

ГРИД-ИНФРАСТРУКТУРА ДЛЯ НАЦИОНАЛЬНОЙ НАНОТЕХНОЛОГИЧЕСКОЙ СЕТИ

А.П. Крюков, В.А. Ильин

Основная цель проекта построения грид-инфраструктуры для Национальной нанотехнологической сети (ГридННС) - это обеспечение доступа географически распределенных научных и инженерных коллективов, участников национальной нанотехнологической сети (ННС), к суперкомпьютерным ресурсам ННС. При этом должен быть обеспечен унифицированный интерфейс к гетерогенным вычислительным ресурсам с высоким уровнем надежности и безопасности.

ГридННС — это комплексное грид-решение, которое включает промежуточное ПО, грид-сервисы и службы, регламенты развертывания и работы и так далее. В частности в него входят следующие сервисы и службы:

- поддержка пользователей, виртуальных организаций, системных администраторов;
- мониторинг и учет ресурсов, заданий;
- регистрация сервисов;
- управление выполнением композитных заданий;
- грид-шлюзы доступа к ресурсам;
- пользовательские интерфейсы.

Таким образом ГридННС представляет собой полнофункциональный грид.

Учитывая специфику работы и администрирования суперкомпьютеров, важным требованием к промежуточному грид-ПО было минимальное вмешательство в работу суперкомпьютерных центров. В частности, не допускалась установка специального ПО на узлы суперкомпьютерных кластеров. В этом существенное отличие данного проекта, например, от грида WLCG[1].

Большинство сервисов и программ были разработаны участниками проекта. Ряд сервисов используют имеющиеся открытые разработки. Например грид-шлюз использует Globus Toolkit-4.2 [2] с модификациями, сделанными участниками разработки. Информационная система построена на основе WS-MDS [3] и использует Glue Scheme-1.3 с некоторыми модификациями.

Основным объектом, с которым работает пользователь в ГридННС является задание [4]. Оно представляет собой набор задач, которые образуют направленный ациклический граф и который определяет порядок выполнения задач. Такая структура задания позволяет представить достаточно широкий класс композитных заданий, практически покрывающих все потребности пользователей. В качестве языка описания заданий используется JSON.

В описании задания и задач, пользователь может указать дополнительные требования, которые необходимо удовлетворить, чтобы выполнение задач прошло успешно. Такая схема позволяет пользователю достаточно гибко описывать необходимые для выполнения задания ресурсы. Однако пользователь при этом не указывает конкретный ресурс. Конкретный ресурс будет выбран автоматически с учетом этих требований.

На практике структура графа задач в задании ГридННС может быть достаточно сложной. Уникальной возможностью ГридННС является то, что при ошибочном завершении одной из задач в задании, задачи на ветвях графа, которые не зависят от задачи завершившейся с ошибкой, продолжают обрабатываться. Это позволяет пользователю исправить ошибку и запустить на повторное выполнение только те части задания, которые не были выполнены. Пользователь может также задать значение кода возврата, который будет рассматриваться как ошибка, что предоставляет дополнительную гибкость в управлении заданиями.

Формат описания задания в ГридННС позволяет объявить некоторый набор задач группой. В этом случае все задачи, принадлежащие одной группе будут запущены на одном и том же ресурсе. Сам ресурс будет выбран автоматически в момент запуска первой задачи группы. Типичным примером использования этой возможности является компиляция задачи на одном шаге и выполнение ее на другом. Необходимость этого может быть связана с тем, что специфика установки не позволяет запустить задачу скомпилированную на другом суперкомпьютере. Выполнение же этих двух шагов в рамках одной задачи часто бывает невозможно. Например, рабочие узлы могут не иметь доступа к компилятору по лицензионным соображениям.

При проектировании ГридННС особое внимание было уделено учету специфики запуска параллельных задач. Каждая такая задача должна запускаться в нужной для нее среде - набора библиотек, параметров окружения и тому подобное. В частности, необходимо обеспечить резервирование нужного для выполнения задачи количества ресурсов и, в первую очередь, ядер.

Для резервирования необходимого количества ядер в задаче используется параметр count. В настоящий момент поддерживается наиболее простая схема отображения процессов параллельной задачи на зарезервированные ядра. А именно, одно ядро на один процесс при использовании максимального количества доступных ядер на рабочем узле суперкомпьютера. В дальнейшем предполагается реализовать более сложные

схемы распределения ядер рабочего узла.

Другим важным моментом запуска параллельных задач является учет окружения, в котором задача должна быть выполнена. В описании задачи пользователь указывает тег необходимого ему прикладного ПО. Однако, окружения на каждом конкретном ресурсе могут быть установлены и настроены по-разному. Поэтому, в момент запуска задания на конкретный ресурс сервис запуска заданий автоматически прописывает те конкретные окружения, которые соответствуют указанному тегу и определены на этом ресурсе. Таким образом, задача будет корректно обработана на любом ресурсе, который удовлетворяет требованиям задачи, а пользователь может не беспокоиться о деталях настройки суперкомпьютера.

Заметим, что скрытие деталей конкретных вычислительных установок (виртуализация ресурса) является одним из главных преимуществ применения грид-технологии.

Управлением выполнением заданиями осуществляет специальный грид-сервис — Pilot [4,5]. Основная его задача принять задание от пользователя, зарегистрировать его и обеспечить запуск задач на ресурсы, которые соответствуют требованиям пользователя. В процессе выполнения задач, сервис Pilot осуществляет контроль за выполнением задач и всего задания в целом. В частности, он обеспечивает правильный порядок запуска задач в соответствии с логикой задания.

Отметим, что этот и ряд других сервисов ГридННС впервые реализованы как RESTful-грид сервисы. Такое решение позволило создать удобный и простой в использовании API этих сервисов. Одной из особенностей сервиса Pilot является использование механизма уведомлений при контроле за ходом выполнения задач, что позволило существенно снизить нагрузку на сервис.

Хотя в вычислительном гриде вопрос управления данными не стоит так остро как в гридах ориентированных на обработку больших объемов данных, этот вопрос является важной частью ГридННС. В ГридННС данные на ресурсы передаются непосредственно перед запуском задач с любых GridFTP серверов, которые зарегистрированы в системе. Данный подход существенно отличается от того, что принято в проекте gLite [6], где данные задачи сначала поступают на сервер управления загрузкой WMS, аналогом которого в ГридННС является Pilot, а потом сервис WMS передает данные на ресурс, на котором будет запущена задача. Разделение потока управления (задания и задачи) и потока данных существенно упростило работу сервиса Pilot и позволило повысить устойчивость его к высоким нагрузкам.

Каждый суперкомпьютер работает под управлением системы пакетного запуска задач. Для наиболее популярных систем в GT-4 есть необходимое ПО, обеспечивающее стык между собственно гридом и системой пакетного запуска. К ним в первую очередь относится PBS/Toque. В рамках проекта ГридННС были дополнительно разработаны стыки для Slurm и российской системы Cleo [7]. При необходимости, этот список может быть достаточно легко расширен.

Ресурсы, подключенные к ГридННС обязаны публиковать информацию о имеющемся оборудовании, его параметрах, установленных операционных системах и прикладных программах и программных окружениях. Это обеспечивает подбор ресурсов в соответствии с требованиями задач и обеспечивать необходимое окружение в процессе выполнения задач на ресурсе.

Для управления функционированием инфраструктурой, а также для автоматического обнаружения ресурсов сервисами ГридННС, реализован специальный сервис регистрации всех компонент ГридННС. Доступ пользователей и/или сервисов разрешен только на зарегистрированные службы и сервисы. Этим обеспечивается дополнительный уровень безопасности и исключает возможность подстановки поддельных (fake) сервисов.

Обслуживание пользователей ГридННС осуществляется по виртуальным организациям (VO) в соответствии с правилами принятыми данной VO. Технология VO позволяет не только упростить процедуру получения доступа пользователя на ресурсы, но и организовать учет потребления ресурсов по VO.

Очень важно обеспечить удобный доступ пользователей к ресурсам. В ГридННС существует два способа запуска заданий пользователя. Первый из них — это интерфейс командной строки (ИКС). Второй — специализированный графический веб-интерфейс. Последний отличается простотой в использовании и возможностью расширения через механизм подключаемых модулей (plug-in).

Интерфейс командной строки особенно удобен если пользователь использует скрипты для автоматизации запуска заданий. ИКС представляет собой набор утилит, которые обеспечивают запуск задания, контроль его выполнения, получение результатов работы. В случае необходимости, пользователь может прекратить выполнение задания.

Большое внимание при разработке ГридННС было уделено организационным вопросам. Сюда входят вопросы регистрации пользователей, VO, сайтов. Были также разработаны регламенты, регулирующие вопросы взаимодействия различных служб и сервисов, а также документация по установке и эксплуатации специального ПО ГридННС. Со всем этим можно ознакомиться на сайте проекта <http://www.ngrid.ru>. Отдельно стоит отметить службу поддержки пользователей. Она включает списки рассылки, через который пользователи могут самостоятельно обсудить свои проблемы, система работы с билетами и специальные страницы ЧАВО на сайте проекта.

Для контроля работы ГридННС регулярно производятся запуски тестовых задач, а также запущен сервис Nagios, который позволяет системным администраторам анализировать состояние своего сервиса или

сайта более детально.

В настоящее время в ГридННС интегрировано в экспериментальном режиме 16 сайтов. Общее количество ядер ЦПУ составляет более десяти тысяч. Среди поддерживаемых ВО отметим ряд прикладных ВО таких как Abinit, nanopchem, nanoprace.

В заключение отметим, что дальнейшее улучшение функциональности ГридННС будет проводиться на основе обобщения практического опыта эксплуатации системы. Но уже сейчас видно, что одним из главных путей наращивания функциональности системы - это разработка проблемно-ориентированных интерфейсов для подготовки композитных заданий в конкретных прикладных областях с ориентацией на конкретные программные системы и комплексы предустановленные в ресурсных центрах.

ЛИТЕРАТУРА

1. WLCG, <http://lcg.web.cern.ch/lcg/>
2. Globus Toolkit 4.2; <http://globus.org>
3. GT 4.0 Release Notes: WS MDS Aggregator Framework. http://www.globus.org/toolkit/docs/4.0/info/aggregator/WS_MDS_Aggregator_Release_Notes.html
4. А.Демичев, В.Ильин, А.Крюков и Л.Шамардин. Реализация программного интерфейса грид-сервиса Pilot на основе архитектурного стиля REST. Вычисленные методы и программирование, с.62-65, т.11, 2010.
5. А.Демичев, А.Крюков и Л.Шамардин. Принципы построения грид с использованием Restful-веб-сервисов. Программные продукты и системы, №4, 2009, г.Тверь
6. gLite, <http://www.glite.org>
7. Cleo batch system. <http://parcon.parallel.ru/cleo-eng.html>