

БРОКЕР РЕСУРСОВ ДЛЯ ПОДДЕРЖКИ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ГРИД-СРЕД

А.В. Шамакина

Статья посвящена созданию методов и алгоритмов планирования ресурсов, а также разработке на их основе брокера ресурсов для поиска оптимальных ресурсов в проблемно-ориентированных грид-средах. Разработанный алгоритм планирования ресурсов учитывает дополнительные знания о специфике предметной области задания и представление о потоке задач. Приведенный алгоритм основан на алгоритме кластеризации доминирующей последовательности DSC. В отличие от оригинального алгоритма для отображения задач на вычислительные ресурсы используется раскраска графа задач, а объединение задач в один кластер производится с учетом наличия свободных слотов на вычислительных ресурсах. Предложены метод двухфазного резервирования ресурсов и учет проблемных параметров задачи для оценки времени ее выполнения. Приведены варианты использования брокера ресурсов, описаны процесс выделения ресурсов и архитектура брокера ресурсов CAEBeans Broker.

Ключевые слова: брокер ресурсов, алгоритмы планирования ресурсов, грид, резервирование, UNICORE.

Введение

В настоящее время перспективным является направление, связанное с применением грид-технологий [1] для решения ресурсоемких научных задач в разных областях: медицине, инженерном проектировании, нанотехнологиях, прогнозировании климата и т.д. Примером системы, предоставляющей доступ к программным системам класса CAE (Computer Aided Engineering [2]) является система CAEBeans [3], которая позволяет выполнять декомпозицию задач на типовые подзадачи; поиск вычислительных ресурсов, согласно требованиям; постановку задач соответствующим базовым компонентам CAE-систем; мониторинг хода решения задач; предоставление результатов решения задания пользователю.

Технологический цикл решения CAE-задания в общем случае предполагает формирование геометрии задачи, генерацию вычислительной сетки, определение граничных условий, проведение компьютерного моделирования, визуализацию и анализ результатов решения. CAE-задания имеют специфику, в рамках которой необходимо учитывать не только характеристики ресурсов, но и наличие установленных инженерных пакетов, количество доступных лицензий на них и др.

Многие грид-среды осуществляют поддержку сложных приложений с потоком задач (workflow [4]), которые обычно моделируют DAG. Можно перечислить такие инструменты, как Condor DAGMan [5], CoG [6], Pegasus [7], GridFlow [8] и ASKALON [9]. Использование дополнительных знаний о специфике области задачи и представление о потоке задач может существенно улучшить эффективность методов планирования ресурсов. Однако, ни один из существующих на сегодняшний день инструментов не учитывает эту специфику.

Цель данной работы состоит в создании методов и алгоритмов планирования ресурсов, а также в разработке на их основе брокера ресурсов для поиска оптимальных ресурсов в проблемно-ориентированных грид-средах.

В настоящей статье рассмотрены алгоритм планирования ресурсов и реализация на его основе брокера ресурсов CAEBeans Broker. Статья организована следующим

образом. В разделе 1 описывается назначение брокера ресурсов и платформа для его реализации, во 2 разделе описываются варианты использования брокера ресурсов, 3 раздел содержит описание архитектуры брокера ресурсов. Процесс выделения ресурсов компонента CAEBeans Broker рассматривается в 4 разделе и в 5 разделе приводится алгоритм планирования ресурсов. В заключении суммируются основные результаты, полученные в данной работе.

1. Брокер ресурсов CAEBeans Broker

CAEBeans Broker — это компонент системы CAEBeans, который принимает задания от пользователя, согласовывает требования к ресурсам и находит наиболее подходящие вычислительные элементы для каждой из задач. Можно выделить следующие основные задачи брокера ресурсов: обработка базы данных ресурсов грид-среды; анализ запросов на предоставление ресурсов, поступающих от внешних клиентов; сбор и предоставление информации об актуальном состоянии грид-среды.

Реализация CAEBeans Broker производится на языке программирования Java в виде сервиса на базе платформы UNICORE. Данный подход обеспечивает независимость компонента от вычислительной платформы, предоставление полной информации о текущем состоянии экземпляра сервиса, а также поддерживает возможность надежного и безопасного исполнения, управление временем жизни; рассылку уведомлений об изменении состояния экземпляра сервиса, управление политикой доступа к ресурсам, управление сертификатами доступа.

2. Варианты использования компонента CAEBeans Broker

Программный компонент CAEBeans Broker представляет собой сервис, предоставляющий интерфейс для выбора наиболее подходящих ресурсов. В процессе работы брокера поиск требуемых ресурсов выполняется для нескольких заданий одновременно. Кроме этого, с брокером ресурсов взаимодействует вспомогательный компонент для сбора информации. Диаграмма вариантов использования компонента CAEBeans Broker приведена на рис. 1.

Вариант использования «Выделить ресурсы» передает запрос от клиента к брокеру ресурсов системы CAEBeans. В роли клиента выступает компонент CAEBeans Server [10], отвечающий за исполнение заданий и их мониторинг. Вариант использования начинается, когда CAEBeans Server указывает в запросе требования к ресурсам необходимым для выполнения задания. Вариант использования «Выделить ресурсы» включает в себя вариант использования «Установить резервирование», который позволяет установить резервирование выбранных ресурсов.

Вариант использования «Освободить ресурсы» уведомляет брокера ресурсов о необходимости освобождения выбранных ресурсов. Данный вариант использования также предполагает удаление заявки на резервирование ресурсов [11]. Соответствующий запрос на удаление ресурсов формируется вариантом использования «Удалить резервирование».

Вариант использования «Получить статус» запрашивает информацию о выделении ресурсов. В случае успешного выделения ресурсов клиент запрашивает список

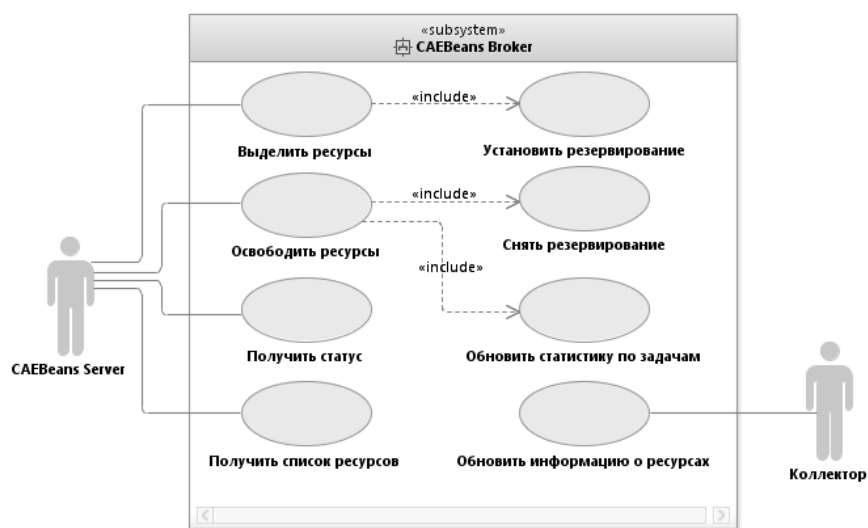


Рис. 1. Диаграмма вариантов использования подсистемы CAEBeans Broker

выделенных для его задания ресурсов с помощью варианта использования «Получить список ресурсов».

Варианты использования «Обновить информацию о ресурсах» и «Обновить статистику по задачам» обновляют информацию о характеристиках ресурсов и статистике выполнения заданий соответственно в каталоге брокера ресурсов.

3. Архитектура брокера ресурсов CAEBeans Broker

Согласно представленной на рис. 1 диаграмме вариантов использования была разработана следующая архитектура брокера ресурсов CAEBeans Broker (рис. 2). CAEBeans Broker состоит из следующих компонентов.

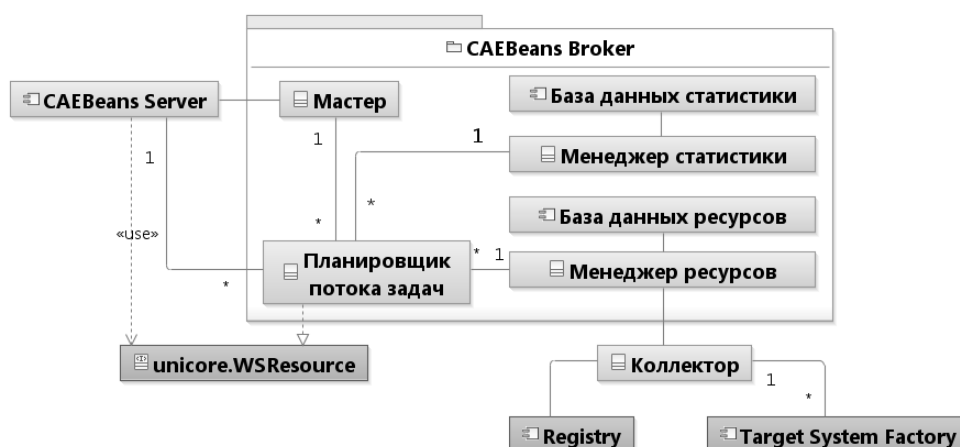


Рис. 2. Архитектура брокера ресурсов CAEBeans Broker

- Мастер принимает запросы от CAEBeans Server и создает экземпляр планировщика потока задач, представляющий собой WS-ресурс в терминах UNICORE.

- Экземпляр планировщика потока задач осуществляет обработку одного запроса. Формирует список требуемых для исполнения задания ресурсов и производит их резервирование.
- Менеджер ресурсов управляет базой данных ресурсов, содержащей информацию о целевых системах и резервировании ресурсов.
- Менеджер статистики управляет базой данных статистики, содержащей информацию о статистике выполнения задач.
- Коллектор работает независимо от CAEBeans Broker и осуществляет сбор информации для базы данных ресурсов.

Темным цветом на рис. 2 выделены компоненты платформы UNICORE, с которыми взаимодействуют компоненты системы CAEBeans.

4. Процесс выделения ресурсов компонента CAEBeans Broker

Взаимодействие компонентов брокера ресурсов, представленных на рис. 2, осуществляется согласно диаграммам последовательности выделения ресурсов в CAEBeans Broker (рис. 3) и сбора информации (рис. 4).

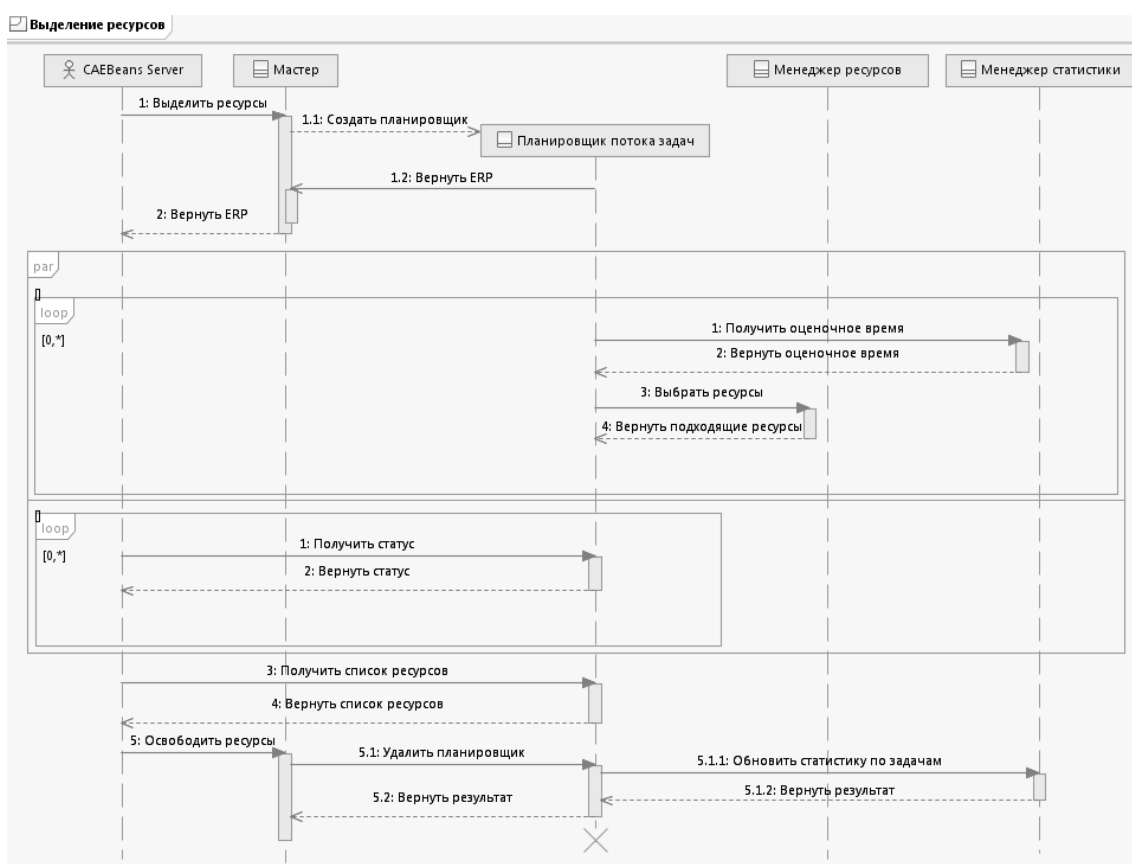


Рис. 3. Диаграмма последовательности выделения ресурсов в CAEBeans Broker

Рассмотрим процесс выделения ресурсов компонентом CAEBeans Broker.

1. Компоненту CAEBeans Server предоставляется доступ к брокеру ресурсов посредством сервиса Мастер. CAEBeans Server подключается к нему с помощью имеющегося сертификата безопасности. Затем CAEBeans Server вызывает метод

- «Выделить ресурсы» для выделения ресурсов заданию и передает методу в качестве входного параметра абстрактный поток задач, содержащий требования к ресурсам для каждой из задач.
2. После вызова компонентом CAEBeans Server метода «Выделить ресурсы» сервис Мастер выполняет следующую последовательность действий.
 - (a) Мастер создает экземпляр планировщика потока задач в виде WS-ресурса. Планировщик потока задач возвращает сервису Мастер свой уникальный идентификатор. В дальнейшем, любую информацию о процессе поиска ресурсов для задания можно получить непосредственно у экземпляра планировщика потока задач, обратившись к нему с помощью данного идентификатора.
 - (b) Мастер записывает уникальный идентификатор созданного экземпляра планировщика потока задач в свой пул.
 - (c) Мастер инициализирует специальный объект — конкретный поток задач, извлекая необходимую информацию из абстрактного потока задач. Конкретный поток задач содержит отображение каждой задачи из задания на конкретный ресурс. Мастер записывает конкретный поток задач с помощью фреймворка Ehcache сначала в кеш, а потом на диск. В качестве ключа к данному объекту используется уникальный идентификатор планировщика потока задач.
 - (d) Мастер возвращает компоненту CAEBeans Server уникальный идентификатор созданного экземпляра планировщика потока задач с помощью метода «Вернуть EPR». В качестве идентификатора выступает адрес конечной точки (Endpoint References, EPR).
 3. После создания планировщик потока задач считывает соответствующий конкретный поток задач из кеша или с диска. Дальнейшие шаги последовательно повторяются до завершения процесса выделения ресурсов.
 - (a) Планировщик потока задач получает оценку времени выполнения для каждой задачи у менеджера статистики, вызвав метод «Получить оценочное время».
 - (b) Планировщик потока задач начинает поиск требуемых ресурсов в базе данных ресурсов помощью метода «Выбрать ресурсы» посредством запроса к менеджеру ресурсов. В случае успешного поиска ресурсов планировщик потока задач получает список ресурсов, в противном случае — получает пустое сообщение. Выбранные ресурсы резервируются.
 - (c) CAEBeans Server проверяет статус выделения ресурсов, обращаясь непосредственно к своему экземпляру планировщика потока задач.
 4. В случае успешного выделения ресурсов CAEBeans Server запрашивает список выделенных для его задания ресурсов с помощью метода «Получить список ресурсов».
 5. После выполнения задания CAEBeans Server отправляет компоненту Мастер запрос на освобождение ресурсов.
 6. После вызова компонентом Мастер метода «Удалить планировщик» экземпляр планировщика потока задач снимает резервирование с выделенных ресурсов и обновляет статистику по выполнившемуся заданию в базе данных статистики.

Процесс сбора информации для базы данных ресурсов представлен на рис. 4. Шаги по обновлению информации повторяются последовательно в течение всего времени исполнения компонента Коллектор. Обновление информации о ресурсах подразумевает сбор информации об аппаратных, программных и лицензионных ресурсах, доступных CAEBeans Broker. Статистика выполнения задач включает информацию о реальном времени выполнения задач с конкретными параметрами на конкретных вычислительных узлах.



Рис. 4. Диаграмма последовательности сбора информации в системе CAEBeans Broker

5. Алгоритм планирования брокера ресурсов

В рамках данной работы предлагается разработать алгоритм планирования ресурсов в распределенных проблемно-ориентированных вычислительных средах, учитывающий дополнительные знания о специфике предметной области задания и представление о потоке работ; использующий резервирование ресурсов; управляющий как аппаратными, программными, так и лицензионными ресурсами распределенной вычислительной сети (РВС).

Предлагается использовать следующие методы:

- метод двухфазного резервирования ресурсов;
- метод кластеризации доминирующей последовательности;
- учет проблемных параметров задачи для оценки ее времени выполнения.

МЕТОД ДВУХФАЗНОГО РЕЗЕРВИРОВАНИЯ РЕСУРСОВ. Резервирование ресурсов в РВС в настоящее время является сложной задачей, поскольку требует наличия в системе управления задачами компонента для централизованного планирования ресурсов и мгновенного принятия решения о резервировании ресурсов. Для решения данной проблемы предлагается использовать метод двухфазного резервирования ресурсов. Данный метод позволяет выполнять на первой фазе предварительное резервирование (маркировку) требуемых ресурсов и одновременно производить поиск оптимального расписания для потока задач. Для планирования отдельного пото-

ка задач отводится некоторое время, ограниченное приоритетом выполнения потока задач. По истечении данного времени первая фаза завершится. В рамках выполнения второй фазы происходит окончательное выделение ресурсов для потока задач или освобождение всех маркированных ресурсов. В обоих случаях операции выделения/освобождения выполняются атомарно.

МЕТОД КЛАСТЕРИЗАЦИИ ДОМИНИРУЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ. Отображение потока задач на вычислительные ресурсы производится с помощью адаптированного метода кластеризации доминирующей последовательности (Dominant Sequence Clustering, DSC [12]). Суть метода DSC сводится к минимизации параллельного времени выполнения потока задач. Однако в оригинальном методе DSC планирование осуществляется в два этапа. На первом этапе происходит объединение нескольких задач в кластеры (группы), на втором этапе — отображение кластеров задач на реальные вычислительные ресурсы. Адаптация данного метода подразумевает использование предварительного резервирования ресурсов на этапе объединения кластеров: поиск свободных слотов на вычислительных ресурсах производится непосредственно при планировании потока задач.

УЧЕТ ПРОБЛЕМНЫХ ПАРАМЕТРОВ ЗАДАЧИ ДЛЯ ОЦЕНКИ ЕЕ ВРЕМЕНИ ВЫПОЛНЕНИЯ. Время выполнения отдельных задач в потоке задач по умолчанию указывается пользователем. Однако в большинстве случаев пользователи указывают существенно завышенное время. Для более точной оценки времени выполнения задач статистика по задачам хранится в отдельной базе данных, которой управляет менеджер статистики. Менеджер статистики предоставляет информацию о точном времени исполнения задач, исходя из имеющихся данных о значениях параметров задачи и архитектуре вычислительных узлов, на которых данная задача исполнялась ранее.

На рис. 5 представлен алгоритм планирования ресурсов CAEBeans Broker. На начальном этапе задание представлено в виде ориентированного ациклического графа. Вершинами графа являются задачи. Каждая вершина является отдельной группой. Необходимо разбить вершины графа на группы таким образом, чтобы количество групп равнялось числу имеющихся вычислительных ресурсов.

Первая фаза алгоритма начинается со строки номер 1. Данная фаза ограничена по времени. В соответствии с оригинальным алгоритмом находим максимальное параллельное время (Parallel Time, PT) и доминирующую последовательность (Dominant Sequence, DS). Затем производим поиск ребра с максимальным весом. Если объединение вершин, соединенных данным ребром, уменьшит параллельное время, то произведем их объединение в одну группу.

Адаптируем оригинальный алгоритм DSC следующим образом. Сопоставим каждому вычислительному ресурсу некоторый цвет. Объединение вершин графа будем производить только в том случае, если для данных вершин существуют подходящие свободные слоты на одном вычислительном ресурсе. Вершины, объединенные в одну группу, раскрасим в один цвет — цвет ресурса. Если подходящих слотов найдено не было, то находим следующее ребро с максимальным временем и повторим процедуру поиска слотов для его вершин.

На шаге 17 начинается вторая фаза алгоритма. Произведем поиск вершин графа, для которых не было зарезервировано слотов на вычислительных ресурсах. Обозначим множество нераспланированных вершин как $V_{nonsched}$. Выделим для каждой

вершины из множества $V_{nonsched}$ первые подходящие свободные слоты на вычислительных ресурсах. Произведем окончательное выделение ресурсов для всех вершин графа.

```

1.  while time > 0 do
2.    while количество групп вершин > количество вычислительных ресурсов do
3.      найти  $PT_{max}$  и соответствующий ему  $DS_{max}$ 
4.      выполнить сортировку весов дуг  $DS_{max}$ , результат записать в массив  $W_E$ 
5.      for i=1 to  $|W_E|$  do
6.        найти вершины  $n_k$  и  $n_m$ , соединяемые i-той дугой
7.        if существуют подходящие слоты для вершин  $n_k$  и  $n_m$ 
8.          на одном вычислительном ресурсе then
9.            объединить вершины в одну группу
10.           раскрасить вершины цветом, соответствующим выбранному ресурсу
11.           break for
12.         end if
13.       end for
14.     end while
15.     time --;
16.  end while
17.  if  $V_{nonsched} == \emptyset$  then
18.    for j=1 to  $|V_{nonsched}|$  do
19.      найти слот  $S_j$ , на котором задача j будет запущена раньше;
20.      зарезервировать слот  $S_j$ ;
21.    end for
22.  end if
23.  выделить ресурсы;

```

Рис. 5. Алгоритм планирования ресурсов CAEBeans Broker

6. Заключение

В данной статье рассмотрен алгоритм планирования ресурсами в распределенных проблемно-ориентированных вычислительных средах, учитывающий дополнительные знания о специфике предметной области задания и представление о потоке задач; использующий резервирование ресурсов; управляющий как аппаратными, программными, так и лицензионными ресурсами РВС. Приведены варианты использования брокера ресурсов, описаны процесс выделения ресурсов и архитектура брокера ресурсов CAEBeans Broker.

В качестве дальнейших направлений работы можно выделить следующее: проведение вычислительных экспериментов для оценки эффективности алгоритма планирования на суперкомпьютерах «СКИФ-Аврора ЮУрГУ» и «СКИФ Урал» Суперкомпьютерного центра ЮУрГУ.

Работа выполнена при поддержке грантов РФФИ № 11-07-00478-а и № 12-07-31076, гранта Президента Российской Федерации МК-1987.2011.9, в рамках государственного задания Министерства образования и науки РФ 8.3786.2011.

Литература

1. Foster, I. The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. – San Francisco: Morgan Kaufman, 2003. – P. 748.
2. Raphael, B. Fundamentals of computer aided engineering / B. Raphael, I. F. C. Smith. – London: John Wiley, 2003. – P. 324.
3. Радченко, Г.И. Сервисно-ориентированный подход к использованию систем инженерного проектирования и анализа в распределенных вычислительных средах / Г.И. Радченко // Параллельные вычислительные технологии (ПаВТ'2011): Труды международной научной конференции (Москва, 28 март. – 1 апр. 2011 г.). – Челябинск: Издательский центр ЮУрГУ, 2011. – С. 606 – 616.
4. Yu, J. A Taxonomy of Workflow Management Systems for Grid Computing / J. Yu, R. Buyya // Grid Computing. – 2005. – V. 3, № 3. – P. 171–200.
5. Condor. High Throughput Computing. URL: <http://www.cs.wisc.edu/condor/> (дата обращения 20.05.2012)
6. Laszewski, G. CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids / G. Laszewski, I.Foster // Java Grande of the ACM. – Jun. 2000. – P. 97–106.
7. Deelman, E. Pegasus: Mapping Scientific Workflows onto the Grid / E. Deelman, J. Blythe // Grid Computing: Second European AcrossGrids Conference (AxGrids 2004). – Jan. 2004. – P. 11–26.
8. Cao, J. GridFlow: Workflow Management for Grid Computing / J. Cao, S. A. Jarvis // International Symposium on Cluster Computing and the Grid (CCGrid'03). – May. 2003. – P. 198–205.
9. Wiczorek, M. Scheduling of Scientific Workflows in the ASKALON Grid Environment / M. Wiczorek, R. Prodan, T. Fahringer // ACM SIGMOD Record. – 2005. – V. 34, № 3. – P. 56–62.
10. Федянина, Р.С. CAEBeans Server: среда выполнения проблемно-ориентированных оболочек над инженерными пакетами / Р.С. Федянина // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 март. – 2 апр. 2010 г.). – Челябинск: Издательский центр ЮУрГУ, 2010. – С. 621 – 628.
11. Mateescu, G. Quality of Service on the Grid via Metascheduling with Resource Co-Scheduling and Co-Reservation / G. Mateescu // High Performance Computing Applications. – 2003. – V. 17, № 3. – P. 209–218.
12. Yang, T. DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors / T. Yang, A. Gerasoulis // IEEE Transactions on Parallel and Distributed Systems. – 1994. – V. 5, № 9. – P. 951–967.

Анастасия Валерьевна Шамакина, старший преподаватель, кафедры «Системное программирование», Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), sham2004@bk.ru.

BROKERING SERVICE FOR SUPPORTING PROBLEM-ORIENTED GRID ENVIRONMENT

A. V. Shamakina, South Ural State University (Chelyabinsk, Russian Federation)

This paper describes scheduling methods and algorithms of resources, and also the development on their base of the broker resource for search optimum resources in problem-oriented grid-environment. The developed scheduling algorithm considers additional knowledge about subject domain specifics of tasks and the representation about a workflow. The algorithm is based on a dominant sequence clustering algorithm (DSC). Unlike the original algorithm is that, for mapping tasks on the computing resources used by task graph coloring and the merging of tasks in a cluster is based on the availability of free slots on computing resources. Proposed a diphasic reservation method of resources and accounting problem of the problem parameters to estimate the time of its execution. Use cases of the broker resource are also given, process of resource allocation and architecture of the resource broker CAEBeans Broker are described.

Keywords: broker resource, scheduling algorithms of resources, grid, reservation, UNICORE.

References

1. Foster I., Kesselman C. The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure. San Francisco: Morgan Kaufman, 2003. 748 p.
2. Raphael B., Smith I.F.C. Fundamentals of Computer-Aided Engineering. London: John Wiley, 2003. 324 p.
3. Radchenko G.I. Servisno-orientirovannyi podkhod k ispol'zovaniyu system inzhenerenogo proektirovaniya i analiza v raspredelennykh vychislitel'nykh sredakh [A Service-Oriented Approach to Using of CAE-systems in Distributed Computing Environments]. Parallelnye vychislitelnye tekhnologii (PaVT'2011): Trudy mezhdunarodnoj nauchnoj konferentsii (Moskva, 28 marta — 1 aprelya 2011) [Parallel Computational Technologies (PCT'2011): Proceedings of the International Scientific Conference (Moscow, Russia, March 28 — April 1, 2011)]. Chelyabinsk, Publishing of the South Ural State University, 2011. P. 606–616.
4. Yu J., Buyya R. A Taxonomy of Workflow Management Systems for Grid Computing. Grid Computing, 2005. V. 3, No 3. P. 171–200.
5. Condor. High Throughput Computing. URL: <http://www.cs.wisc.edu/condor/>
6. Laszewski G., Foster I. CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids. Java Grande of the ACM. 2000. P. 97–106.
7. Deelman E., Blythe J. Pegasus: Mapping Scientific Workflows onto the Grid. Grid Computing: Second European AcrossGrids Conference (AxGrids 2004). 2004. P. 11–26.
8. Cao J., Jarvis S.A. GridFlow: Workflow Management for Grid Computing. International Symposium on Cluster Computing and the Grid (CCGrid'03). 2003. P. 198–205.

9. Wicczorek M., Prodan R., Fahringer T. Scheduling of Scientific Workflows in the ASKALON Grid Environment. ACM SIGMOD Record, 2005. V. 34, No 3. P. 56–62.
10. Fedyanina R.S. CAEBeans Server: sreda vypolneniya problemno-orientirovannykh obolokhek nad inzhenernymi paketami [CAEBeans Server: the Runtime Environment of Problem-oriented Shells over Engineering Packages]. Parallelnye vychislitelnye tekhnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferentsii (Ufa, 29 marta — 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March 29 — April 2, 2010)]. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 621–628.
11. Mateescu G. Quality of Service on the Grid via Metascheduling with Resource Co-Scheduling and Co-Reservation. High Performance Computing Applications, 2003. V. 17, No 3, P. 209–218.
12. Yang T., Gerasoulis A. DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors. IEEE Transactions on Parallel and Distributed Systems, 1994. V. 5, No 9, P. 951–967.

Поступила в редакцию 6 августа 2012 г.