

ПРОБЛЕМЫ МОДЕЛИРОВАНИЯ GRID-СИСТЕМ И ИХ РЕАЛИЗАЦИИ

О.И. Самоваров, Н.Н. Кузюрин, Д.А. Грушин, А.И. Аветисян, Г.М. Михайлов, Ю.П. Рогов

ВВЕДЕНИЕ

В последние годы за рубежом и в нашей стране активно ведутся работы, связанные с объединением глобальных вычислительных ресурсов в единую распределенную систему вычислений, хранения и передачи информации. Такая система (Grid [1]) обеспечивает доступ пользователей к миллионам географически распределенных компьютерных ресурсов.

В настоящее время уже создана техническая база (телекоммуникационные сети большой пропускной способности) и сделаны шаги по созданию программного обеспечения для работы в среде Grid. Создан свободно распространяемый программный пакет Globus Toolkit [2] с открытым исходным кодом, который предлагает базовые средства для создания Grid-инфраструктуры: средства обеспечения безопасности в распределенной среде, средства надежной передачи больших объемов данных, средства запуска и получения результатов выполнения задач на удаленных вычислительных ресурсах. На базе пакета Globus Toolkit так же создаются промышленные версии реализаций Grid-инфраструктуры.

Несмотря на уже существующие средства создания Grid-инфраструктур, существует ряд важных научных задач, без решения которых невозможно полномасштабное использование возможностей Grid- технологий в науке и промышленности. Одной из таких актуальных задач является эффективное управление ресурсами в распределенной среде. Отсутствие хорошего планировщика, обеспечивающего управление потоком задач, не только значительно снижает эффективность использования всей Grid-инфраструктуры, но может сделать необоснованным ее создание. Разработка такого планировщика должна опираться не только на анализ существующих Grid-инфраструктур, но и на изучение тенденций их развития на основе тестовых стендов (сегментов Grid) и высокоуровневых моделей, адекватно отражающих свойства таких сложных систем.

В связи с этим одной из актуальных задач является создание системы моделирования распределенной вычислительной Grid-инфраструктуры, которая позволит адекватно оценивать ее поведение при изменяющихся условиях и оптимизировать стратегию управления потоками задач. В настоящее время существует несколько проектов по разработке систем моделирования распределенных Grid-систем. Среди них наиболее известными являются Bricks [3], MicroGrid[4], OptorSim[5], SimGrid[6]. Каждая из них имеет свои достоинства и недостатки. Среди недостатков можно отметить узкую специализацию систем, отсутствие публично доступных версий, а также ограниченность моделируемых архитектур Grid-систем. Особенности реализаций некоторых из них накладывают ограничения на количество одновременно существующих элементов в Grid-системе и требуют от пользователя знания специальных языков программирования, что существенно снижает эффективность их использования.

В ИСП РАН в рамках работ по параллельным вычислениям и Grid-технологиям разрабатывается среда моделирования распределенных вычислительных систем. Кроме того, совместно с ВЦ РАН и МФТИ на базе существующих вычислительных ресурсов ведутся работы по созданию стенда как для работ проведения по верификации модели, так и для поддержки учебного процесса базовых кафедр МФТИ и ВМК МГУ.

МОДЕЛЬ GRID-СИСТЕМ

Архитектура среды моделирования Grid-систем состоит из следующих основных компонентов: графический интерфейс пользователя, модель Grid-системы, транслятор, выполняемая модель, монитор выполнения, модуль статистики.

Основной вариант использования такой модели можно описать следующим образом:

1. Пользователь с помощью графического интерфейса задает характеристики моделируемой Grid-системы и определяет параметры моделирования.
2. Транслятор преобразует это описание Grid-системы в код программы моделирования. Получение выполняемой модели происходит автоматически, от пользователя не требуется никаких дополнительных действий. Если описание содержит ошибки, транслятор сообщает об этом пользователю.
3. Пользователь запускает модель. Во время работы программы монитор осуществляет сбор информации о происходящих событиях. Данная информация сохраняется в профиле выполнения, анализируется модулем статистики и выводится пользователю. После завершения выполнения профиль может анализироваться отдельно.

Описание моделируемой среды производится с помощью специального языка, имеющего как графическую, так и текстовую нотацию. В пользовательском интерфейсе реализована возможность просмотра и редактирования

различных аспектов модели: списка кластеров, потоков задач, характеристик отдельной задачи, брокеров, сетевых соединений и т.д.

Правила описания определяются мета-моделью. Мета-модель Grid-системы включает в себя следующие основные компоненты: поток задач, брокер, кластер, сетевое соединение, хранилище данных.

ОПИСАНИЕ ПОТОКА ЗАДАЧ

Поток задач может быть описан, например, одним из следующих способов. Реальный поток (“workload”) позволяет задавать поток задач на основе собранных статистических данных использования уже существующей Grid-системы. Для порождения реального потока задач в качестве исходных данных используется текстовый файл в формате “Workload” [7]. Каждая строка такого файла содержит информацию о свойствах задачи, времени ее порождения и работы. Синтетический поток задач позволяет определить в модели поток задач по следующим начальным параметрам:

- свойствам задачи;
- периоду - времени до порождения очередной задачи;
- количеству порождаемых задач;
- задержке перед началом порождения первой задачи.

Пользователь имеет возможность задать определение своего собственного потока задач, предоставив Java-класс, реализующий специальный интерфейс на определенной модели.

ОПИСАНИЕ АЛГОРИТМОВ РАСПРЕДЕЛЕНИЯ

Среда моделирования Grid-систем позволяет описать алгоритмы распределения потока заданий между ресурсными центрами распределенной среды брокером, а также алгоритмы распределения заданий локальными планировщиками на узлы каждого кластера. Алгоритм может быть описан либо с помощью встроенного языка, либо подключен в виде Java-класса, определяемого пользователем и реализующего интерфейс LocalScheduler для локального планировщика или MetaScheduler для брокера.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ

По данной модели был разработан и реализован прототип системы моделирования распределенных Grid-систем. Существенными особенностями реализации является:

1. использование архитектуры MDA [8], когда описание моделируемой среды производится на специальном языке предметной области и затем автоматически транслируется в код программы, которая представляет собой выполняемую модель. Это позволяет пользователю быстро задать характеристики моделируемой среды и получить результат. Также следует отметить возможность существования нескольких реализаций выполняемой модели, например, на языках Java или C++.
2. Система позволяет моделировать различные Grid-архитектуры: одно- и двухуровневые системы с одним или несколькими брокерами, добавлять хранилища данных, определять топологию сетевых соединений и т.д.
3. Выполняемая модель реализована в виде конечных автоматов. Это существенно увеличивает производительность и масштабируемость системы по сравнению с использованием отдельных Java-потоков для каждого элемента моделируемой среды.
4. Система предоставляет возможность для быстрого описания эвристик распределения задач с помощью набора правил. При моделировании распределения задач в Grid-системе часто требуется проверить несколько эвристик, незначительно отличающихся друг от друга, например, сортировкой входного потока задач способом выбора очередной задачи или ресурса и т.п. С помощью представленного языка правил возможно наиболее быстро описать эвристику распределения и проверить ее работу.
5. В системе поддерживается возможность проведения серии экспериментов, состоящей из последовательных запусков выполняемой модели с изменением некоторых параметров при каждом следующем запуске. Например, может изменяться поток задач, конфигурация кластеров, сетевых соединений и т.п. Это позволяет в рамках одного эксперимента просмотреть динамику изменения эффективности системы и определить узкие места.
6. В системе реализован удобный механизм обработки результатов моделирования. Результат выполнения модели хранится в отдельном профиле и может обрабатываться независимо. Пользователь может использовать свой шаблон для выбора и визуализации только необходимой в данный момент информации. Это позволяет нескольким исследователям провести моделирование один раз, а затем независимо анализировать полученную информацию.
7. Система включает в себя готовые шаблоны для отображения:

- загруженности системы – общей и с разбивкой по отдельным кластерам,
- времени ожидания задач в очереди – среднего и пикового с разбивкой по классам задач,
- пропускной способности, как по количеству задач, так и с использованием интегральной оценки.

Система моделирования Grid реализована на языке Java и на основе платформы Eclipse. Это дает возможность интеграции с другими Eclipse-приложениями, например, средой разработки Java, системами контроля версий и позволяет использовать систему моделирования под операционными системами Linux, Windows, Solaris и др.

АПРОБАЦИЯ И ТЕСТИРОВАНИЕ

Цель экспериментов заключалась в следующем. С использованием реализованного прототипа среды смоделировать поведение реально существующей распределенной Grid- системы при различных условиях распределения потока заданий. В качестве распределенной Grid-системы была выбрана сеть Sharcnet[9]. Распределенная Grid-система Sharcnet (Shared Hierarchical Academic Research Computing Network) – это консорциум из 16 колледжей и университетов в юго-западной части провинции Канады Онтарио, вычислительные ресурсы которых объединены высокоскоростной оптической сетью (Рис. 2).

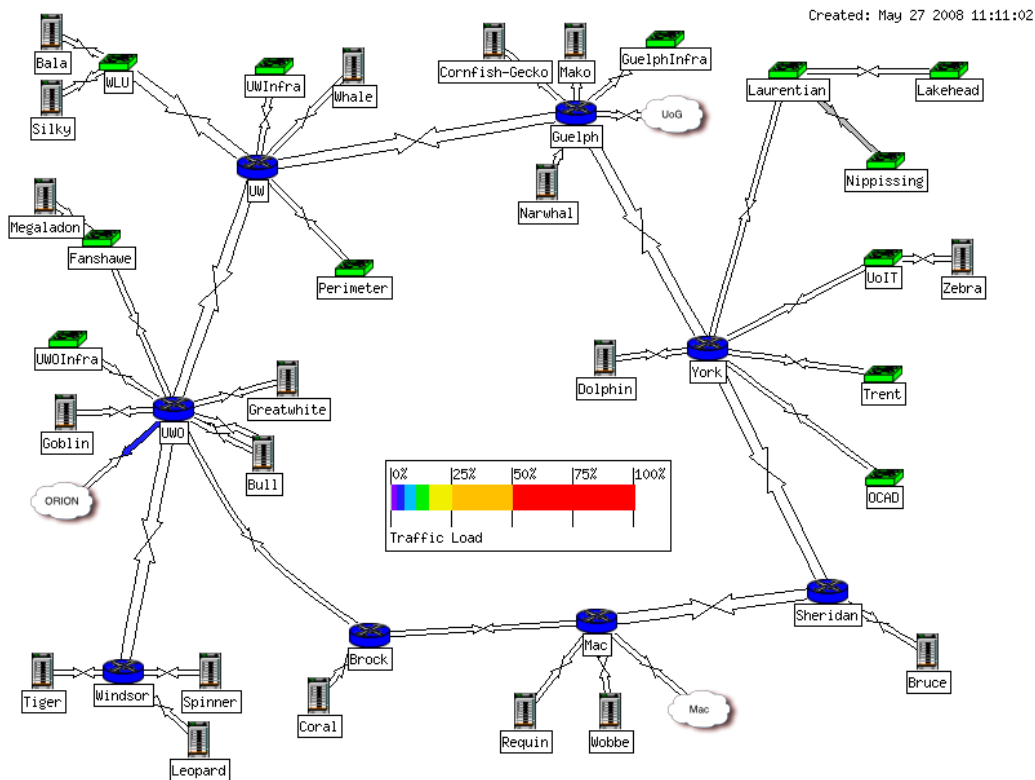


Рис. 1. Схема распределенной Grid- системы Sharcnet.

Характеристики вычислительных ресурсов сети Sharcnet представлены в таблице 1.

Название кластера	Общее число вычислительных ядер	Число узлов
bruce	128	32 x 4 x Opteron
narwhal	1068	267 x 4 x Opteron dual core
tiger	128	32 x 4 x Opteron
bull	384	96 x 4 x Opteron
megaladon	128	32 x 4 x Opteron
dolphin	128	32 x 4 x Opteron
requin	1536	768 x 2 x Opteron
whale	3072	768 x 4 x Opteron
zebra	128	32 x 4 x Opteron
bala	128	32 x 4 x Opteron

Таблица 1: Вычислительные ресурсы сети Sharcnet.

В качестве входных данных была использована запись реальных задач (“workload”- поток), выполнявшихся на указанных Grid-кластерах с декабря 2005 по январь 2007 года. Особенность потока состоит в том, что задачи поступали на кластеры непосредственно, т.е. для распределения не использовался брокер. Мы применили разработанную систему моделирования и сравнили эффективность распределения задач в сети Sharcnet в оригинальном случае (без брокера) с распределением, получаемым с помощью брокера:

Без брокера задачи поступали на кластеры в оригинальной последовательности, указанной в файле загрузки. На каждом кластере использовалась эвристика ”первый подходящий“. Задачи, находящиеся в очереди на кластере, перебираются в том порядке, в котором они поступили. Если задачу возможно в данный момент разместить на кластере, то она размещалась.

С брокером задачи поступают в очередь брокера. Брокер распределяет задачи по отдельным кластерам. На каждом кластере использовалась эвристика ”первый подходящий“. На брокере применялась эвристика ”ширина задачи“. Брокер последовательно обрабатывает задачи, поступившие к нему, и направляет каждую из них на кластер, на котором отношение суммы требуемого числа узлов всех задач в очереди к числу узлов кластера минимальны.

Всего было проведено семь экспериментов. На графиках номера e1, e2 и т.д. будут обозначать номера экспериментов.

e1. Задачи распределялись на кластеры согласно файлу загрузки.

e2. Задачи направлялись на брокер, который затем распределял их на кластеры. На брокере использовалась эвристика NW, т.е. при распределении очередной задачи брокер выбирает кластер, для которого значение выраже-

ния $M = \frac{N}{W}$, где N - число задач в очереди кластера, W - ширина кластера, минимально.

$$M = \frac{\sum W_j + W_t}{W}$$

e3. На брокере использовалась эвристика WW - минимально, где W_j - сумма ширин задач в очереди кластера, W_t - ширина задачи.

$$M = \frac{\sum S_j + S_t}{W}$$

e4. На брокере использовалась эвристика SW - минимально, где S_j - сумма площадей задач в очереди кластера, S_t - площадь задачи.

e5, e6, e7. Использовались эвристики NW, WW, SW соответственно, но на брокер направлялись только задачи единичной ширины.

Некоторые результаты экспериментов представлены на рисунках 3, 4, 5, 6.

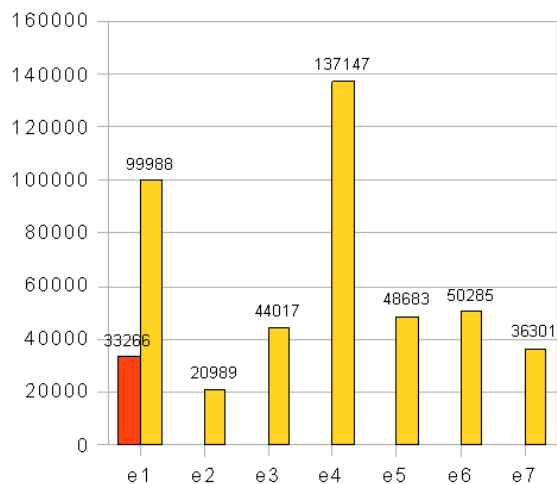


Рис. 2. Общее среднее время ожидания заданий в очереди (секунды).

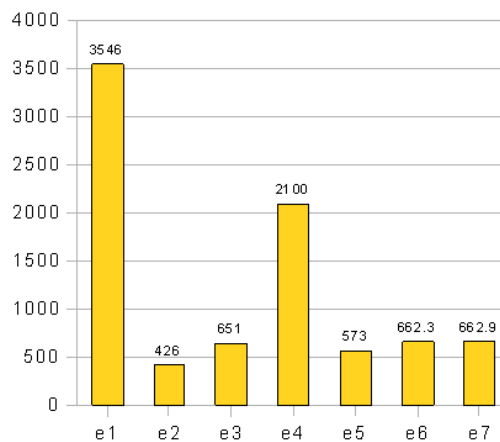


Рис. 3. Общее среднее число задач в очереди

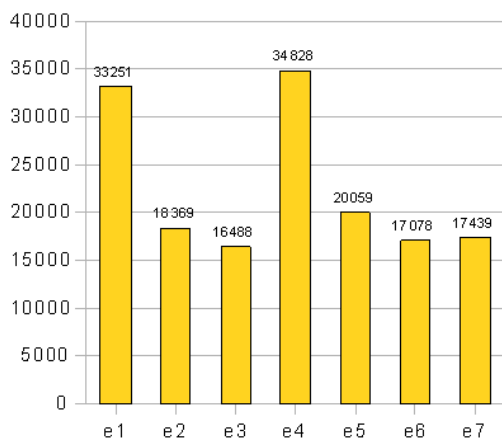


Рис. 4. Общее пиковое число задач в очереди

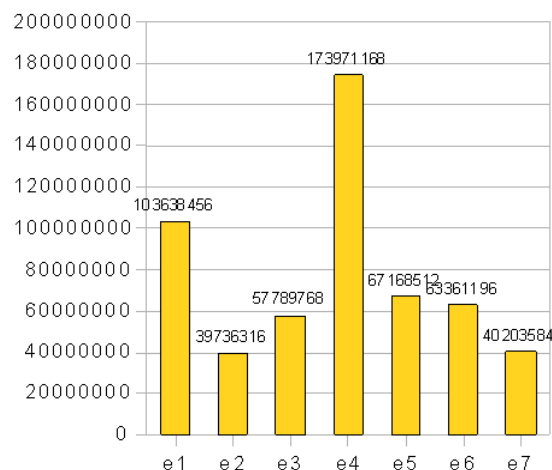


Рис. 5. Общая средняя реальная площадь очереди

Из диаграммы видно, что для данной модели распределенной Grid-системы распределение потока заданий через брокер с использованием эвристик, кроме случая e4, дает значительный эффект и снижает среднее время ожидания заданий в очереди, а также обеспечивает более равномерную загрузку вычислительных ресурсов.

ЗАКЛЮЧЕНИЕ

В статье представлена среда моделирования, которая позволяет адекватно оценивать поведение распределенных Grid-систем при изменяющихся условиях и оптимизировать стратегии управления потоками задач. Также представлены результаты использования реализованного прототипа данной среды на моделировании реально существующей Grid-системы Sharcnet.

ИСП РАН при сотрудничестве с ВЦ РАН а также с МФТИ на базе существующих вычислительных ресурсов ведет работы по созданию распределенного учебного лабораторного стенда для проведения практических занятий по подготовке специалистов по параллельному программированию на кафедрах системного программирования МФТИ и ВМК МГУ. Кроме того, при сотрудничестве специалистов ИСП РАН и ВЦ РАН ведутся работы по созданию Grid-сегмента для выполнения научных расчетов. Среди прикладных задач, выполняемых на данном Grid-сегменте, можно отметить выполнение расчетов с использованием разработанного специалистами ВЦ РАН параллельного варианта алгоритма отыскания глобального экстремума функции многих переменных, основанного на методе неравномерных покрытий, предложенном академиком Ю.Г. Евтушенко [10].

Дальнейшее направление работ по развитию распределенной Grid-системы базируется на модели, представленной в данной работе.

ЛИТЕРАТУРА:

1. Foster I., Kesselman C. The GRID: Blueprint for a New Computing Infrastructure. Second edition, Morgan Kaufmann Publishers, 2004.
2. Globus Toolkit Home Page. <http://www.globus.org>
3. Overview of a performance evaluation system for global computing scheduling algorithms / A. Takefusa, S. Matsuoka, K. Aida et al. // Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing (HPDC'99), 1999, Pp. 97–104.
4. The microgrid: Using emulation to predict application performance in diverse grid network environments / H. Xia, H. Dail, H. Casanova, A. Chien // In Proceedings of the Workshop on Challenges of Large Applications in Distributed Environments (CLADE'04), IEEE Press. 2004.
5. Optorsim - a grid simulator for studying dynamic data replication strategies / W. Bell, D. Cameron, L. Capozza et al.
6. Legrand A., Marchal L., Casanova H. Scheduling distributed applications: The simgrid simulation framework // Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid 2003 (CCGrid2003), 2003, Pp. 138–145.
7. Parallel workloads archive <http://www.cs.huji.ac.il/labs/parallel/workload/>
8. Soley R. Model driven architecture. object management group white paper. Draft 3.2, 2000, November.
9. The shared hierarchical academic research computing network. <http://www.sharcnet.ca>. 2007.
10. Ю.Г. Евтушенко, В.У. Малкова, А.А. Станевичюс "Распараллеливание процесса поиска глобального экстремума"// Автоматика и телемеханика. 2007. № 5.