

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 03.02.2021 14:26:34

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Юго-Западный государственный университет

(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ

Первый проректор –
проректор по учебной работе

Е.А. Кудряшов

«15» февраля

2012г.



ПРИНЦИПЫ ОРГАНИЗАЦИИ ГРИД-СИСТЕМ НА ПЛАТФОРМЕ VOINC

Методические указания по выполнению лабораторной работы
для студентов технических специальностей

Курск 2012

УДК 621.3

Составитель Э.И. Ватутин

Рецензент

Кандидат технических наук, доцент *М.И. Труфанов*

Принципы организации грид-систем на платформе BOINC: методические указания по выполнению лабораторных работ по дисциплинам «Сети ЭВМ и средства телекоммуникаций», «Специальные процессоры, машины и системы», «Параллельное программирование» / Юго-Зап. гос. ун-т; сост. Э.И. Ватутин; Курск, 2012. 12 с.: ил. 1. Библиогр.: с. 12.

Методические рекомендации содержат описание архитектуры и основных принципов организации грид-систем на платформе BOINC. Приведено описание актуальных вычислительно сложных проектов из различных областей науки. Приведено описание настройки клиентской расчетной части, показаны основные этапы параллельной распределенной обработки заданий.

Предназначены для студентов технических специальностей.

Текст печатается в авторской редакции

Подписано в печать _____. Формат 60x84 1/16.

Усл. печ. л. _____. Уч. – изд. л. _____. Тираж 50 экз. Заказ _____. Бесплатно.

Юго-Западный государственный университет.

305040, Курск, ул. 50 лет Октября, 94.

Содержание

Введение	4
1. Принципы организации грид-инфраструктуры	6
2. Основные направления исследований	9
3. Порядок выполнения работы	11
4. Контрольные вопросы	11
Библиографический список	12

Введение

В связи с успешным решением проблемы «последней мили» в течение последнего десятилетия Интернет получает все большую распространенность, что открывает широкие перспективы и возможности для развития ряда направлений науки и техники, включая концепцию «умного дома», социальные сети, потоковую трансляцию видео в реальном времени, удаленный доступ к корпоративным и домашним сетям с использованием технологии VPN и т.д. Одним из бурно развивающихся направлений в рамках данной тенденции являются грид-системы, применяемые для параллельной распределенной обработки информации.

Существует целый ряд вычислительных средств, ориентированных на решение вычислительно сложных задач различных классов.

Суперкомпьютер – специализированная вычислительная машина, значительно (в тысячи и более раз) превосходящая по производительности настольные ЭВМ.

Вычислительный кластер (англ. Computer cluster) – группа вычислительных узлов (обычно компьютеров общего назначения), объединенных быстрыми низко латентными каналами связи и представляющих с точки зрения пользователя единый аппаратный ресурс. Вычислительные кластеры по стоимости и производительности занимают промежуточное место между настольными ЭВМ и суперкомпьютерами.

Распределенные вычисления (англ. Distributed computing) – способ решения вычислительно сложных задач с использованием множества компьютеров, объединенных сетевой подсистемой и в совокупности участвующих в параллельной обработке информации.

Грид-система (от англ. Grid – сеть, решетка) – одна из разновидностей распределенных вычислений, при которой виртуальный суперкомпьютер представлен совокупностью географически удаленных гетерогенных вычислительных узлов (компьютеров), работающих над совместным решением поставленной задачи, допускающей разбиение на большое количество (десятки – сотни тысяч и более) параллельно выполняемых подзадач. Для распределенных вычислительных

систем характерно наличие хорошей масштабируемости – близкого к линейному росту производительности с ростом числа вычислительных узлов. Целью данной работы является ознакомление с принципами организации и участия в современных проектах грид-вычислений на платформе BOINC.

Приведенные виды параллельных вычислительных систем применяются для решения различных вычислительно сложных задач, допускающих возможность распараллеливания и относящихся к разным классам:

- сильносвязанные задачи – задачи, подразумевающие интенсивный обмен данными между подзадачами и наличие большого числа вычислительных зависимостей по данным или управлению;
- слабосвязанные задачи – задачи, при параллельной обработке которых между подзадачами отсутствует необходимость в обмене информацией либо подобные обмены сведены к минимуму.

Классическими примерами сильносвязанных задач является решение систем линейных уравнений большой размерности (необходимое при решении ряда практических задач и входящее в состав теста LINPACK, используемого для тестирования производительности суперкомпьютеров и составления рейтинга TOP500) или решение дифференциальных уравнений сеточными методами. Сильносвязанные задачи допускают эффективную параллельную реализацию только на многопроцессорных/многоядерных вычислительных архитектурах, вычислительных кластерах или суперкомпьютерах.

Примерами слабосвязанных задач являются задачи вычислительной математики и теории чисел (проверка математических гипотез для различных наборов входных значений, поиск простых чисел различного вида и др.), астрофизические задачи (анализ данных оптических, гравитационных и радиотелескопов, моделирование взрывов сверхновых звезд или эволюции планетных систем при различных начальных условиях и др.), задачи вычислительной физики (например, моделирование движения заряженных частиц в неоднородном магнитном поле), а также ряд задач вычислительной химии и биологии (фолдинг, докинг, скрининг, секвенирование). Слабосвязанные задачи допускают эффективную параллельную обработку на большинстве

вычислительных архитектур (за исключением некоторых случаев узкоспециализированных архитектур), но наиболее эффективным способом их решения с позиции стоимости, производительности и масштабируемости является их реализация в составе грид-систем.

Наиболее крупной грид-системой является платформа BOINC (англ. Berkeley Open Infrastructure for Network Computing – открытая сетевая инфраструктура университета в Беркли для распределенных вычислений), разработанная в 2002–2004 гг. в Калифорнийском университете в Беркли в рамках проекта SETI@Home. Клиентское и серверное программное обеспечение платформы представляет собой свободно распространяемое программное обеспечение с открытыми исходными кодами. По состоянию на 2011 год совокупная производительность всех проектов на данной платформе составляет 5,5 петафлопс, что сопоставимо с производительностью самых производительных суперкомпьютеров; платформа объединяет около 60 проектов распределенных вычислений, в совокупности объединяя более 6 млн. компьютеров из 273 стран. Отличительной особенностью функционирования платформы является наличие в ее составе не только крупных организаций с мощными вычислительными кластерами, но и большого числа добровольных участников (более 2 млн.), выполняющих вычисления на своих компьютерах в фоновом режиме с приоритетом простоя и не мешающих выполнению основных задач компьютеров, вычислительные ресурсы большинства из которых по статистике загружены менее чем на 5%. Основной особенностью проектов распределенных вычислений, функционирующих на платформе BOINC, является некоммерческий научный характер выполняемых в их рамках расчетов.

Крупные страны имеют различные собственные грид-системы (SZTAKI в Венгрии, EGEE среди стран Евросоюза, Almere GRID в Нидерландах, Extermadura и IBERCIVIS в Испании, Westminster в Великобритании и др.), в настоящее время имеющие интеграцию с BOINC.

1. Принципы организации грид-инфраструктуры

Параллельная обработка информации, ориентированная на грид, начинается с разбиения вычислительно сложной задачи на

множество более мелких подзадач. Данная задача выполняется организаторами проекта прозрачно для его участников. Множество подзадач (англ. workunits) добавляется в соответствующие таблицы базы данных сервера проекта. Для каждого задания серверным программным обеспечением формируется несколько идентичных копий исходных расчетных данных, число которых называется кворумом. Подобная избыточность уменьшает интегральную полезную производительность проекта, но в то же время позволяет защититься от некорректных результатов вычислений, которые могут быть результатом умышленных действий некоторой категории пользователей, следствием ошибок в аппаратной части компьютера (например, из-за перегрева или завышенного значения тактовой частоты) либо являться следствием некорректного поведения метода при реализации вычислений на различных аппаратно-программных платформах (с использованием векторных расширений системы команд процессора или без, с использованием вычислительных возможностей видеокарт различной архитектуры и т.д.). Значение кворума выбирается разработчиками индивидуально для каждого из проектов как компромисс между производительностью и надежностью.

В ответ на обращение клиентского компьютера к серверу ему выдается набор расчетных заданий в соответствии с текущими настройками и вычислительными возможностями процессора или видеокарты клиента. Исходные данные обычно передаются в формате XML, но также возможен обмен бинарными файлами. На клиенте осуществляется расчет полученного задания, результаты которого отправляются на сервер.

При получении всех копий результатов расчета, число которых совпадает с кворумом, для данного задания производится т.н. валидация – сравнение полученных результирующих данных (в простейшем случае путем сопоставления двоичного содержимого полученных файлов). В случае совпадения результатов задание считается успешно выполненным, а считавшим его пользователям производится начисление кредита, который приблизительно характеризует объем проделанных вычислений. Если валидация не завершается успешно, обычно производится выдача ряда дополнительных заданий с целью выявления среди множества полученных результатов корректных и некорректных.

Для каждого из расчетных заданий разработчиками проекта производится настройка допустимого времени (англ. deadline) на его вычисление в зависимости от специфики проекта и среднего объема времени, необходимого для обсчета задания. Если в течение отведенного срока результат вычислений не получен (например, из-за отказа клиентской машины), задание считается просроченным и выдается повторно другому пользователю по его запросу.

Схематично процесс обмена данными в ходе вычислений показан на рис. 1.

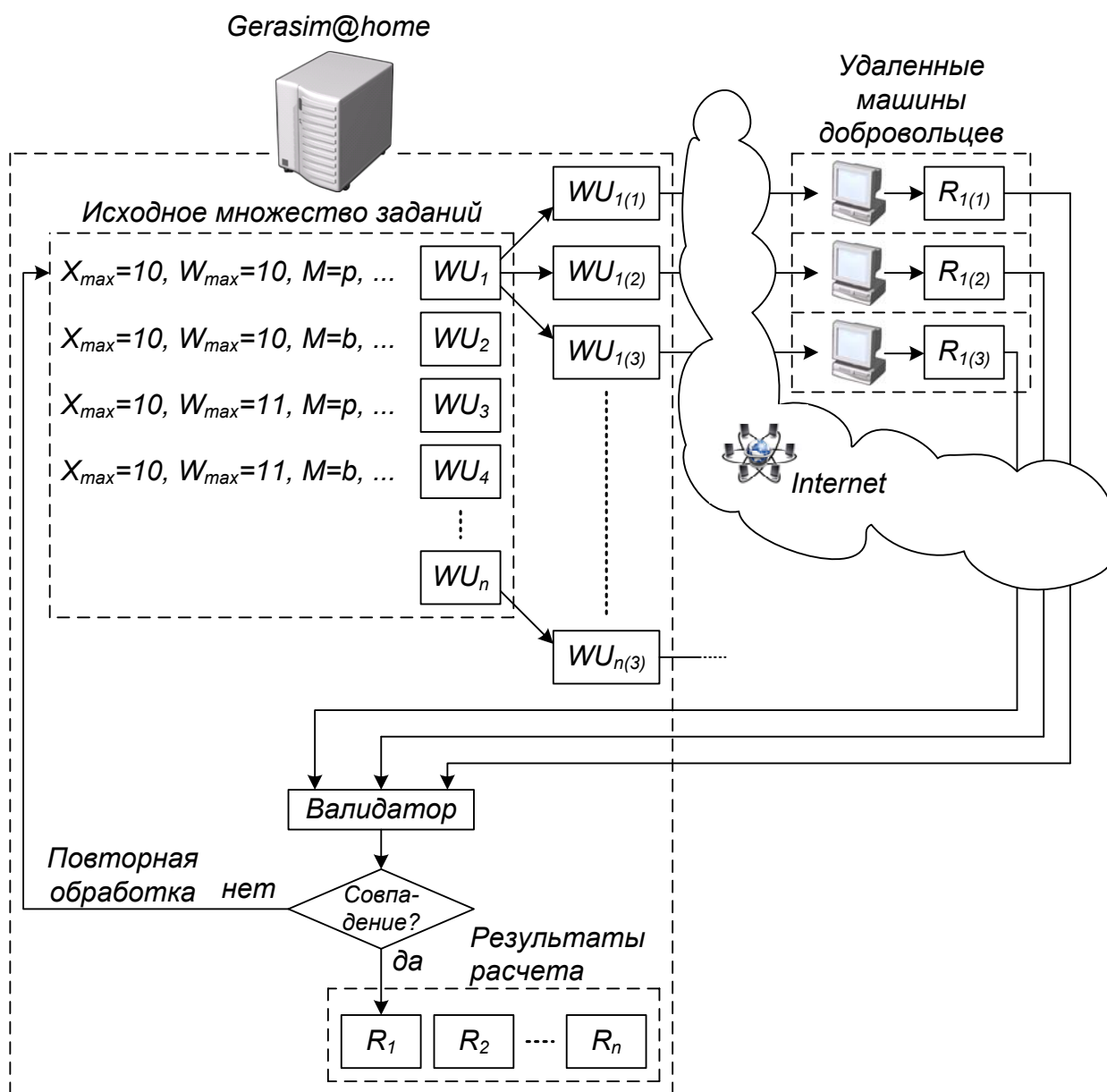


Рис. 1. Процесс расчета для проекта Gerasim@home

Расчетные модули различных проектов характеризуются существенно различным временем расчета, объемами входящего и

исходящего трафика, затратами оперативной памяти и места на жестком диске, кворумом и разрешенным временем вычисления.

2. Основные направления исследований

Различные области науки обладают рядом вычислительно сложных задач, решаемых в рамках проектов распределенных вычислений. К ним относятся:

- естественные науки (физика, астрономия, вычислительная техника):
 - поиск внеземного разума в ходе анализа узкополосных радиосигналов с радиотелескопов – проект SETI@Home;
 - поиск пульсаров и гравитационных волн – проект Einstein@home;
 - оптимизация траектории движения пучков заряженных частиц в магнитном поле Большого адронного коллайдера – проект LHC@Home;
 - проектирование ускорительных стадий мюонного коллайдера Neutrino Factory – подпроект Muon проекта uoуo@home;
 - создание высокоточной трехмерной модели движения звезд в галактике Млечный путь – проект MilkyWay@Home;
 - исследования в области наноманетизма – проекты Spinhenge@home, Magnetism@home;
 - исследование поведения плазмы при ее магнитном удержании в термоядерном реакторе – проект EDGeS@Home;
 - проектирование мембран фильтров для очистки воды – подпроект Computing for Clean Water проекта World Community Grid;
 - моделирование эволюции Вселенной от момента Большого взрыва, поиск наиболее адекватной модели по результатам наблюдения реликтового излучения – проект Cosmology@home;
 - моделирование климата, поиск адекватных климатических моделей – проект Climate Prediction Dot Net;
 - слежение за орбитами малых тел в окрестности Земли – проект Orbit@home;

- распределенный мониторинг землетрясений и радиационной обстановки – проекты Quake Catcher Network, Radioactive@Home;
- биология, химия, медицина:
 - моделирование пространственной структуры белков (фолдинг) – проекты Rosetta@home, Folding@home, World Community Grid, DrugDiscovery@Home, Proteins@home, ROEM@HOME, CAS@Home, evo@home, Correlizer;
 - моделирование протекания химических реакций, квантовая химия – проекты Hydrogen@home, Eon, QMC;
 - исследование и систематизация ДНК и РНК – проекты RNA World, DNA@Home;
 - исследования генома на предмет поисков корреляций с различными заболеваниями – проекты World Community Grid, Superlink@Technion;
 - систематизация свойств и структуры протеинов – проект SIMAP;
 - поиск перспективных материалов для создания солнечных батарей с высоким КПД – подпроект Clean Energy Project проекта World Community Grid;
- математика и криптография:
 - поиск простых чисел специального вида – проекты PrimeGrid, GIMPS, Mersenne@home;
 - факторизация чисел различного вида – проекты NFS@Home, Primaboinca;
 - проверка математических гипотез (гипотеза Коллаца, гипотеза Римана, ABC-тройки, проблемы Ризеля и Серпинского) – проекты ABC@Home, Collatz Conjecture, PrimeGrid;
 - проверка криптостойкости криптосистем – проекты SAT@Home, Moo Wrapper;
 - расшифровка немецких сообщений времен Второй мировой войны – проект Enigma@home;
 - поиск оптимальных линеек Голомба – подпроект ORG проекта yoYo@home;
 - решение задач многокритериальной дискретной и непрерывной оптимизации – проекты OPTIMA@Home и Gerasim@Home;

- исследования в области полей Галуа – проект NumberFields@home.

3. Порядок выполнения работы

1. Установить на компьютер программное обеспечение BOINC Manager.
2. Подключиться к одному или нескольким проектам распределенных вычислений исходя из личных предпочтений и рекомендаций преподавателя.
3. Определить пропускную способность и латентность сетевого канала выхода в Интернет.
4. Определить средний объем входящего трафика для расчетных заданий выбранного проекта.
5. В результате наблюдения за процессом вычисления нескольких заданий определить минимальное и максимальное время их вычисления, затраты оперативной памяти и жесткого диска.
6. Определить время, в течение которого необходимо отправить результаты расчета.
7. Определить средний объем исходящего трафика для расчетных заданий выбранного проекта.
8. Убедиться в успешности валидации расчетных заданий, определить средний объем очков, выдаваемых 1 ядру процессора за 1 час работы.
9. Определить интегральный объем вычислений, выполненный пользователем в рамках выбранного проекта.

4. Контрольные вопросы

1. В чем заключается проблема «последней мили»?
2. В чем отличия между настольными ЭВМ, вычислительными кластерами, суперкомпьютерами и грид-системами?
3. Чем отличаются сильно- и слабосвязанные задачи?
4. Какие современные актуальные вычислительно сложные задачи допускают эффективную параллельную реализацию с использованием грид-систем?
5. Что подразумевается под определением кворума при организации параллельных вычислений на платформе BOINC?

6. С какой целью выполняется валидация полученных результатов?

Библиографический список

1. <http://ru.wikipedia.org>
2. <http://boincstats.com>
3. <http://boinc.berkeley.edu>
4. <http://boinc.ru>