

На правах рукописи

Мьё Тун Тун

**РАЗРАБОТКА РАСПРЕДЕЛЕННЫХ ГЕТЕРОГЕННЫХ  
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ЗАПУСК ПРИЛОЖЕНИЙ В  
РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ**

Специальность: 05.13.18 – Математическое моделирование,  
численные методы и комплексы программ  
(технические науки)

**А В Т О Р Е Ф Е Р А Т**  
диссертации на соискание ученой степени  
кандидата технических наук

Санкт-Петербург - 2011

Работа выполнена в Санкт-Петербургском государственном морском техническом университете «СПбГМТУ»

**Научный руководитель –**

доктор физико-математических наук, профессор Богданов Александр Владимирович

**Официальные оппоненты:**

доктор физико-математических наук, профессор Горбачев Юрий Евгеньевич  
кандидат технических наук, профессор Руковчук Владимир Павлович

**Ведущее предприятие –** Факультет прикладной математики и процессов управления Санкт-Петербургского Государственного университета.

Защита диссертации состоится «14» сентября 2011г. в 15 часов на заседании совета по защите докторских и кандидатских диссертаций Д212.238.01 Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) по адресу: 197376, Санкт-Петербург, ул. Проф. Попова, 5.

С диссертацией можно ознакомиться в библиотеке Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».

Автореферат разослан «\_\_\_\_\_» \_\_\_\_\_ 2011 г.

Ученый секретарь совета  
по защите докторских и  
кандидатских диссертаций, к.т.н.

Н.Л. Щеголева

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

### **Актуальность темы.**

Рациональной альтернативой созданию собственного суперкомпьютерного центра является аренда вычислительных и программных ресурсов в режиме удаленного доступа у центров коллективного пользования, функционирующих при крупных университетах, академических институтах и других организациях. Однако при этом возникает целый комплекс проблем, связанных с обеспечением безопасности вычислительных систем и данных. Указанный комплекс проблем можно решить посредством применения концепции грид вычислений (Grid Computing) и родственной ей концепции облачных вычислений (Cloud Computing) в соответствии с которыми, пользователю предоставляется конечный проблемно-ориентированный сервис, обеспечивающий решение задач на базе ресурсов распределенных вычислительных систем. В соответствии с этим актуальной является задача разработки сервисно ориентированных методов использования систем анализа в распределенных вычислительных средах. В настоящее время эффективные комплексные решения в этой области отсутствуют.

В диссертации рассматриваются эксперименты с различными способами запуска приложений в распределенных системах, варьируя возможности как параллельных библиотек, так и операционной среды распределенной системы. Предложен набор математических моделей, на основе которых строится синтетический тест, позволяющий определить эффективность вычислительных сред.

**Целью работы** является создание набора программных средств для объединения гетерогенных вычислительных ресурсов в единый вычислительный комплекс и подбор (и программная реализация) комплекса математических моделей для тестирования и оптимизации получающихся распределенных вычислительных комплексов.

Для достижения поставленной цели в работе решены следующие **задачи**:

1. Тестирование Grid программного обеспечения в новых платформах;
2. Исследование алгоритмов виртуализации задачи и управления распределенными ресурсами;
3. Создание комплекса программ реализации Grid системы для запуска комплексных задач.
4. Разработка математических моделей, на основе которых строится синтетический тест, позволяющий определить эффективность вычислительных сред.

**Методы исследований** базируются на технологиях проектирования

программного обеспечения, теории надежности информационных систем, теории случайных процессов и потоков.

**Научной новизной** обладают следующие результаты, полученные автором в процессе выполнения работы:

1. Новый подход к системным интеграции гетерогенного вычислительного комплекса.
2. Новый метод интеграции разноуровневых приложений на базе промежуточного программного обеспечения, предоставляемого платформой SGE.
3. Новые модели и алгоритмы интегрированной работы приложений виртуального полигона в условиях распределенной вычислительной среды.

**Достоверность научных результатов и выводов** подтверждена результатами тестирования алгоритмов и программного обеспечения, а также практическим использованием разработанных алгоритмических и программных методов и средств.

**Основные научные результаты.**

1. Осуществлена системная интеграция гетерогенного вычислительного комплекса, состоящего из процессоров разных архитектур.
2. Предложен набор программных продуктов, реализующих распределенный гетерогенный вычислительный комплекс.
3. Реализован набор вычислительных моделей, с помощью которых проведено тестирование созданного комплекса.

**Научные положения, выносимые на защиту:**

1. Подход к системной интеграции распределенных гетерогенных вычислительных структур.
2. Программные продукты для создания операционного окружения в распределенной гетерогенной среде.
3. Адаптированные модели grid NAS Benchmark для тестирования полученных гетерогенных сред.

**Практическая значимость составляют:**

1. Набор программных продуктов, осуществляющих системную интеграцию в гетерогенной распределенной вычислительной среде.
2. Набор моделей, осуществляющих тестирование и оптимизацию в гетерогенной распределенной вычислительной среде.

**Внедрение результатов работы.** Результаты диссертационной работы внедрены для интеграции ресурсов в вычислительном комплексе ПМ-ПУ СПбГУ и использованы в учебном процессе кафедр ИВТ СПбГМУ и ВТ СПбГЭТУ «ЛЭТИ».

**Апробация работы.** Основные результаты работы докладывались и обсуждались на национальных и международных научно-технических

конференциях:

- Национальная конференция МОРИНТЕХ 2009, Санкт-Петербург, 2009 г.
- Международная конференция «Computer Science & Information Technologies», Yerevan, Armenia, 2009 г.
- Научно-методическая конференция «Телематика-2009», Санкт-Петербург, 2009 г.
- 4-ая Международная конференция «Distributed Computing and Grid-Technologies in Science and Education». Dubna, 2010 г.
- Известия СПбГЭТУ «ЛЭТИ». Санкт-Петербург, 2011 г.

### **Публикации**

Основные теоретические и практические результаты диссертации опубликованы в 6 статьях и докладах, из них по теме диссертации 6, среди которых 1 публикация в ведущем рецензируемом издании, рекомендованном в действующем перечне ВАК. Доклады доложены и получили одобрение на 5 международных, всероссийских и межвузовских научно-практических конференциях перечисленных в конце автореферата.

**Структура и объем диссертации.** Диссертационная работа состоит из введения, шести глав с выводами, заключения, одного приложения и списка литературы, включающего 125 наименований. Основная часть работы изложена на 127 страницах машинописного текста. Работа содержит 54 рисунка.

## **СОДЕРЖАНИЕ РАБОТЫ**

**Во введении** обоснована актуальность темы диссертации.

**В первой главе** рассмотрены технологии создания распределенных вычислительных сред и выполняется анализ их применимости к комплексным задачам в качестве базовой технологии.

В первой части дается обзор современных технологий распределенных вычислений. До середины 70-х годов прошлого века по причине высокой стоимости телекоммуникационного оборудования и относительно слабой мощности вычислительных систем доминировала централизованная модель вычислений. В конце 70-х годов появление систем разделения времени и удаленных терминалов, явилось предпосылкой возникновения клиент-серверной архитектуры, обеспечивающей предоставление ресурсов мейнфреймов конечным пользователям посредством удаленного соединения.

Разработанная в конце 1990-х концепция грид ориентирована на стандартизованное совместное использование автономных географически

распределенных компьютерных ресурсов в зависимости от их доступности, производительности, цены и иных характеристик, важных для конечного пользователя. Грид-системы ориентированы на формирование виртуального пространства для прозрачного совместного использования распределенных ресурсов в рамках виртуальных организаций. Виртуальная организация – это ряд людей и/или организаций, объединенных общими правилами коллективного доступа к определенным вычислительным ресурсам.

В 2000-2005 гг. произошло смещение тренда разработки распределенных приложений на сервисно ориентированную парадигму. Сервисами называются открытые, самоопределяющиеся программные компоненты, обеспечивающие прозрачную сетевую адресацию и поддерживающие быстрое построение распределенных приложений. В последствие сервисно ориентированный подход был принят разработчиками грид-систем и реализован в виде архитектуры OGSA (Open Grid Service Architecture).

В начале 2000 г., получила развитие концепция P2P-сетей (от англ. peer-to-peer), обеспечивающая формирование сетей на базе принципов децентрализации. В результате анализа выявлено, что одноранговые сети не ориентированы на безопасное, стандартизованное предоставление вычислительных ресурсов крупных организаций и сообществ.

В 2008 г. сформировалась новая концепция предоставления вычислительных ресурсов, названная «облачными вычислениями». Облаком называется пул виртуальных ресурсов (таких как аппаратное обеспечение, платформы разработки или сервисы), для которых обеспечены легкий доступ и простота использования.

Облачные вычисления и грид вычисления - масштабируемые. Масштабируемость достигается через балансировку нагрузки прикладных потоков, отдельный прогон на множестве операционных систем и связи через Веб-службы. Центральный процессор и сети, а так же полоса пропускания локализованы и не локализованы по требованию. Объем памяти системы изменяется в зависимости от числа пользователей и количества данных, переданных на данном цикле.

Оба типа вычисления порождают мульти задачу, что означает, что многие клиенты могут выполнить различные задачи, получая доступ к единственному или многократным копиям прикладного программного обеспечения. Разделение ресурсов приводит к сокращению затрат на инфраструктуру и увеличение способности к пиковой нагрузке.

**Во второй главе** производится анализ вычислительных Grid.

Необходимо объединить в единую систему различные по технической реализации и различные по типам компьютерные ресурсы (вычислительные ресурсы, ресурсы хранения и передачи информации) и донести совокупный ресурс до потребителя.

Потребитель должен получать услуги от системы в целом, ему не важно знать, где и какая установка хранит или обрабатывает его информацию, какого типа данная установка, какие линии передачи информации при этом задействованы и т. п. Также как и в электрических сетях, высокопроизводительная вычислительная GRID-система должна обеспечивать эффективное использование всей совокупности ресурсов (вычислительных, ресурсов хранения информации, ресурсов исходных данных — то есть, источников данных), нейтрализацию последствий аварий на линиях передачи, в устройствах хранения или обработки информации. Тем самым, создание подобных систем должно кардинально улучшить эффективность использования совокупных компьютерных ресурсов.

Архитектура Grid задает требования для основных классов компонент технологии (протоколов, сервисов, API, SDK), однако она не предоставляет строгий набор спецификаций, оставляя возможность их свободного развития в рамках представленной концепции. Архитектура Grid протоколов разделена на уровни такие как базовый уровень, связывающий уровень, ресурсный уровень, коллективный уровень и прикладной уровень. Компоненты на каждом уровне могут использовать возможности и компоненты любого из нижерасположенных уровней.

В терминологии Grid совокупность людей и организаций, решающих совместно ту или иную общую задачу и предоставляющих друг другу свои ресурсы, называется виртуальной организацией (ВО). Например, виртуальной организацией может быть совокупность всех людей, участвующих в каком-либо научном сотрудничестве. Виртуальные организации могут различаться по составу, масштабу, времени существования, роду деятельности, целям, отношениям между участниками (доверительные, не доверительные) и т.д.

Причины внедрения Grid, как и WEB технологий, имеют те же тенденции в смысле масштаба и развертывания. Следует, однако, понимать большую разницу между Web и Grid: несмотря на крупный масштаб интернета количество хост-машин, вовлеченных в типичную транзакцию на Web, все еще незначительно и намного меньше, чем предусмотрено для многих приложений Grid.

Конвергенция гридов и Web-служб является фундаментальным шагом навстречу реализации, общепринятой социумом, ориентированной на

службы инфраструктуры, которая позволяет людям создавать, предоставлять, получать доступ и использовать массу интеллектуальных услуг где угодно и когда угодно, надёжным безопасным способом и с минимальными затратами.

Конвергенция средств виртуализации, технологий организации распределенных вычислений и сервис-ориентированных архитектур формирует критически важную точку перегиба в развитии корпоративной ИТ-архитектуры. Сочетание возможностей данных технологий создает условия для решения наиболее важных задач, стоящих сегодня перед предприятиями. Внедрение решений, основанных на стандартах и использующих технологии Grid и SOA, позволяет ИТ-организациям улучшить качество обслуживания пользователей.

Сервис-ориентированная архитектура (SOA) - это такая архитектура приложения, в которой компоненты или «сервисы», имея согласованные общие интерфейсы, используют единые правила (контракты) для определения того, как вызывать сервисы и как они будут взаимодействовать друг с другом. При высокой степени интеграции grid, SOA и технологий виртуализации открытые стандарты позволяют ИТ-сервисам и бизнес-сервисам выходить за границы предприятия. Internet-провайдеры и поставщики услуг связи играют весьма существенную роль в деле размывания границ между предприятиями и формирования глобальных экосистем.

Появление концепции SOA способствует разворачиванию возможностей грид-ресурсов через стандартные интерфейсы, определённые как составная часть расширений соответствующих служб. Это дает возможность интеграции ресурсов с использованием интерфейсов, специфицированных в открытых стандартах. Кроме того, операции на каждом уровне грид-архитектуры могут быть разработаны таким образом, чтобы обеспечить достаточно лёгкую интеграцию всех уровней архитектуры.

SOA является областью, в которой грид-компьютинг начинает играть главную роль. Цена адаптации грида может сильно зависеть от требований к интеграции среды, например, функционирования в пределах предприятия, или с включением внешних партнёров. Значительным фактором является также сложность виртуализации, зависящая от сложности грид-приложений.

В настоящее время сосуществует множество различных grid-технологий, которые стимулируют креативность научно-исследовательского сообщества. Однако, возможно, что в конечном счёте будет один Grid, базирующийся на согласованных интерфейсах и протоколах, точно также, как это имеет место в Web. Внутри этой среды возможно сосуществование



виртуальных организаций, которые будут безопасно развиваться и взаимодействовать друг с другом. Такой подход позволит избежать распространения несовместимых гридов, которые препятствуют широкому благожелательному отношению к grid-технологиям.

**В третьей главе** рассматриваются алгоритмы и технологии программного комплекса. В последнее время при решении ряда научных и промышленных задач параллельные численные расчеты производятся с использованием GRID-систем. При этом обеспечение качества обслуживания сети является ключевым моментом. Реализация этого механизма заложена в основу сетей нового поколения, которые позволят создавать быстрые и надежные GRID-системы. В работе описываются алгоритмы и технологии распределенных гетерогенных вычислительных систем.

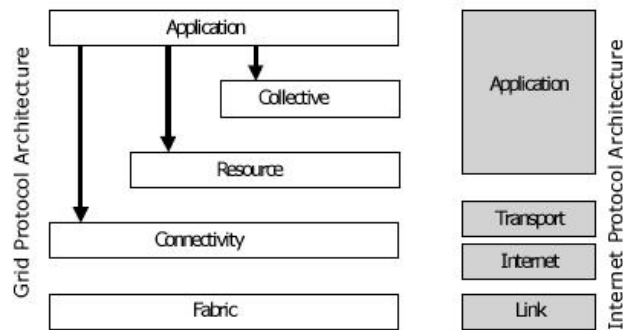


Рис.1. Уровни архитектуры Грид протоколов и их соответствие уровням архитектуры протоколов Internet

В последние годы стало ясно, что есть значительное перекрытие между целями вычислительного GRID и преимуществ сред, основанных на SOA и веб-сервисах. Быстрый прогресс в технологии веб-сервисов и разработке соответствующих стандартов обеспечили эволюционный путь от жесткой и узко-направленной архитектуры GRID - систем первого поколения к стандартизированным, сервис-ориентированным Гридам, гарантирующим стабильно-высокое качество обслуживания пользователей (Грид промышленного уровня).

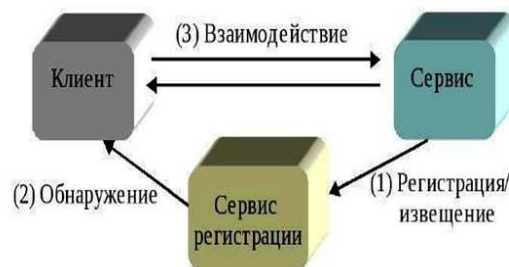


Рис.2. Простой цикл взаимодействия сервисов  
Появление стандарта OGSA стала ключевым моментом в развитии

программного обеспечения Грид. Во первых многие аспекты были стандартизированы, и уже можно сказать произошел переход на единый стандарт OGSA. Во вторых, помимо инструментальных средств, новое программное обеспечение Грид, будет включать комплект служб, которые не только поддерживают дистанционные операции, но и обеспечивают функционирование Грид как операционной среды (мониторинг всей инфраструктуры, управление заданиями и распределение ресурсов).

Также появление Грид позволило в полной мере решить главные задачи: виртуализация ресурсов и интероперабельность. Теперь не нужно сосредотачивать отдельное внимание на физические, логические ресурсы. Все ресурсы представляются в виде служб, и таким образом, обеспечивается унификация работы с ресурсами. Задача интероперабельности решается за счет того, что в OGSA используются стандартные протоколы, понимаемые и поддерживаемые всеми аппаратно-программными платформами.

**В четвертой главе** приводится описание разных Грид продуктов, использованных в работе. Для решения проблемы выбора программных средств для реализации Грид среды, мы разделим средний уровень Грид на два логических компонента: систему доступа и систему управления распределенными ресурсами. Это сделано по двум причинам. Во первых, с целью разделить функциональность, связанную с аутентификаций, авторизацией, политиками безопасности, управления виртуальными организациями, обеспечения единого окна и т.д. и функциональность, связанную непосредственно с управлением ресурсами Грид. Вторая причина – это наличие ярко выраженных продуктов, которые предназначены для решения именно конкретных задач.

В качестве системы доступа выбор пал на UNICORE (Uniform Interface to Computing Resources). По сути, выбор этот оказался безальтернативным, потому что по сравнению с другими подобными системами, например Condor, только UNICORE предоставляет доступ непосредственно к ресурсу. Во вторых, европейские партнеры по проекту DEISA, с которыми ведется сотрудничество по вопросам организации Грид, используют именно UNICORE, и поэтому при интеграции с европейскими суперкомпьютерными центрами, необходимо использовать именно это продукт.

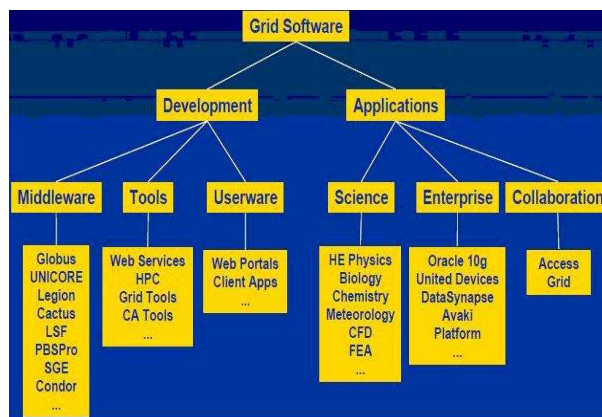


Рис.3. Классификация программного обеспечения Грид

Выбор системы управления распределенными ресурсами более богатый. Существует целый класс компонент, который так и называется – Distributed Resources Management System (DRMS). В состав этого класса входят такие продукты, как Sun Grid Engine, Torque, LSF, LoadLeveler и т.д. В качестве DRMS мы использовали Sun Grid Engine от корпорации Sun Microsystems.

Платформа Sun Grid Engine (SGE) основана на программном обеспечении, разработанном фирмой Genias, известном как Codine/GRM. В SGE задания находятся в зоне ожидания, а очереди на серверах обеспечивают сервисы для заданий. Она

- Позволяет объединить несколько серверов или рабочих станций в единый вычислительный ресурс, который может быть использован как для пакетных задач, так и для высокопроизводительных пакетных вычислений.
- Потребитель вводит задание в SGE и объявляет профиль необходимых требований для его выполнения.
- SGE определяет заданию соответствующую очередь и распределяет его либо с высшим приоритетом, либо с самым длинным временем ожидания, пробуя запускать новые задания на наиболее соответствующей или наименее загруженной очереди.

Продукт Sun Grid Engine предназначен, прежде всего, для сетей среднего размера, охватывающих отдел или небольшое предприятие. Этот продукт предназначен для сетей класса Cluster Grid и доступен бесплатно.

При изучении Sun Grid Engine инфраструктуры легко распознать трудные отношения, которые связывают SGE со знакомой окружающей средой UN\*X. Sun Grid Engine объединяет вычислительные ресурсы для решения особо сложных задач. Объединяя отдельные компьютеры в единый вычислительный комплекс, Sun Grid Engine предлагает эффективный подход для решения таких задач.

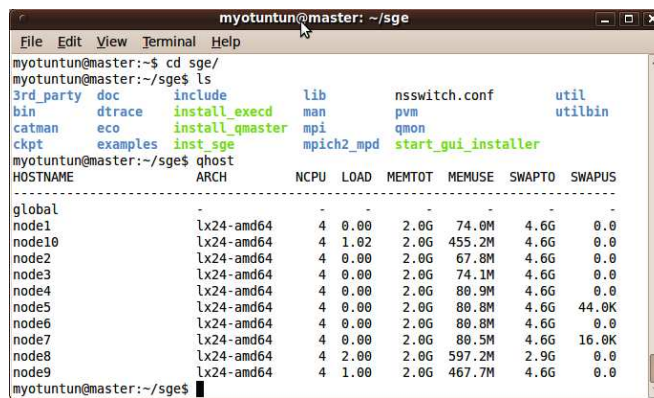
Это решение может иногда казаться грубым и ограниченным, но оно вероятно подходит для большинства потребителей, особенно в области научных исследований, где относительно небольшое количество людей

потребляют большое количество разнородных ресурсов с непредсказуемым заранее расписанием.

Основной вывод из проведенного исследования состоит в рекомендации — не использовать простые пути, связанные с использованием SGE только для объединения ресурсов, а затем применения стандартных параллельных библиотек (это почти никогда не получается на гетерогенных системах), а наоборот, использовать для запуска и мониторинга задач весь мощный инструментарий привычной операционной среды UN\*X, который, в основном, доступен и в SGE.

**В пятой главе** приведено описание развертывания тестовой Грид среды, опыт которой будет использован в дальнейшем для развертывания системы. Многие современные прикладные задачи настолько сложны, что для их решения необходима консолидация вычислительных мощностей нескольких суперкомпьютеров, серверов, рабочих станций. Такие задачи нередко встречаются в нанотехнологиях, науках о материалах, биоинформационных технологиях, криптоанализе и других областях. Как правило, в распоряжении исследователя имеются разнородные вычислительные ресурсы различных типов: это рабочие станции, небольшие многопроцессорные системы, доступные в монопольном режиме, суперкомпьютеры коллективного доступа. Организация решения вычислительных задач в распределенной среде, состоящей из ресурсов перечисленного типа, сопряжена со значительными сложностями, обусловленными следующими факторами:

- Неоднородность программного обеспечения и средств доступа к вычислительным ресурсам;
- Относительно высокая вероятность отказа компонента распределенной системы или канала связи с ним;
- Существенные различия в производительности между узлами распределенной системы, осложняющие задачу оптимизации распределения вычислительной нагрузки между ними.



```
myotuntun@master: ~/sge
File Edit View Terminal Help
myotuntun@master:~$ cd sge/
myotuntun@master:~/sge$ ls
3rd_party doc include lib nsswitch.conf util
bin dtrace install_execd man pvm utilbin
catman eco install_qmaster mpi qmon
ckpt examples inst_sge mpich2_mpd start_gui_installer
myotuntun@master:~/sge$ qhost
HOSTNAME ARCH NCPUS LOAD MEMTOT MEMUSE SWAPTO SWAPUS
-----
global - - - - -
node1 lx24-amd64 4 0.00 2.0G 74.0M 4.6G 0.0
node10 lx24-amd64 4 1.02 2.0G 455.2M 4.6G 0.0
node2 lx24-amd64 4 0.00 2.0G 67.8M 4.6G 0.0
node3 lx24-amd64 4 0.00 2.0G 74.1M 4.6G 0.0
node4 lx24-amd64 4 0.00 2.0G 80.9M 4.6G 0.0
node5 lx24-amd64 4 0.00 2.0G 80.8M 4.6G 44.0K
node6 lx24-amd64 4 0.00 2.0G 80.8M 4.6G 0.0
node7 lx24-amd64 4 0.00 2.0G 80.5M 4.6G 16.0K
node8 lx24-amd64 4 2.00 2.0G 597.2M 2.9G 0.0
node9 lx24-amd64 4 1.00 2.0G 467.7M 4.6G 0.0
myotuntun@master:~/sge$
```

Рис.4. Хосты в AMD кластере

Учет ненадежности и различий в производительности при решении комбинаторных и оптимизационных задач рассмотрен во многих работах. Основной результат этой главы состоит в интеграции разнородных вычислительных ресурсов. Основной подход к этой проблеме заключается в разработке универсальных средств системного уровня, обеспечивающих прозрачный для приложений способ интеграции. К этой категории можно отнести различные версии MPI, позволяющие переносить параллельные программы, предназначенные для выполнения на многопроцессорных вычислительных системах, в грид-среду, объединяющую несколько кластеров.

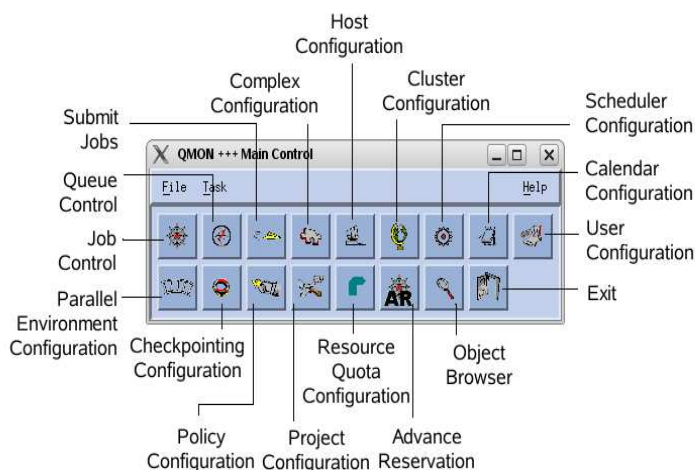


Рис.5. Параллельное операционное окружение (PE)

Параллельное операционное окружение (PE) является пакетом программного обеспечения, который позволяет параллельное вычисление на параллельных платформах в сетевом окружении. Множество систем такого рода развились за последние годы в жизнеспособную технологию для распределенной и параллельной обработки на различных платформах аппаратных средств. SGE может непосредственно запустить некоторое количество задач с параллельным окружением типа PVM или MPI. Чтобы планировать такие задания, SGE обеспечивает простой интерфейс, определяемый Параллельным Окружением.

- MPI – Message Passing Interface ( Message Passing Interface Forum )
- PVM – Parallel Virtual Machine ( Oak Ridge National Laboratories )

Любой вид отношений между заданиями отдельных PE может быть отобран при планировании работы, используя соответствующие инструментальные средства PE, обеспеченные SGE.

```

myotuntun@master: ~/sge/mpich2_mpd
File Edit View Terminal Help
myotuntun@master:~/sge/mpich2_mpd$ ls
bin      mpich2_mpd.sh  README      rsh  startmpich2.sh
hostname mpich2.template README.atm  src  stopmpich2.sh
myotuntun@master:~/sge/mpich2_mpd$ qsub mpich2
mpich2_mpd.sh  mpich2.template
myotuntun@master:~/sge/mpich2_mpd$ qsub mpich2_mpd.sh
Your job 256 ("mpich2_mpd.sh") has been submitted
myotuntun@master:~/sge/mpich2_mpd$ qstat -f
queuename
-----
all.q@node10.local  BIP  0/0/4    1.00  lx24-amd64  E
-----
all.q@node2.local   BIP  0/0/4    0.00  lx24-amd64
-----
all.q@node3.local   BIP  0/0/4    0.00  lx24-amd64
-----
all.q@node4.local   BIP  0/0/4    0.00  lx24-amd64
-----
all.q@node5.local   BIP  0/0/4    0.00  lx24-amd64
-----
all.q@node6.local   BIP  0/0/4    0.00  lx24-amd64
-----
all.q@node7.local   BIP  0/0/4    0.00  lx24-amd64
-----
all.q@node8.local   BIP  0/0/4    2.00  lx24-amd64
-----
all.q@node9.local   BIP  0/0/4    1.00  lx24-amd64
-----
#####
- PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
#####
220 0.00000 simple.sh myotuntun  Eqw 10/06/2009 23:19:24 1
256 0.00000 mpich2_mpd myotuntun qw  04/21/2010 02:13:00 1
myotuntun@master:~/sge/mpich2_mpd$

```

Рис.6. Запуск MPICH2 и SGE в распределенной вычислительной среде

Кроме того эта система предлагает альтернативный способ для запуска параллельной задачи: большинство параллельных технологий фактически не обеспечивают надежного управления гетерогенным ресурсом, хотя, достаточно часто, они более гибкие по сравнению с инфраструктурой grid, поэтому в SGE есть возможность выполнить тесную интеграцию с параллельным окружением. Тесная интеграция означает, что ответственность за запуск задачи передается с уровня параллельных библиотек на уровень операционного окружения среды grid SGE, гарантируя все отмеченные выше особенности управления ресурсом.

Другим примером интеграции разнородных вычислительных ресурсов на системном уровне является технология программных «мостов» между сервисными гридами и гридами рабочих станций. Эти технологии позволяют использовать ресурсы сервисных гридов для выполнения заданий из гридов рабочих станций. Возможна также интеграция в обратном направлении.

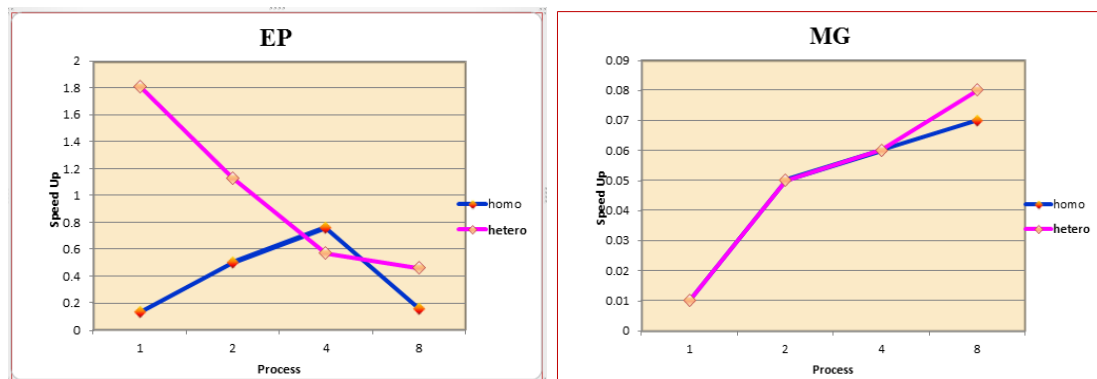
**В шестой главе** рассматриваются тестовые программы взятые из разных предметных областей и представляющие собой либо модельные, либо реальные промышленные приложения. Такие тесты позволяют действительно оценить производительность компьютера на реальных задачах и получить наиболее полное представление об эффективности работы компьютера с конкретным приложением.

Наиболее распространенными тестами, построенными по этому принципу, являются: набор из 24 Ливерморских циклов ( The Livermore Fortran Kernels, LFK) и пакет NAS Parallel Benchmarks (NPB), в состав которого входят две группы тестов, отражающих различные стороны реальных программ вычислительной гидродинамики. NAS тесты являются альтернативной Linpack, поскольку они относительно просты и в то же

время содержат значительно больше вычислений, чем, например, Linpack или LFK. Однако при всем разнообразии тестовые программы не могут дать полного представления о работе компьютера в различных режимах.

Особенностью реализации теста является необходимость компиляции программы для каждого класса задач. Это вызвано тем, что стандарт языка fortran-77 не поддерживает динамическое распределение памяти. Результаты NPВ получаются в миллионах действий в секунду. Задачи были выбраны после оценки множества больших прикладных программ вычислительной гидродинамики, решаемых в NASA. Задачи пакета NPВ содержат значительно больше вычислений, чем использовавшиеся ранее тесты, например такие, как Livermore Loops или LINPACK, поэтому они более приемлемы для оценки параллельных машин. С другой стороны, эти задачи относительно просты, что позволяет ставить эти задачи на новых вычислительных системах без значительных усилий и задержек.

Основные проблемы, с которыми пришлось столкнуться, это невозможность использования стандартных кластерных тестов для гридовских окружений, а, с другой стороны, с невозможностью оценить накладные расходы для Грид-варианта тестов. Поэтому, основываясь на концепции Главы 5, были скомпилированы новые наборы тестов в разработанных PSE, и именно они использовались для оптимизации распределенных вычислительных систем.



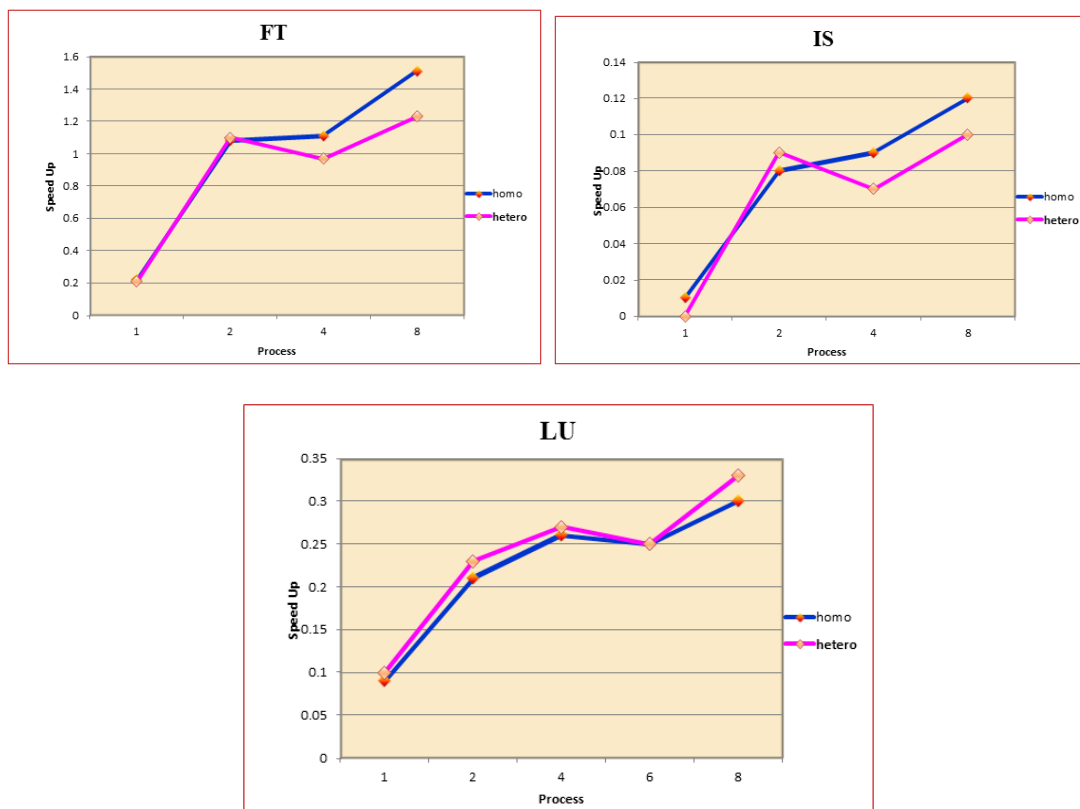


Рис.7. Сравнение тестов дальнего взаимодействия NPВ для однородной и гетерогенной распределенных вычислительных сред

Проведенные тесты показали высокую эффективность предлагаемых подходов - масштабируемость приложений в гетерогенных системах вызывает удивление.

По результатам данного исследования, основным фактором, ограничивающим производительность, следует признать ЛВС. При использовании сетевой технологии с меньшими задержками и большей пропускной способностью возможен значительный выигрыш в производительности.

## ЗАКЛЮЧЕНИЕ

Таким образом, в настоящей диссертации была разработана методика создания и оптимизации распределенных гетерогенных вычислительных комплексов. Основная идея состоит в объединении трех последовательных шагов. Сначала с помощью специально подобранного промежуточного программного обеспечения осуществляется системная интеграция вычислительных систем, входящих в гетерогенный комплекс. Затем на основе специальных алгоритмов создается операционное окружение для



удобной работы пользователя в распределенной вычислительной среде. И, наконец, на основе набора простых математических моделей строится синтетический тест, позволяющий определять производительность созданного комплекса и, тем самым осуществлять его оптимизацию.

Особые возможности достигаются при применении технологии виртуализации, позволяющей балансировать производительности машин и создавать сплошную адресацию ячеек памяти, тем самым устраняя основные проблемы, ограничивающие масштабирование комплекса.

Всего было построено три варианта распределенной вычислительной среды на процессорах разной архитектуры (включая высокопроизводительные и многопоточные), которые демонстрировали высокую эффективность предлагаемых методик. Все это позволяет рекомендовать предложенные программные продукты и синтетическую модель для широкого использования в распределенных гетерогенных вычислительных средах.

**Опубликованные научные работы по теме диссертации  
в изданиях, определенных ВАК:**

1. А. В. Богданов, Ла Мин Хтут, Мьё Тун Тун, Виртуальная организация (ВО): Концептуальный анализ системы входа для пользователей в развитых распределённых вычислительных средах, Известия СПбГЭТУ «ЛЭТИ». – 2011.– Вып. 4.– С. 48-52.

**и в других изданиях:**

2. A.V. Bogdanov, A.A. Lazarev, Myo Tun Tun, La Min Htut, Разработка распределенных вычислительных систем Grid и запуск приложений в гетерогенной вычислительной среде, Distributed Computing and Grid-Technologies in Science and Education: Proceedings of the 4th Intern. Conf. (Dubna, June28-July 3, 2010). –Dubna: JINR, -p.69-74
3. A.V.Bogdanov, A.A. Lazarev, La Min Htut, Myo Tun Tun, Создание системы доступа для пользователей в Грид среде, Distributed Computing and Grid-Technologies in Science and Education: Proceedings of the 4th Intern. Conf. (Dubna, June28-July 3, 2010). –Dubna: JINR, -p.63-69
4. Богданов А.В., Мьё Тун Тун, Разработка распределенных вычислительных систем и запуск приложений в гетерогенной вычислительной среде. Санкт-Петербург, 22-25 июня 2009 г. XVI Всероссийская научно-методическая конференция "Телематика'2009", С.425-427.

5. Мьё Тун Тун, Конфигурация Sun Grid Engine для гибридных систем // Сборник докладов 5-я Общероссийская конференция молодых и специалистов по морским интеллектуальным технологиям «Моринтех-юниор 2009». Санкт-Петербург.10-12 ноября 2009 г. С.91-93.
6. A.V. Bogdanov, Thurein Kyaw Lwin, Myo Tun Tun, La Min Htut, Система интеграции гетерогенных комплексов для научных вычислений, основанных на использовании технологии DB2 // Proceedings of International Conference «Computer Science & Information Technologies», 28 September - 2 October, 2009, Yerevan, Armenia, pp.397-399.