ГРИД-ДИСПЕТЧЕР: РЕАЛИЗАЦИЯ СЛУЖБЫ ДИСПЕТЧЕРИЗАЦИИ ЗАДАНИЙ В ГРИД*

В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин, Э.З. Любимский, Е.В. Хухлаев, О.Н. Шорин

Институт прикладной математики им. М.В.Келдыша РАН;
Россия, 125047, Москва, Миусская пл. 4; тел. (095)250-79-82,
kvn@keldysh.ru, kei@keldysh.ru, koryagin@keldysh.ru, ljubimsk@keldysh.ru,
huh@keldysh.ru, shorin@keldysh.ru

Введение

Одной из задач, которые появились в контексте развития Грид, является создание диспетчеров – посредников между потребителями и поставщиками ресурсов, осуществляющих прием пользовательских заданий, распределение и запуск их по доступным ресурсам.

В ИПМ им. М.В. Келдыша реализован программный комплекс Грид-диспетчер, являющийся развитием системы Метадиспетчер [1], функция которого – управление заданиями, в том числе и планирование. В Грид-диспетчере предложен новый способ диспетчеризации заданий и алгоритм планирования, основанный на следующих достаточно общих предположениях:

- Грид образован из кластеризованных неотчуждаемых ресурсов, то есть на машины кластеров поступает независимый от Грид-диспетчера поток *локальных* заданий;
- интенсивность поступления *глобальных* заданий Грид-диспетчера на ресурсы регулируется соглашениями между поставщиками ресурсов и потребителями, в качестве посредника выступает Грид-диспетчер;
- задания, поступившие Грид-диспетчеру, образуют приоритизированную очередь. Порядок распределения заданий осуществляется в соответствии с приоритетами заданий;
- условием предоставления ресурса является его цена, вычисляющаяся динамически в зависимости от того, какова приоритетность локального задания, которое может занять данный ресурс.

Жизненный путь задания

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

Задание, поступившее в Грид-диспетчер, попадает в глобальную очередь, упорядоченную по плате за задания, которую назначают пользователи. Пока задание находится в очереди, пользователь может изменить параметры своего задания,

*Работа выполнена при поддержке Российского фонда фундаментальных исследований (проекты 02-01-00282 и 04-07-90299).

используя идентификатор, полученный при посылке 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 распределения на какой-либо ресурс, инициируется процесс запуска задания: производится доставка необходимых файлов и совершается непосредственный старт задания в распланированном кластере.

После того, как задание закончится, пользователь может получить результаты его работы. На какой бы стадии выполнения задание не находилось, пользователь всегда может узнать его статус, а также отменить его.

Схема работы Грид-диспетчера

Управление заданиями в Грид-диспетчере является циклическим процессом, который инициируется в ответ на происходящие события. События, приходящие в Грид-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, Грид-диспетчер выбирает очередное сообщение из очереди и, в зависимости от выбранного сообщения, предпринимает те или иные действия.

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

Буферизация приходящих сообщений позволяет минимизировать время ответа отправителю сообщения, а также содержать базу данных Грид-диспетчера в непротиворечивом состоянии.

Особенности диспетчеризации в Грид

Задача диспетчеризации решалась многократно и для разных условий, так что имеет смысл сравнить условия управления заданиями в Грид с наиболее близким аналогом – кластерными системами. Диспетчеризация в кластерных системах происходит в виде непосредственной реакции на события, из которых основными являются освобождение ресурсов и появление новых заданий. Однако, реализовать точно такую же схему на уровне Грид невозможно по нескольким причинам:

- шаг диспетчеризации требует существенного времени: должно быть выполнено планирование и осуществлена доставка задания на ресурс. Даже не учитывая затрат на планирование, можно утверждать, что доставка заданий, а в Грид они предполагаются большими, с такой скоростью, чтобы диспетчер мог непосредственно реагировать на события, невозможна;
- число событий в Грид будет на порядки больше, чем в локальных вычислительных комплексах, просто из-за большего количества объектов – ресурсов и заданий.

На основании перечисленных выше причин, распределение заданий в Грид-диспетчере осуществляется с упреждением по отношению к моментам их запуска. Для реализации такого подхода был разработан механизм составления прогноза занятия/освобождения ресурсов в кластере, который опирается на возможности планировщика Maui [2].

Поток информации между кластером и Грид-диспетчером направлен снизу-вверх: происходящее изменение (начало, окончание задания) детектируется на уровне системы управления кластером, производится моделирование прогноза – расписания, которое передается Грид-диспетчеру. Расписание, пересылаемое Грид-диспетчеру, в данном случае рассматривается, как предложение кластера по предоставлению своих ресурсов Грид-диспетчеру. Это предложение действительно до поступления следующего расписания.

Структурная схема Грид-диспетчера

Грид-диспетчер состоит из трех основных компонент:

- интерфейса пользователя, который позволяет отправить задание в Грид-диспетчер, узнать статус уже отправленного задания, изменить параметры задания, отменить задание и получить результаты работы задания;
- кластерного Агента, который составляет прогноз занятия/освобождения ресурсов в кластере;
- а также серверной части Грид-диспетчера, которая осуществляет прием сообщений от Агентов и пользователей, распределение заданий по ресурсам и их запуск.

Коммуникации распределенных компонентов друг с другом реализованы с помощью аппарата Грид-служб [3], появившегося в инструментальной среде Globus Toolkit [4] версии 3.0. Грид-службы являются стандартизованным средством взаимодействия отдельных компонентов Грид.

Серверная часть Грид-диспетчера в свою очередь содержит в себе:

- ядро Грид-диспетчера, которое обрабатывает поступающие события;
- службу запуска заданий в кластер и управления запущенными заданиями – Job Control;
- Грид-службы, принимающие сообщения от Агентов и пользователей;
- базу данных, в которой хранится очередь заданий и информация о ресурсах;
- очередь сообщений.

Постановка задачи планирования

Основной задачей ядра Грид-диспетчера является осуществление планирования. Задача планирования заключается в составлении глобального расписания запуска заданий на основе множества локальных расписаний с разных кластеров и информации о параметрах предоставляемых ресурсов. Результат - глобальное расписание, содержит в себе информацию о запусках как локальных, так и глобальных заданий. Алгоритм планирования учитывает некоторые соглашения и ограничения:

- глобальные задания упорядочены по плате. Менее приоритетное задание не может стартовать раньше более приоритетного задания;
- между кластерами и Грид-диспетчером существует соглашение о предоставлении кластером своих ресурсов заданиям Грид-диспетчера. Это соглашение основано на сравнении платы заданий, глобальных и локальных, претендующих на ресурс;
- глобальному заданию соответствует ресурсный запрос, написанный пользователем на языке описания ресурсов RSL. При распределении

глобальных заданий по ресурсам должно учитываться соответствие ресурсов запросам заданий;

- распределение глобальных заданий должно производиться с учетом прав доступа владельца задания.

На данный момент реализован вариант алгоритма планирования, который основывается на предположении неделимости ресурсов, т.е. на одном ресурсе может находиться только одно задание.

Алгоритм планирования

Как было сказано ранее, схема работы Грид-диспетчера является циклической: во время очередного цикла все поступаемые сообщения буферизуются в очереди сообщений, после окончания текущего цикла из очереди выбирается первое сообщение и начинается новый цикл.

События, которые могут повлиять на составление глобального расписания, можно разделить на три группы:

- к первой группе относятся события, изменяющие множество глобальных заданий. В эту группу входят события поступления нового задания в очередь Грид-диспетчера и изменения параметров уже существующего задания;
- вторая группа состоит из событий, которые изменяют множество локальных расписаний;
- третья группа – из событий, которые изменяют параметры ресурсов. В частности, к этой группе относятся события изменения состава ресурсов в кластере или изменение соглашений предоставления кластером своих ресурсов глобальным заданиям Грид-диспетчера.

При возникновении любого из событий, способного привести к изменению глобального расписания, ядро Грид-диспетчера начинает планирование.

Первым шагом планирования является удаление из базы данных Грид-диспетчера локальных расписаний, которые к этому моменту устарели. Далее происходит построение глобального расписания, которое основано на сопоставлении спрогнозированных локальных заданий с глобальными заданиями, претендующими на ресурсы, занятые локальными. При этом порядок выбора глобальных заданий для сопоставления осуществляется с учетом их платы, а порядок выбора прогнозируемых локальных заданий происходит в соответствии с прогнозируемым для них временем старта.

Сопоставление глобального задания, претендующего на некоторый ресурс, и локального задания, запуск которого спрогнозирован на рассматриваемом ресурсе через некоторое время, заключается в проверке нескольких условий:

- Сначала сравниваются платы за глобальное и локальное задания. Поскольку каждый кластер имеет свою политику предоставления ресурсов, для сохранения автономности администратор каждого кластера составляет функцию, с помощью которой будет выполняться пересчет платы глобального задания в локальный приоритет. Используя функцию пересчета, планировщик сравнивает вычисленный приоритет глобального задания и приоритет локального. Если плата за глобальное задание оказывается выше, то осуществляется проверка следующих условий.
- На следующем шаге алгоритма проверяется: имеет ли пользователь глобального задания право запускать задания на рассматриваемом ресурсе.

Это соответствие устанавливается с использованием информации, хранящейся в базе данных Грид-диспетчера.

- На последнем шаге сравнения определяется: удовлетворяет ли рассматриваемый ресурс запросу глобального задания. Для этого используется информация из ресурсного запроса глобального задания и информация о параметрах рассматриваемого ресурса, которая хранится в информационной базе Грид-диспетчера.

Если все 3 условия удовлетворены, это означает, что рассматриваемый ресурс удовлетворяет ресурсному запросу глобального задания, и не нарушается соглашение о предоставлении локальных ресурсов глобальным заданиям, поскольку плата за глобальное задание оказалась выше платы прогнозируемого локального задания. В глобальное расписание заносится информация о том, что глобальное задание планируется запустить на рассмотренном ресурсе вместо спрогнозированного локального задания.

Как можно видеть, при сопоставлении двух заданий осуществляется проверка 3 условий. Порядок их проверки не влияет на результат сравнения, но в целях увеличения скорости алгоритма планирования сначала проверяются условия, требующие меньших затрат.

Запуск и управление заданиями

После составления глобального расписания, ядро Грид-диспетчера инициирует запуск глобальных заданий в соответствии с построенным расписанием. Для этого ядро дает команду службе Job Control на запуск заданий, указывая при этом какие глобальные задания надо запустить в тот или иной кластер. Служба Job Control инициирует процесс запуска заданий, используя при этом стандартные средства инструментальной среды Globus Toolkit [4]. После запуска заданий Job Control может по запросу пользователя получать информацию о статусе заданий, занося эту информацию в базу данных. Если пользователь захочет отменить какое-нибудь из уже запущенных заданий, то этот запрос попадет в службу Job Control, которая и прервет выполнение задания.

Сравнение Грид-диспетчера и GRB

После того, как описаны схема и алгоритм работы Грид-диспетчера, сравним Грид-диспетчер с известным брокером – GRB [5]. Проведем сначала структурное сравнение обоих систем. В GRB выделяются следующие компоненты:

- Resource Broker Master (RBM) – этот основной процесс, который принимает от пользователей все запросы;
- Resource Broker Agent (RBA) – это процесс, порождаемый Resource Broker Master для обработки определенного запроса от пользователя, который осуществляет функцию распределения задания по ресурсам;
- Job Registry (JR) – база данных, в которой хранится вся информация, требующаяся GRB;
- Job Submission Service (JSS) – служба запуска заданий в кластер и их мониторинга.

В Грид-диспетчере функции RBM и RBA выполняет Грид-служба, принимающая запросы от пользователей. Также как и у GRB Грид-диспетчер имеет базу данных. Служба Job Control в Грид-диспетчере эквивалентна службе JSS в GRB.

Отличие Грид-диспетчера от GRB состоит в наличии глобальной приоритизированной очереди заданий, что позволяет управлять очередностью распределения заданий по ресурсам, а также в наличии механизма прогнозирования занятия/освобождения локальных ресурсов. Этими функциональными отличиями и объясняется появление в составе Грид-диспетчера таких компонент, как Агенты и Грид-службы для работы серверной части с Агентами.

Пути дальнейшего развития

Основные пути для дальнейшего развития проекта мы видим в следующем:

- Использование механизма предварительного резервирования, что позволит Грид-диспетчеру гарантировать запуск заданий в соответствии с построенным глобальным расписанием;
- Введение параметров в алгоритмы планирования, создания прогноза и доставки обновлений. Варьирование этими параметрами позволит системе Грид-диспетчер адаптироваться к таким характеристикам, как пропускная способность сети, темп поступления событий. Эти механизмы повысят масштабируемость, то есть позволят обслуживать большее число кластеров;
- Автоматизация процесса регистрации обслуживаемых кластеров в базе данных Грид-диспетчера. На данный момент эту операцию приходится делать вручную.

Литература

[1] С.А. Богданов, В.Н. Коваленко, Е.В. Хухлаев, О.Н. Шорин, «Метадиспетчер: реализация средствами метакомпьютерной системы Globus.» Препринт ИПМ РАН, №30, Москва, 2001.

[2] http://www.supercluster.org/maui/

[3] Open Grid Service Infrastructure Primer
http://www.gridforum.org/Meetings/GGF11/Documents/OGSI_Primer_Final.pdf

[4] http://www.globus.org

[5] S. Cavalieri, S. Monforte, “Resource Broker Architecture and APIs”
http://server11.infn.it/workload-grid/docs/20010613-RBArch-2.pdf