

ОРГАНИЗАЦИЯ БРОКЕРА РЕСУРСОВ В СИСТЕМЕ CAEBEANS¹

А.В. Шамакина

В настоящее время все большее распространение получают глобальные вычислительные сети Grid. Они были предложены в качестве новой парадигмы для решения крупномасштабных вычислительных задач в науке, технике и бизнесе. Это связано с тем, что сети Grid дают возможность одновременного использования миллионов вычислительных ресурсов, принадлежащих различным организациям и расположенных в различных административных областях. Системы Grid объединяют разнородные вычислительные ресурсы (персональные компьютеры, рабочие станции, кластеры, суперкомпьютеры), используя разные стратегии доступа к ним, выполняя различные приложения (научные, инженерные и коммерческие), предъявляющие к системе различные требования. Ресурсы принадлежат различным организациям, имеющим свои правила управления ресурсами, их использования и определения их стоимости для различных пользователей в различное время. Доступность и загруженность ресурсов также может динамически изменяться во времени.

В окружении Grid владельцы и потребители ресурсов имеют различные цели, используют различные стратегии и экономические схемы регулирования спроса и предложения. Таким образом, актуальной проблемой является разработка систем управления ресурсами Grid, нацеленных на оптимизацию отношений между владельцами ресурсов и пользователями в соответствии с выбранными ими стратегиями. Для решения этой проблемы создаются системы управления ресурсами Grid.

Однако ни одна из ранее созданных систем управления ресурсами не удовлетворяет специфике системы CAEBeans, в рамках которой необходимо учитывать не только характеристики ресурсов, но и наличие установленных инженерных пакетов, количество доступных лицензий на них и др. В системе CAEBeans в качестве Grid Middleware был выбран Unicore 6.0.

Брокер ресурсов принимает задания от пользователя, согласовывает требования к ресурсам и направляющий задания на подходящий вычислительный элемент. Можно выделить следующие основные задачи брокера ресурсов: обработка каталога ресурсов грид-среды; анализ запросов на предоставление ресурсов, поступающих от компонентных CAEBean; сбор и предоставление информации об актуальном состоянии грид-среды. В работе предложены новая архитектура брокера ресурсов и алгоритмы планирования.

В процедуре проекции подзадачи с физического на системный слой брокер ресурсов выступает в качестве промежуточного звена, обеспечивающего поиск и предоставление ресурсов, оптимальных для решения данной подзадачи (см. рис. 1).

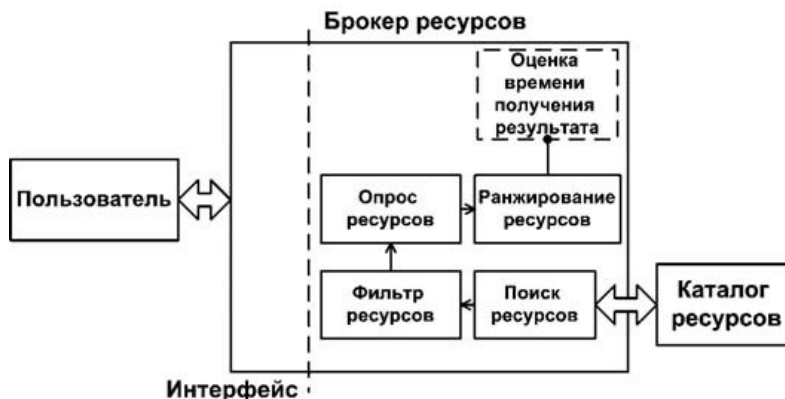


Рис. 1. Архитектура брокера ресурсов

Рассмотрим более подробно алгоритм выделения и использования ресурсов грид-среды с участием брокера ресурсов.

1. Компонентный CAEBean производит запрос к брокеру на предоставление ресурса.
2. Брокер производит запрос к каталогу ресурсов, формируя список узлов, характеристики которых удовлетворяют заявленным обязательным требованиям и правам пользователя, поставившего задачу.

¹ Работа выполнена при финансовой поддержке Федерального агентства по науке и инновациям (контракт 2007-4-1.4-20-01-026) и программы СКИФ-ГРИД (контракт СГ-1/07).

3. Брокер производит опрос узлов из выбранного списка, для уточнения их доступности и определения значений их динамических характеристик (состояние очереди задач, доступный объем оперативной памяти и т.п.).
4. Брокер оценивает общее время получения результата (TTD — Total Time to Delivery) для каждого из анализируемых узлов и выбирает узел, с наименьшим значением данного параметра.
5. Брокер передает адрес предоставляемого ресурса компонентному CAEBean.
6. Компонентный CAEBean производит захват предоставленного ресурса, а брокер маркирует данный ресурс как занятый.
7. Дальнейший процесс постановки, решения и получения результатов происходит напрямую, между компонентным CAEBean и предоставленным ресурсом без участия брокера.
8. После получения результатов вычисления компонентный CAEBean производит освобождение ресурса и извещение брокера об его освобождении.

Одной из наиболее важных задач брокера ресурсов является оценка общего времени получения результата (TTD — Total Time to Delivery) при постановке задачи пользователем конкретному вычислительному ресурсу. В дальнейшем, на основе этой оценки, определяется ресурс, обеспечивающий оптимальное время исполнения поставленной задачи. При оценке TTD необходимо учитывать, что от постановки до получения конечных результатов задача, как правило, проходит следующие стадии решения.

1. *Постановка.* Производится передача исходных файлов и сценариев решения задачи выбранному ресурсу.
2. *Ожидание.* Это время, необходимое для запуска задачи. Состоит из времени ожидания задачи в очереди задач выбранного ресурса, времени постановки задачи и т.п.
3. *Исполнение.* Проведение вычислений на выбранном ресурсе.
4. *Завершение.* Передача файлов результатов в требуемое хранилище.

Необходимо отметить, что время ожидания обычно определяется как максимум из времени на постановку задачи и времени на все остальные операции, необходимые для того, чтобы запустить задачу на исполнение. Это связано с тем, что процесс передачи исходных файлов может происходить независимо от всех остальных процессов подготовки задачи. Таким образом, TTD может быть оценено как сумма оценок времени на проведение операций ожидания, исполнения и завершения.

При формировании данных оценок, брокер использует информацию о подзадаче, передаваемую соответствующим компонентным CAEBean. В процессе запроса вычислительных ресурсов компонентный CAEBean может предоставить брокеру такую информацию, как размер исходных данных; вычислительная сложность рассматриваемой подзадачи; масштабируемость подзадачи и т.п. На основе этих данных и информации о текущих характеристиках грид-среды брокер ресурсов может произвести оценку общего времени получения результата.

Представленный подход разработан для поддержки технологии CAEBeans, ориентированной на сервисно-ориентированное представление ресурсов CAE-пакетов в рамках гетерогенных вычислительных сред. В качестве аппаратной поддержки технологии CAEBeans предполагается задействовать вычислительные ресурсы суперкомпьютерных систем, входящих в грид-среду СКИФ-Полигон, созданную в рамках Российско-Белорусского суперкомпьютерного проекта СКИФ-Грид.

ЛИТЕРАТУРА:

1. I. Foster, C. Kesselman The Grid 2: Blueprint for a New Computing Infrastructure. Second edition. –San Francisco: Morgan Kaufmann, 2003.
2. E. Elmroth, J. Tordsson A Grid resource broker supporting advance reservations and benchmark-based resource selection // State-of-the-art in Scientific Computing, Springer-Verlag, LNCS. – 2006. – № 3732. – P. 1061-1070.
3. A. Grimshaw, W. Wulf The Legion Vision of a Worldwide Virtual Computer // Comm. ACM. — 1997. — 40(1). –P. 39–45.
4. А.И. Аветисян, С.С. Гайсарян, Д.А. Грушин, Н.Н. Кузюрин, А.В. Шокуров Эвристики распределения задач для брокера ресурсов Grid // Труды Института системного программирования РАН, т. 5, 2004, стр. 41-62.
5. А.Е. Дорошенко, О.В. Алистратов, Ю.М. Тырчак, А.П. Розенблат, К.А. Рухлис Системы Grid-вычислений - перспектива для научных исследований // Пробл. программирования. –2005. –N 1. –С. 14-38.
6. Г.И. Радченко, Л.Б. Соколинский CAEBeans: иерархические системы структурированных проблемно-ориентированных оболочек над инженерными пакетами // Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ: тр. Всерос. науч. конф. (24-29 сентября 2007 г., г. Новороссийск). -Новороссийск, 2007. –С. 54-57.
7. Г.И. Радченко, Л.Б. Соколинский, И.С. Кутепов BeanShells: интеграция CAE-пакетов в GPE // Параллельные вычислительные технологии (ПаВТ'2007): тр. Междунар. науч. конф. (Челябинск, 29 янв. – 2 февр. 2007 г.). -Челябинск, 2007. –Т.2. –С. 15.
8. Condor High Throughput Computing: [<http://www.cs.wisc.edu/condor/>], 23 June 2008.
9. Uniform Interface to Computing Resources: [<http://www.unicore.eu/>], 19 Jun 2008.