Green L. Comparing operating characteristics of queues in which customers require a random number of servers // Management Science. 1980. Vol. 27, no. 1. Pp. 65–74.

15. Whitt W. Blocking when service is required from several facilities simultaneously // AT&T Technical Journal. 1985. Vol. 64, no. 8. Pp. 1807–1856.

16. Kaufman J. Blocking in a shared resource environment // IEEE Transactions on Communications. 1981. Vol. 29, no. 10. Pp. 1474–1481.

17. Морозов Е., Румянцев А. Некоторые модели многопроцессорных систем обслуживания с тяжелыми хвостами // Параллельные вычислительные технологии 2011: сборник трудов Международной научной конференции. Челябинск: ЮУрГУ, 2011. С. 555–566.

18. Румянцев А. О стохастическом моделировании вычислительного кластера // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: Тезисы докладов Всероссийской конференции с международным участием (18–22 апреля 2011). Москва: РУДН, 2011. С. 46–47.

19. Морозов Е. В., Румянцев А. С. Вероятностные модели многопроцессорных систем: стационарность и моментные свойства // Информатика и ее применения. 2012. Т. 6. №3. С. 99–106.

20. Alejandro Luciero. Simulation of batch scheduling using real production-ready software tools: URL: www.bsc.es/media/4856.pdf (дата обращения 06.12.2012)

21. Green Grid metrics: describing data center power efficiency // Whitepaper. Green Grid, 2007.

22. Shmueli E., Feitelson D.G. On Simulation and Design of Parallel-Systems Schedulers: Are We Doing the Right Thing? // IEEE Transactions on Parallel and Distributed Systems. 2009. Pp. 983–996.

23. ЦКП КарНЦ РАН "Центр высокопроизводительной обработки данных": URL: http://cluster.krc.karelia.ru (дата обращения 06.12.2012)