

УДК 004.7

УПРАВЛЕНИЕ ПАРАЛЛЕЛЬНЫМИ ЗАДАНИЯМИ В GRID

Евдокимов А.А., Вороной С.М.

Донецкий национальный технический университет

1 Вычислительный GRID

GRID – это программно-аппаратная среда, которая построена из вычислительных установок, находящихся в различных административных доменах телекоммуникационной сети, и которая позволяет дистанционно использовать любое количество ресурсов этих установок – процессорных, оперативной и постоянной памяти, программ и данных.

2 Структура GRID диспетчера на основе агентного подхода

Анализ существующих средств планирования и балансирования загрузки в GRID системах свидетельствует о том, что эти методы и средства в настоящее время развиты недостаточно. Существующие средства планирования и управления ресурсами не удовлетворяют двум основным требованиям одновременно: масштабируемости и адаптивности [1].

Перспективным направлением повышения эффективности организации GRID является использование агентного подхода.

Агент – это сущность, которая может принимать информацию из внешней среды и реагировать на внешние возмущения. Агенты благодаря способности обмена информацией между собой могут в большой степени улучшить планирование задач в GRID. Каждый агент рассматривается как представитель GRID ресурса на метауровне, т.е. можно рассматривать агента как провайдера сервиса [2]. На основе агентного подхода можно построить структуру GRID, которая приведена на рис. 1.

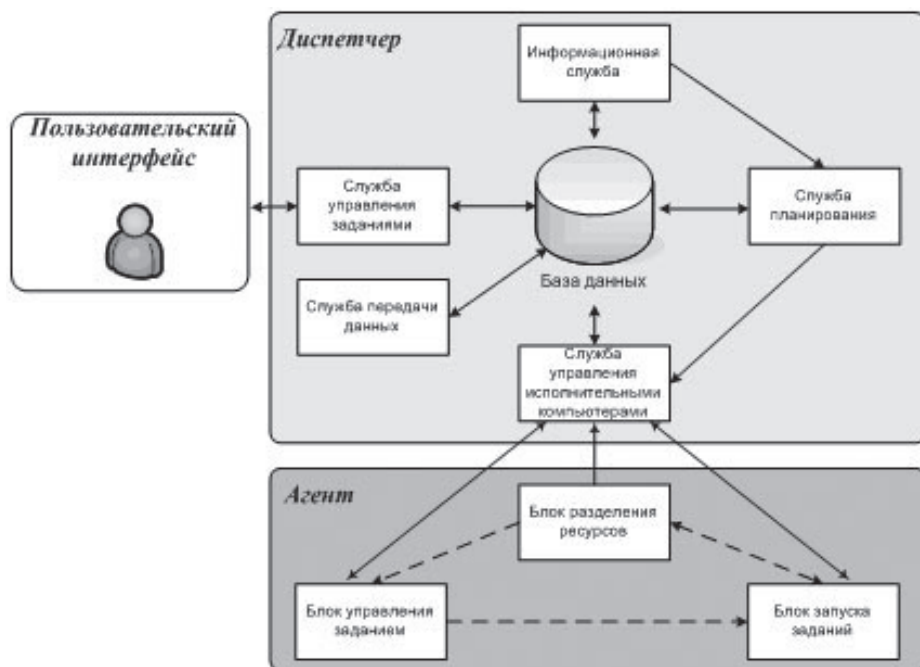


Рисунок 1 – Архитектура GRID сети на основе агентного подхода

GRID сеть состоит из агента диспетчеризации, который устанавливается на сервере GRID сети и агентов вычислительных ресурсов, которые устанавливаются на сервер вычислительного кластера или на вычислительные ресурсы узлов. Агент диспетчеризации принимает заказы от пользователей, бронирует вычислительные ресурсы, регистрирует вычислительные ресурсы (входящие в GRID сеть), ведет статистику выполнения всех заданий, а также следит за ходом выполнения заданий на вычислительном ресурсе.

Агенты вычислительных ресурсов устанавливаются на серверах кластеров и на вычислительных ресурсах. Эти агенты контролируют выполнение задания: они отвечают за запуск задания, следят за ходом выполнения задания, а также за

корректным разделением ресурсов, вычисляют время выполнения задания (осуществляется поиск более подходящего ресурса для задания).

3 Жизненный путь задания

Пользователь, желающий отправить свое задание на выполнение, посылает задание на сервер GRID–диспетчера, используя пользовательский интерфейс. В ответ пользователь получает идентификатор, с помощью которого впоследствии будет осуществлять управление своим заданием.

Задание, поступившее в GRID–диспетчер, попадает в глобальную очередь, упорядоченную по плате за задания, которую назначают пользователи. Пока задание находится в очереди, пользователь может изменить параметры своего задания, используя идентификатор, полученный при отправке задания. Варьируя платой, он может повлиять на то, как быстро оно будет обработано.

Над очередью заданий периодически осуществляется процедура планирования, которая распределяет задания по ресурсам. В случае успешного назначения, на какой–либо ресурс, инициируется процесс запуска задания: производится доставка необходимых файлов и совершается непосредственный старт задания в выбранном кластере или вычислительном ресурсе узла [3].

4 Схема работы GRID диспетчера

Управление заданиями в GRID диспетчере является циклическим процессом, который инициируется в ответ на происходящие события. События, приходящие в GRID диспетчер могут быть различными: поступил запрос от пользователя, обновилась информация о доступных ресурсах, задание успешно закончилось и т.д.

Все поступающие события буферизуются в очереди

сообщений. Начиная цикл диспетчеризации, GRID диспетчер выбирает очередное сообщение из очереди и, в зависимости от выбранного сообщения, предпринимает те или иные действия.

Для хранения очереди заданий, очереди поступающих сообщений и информации о ресурсах, в GRID-диспетчере используется реляционная база данных.

Буферизация входящих сообщений позволяет минимизировать время ответа отправителю сообщения, а также содержать базу данных GRID-диспетчера в непротиворечивом состоянии [1].

5 Специфика планирования в GRID

Планирование в GRID, хотя и имеет специфику, относится к широкому классу задач планирования, общая постановка которых характеризуется наличием множества ресурсов, множеством их потребителей и определенными условиями выделения ресурсов потребителям.

Известно, что способы планирования в зависимости от этих условий, целей и других факторов могут быть совершенно разными [4].

6 Гетерогенность ресурсов и их отбор для заданий

Задание GRID представляет собой обычный исполняемый файл (скрипт, программный код). Службами GRID оно доставляется на исполнительные ресурсы, а собственно выполнение происходит в среде операционной системы этих ресурсов. Как правило, программа, подготовленная на определенном компьютере, не требует каких-либо модификаций для использования в GRID. Однако любая программа рассчитана на определенную среду выполнения - ОС, архитектуру компьютера, объемы и характеристики его ресурсов.

При запуске через систему планирования требования задания оформляются в виде ресурсного запроса. Используемые

на практике формализмы записи ресурсного запроса, хотя и разнообразны, но ориентированы на то, чтобы рассматривать ресурсы унифицированным способом, предполагая, что набор классов является стандартизированным (с возможностью расширения).

7 Существующие методы планирования

В современных параллельных архитектурах и кластерных системах используются:

- методы разделения времени (Gang scheduling) – основаны на идее разделения процессоров между несколькими заданиями;
- методы разделения ресурсов (FCFS, First-Fit, Backfill) – основаны на идее разделения пространства ресурсов между заданиями.

Простейшим методом планирования является метод FCFS (First Come First Served), согласно которому задание, поступившее в очередь раньше других, имеет самый высокий приоритет и, следовательно, должно быть запущено первым. Если для его запуска ресурсов оказывается недостаточно, ожидается момент времени, когда накопится требуемый ему объём свободных ресурсов, и задание будет запущено.

Модификация «первый подходящий» (First-Fit) метода FCFS допускает нарушение порядка очереди, позволяя низкоприоритетным заданиям занимать ресурсы, оставшиеся незагруженными. Это повышает эффективность их использования, однако приводит к тому, что большая часть процессоров занимается мелкими заданиями – возникает фрагментация ресурсов.

Более аккуратный способ решения предлагает метод обратного заполнения Backfill. В отличие от FCFS он требует от пользователей оценку времени выполнения их заданий, что позволяет выделять ресурсы заданиям не непосредственно в момент освобождения, а заблаговременно. Для этого строится

план распределения ресурсов – расписание запусков заданий. При построении расписания ресурсы выделяются заданиям в порядке их приоритетов, причём задание может получить некоторые ресурсы только при условии, что они уже не отведены более приоритетным заданиям (в консервативном варианте) или самому приоритетному заданию (в агрессивном варианте) [3].

8 Новый метод управления заданиями в GRID

Новый метод планирования разработан для практически важной формы грида:

- ресурсы организованы в кластеры;
- ресурсы используются совместно с владельцами.

В разработанном подходе планировщик использует методы динамического программирования. Планировщик рассматривает очередь целиком, и пытается найти множество заданий, которые совместно максимально эффективно загрузили бы ресурсы.

Такой подход является решением проблемы фрагментации ресурсов, которая возникает, когда планировщики рассматривают задания из очереди по порядку, одно за другим.

Для анализа эффективности рассмотренных алгоритмов планирования была разработана экспериментальная среда, основанная на имитационном моделировании GRID.

Целью экспериментов являлось определение стандартных показателей эффективности планирования:

- степень загрузки ресурсов;
- время ожидания выполнения;
- время выполнения задания.

На основе показателей эффективности проведено сравнение алгоритмов диспетчеризации ресурсов.

Литература

- [1] Березовский П.С., Коваленко В.Н. Состав и

- функции системы диспетчеризации заданий в гриде с некластеризованными ресурсами, ИПМ им. М.В.Келдыша РАН, Москва, 2007. Электронный ресурс. Режим доступа: http://www.keldysh.ru/papers/2007/prep67/prep2007_67.html
- [2] Шелестов Ю.А., Скакун С.В., Куссуль О.М. Агентный подход к реализации модели поведения пользователей GRID-систем, Институт космических исследований НАНУ-НКАУ, Москва, 2004. Электронный ресурс. Режим доступа: <http://nbuv.gov.ua/portal/natural/Npdntu/2008/ikot/08saypgs.pdf>
- [3] Коваленко В.Н., Коваленко Е.И., Корягин Д.А., Семячкин Д.А. Управление параллельными заданиями в Гриде с неотчуждаемыми ресурсами, ИПМ им. М.В.Келдыша РАН, Москва, 2007.
- [4] Коваленко В.Н., Коваленко Е.И., Шорин О.Н. Разработка диспетчера заданий грид, основанного на опережающем планировании. Препринт № 133. 2005. Москва: ИПМ РАН, с. 1–28. Электронный ресурс. Режим доступа: http://http://www.gridclub.ru/library/publication.2006-01-10.2132656354/publ_file/