

14. Maksimov E. Yu., Yurkov N. K., Yakimov A. N. A finite-element model of the thermal influences on a microstrip antenna. *Measurement Techniques*, 2011, vol. 54, no. 2, May, pp. 207–212.

15. Shishulin D. N., Yurkov N. K., Yakimov A. N. Modeling the Radiation of a Mirror Antenna taking Vibration Deformations into Account. *Measurement Techniques*, 2014, vol. 56, no. 11, February, pp. 1280–1284.

УДК 004.94, 519.872

**ДЕТЕРМИНИРОВАННАЯ ИМИТАЦИОННАЯ МОДЕЛЬ
КЛАСТЕРОВ ГРИД-СИСТЕМЫ ДЛЯ СРАВНЕНИЯ
ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ ЭВРИСТИК
РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ**

Гаевой Сергей Владимирович, аспирант, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, e-mail: gaevserge@mail.ru

Аль-Хадша Фарес Али Хуссейн, аспирант, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, e-mail: alhadsha@mail.ru

Фоменков Сергей Алексеевич, доктор технических наук, профессор, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, e-mail: saf@vstu.ru

Лукьянов Виктор Сергеевич, доктор технических наук, профессор, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28

В работе осуществлено прогнозирование выполнения заданий на вычислительной Грид-системе (Grid-системе), которая состоит из набора кластеров. Для достижения этой цели авторами была разработана детерминированная имитационная модель. В рамках этой модели кластерная система была представлена в виде системы массового обслуживания (СМО). Каждое из них имеет набор следующих параметров: ширина – количество вычислительных машин, необходимое для выполнения работы; длина – время выполнения задания; площадь (сложность) – произведение длины и ширины. Неограниченная по размеру очередь позволяет выполнить все поступающие на вход системы задания.

Такой подход позволяет определить среднюю длину очереди, среднее время ожидания задания и т.д. В настоящее время авторами исследуются методы оптимизации распределения работы между кластерами. Предполагается, что задание выполняется внутри только одного (любого) кластера GRID-системы. Каждый кластер имеет свой собственный входящий поток заданий от пользователей, отделов, организаций и т.п. Межкластерные соединения позволяют кластерам выполнять задания друг друга.

Процедура выбора кластера для исполнения задания требует введения некоторых критериев. В общем случае не существует ответа на вопрос «Какой критерий выбора кластера следует использовать?», нет и аналитического решения. Поэтому целесообразно промоделировать реальные кластеры и испытать различные способы распределения заданий. Из этих способов нужно выбрать оптимальные решения, удовлетворяющие заданным требованиям.

Авторами построена детерминированная имитационная модель для такой вычислительной системы, предложены эвристики для распределения заданий между кластерами Грид-системы и произведено сравнение эффективности этих эвристик для конкретного случая.

Ключевые слова: Грид-системы, брокер заданий, имитационное моделирование, детерминированная имитационная модель, распределение заданий, эвристики распределения заданий, системы массового обслуживания, вычислительный кластер, parallel workloads

DETERMINISTIC SIMULATION MODEL OF CLUSTERS OF A GRID-SYSTEM FOR COMPARISON OF HEURISTICS FOR TASK DISTRIBUTION

Gaevoy Sergey V., post-graduate student, Volgograd State Technical University, 28 Lenin av., Volgograd, 400005, Russian Federation, e-mail: gaevserge@mail.ru

Al-Khadsha Fares A.Kh., post-graduate student, Volgograd State Technical University, 28 Lenin av., Volgograd, 400005, Russian Federation, e-mail: alhadsha@mail.ru

Fomenkov Sergey A., D.Sc. (Engineering), Professor, Volgograd State Technical University, 28 Lenin av., Volgograd, 400005, Russian Federation, e-mail: saf@vstu.ru

Lukyanov Viktor S., D.Sc. (Engineering), Professor, Volgograd State Technical University, 28 Lenin av., Volgograd, 400005, Russian Federation

In paper an attempt to predict job execution on a Grid-system have been made. The Grid-system consists of computing clusters. To achieve this goal we have developed a deterministic simulation model. The cluster system is represented as a queueing system (QS) that executes incoming jobs. Each job has following parameters. The width is an amount of faculty units being required to execute the job. The length is the time the job needs to be executed. Square (difficulty) is a product of the length and the width. Unlimited queue is assumed because of the requirement to execute all jobs.

That approach allows us to determine average queue length, average waiting time of the job etc. Now we investigate a method of job distribution between clusters. It is assumed that a task is executed by a single cluster.

Each cluster of the GRID-system has his own jobs incoming from users, departments, some organizations etc. The connection allows clusters to execute jobs of each other. That policy makes a criteria necessary. And we need to answer the question «What criteria must be used?». There is neither an analytic solution nor a general answer. One has to model the actual clusters and try different job distributions. From those distributions one needs to select solutions that satisfy our requirements.

The authors have created a deterministic simulation model of such calculating system, proposed some heuristics for the job distribution and compared these heuristic for a chosen example.

Keywords: Grid-systems, resource broker, simulation modeling, simulation, deterministic simulation model, job distribution, the heuristics of a job distribution, queueing systems, a computing cluster, parallel workloads

Введение. Грид-системы строятся на базе кластеров и позволяют балансировать нагрузку между ними. Оптимизация работы таких систем может быть обеспечена рациональной балансировкой нагрузки. При этом для формирования оптимальных правил распределения заданий могут быть использованы методы имитационного моделирования.

Вопросы моделирования работы Грид-систем уже поднимались авторами ранее в [2, 3, 5, 6]. При этом был создан программный продукт [8]. В данной статье рассматривается два аспекта указанной проблемы, а именно – процесс выполнения заданий внутри кластеров (узлов Грид-системы) и способы распределения заданий между узлами системы. Целесообразно рассмотреть процесс обслуживания, когда задание может использовать сразу несколько каналов обслуживания (в данном случае, несколько вычислительных машин).

Предлагаемая модель является обобщенной: она может быть использована при передаче не только сообщений в сетях, но и в иных случаях, например, при обработке документации, обслуживании клиентов в организациях, в цехах и т.д.

Целью данной работы было создание программного средства и демонстрация возможностей его использования, а также оценка эффективности найденных в источниках и предложенных авторами эвристик на конкретном примере, поскольку определение универсальной эвристики невозможно. Применение детерминированной дискретно-событийной имитацион-

ной модели позволяет избавиться от необходимости анализа стохастических характеристик логов заданий, так как это в общем случае может создать определенные трудности.

Характеристика материала, использованного для имитационного моделирования. Данные, использованные для моделирования, были взяты нами из [12], где они хранятся в деперсонализованном виде. Это данные реальных вычислительных кластеров, предназначенные для использования на безвозмездной основе. Характеристики данных представлены в [12], поэтому в данной статье нет смысла их повторять. На каждый лог-файл предоставляется большое количество различных характеристик (в основном в графической форме). В общем случае анализ входных данных подобных систем весьма затруднителен – существуют различные подходы, иногда отрицающие друг друга. Коллекция таких подходов может быть найдена по тому же адресу [12]. Входные потоки заданий обычно не являются пуассоновскими (коэффициент вариации времени между приходами заданий больше пяти, а иногда больше десяти). Хотя в некоторых моделях все-таки делается допущение пуассоновских входных потоков (стационарных или динамических). Длина и ширина являются зависимыми величинами, поэтому необходимо установление зависимости между ними.

Моделирование распределения заданий.

А. Постановка задачи.

Грид-система состоит из узлов, именуемых кластерами. Каждый кластер в свою очередь построен из вычислительных машин (каналов обслуживания). С практической точки зрения [4] удобно создавать кластеры так, чтобы вычислительные машины внутри них обладали одинаковыми характеристиками производительности. Рассматривать иные ситуации в данном контексте не имеет смысла.

В настоящее время каждое ядро многоядерного процессора можно рассматривать как вычислительную машину, так как в кластере на разных ядрах одного процессора могут выполняться разные задачи. Измерять производительность машин можно в разных величинах. Воспользуемся приемом из [11] и введем понятие эталонной машины, производительность которой обозначим за условную единицу. Производительности остальных машин будем измерять в этих условных единицах.

В систему для каждого кластера поступают задания, которые надо обработать. Вся обработка ведется в пределах одного кластера [11]. Сложность [4] задания будем измерять временем выполнения на одной эталонной машине. Такой подход оказался на практике лучше, чем предложенный в [2, 3, 5, 6, 8].

Каждое задание может исполняться параллельно на нескольких машинах. Количество вычислительных машин, требуемое заданием, прописано в нем в момент создания. Назовем эту величину шириной задания [7, 12]. Длиной [7, 12] задания назовем время его выполнения на требуемом числе эталонных машин (с единичной производительностью). Площадью [7] задания назовем произведение длины на ширину. Очевидно, что площадь – это приближенно сложность задания.

В данной работе кластер будет представлен в виде обслуживающего блока с некоторым количеством каналов обслуживания и единой не приоритетной, не ограниченной по размеру очередью. Это обеспечит нам обслуживание всех входящих заданий. Задание может быть поставлено (передано) на исполнение только в случае наличия достаточного для его обработки количества свободных машин.

Шириной (площадью, сложностью) очереди назовем сумму ширин (площадей, сложностей) входящих в нее заданий. Длиной очереди назовем число заданий в ней.

В. Метод моделирования.

Для расчета параметров вычислительных систем воспользуемся методом дискретно-событийного имитационного моделирования. В отличие от методов, которыми мы пользо-

вались ранее в [2, 3, 5, 6, 8], построим не стохастическую, а детерминированную модель [9, 10]. Данный вариант рационален, так как мы располагаем реальными входными данными реальных вычислительных кластеров (логами) и собираемся их использовать.

Событиями модели являются следующие.

1. Поступление задания. Времена прихода заданий жестко заданы в виде списка в лог-ге. Там же указаны их длина и ширина. Появившиеся задания передаются кластеру в соответствии с эвристикой. Для всякого пришедшего задания надо в первую очередь проверить, нельзя ли его сразу поставить на исполнение (это может быть возможно даже при наличии очереди, так как задания имеют разную ширину). Если это возможно, то задание ставится на исполнение, в противном случае – помещается в конец очереди.

2. Завершение обслуживания задания. Задание удаляется из системы как обслуженное. После этого производится попытка передачи из очереди на исполнение тех заданий, которые в ней находятся. С этой целью очередь просматривается вся (от начала до конца), так как извлекаемых заданий может быть сразу несколько, поскольку ушедшее задание может освободить несколько каналов обслуживания. Задание, для которого имеется достаточно свободных машин, ставится на исполнение, и просмотр очереди продолжается со следующего за ним задания. Данная политика считает более приоритетной в отношении исполнения задание, пришедшее первым.

Отметим, что этот алгоритм может быть не оптимальным вариантом с позиций полноты загрузки системы, так как используя первое в очереди задание можно занять не все свободные (доступные) каналы. В то же время при этом в очереди останутся ждать задания, которые в одиночку или совместно смогут одновременно занять все доступные каналы обслуживания. Таким образом, простейший алгоритм может не обеспечить максимальную загрузку.

Если разместить все времена завершения в бинарной куче (binary heap), то время поиска следующего события сокращается с $O(n)$ до $O(\log(n))$, где n – число событий, что значительно ускоряет процесс моделирования. На каждом шаге мы выбираем минимум из времени прихода следующего задания и головного элемента кучи, что приводит к соответствующему событию. Моделирование останавливается, когда будет обслужено последнее задание (это время выполнения всего пакета поданных на вход системы заданий).

Измерение показателей данной системы проводится как измерение показателей системы массового обслуживания (СМО) – эти вопросы рассматриваются в [9, 10]. При этом используется подход, аналогичный выводу формулы Литтла [1].

Следующие параметры можно рассчитывать как для всего кластера, так и Грид-системы в целом.

Вспомогательные параметры:

$$Z_e = \sum_{i=1}^N T e_i, Z_q = \sum_{i=1}^N T q_i, Z_s = \sum_{i=1}^N T s_i, \forall i = \overline{1, N} : T s_i = T e_i + T q_i,$$

$$Z_s = Z_e + Z_q, Z_v = \sum_{i=1}^N W_i T e_i, Z_w = \sum_{i=1}^N W_i T q_i.$$

Основные параметры:

$$T_q = \frac{Z_q}{N}, T_e = \frac{Z_e}{N}, T_s = \frac{Z_s}{N}, T'_q = \frac{Z_q}{L} = \frac{T_q}{F_q},$$

$$Q = \frac{Z_q}{T}, E = \frac{Z_e}{T}, S = \frac{Z_s}{T}, W = \frac{Z_w}{T}, F_q = \frac{L}{N}, F_T = \frac{T_q}{T_s}.$$

Параметры, применимые только к кластерам:

$$V = \frac{Z_v}{T}, U = \frac{V}{M}.$$

Здесь введены обозначения: Te_i – время выполнения i -го задания; Tq_i – время ожидания в очереди i -го задания; Ts_i – время пребывания в системе i -го задания; W_i – ширина i -го задания; N – число заданий, поступивших на исполнение; M – число каналов обслуживания (вычислительных машин) кластера; L – число заданий, побывавших в очереди; T – время моделирования; Z_e, Z_q, Z_s, Z_v, Z_w – площади под гистограммами (интегралы функций по интервалу моделирования) числа обслуживаемых заявок, длины очереди, числа заданий в системе, числа занятых каналов, ширины очереди, соответственно; T_q – среднее время ожидания в очереди; T_e – среднее время выполнения; T_s – среднее время пребывания в системе; T_q' – среднее время ожидания среди побывавших в очереди; Q – средняя длина очереди; E – среднее число исполняемых заданий; S – среднее число заданий в системе; V – среднее число занятых каналов; W – средняя ширина очереди; U – использование системы; F_q – доля заданий, побывавших в очереди; F_T – доля времени в системе, истраченного на ожидание.

С. Стратегии распределения заданий между кластерами.

Такие стратегии являются эвристиками, то есть позволяют определить кластер путем эмпирических оценок [7, 11]. Все стратегии, кроме первой, работают так. Каждому узлу присваивается стоимость исполнения задания. Узлы, которые не имеют достаточного числа машин для исполнения задания, получают оценку, равную бесконечности. Выбирается узел с наименьшей оценкой (стоимостью исполнения). При совпадении оценок двух или более кластеров будем выбирать кластер с меньшим номером. Предполагается, что для всякого задания существует хотя бы один кластер, способный его выполнить. Задания берутся из логов работы реальных кластеров [12], поэтому достаточно широким является как минимум тот кластер, в логе которого оно было указано.

Охарактеризуем рассматриваемые далее эвристики.

1. Rotate. Список кластеров просматривается циклически, начиная со следующего за тем, который получил последнее задание. Если кластер не обладает достаточным числом машин для исполнения задания, то он пропускается.

2. FreeExec. Скорректируем оценку из [7] – $W/(N+W)$, где W – количество машин кластера; N – количество свободных машин кластера. Используется при быстром выполнении заданий в отсутствие очереди.

3. QueueWidth. Оценка из [7] $\left(w_{зад} + \sum_j w_{задj} \right) / W$, где $w_{зад}$ – ширина задания, для которого определяется кластер; $w_{задj}$ – ширина j -го задания в очереди кластера; W – количество машин кластера.

4. QueueLen. Модификация оценки QueueWidth $(1+Q)/W$, где Q – длина очереди кластера; W – количество машин кластера.

5. QueueDif. Оценка из [7] $\left(c_{зад} + \sum_j c_{задj} \right) / W$, где $c_{зад}$ – сложность задания, для которого выбирается исполнитель; $c_{задj}$ – сложность j -го задания в очереди кластера; W – количество узлов кластера.

6. QueueProd. Наша модификация оценки № 5 – $\left(c_{зад} + \sum_j c_{задj} \right) / (W \cdot P)$, где $c_{зад}$ – сложность задания, для которого выбирается исполнитель; $c_{задj}$ – сложность j -го задания в очереди кластера; W – количество узлов кластера; P – производительность узла.

7. MaxProd – $(W \cdot P)^{-1}$, где P – производительность узла; W – количество узлов кластера.

8. EqualLoading – $\sum_j c_{задj}$, где $c_{задj}$ – сложность j -го задания в очереди кластера.

9. EqualQueueLen – Q , где Q – длина очереди кластера.

10. EqualQueueWidth – $\sum_j w_{задj}$, где $w_{задj}$ – ширина j -го задания в очереди кластера.

D. Обработка логов.

В [12] представлены реальные логи работы вычислительных систем (кластеров). Использование логов «как есть» нерационально по двум причинам: во-первых, приток заданий обладает определенной циклической закономерностью в течение рабочей недели ($w = 7 \text{ дн.}$), а, во-вторых, логи имеют разную длину. Из-за такой цикличности (рис. 1) кластеры будут испытывать пики нагрузки синхронно – этот фактор надо учесть при моделировании.

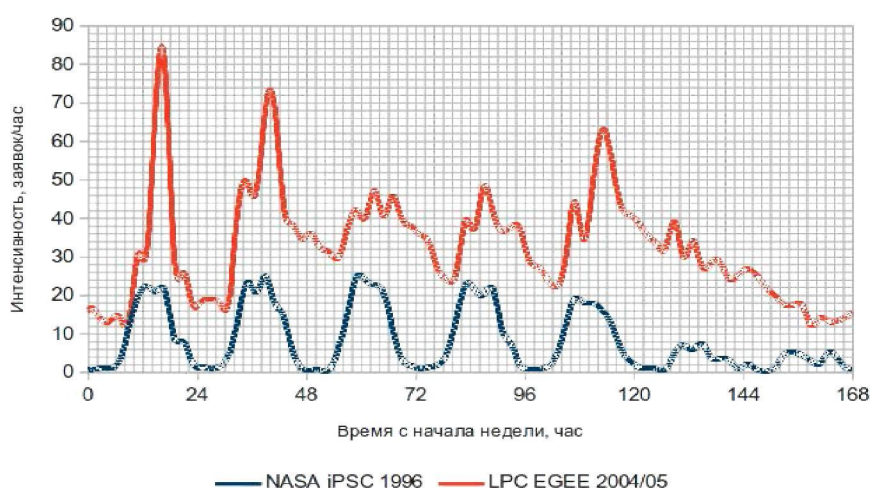


Рис. 1. Пример изменения интенсивности потока заданий в течение недели для кластеров NASA iPSC и LPC EGEE

Пусть неотрицательные числа $\theta_i^{(j)}, i = \overline{1, n_j}, j = \overline{1, m}$ – это времена прихода i -го задания по логам j -ых кластеров. Введем параметр δ_j – время, прошедшее с последней полуночи с воскресенья на понедельник до момента времени, принятого в логе j -го кластера в качестве нулевого ($0 \leq \delta_j < w$). Как правило, нулевым моментом является приход первого задания. Исправленными временами прихода заданий будут $\tau_i^{(j)} = \theta_i^{(j)} + \delta_j, i = \overline{1, n_j}, j = \overline{1, m}$. Теперь необходимо продолжить логи кластеров на неограниченное время, сохранив цикличность. Примем $\tau_{i+n_j}^{(j)} = \tau_i^{(j)} + \zeta_j$, где $\zeta_j = w \cdot \text{ceil}(\theta_{n_j}^{(j)} / w)$. Схема преобразования логов дана на рис. 2. Выберем для моделирования все такие задания, для которых $\tau_i^{(j)} \leq \max_{j=1}^m \tau_{n_j}^{(j)}$, то есть возьмем за время моделирования время прихода последнего неповторяющегося задания. Это обеспечит нам присутствие каждого задания в системе хотя бы один раз.

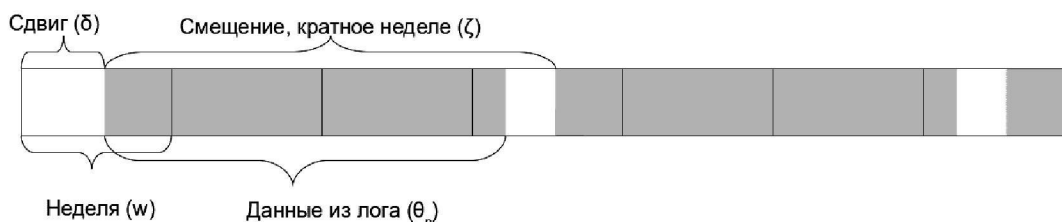


Рис. 2. Схема преобразования логов

Е. Пример использования стратегий.

Возьмем длины и ширины заданий из логов [12], а производительности машин из таблицы. Исследования проводятся в двух вариантах. Вариант, заявленный выше (когда длина и ширина берутся из логов), назовем Parallel. Вариант Sequential – исполнение с выделением для каждого задания только одной машины при сохранении площади. Введем еще одну стратегию Self, когда каждый кластер исполняет свои задания без обмена заданиями с другими кластерами.

Таблица

Принимаемые производительность машин

Кластер	Количество вычислительных машин	Производительность одной машины
NASA-iPSC-1993-3.1-cln.swf	128	0,7
LANL-CM5-1994-4.1-cln.swf	1024	0,8
SDSC-Par-1995-3.1-cln.swf	400	0,9
SDSC-Par-1996-3.1-cln.swf	400	1,0
CTC-SP2-1996-3.swf	338	1,1
HPC2N-2002-2.1-cln.swf	240	1,2
LPC-EGEE-2004-1.2-cln.swf	140	1,3

Нас интересуют: время выполнения пакета всех заданий (рис. 3); среднее время ожидания результата по одному заданию (рис. 4); средняя длина очереди (рис. 5).

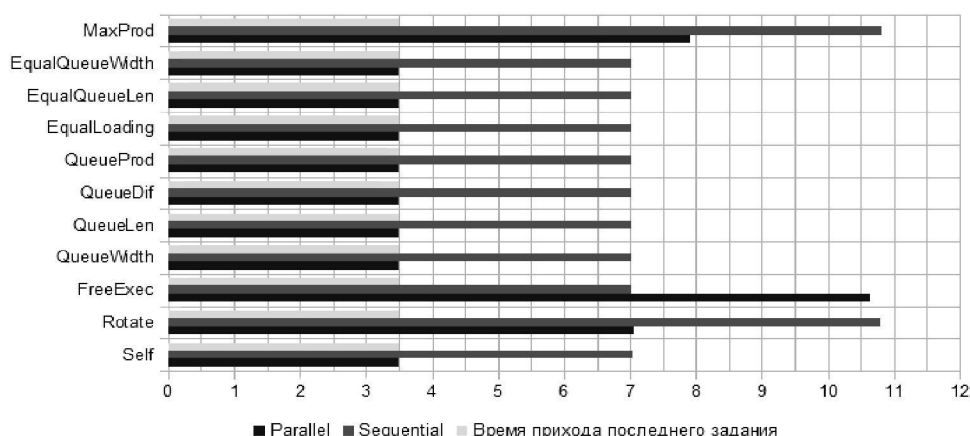


Рис. 3. Время выполнение всего пакета заданий для различных стратегий (лет)

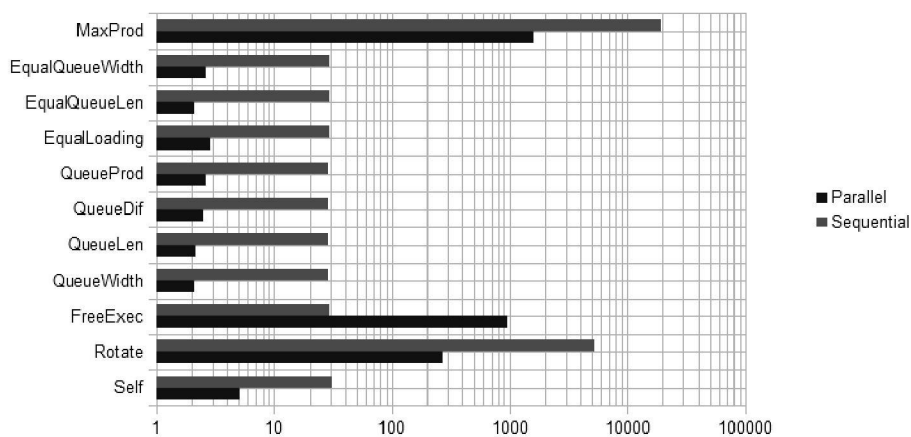


Рис. 4. Среднее время ожидания результата для различных стратегий, час

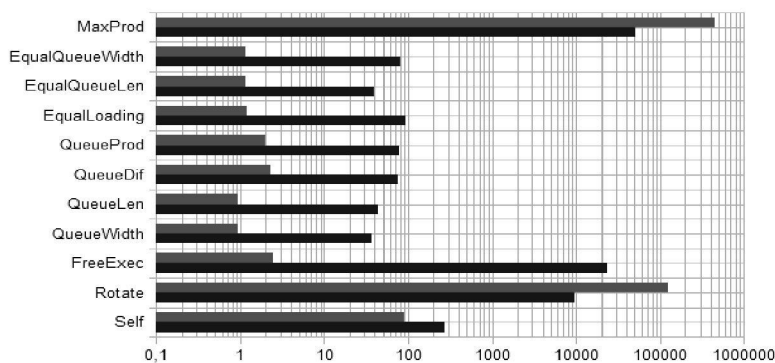


Рис. 5. Средняя длина очереди для различных стратегий

На основе представленных выше результатов (показателей) можно сделать следующие выводы.

1. В большинстве случаев распараллеливание выполнения заданий оправдывает себя. Sequential сокращает длину очереди лишь за счет того, что задания занимают машины дольше.

2. Стратегии Rotate и MaxProd не оправдывают себя. В случае MaxProd это значит, что систему нельзя заменить лишь ее самым производительным звеном, что сильно упростило бы задачу.

3. Стратегия FreeExec предназначена для использования при отсутствии очереди. Поэтому при Sequential результат хороший, а при Parallel – очень плохой.

4. При Sequential ширина и длина очереди совпадают, поэтому $QueueWidth = QueueLen$ и $EqualQueueWidth = EqualQueueLen$, а времена ожидания результата у большинства стратегий практически совпадают. С точки зрения средней длины очереди $QueueWidth/Len$ немного выигрывают у $EqualQueueWidth/Len$. На третьем месте – EqualLoading.

5. Самые хорошие результаты при Parallel дают (в порядке незначительного ухудшения результатов) $QueueWidth$, $EqualQueueLen$ и $QueueLen$.

6. Наилучшие результаты дает наиболее легко вычисляемый критерий (показатель). Это делает ненужным предсказание сложности заданий, поскольку учет дополнительных показателей (например, $QueueDif$, $QueueProd$) лишь ухудшает результат.

Необходимо отметить, что сделанные здесь выводы не являются универсальными. Результаты могут сильно измениться в зависимости от потока заданий и структуры Grid-системы. Поэтому в каждом конкретном случае надо моделировать конкретную Грид-систему и анализировать соответствующие ей полученные данные.

Заключение. Таким образом, были предложены эвристики распределения заданий между несколькими устройствами обслуживания (кластерами). Для оценки качества эвристик была построена детерминированная имитационная модель и получены показатели их производительности для конкретного примера. Это позволило продемонстрировать работоспособность созданной нами системы имитационного моделирования, которая может быть применена для оценки эвристик в каждом конкретном случае с использованием реального потока заданий кластеров системы. Иными словами, перед моделированием необходимо располагать логами входных заданий исследуемых кластеров.

Список литературы

1. Вывод формулы Литтла // Проект «Моделирование систем». – Режим доступа: <http://sardismusic.com/t2r11part1.html> (дата обращения 23.10.2013), свободный. – Заглавие с экрана. – Яз. рус.
2. Лукьянов В. С. Имитационная модель гетерогенной вычислительной системы / В. С. Лукьянов, Д. Н. Жариков, С. В. Гаевой, Д. С. Попов // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах» : межвуз. сб. науч. ст. – Волгоград, 2011. – Вып. 11, № 9. – С. 85–88.
3. Имитационное моделирование грид-систем : моногр. / В. С. Лукьянов, А. Е. Андреев, Д. Н. Жариков, А. А. Островский, С. В. Гаевой. – Волгоград, 2012. – 215 с.
4. Интернет-портал по грид-технологиям :: GRIDCLUB.RU. – Режим доступа: <http://gridclub.ru/> (дата обращения 12.11.2011), свободный. – Заглавие с экрана. – Яз. рус.
5. Лукьянов В. С. Моделирование отказоустойчивых GRID-систем / В. С. Лукьянов, Д. Н. Жариков, С. В. Гаевой, О. В. Шаповалов // Инновации на основе информационных и коммуникационных технологий : мат-лы Междунар. науч.-практ. конф. (Россия, г. Сочи, 1–10 октября 2010 г.) / Московский гос. ин-т электроники и математики МИЭМ (ТУ) [и др.]. – Москва, 2010. – С. 253–254.
6. Моделирование GRID-систем / В. С. Лукьянов, Д. Н. Жариков, С. В. Гаевой, Ю. В. Шафран // Информационные технологии моделирования и управления. – 2009. – № 5. – С. 669–677.

7. Проблемы моделирования GRID-систем и их реализация / О. И. Самоваров [и др.] // Портал «Информационно-коммуникационные технологии в образовании». – Режим доступа: <http://www.ict.edu.ru/vconf/files/9451.pdf> (дата обращения 10.05.2010), свободный. – Заглавие с экрана. – Яз. рус.
8. Свид. о гос. регистрации программы для ЭВМ № 2010610693 от 20 янв. 2010 г. РФ, МПК (нет). Имитационная модель грид-системы (GridModel) / В. С. Лукьянов, Д. Н. Жариков, С. В. Гаевой, Ю. В. Шафран ; ВолгГТУ. – 2010.
9. Фоменков С. А. Математическое моделирование системных объектов : учеб. пос. / С. А. Фоменков, Д. А. Давыдов, В. А. Камаев. – Волгоград : РПК «Политехник», 2006. – 180 с.
10. Шеннон Р. Имитационное моделирование систем – искусство и наука / Р. Шеннон ; под ред. Е. К. Масловского. – Москва : Мир, 1978. – 418 с.
11. Эвристики распределения задач для брокера ресурсов Grid / А. И. Аветисян [и др.]. – Режим доступа: <http://www.citforum.ru/nets/digest/grid/index.shtml> (дата обращения 02.10.2009), свободный. – Заглавие с экрана. – Яз. рус.
12. Parallel Workloads Archive // The Rachel and Selim Benin School of Computer Science and Engineering. – Available at: <http://www.cs.huji.ac.il/labs/parallel/workload/> (accessed 03.05.2014).

References

1. Little's formula proof. Project "System Simulation". Available at: <http://sardismusic.com/t2r11part1.html> (accessed 23 October 2010). (In Russ.).
2. Lukyanov V. S., Zharikov D. N., Gaevoy S. V., Popov D. S. Imitatsionnaya model geterogennoy vychislitelnoy sistemy [Simulation model of heterogeneous computing system]. *Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya «Aktualnye problemy upravleniya, vychislitelnoy tekhniki i informatiki v tekhnicheskikh sistemakh»: mezhvuzovskiy sbornik nauchnykh statey* [News of Volgograd State Technical University. Series "Actual problems of management, computer science and informatics in technical systems": Interuniversity Collection of Scientific Articles], 2011, vol. 11, no. 9, pp. 85–88.
3. Lukyanov V. S., Andreev A. Ye., Zharikov D. N., Ostrovskiy A. A., Gaevoy S. V. *Imitatsionnoe modelirovanie grid-sistem* [Simulation modeling of Grid-systems]. Volgograd, 2012. 215 p.
4. Internet portal on Grid-technologies :: GRIDCLUB.RU. Available at: <http://gridclub.ru/> (accessed 12 November 2010). (In Russ.).
5. Lukyanov V. S., Zharikov D. N., Gaevoy S. V., Shapovalov O. V. Modelirovanie otkazoustoychivyykh GRID-sistem [Modeling of fault-tolerant Grid-systems]. *Innovatsii na osnove informatsionnykh i kommunikatsionnykh tekhnologiy: materialy Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Innovations based on information and communication technologies: Proceedings of the International Scientific and Practical Conference] (Russia, Sochi, 1–10 October, 2010). Moscow, 2010, pp. 253–254.
6. Lukyanov V. S., Zharikov D. N., Gaevoy S. V., Shafran Yu. V. Modelirovanie GRID-sistem [Grid-system modeling]. *Informatsionnye tekhnologii modelirovaniya i upravleniya* [Information technologies of modeling and management], 2009, no. 5, pp. 669–677.
7. Samovarov O. I. et al. Problemy modelirovaniya GRID-sistem i ikh realizatsiya [Problems of simulation of Grid-systems and their implementation]. Portal «Informatsionno-kommunikatsionnye tekhnologii v obrazovanii» [Portal "Information and communication technologies in education"]. Available at: <http://www.ict.edu.ru/vconf/files/9451.pdf> (accessed 10 May 2010).
8. Lukyanov V. S., Zharikov D. N., Gaevoy S. V., Shafran Yu. V. Certificate of State Registration of the computer program no. 2010610693, January 20th, 2010. Russia, IPC (no). Simulation model of Grid-system (GridModel)], 2010. (In Russ.).
9. Fomenkov S. A., Davydov D. A., Kamaev V. A. *Matematicheskoe modelirovanie sistemnykh obektov* [Mathematical modeling of system objects]. Volgograd, RPK «Politehnik», 2006. 180 p.
10. Shannon, R., Maslovskiy Ye., K. (ed.) *Imitatsionnoe modelirovanie sistem – iskusstvo i nauka* [Imitation system modeling – art and science]. Moscow, Mir, 1978. 418 p.
11. Avetisyan A. I. et al. Heuristics of problem distribution for Grid resource broker. Available at: <http://www.citforum.ru/nets/digest/grid/index.shtml> (accessed 2 October 2009). (In Russ.).
12. Parallel Workloads Archive. The Rachel and Selim Benin School of Computer Science and Engineering. Available at: <http://www.cs.huji.ac.il/labs/parallel/workload/> (accessed 3 May 2013).