

Судя по рис. 6, увеличение активного сопротивления проводов ЛЭП особенно опасно для протяженных линий электропередачи. А малая собственная индуктивность этих проводов для протяженных ЛЭП, как следует из рис. 7, крайне нежелательна и способствует существенному увеличению затухания напряжения.

Аналогичные зависимости можно построить и для тока. Следует помнить, что здесь идет речь о погонных параметрах линии.

Предлагаемые зависимости напряжения и тока вдоль однопроводной линии от ее погонных параметров построены на основании уравнений (2). Этот факт подтверждает универсальность этих уравнений, которые могут быть использованы не только для прогнозирования, спектрального состава напряжения и тока в любом месте ЛЭП, но и для определения оптимальных значений первичных параметров линии. Но только однопроводной линии электропередачи. Для многопроводных линий и, тем более, при передаче электроэнергии пониженного качества эти уравнения не справедливы.

Таким образом, качественные и количественные характеристики распределения электрической энергии зависят от величин активных сопротивлений проводов, собственных индуктивностей, емкостных и активных проводимостей между проводом и заземленными конструкциями ЛЭП. Проследив эти зависимости, можно сделать заключение об оптимальных величинах параметров исследуемой линии.

Выводы. Количественные и качественные оценки распространения электрической энергии вдоль однопроводной ЛЭП зависят от ее вторичных и, в конечном счете, от ее первичных параметров: увеличение емкости между проводом и заземленными конструктивными элементами приводит к возрастанию напряжения и тока на протяженных ЛЭП, а изменение активной проводимости между проводом и заземленными конструктивными элементами линии электропередачи в диапазоне от 0 до 1 См практически не оказывает влияния на оценку распространения электрической энергии вдоль однопроводной ЛЭП; увеличение активного сопротивления провода ЛЭП особенно опасно для протяженных линий электропередачи, а малая собственная индуктивность этих проводов для протяженных ЛЭП крайне нежелательна и способствует существенному увеличению затухания напряжения и тока.

Литература

1. Большанин, Г.А. Распределение электрической энергии пониженного качества по участкам электроэнергетических систем. В 2 кн. / Г.А. Большанин. – Братск: БрГУ, 2006. 807 с.

References

1. Bol'shanin G.A. Down-graded electric energy distribution along the power systems sections. V 2 kn. / G.A. Bol'shanin. Bratsk, BrGU, 2006. 807 p.

УДК 681.3,681.5

Распределение параллельных программ между ролями-сотрудниками в облачном кластере

Ю.А. Шичкина¹

¹Братский государственный университет, Макаренко 40, Братск, Россия. E-mail: strange.y@mail.ru
Статья поступила 20.12.2011, принята 19.05.2012

В последнее время набирают популярность услуги по предоставлению вычислительных мощностей в «облаках». Исследования показывают, что методы оптимизации параллельных алгоритмов могут быть успешно применены для автоматизации процесса выделения вычислительных ресурсов в «облаке» по запросу пользователя. В данной статье рассматривается архитектура приложения в облаке, включающая: веб-роли, предназначенные для получения и обработки входящих запросов по протоколу HTTP через службы веб-сервера; роли-сотрудники, получающие данные из хранилища данных и обменивающиеся информацией с веб-ролями посредством запросов, размещаемых в очередях; диспетчер очереди, осуществляющий распределение задач из очереди по ролям-сотрудникам; диспетчер программ, необходимый для выбора программы из базы данных и ее параметров и передачи в очередь; а также очередь с информацией о задачах, поступивших на выполнение, и базу данных программ. Показано, что при использовании схем с применением диспетчеров очереди и программ вероятность превышения допустимого времени решения задачи становится намного меньше в сравнении с другими способами. Для получения оптимального распределения нагрузки среди ролей-сотрудников применялся аппарат матричной алгебры, теории множеств и теории графов. В дальнейшем взаимосвязь задач и процессоров можно представить с помощью взвешенного ориентированного графа, в котором весами ребер будут количества процессоров, необходимых для решения задачи, а именами узлов – номера решаемых задач. На основе взвешенного графа построен алгоритм распределения задач по ролям-сотрудникам в диспетчере очереди, основанный на теории множеств.

Ключевые слова: параллельный алгоритм, облачные вычисления, диспетчер очереди, диспетчер задач, роли-сотрудники.

Concurrent programs distribution among collaborators roles in a cloud hosting

Yu. A. Shichkina¹

¹Bratsk State University, 40 Makarenko str., Bratsk, Russia. E-mail: strange.y@mail.ru

The article received 20.12.2011, accepted 19.05.2012

The services for providing computing systems in “clouds” are currently gaining popularity. The studies demonstrate that the optimization methods for parallel algorithms can be successfully applied to automate the process of allocating computational resources in the “cloud” on user’s demand. The article considers the application architecture in the cloud including: web roles aimed at receiving and processing the HTTP protocol requests-in through the web server services; collaborators roles receiving data from the data warehouse and sharing information with web roles by means of demands placed in queues; the Queue Execution Manager implementing distribution of tasks out of the queue according to the collaborators roles; the Program Manager necessary to select a program and its parameters from the data base and pass to the queue; the queue containing information about the tasks arrived to be performed as well as the programs data base. It has been demonstrated that when employing the structures applying the Queue Execution Manager and the Program Manager, the possibility of the problem time overshoot becomes much less in comparison with other methods. To get optimum load distribution among collaborators roles, the apparatus of matrix algebra, set theory and graph theory was applied. Subsequently, the interrelation between tasks and processors can be presented with the help of a weighted directed graph where the weights of the edges will be the number of processors necessary to solve the problem, and the graphs junction names – the numbers of the problems being solved. On the basis of a weighted graph, the algorithm for tasks distribution among collaborators roles in the Queue Execution Manager based on the set theory has been developed.

Key words: parallel algorithm, cloud calculations, Queue Execution Manager, Task Manager, collaborators roles.

Введение. Кластерная архитектура занимает лидирующие позиции среди суперкомпьютеров, в основном благодаря низкой по сравнению с традиционными параллельными архитектурами стоимости единицы производительности. Тем не менее, далеко не каждая организация, нуждающаяся в высокопроизводительных вычислениях, может позволить себе приобрести кластер, способный обеспечить необходимый для нее уровень производительности.

Еще одним фактором, влияющим на выбор высокопроизводительного компьютерного комплекса, является то, что современные кластеры, приобретаемые с запасом масштабируемости для решения самых больших задач организации, обычно большую часть времени простаивают, либо загружены незначительно.

Альтернативным путем получения вычислительных мощностей является использование технологий обработки данных, подразумевающих предоставление вычислительных ресурсов пользователю, как интернет-сервиса, т. е. ресурсов в «облаках».

Важными требованиями для построения «облака» являются высокая степень автоматизации процесса выделения ресурсов по запросу пользователя, обеспечение постоянного запаса вычислительных мощностей для удовлетворения возрастающих нужд заказчиков и высокого уровня доступности используемых систем.

На базе облачного решения можно создать «облачный кластер» для выполнения параллельных программ [1]. В целом, на практике границы между грид- и облачными вычислениями (grid и cloud) достаточно размыты. Сегодня с успехом можно встретить «облачные» системы на базе модели распределенных вычислений, и наоборот.

Распределения нагрузки среди ролей-сотрудников в «облачном кластере». Для упрощения процесса

разработки параллельных программ необходимы высокоуровневые средства, позволяющие просто проектировать, реализовывать и выполнять запуски сложных программ на платформе систем «облачных вычислений». При разработке таких высокоуровневых средств можно применять аппарат матричной алгебры, теории множеств и теории графов, изложенные выше, при решении задач:

- распараллеливания отдельных программ: определения числа ролей-сотрудников и времени, которое они будут заняты;

- оптимального распределения нагрузки среди ролей-сотрудников: нагрузка на роли-сотрудники в данном случае, при неограниченном времени выполнения задачи, может быть неравномерной, но без «пузырей». Роли-сотрудники, освободившиеся раньше завершения решения задачи, могут быть использованы для решения других задач, стоящих в очереди;

- оптимального проектирования структуры хранилища данных и программ, очереди запросов.

Пример. В нижеприведенной таблице представлены сведения о задачах, поступивших за время $T=4$ в очередь, и параметры задач, такие, как: i – порядковый номер задачи; n – число процессоров, необходимое для решения задачи; t_{i1} – время решения задачи при работе n процессоров; t_{i2} – минимальное время, за которое может быть решена задача; t_{i3} – время, за которое должна быть решена задача; t_{i4} – время поступления задачи в очередь.

Необходимо определить задачу роли-сотруднику с M процессорами так, чтобы весь путь от запроса пользователя до возврата ему результатов занял время $t \leq t_{i3}$. Всего процессоров в облачном кластере N . Решить данную задачу можно по-разному.

Схема решения задачи «естественным путем», по мере поступления задач в очередь и наличия необходимых вычислительных ресурсов облака, приведена на рис. 1.

Если же в схему облачных вычислений добавить еще один компонент (рис. 2) – диспетчер очереди, в

задачу которого входит анализ вычислительных ресурсов и условий задачи и выбор задания для решения не в порядке поступления, а с учетом эффективности работы облака, то временная диаграмма решения задач в облаке может быть более сжатой.

Таблица 1

Задачи, поступившие в очередь запросов

№ задачи	n	t_{i1}	t_{i2}	t_{i3}	t_{i4}
1	12	4	3	8	0
2	15	6	5	–	1
3	13	3	3	–	2
4	9	5	4	7	3

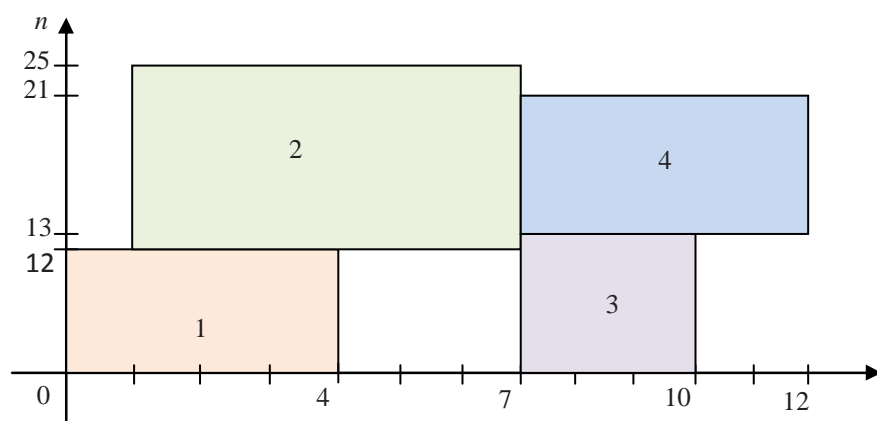


Рис. 1. Схема решения задач по мере их поступления в очередь.

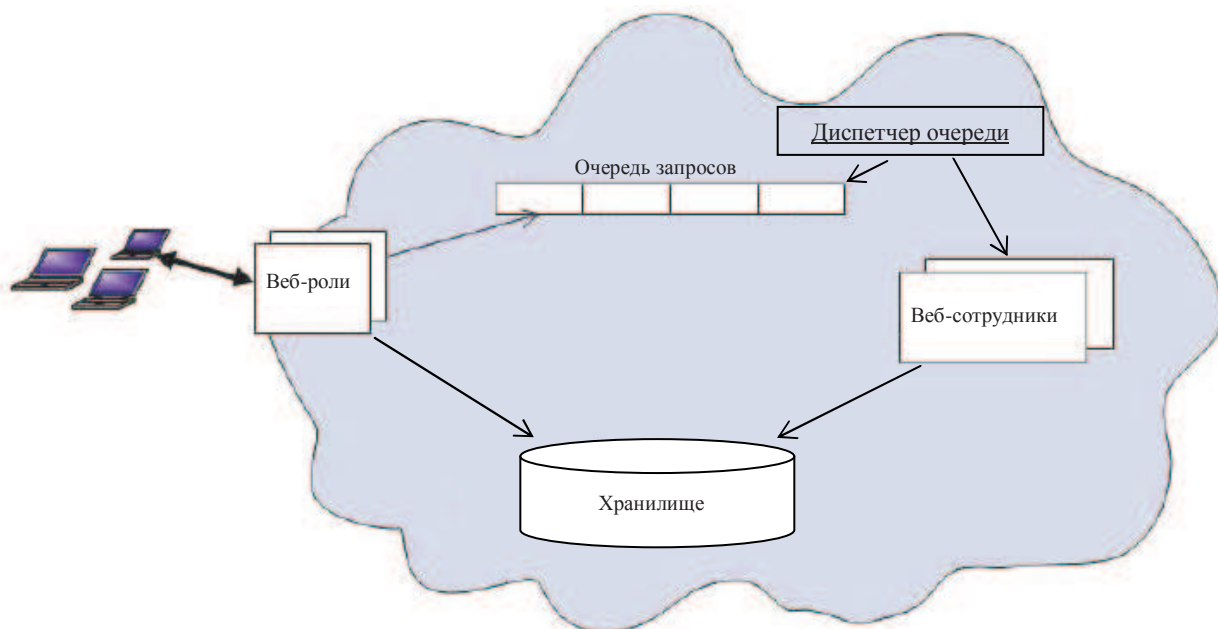


Рис. 2. Пример архитектуры приложения в облаке.

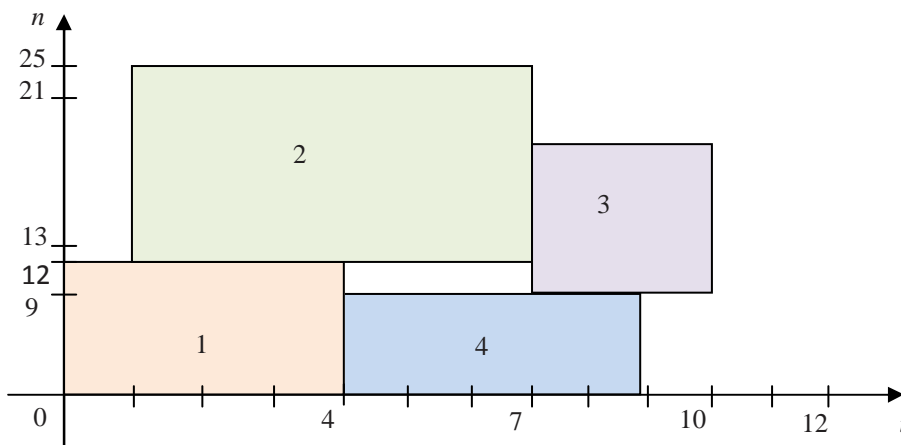


Рис. 3. Схема решения задач с учетом их условий.

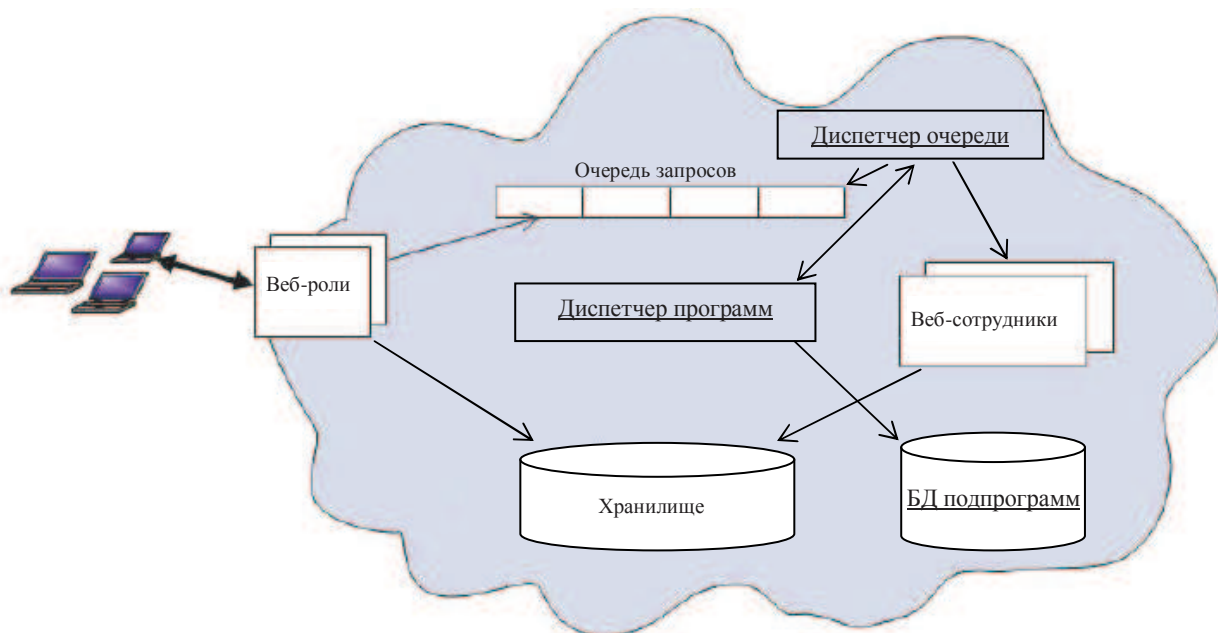


Рис. 4. Пример архитектуры приложения в облаке.

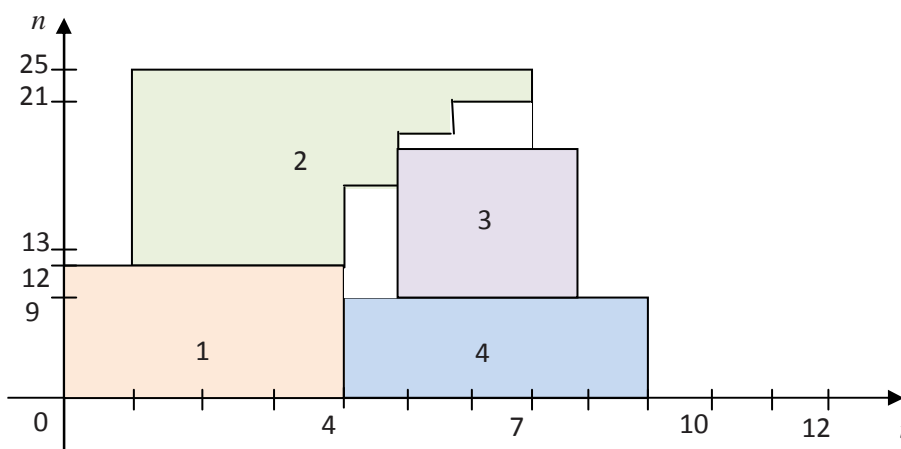


Рис. 5. Схема решения задач при наличии диспетчера программ.

Процесс решения задач с применением диспетчера очереди заметно сокращается и позволяет избегать ситуаций, когда время решения задачи ограничено, а вы-

числительных ресурсов в момент поступления задачи недостаточно.

Для дальнейшей оптимизации работы облака в него можно добавить еще несколько компонентов – базу данных подпрограмм и их спецификаций, а для работы с этой базой данных – диспетчер программ.

В функции диспетчера программ входит, как минимум, анализ тех сведений о программе, которые хранятся в базе данных подпрограмм, а возможно (или при наличии определенных требований и достаточного времени для решения задачи) – поиск оптимального параллельного алгоритма решения задачи с применением графовых алгоритмов [2]. Так, например, если известны временные диаграммы или расписание задачи 2 (таблица 1), то общая схема решения задач в облаке с применением диспетчера программ может быть следующей (рис. 5).

У отдельных задач время их решения может быть регламентировано, например, у задач 1 и 4 (таблица 1). Для решения задачи 4 схема поочередного последовательного решения (рис. 3) не подходит, т. к. время ожидания вычислительных ресурсов в очереди больше допустимого по условию времени: $t_i - t_{i4} + t_{i1} > t_{i3}$, где t_i – время начала решения задачи.

При использовании схем с применением диспетчеров очереди и программ вероятность превышения допустимого условием времени решения задачи становится намного меньше и, как следствие, эффективность облачного кластера выше.

Результаты. Процедура распределения задач по процессорам в целом аналогична (таблица 2) распределению операций параллельного алгоритма на распределенных вычислительных системах.

Таблица 2

Сравнение распределенных и облачных вычислений

Признак	Вычисления	
	Распределенные	Облачные
Взаимозависимость объектов вычислений	Операции параллельного алгоритма часто зависят друг от друга, вследствие чего необходимо строгое соблюдение порядка следования операций по времени на разных процессорах	Решаемые задачи часто не зависят друг от друга, но зависят от времени поступления и допустимого времени решения. Вследствие этого необходимо также строгое соблюдение порядка распределения задач из очереди по ролям-сотрудникам
Зависимость от времени вычислений	Каждая операция выполняется определенное время t	Каждая задача выполняется определенное время в целом t , но вычислительные ресурсы заняты по-разному $\{t_1, t_2, \dots, t_n\}$
Зависимость от вычислительных ресурсов	Одним из параметров оптимальности параллельного алгоритма является минимальное количество задействованных вычислительных единиц при заданном фиксированном времени выполнения алгоритма. Решаемая задача $F(t, n) \rightarrow \min$.	Для решения задачи при фиксированном времени выполнения требуется определенное количество вычислительных единиц, которое не всегда может быть в наличии. Решаемая задача $F(t, n, tp) \rightarrow \min$, где tp – время в очереди в ожидании n вычислительных единиц.

В связи с этим методы, разработанные в распределенных вычислениях, можно адаптировать для организации вычислений в облачном кластере.

Для наглядного представления взаимосвязей процессоров и задач и поиска формализованного метода управления процессом распределения задач из очереди между ролями-сотрудниками можно, как и в распределенных вычислениях, применить элементы теории графов.

Литература

1. Авдиенков О.А., Маланин В.Н. Архитектурные аспекты использования облачной платформы Microsoft Windows Azure Platform для параллельного программирования // Высокопроизводительные

параллельные вычисления на кластерных системах: материалы VI Междунар. науч.-практ. семинара, С.-Петербург, 12-17 дек. 2006 г. СПб., 2007. Т.1. С.253-258.

2. Шичкина Ю.А. Комбинированный метод многопараметрической оптимизации взвешенного информационного графа // Системы. Методы. Технологии. 2010. №3(7). С. 76-82.

References

1. Avdienkov O.A., Malanin V.N. Architectural aspects of a cloud Microsoft Windows Azure Platform application for parallel programming. // Vysokoproizvoditel'nye parallel'nye vychisleniya na klasternykh sistemakh: materialy VI mezhdunar. nauch.-prakt. seminar, S.-Peterburg, 12-17 dek. 2006. SPb: S.-Peterb. gos. un-t, 2007. T. 1. P. 253-258.

2. Shichkina Yu.A. Combined method of multiparametric optimization for a weighted information graph. Sistemy. Metody. Tekhnologii. 2010, №3 (7). P. 76-82.