БУ ВО

«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Ханты-Мансийского автономного округа – Югры»

Кафедра автоматизированных систем обработки информации и управления (АСОИУ)

РЕФЕРАТ

по теме: «GRID-технологии»

Выполнил студент гр. 606-71м

Боровков Дмитрий Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Преподаватель:

Горбунов Дмитрий Владимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Сургут, 2018

# Понятие GRID

Грид - согласованная, открытая и стандартизованная среда, которая обеспечивает гибкое, безопасное, скоординированное разделение (общий доступ) ресурсов в рамках виртуальной организации".

Также как и электрические сети, грид это соединение технологии, инфраструктуры и стандартов. Технология – это специальное программное обеспечение, которое позволяет организациям или частным лицам предоставлять ресурсы (компьютеры, хранилища данных, сети и другие) в общее пользование, а потребителям – использовать их, когда необходимо. Инфраструктура состоит из аппаратных средств и служб (на основе людских и программных ресурсов), которые должны быть организованы, и постоянно поддерживаться для того, чтобы ресурсы могли совместно использоваться. Наконец, стандарты должны определять формат и протоколы обмена сообщениями, как между службами, так и между службами и пользователями, а также правила работы грида.

2. Общие задачи грида

Вычислительные ресурсы предоставляют пользователю грид-системы процессорные мощности.

При всем разнообразии архитектур любая вычислительная система может рассматриваться как потенциальный вычислительный ресурс грид-системы. Необходимым условием для этого является наличие ППО, реализующего стандартный внешний интерфейс с ресурсом и позволяющего сделать ресурс доступным для грид-системы. Основной характеристикой вычислительного ресурса является производительность.

Ресурсы хранения также используют ППО, реализующее унифицированный интерфейс управления и передачи данных. Как и в случае вычислительных ресурсов, физическая архитектура ресурса памяти не принципиальна для грид-системы, будь то жесткий диск на рабочей станции или система массового хранения данных на сотни терабайт. Основной характеристикой ресурсов хранения данных является их объем.

Информационные ресурсы и каталоги являются особым видом ресурсов хранения данных. Они служат для хранения и предоставления метаданных и информации о других ресурсах грид-системы.

Сетевой ресурс является связующим звеном между распределенными ресурсами грид-системы. Основной характеристикой сетевого ресурса является скорость передачи данных.

Задачи:

1. Обеспечение распределенных вычислений и обработки данных (удаленный доступ к вычислительным ресурсам). На время выполнения задачи или проекта есть возможность использовать ресурсы множества персональных компьютеров, рабочих станций, кластеров, может быть даже и суперкомпьютеров, а также хранилищ данных, размещенных в разных точках мира и принадлежащих разным людям и учреждениям.
2. Повышение эффективности компьютерных ресурсов.

3. Типы грид-систем с точки зрения решаемых задач

1. Вычислительный грид (Computational Grid) ‑ достижение максимальной скорости вычислений за счет глобального распределения этих вычислений между тысячами компьютеров.

2. Грид для интенсивной обработки данных(Data Grid) ‑ обработка огромных объемов данных относительно несложными программами.

3. Семантический Грид для оперирования данными из различных баз данных (Semantic Grid).

1.2.3 Задачи грида и задачи суперкомпьютеров (сходство и различие)

1. грид не слишком подходит для параллельных вычислений с интенсивным межпроцессорным обменом. частое общение невозможно обеспечит для географически распределенных и, возможно, аппаратно-неоднородных ресурсов в грид-среде.

2. Грид-технология не является технологией параллельных вычислений, она предназначена для удаленного запуска отдельных задач на территориально распределенные ресурсы. Поэтому если громоздкая задача, которую необходимо решить, может быть разбита на большое количество маленьких, независимых (не обменивающихся никакими данными) частей, - грид-технология оказывается особенно эффективным и относительно дешевым решением. Напротив, суперкомпьютеры оказываются для таких вычислений неоправданно дорогим и неэффективным решением.

1.4.3 Общие принципы архитектуры грид-систем

1.4.3.1 Структура (стек) протоколов глобального грида



В левой части рис. 2 показаны уровни стека грид-протоколов, а справа - четыре аналогичных им уровней модели OSI (всего в стеке OSI семь уровней).

1.4.3.1.1 Аппаратный уровень: управление локальными ресурсами

Компоненты аппаратного уровня реализуют локальные операции, специфические для каждого конкретного ресурса (логического или физического). По существу, представляет собой набор интерфейсов для управления локальными ресурсами.

1.4.3.1.2 Связывающий уровень: коммуникации и безопасность

Коммуникационные протоколы связывающего уровня должны обеспечивать надежный транспорт и маршрутизацию сообщений, а также присвоение имен объектам сети, а протоколы аутентификации этого уровня предоставляют криптографические механизмы для идентификации и проверки подлинности пользователей и ресурсов.

1.4.3.1.3 Ресурсный уровень: совместное использование ресурсов

Ресурсный уровень с помощью коммуникационных и аутентификационных протоколов, входящих в нижележащий связывающий уровень, проводит согласование методов безопасности, инициализацию и мониторинг ресурсов, и управление ими. Для доступа к локальным ресурсам и дальнейшего управления ресурсный уровень вызывает соответствующие функции аппаратного уровня. Заметим, что протоколы ресурсного уровня предназначены исключительно для работы с локальными ресурсами, они не учитывают глобальное состояние системы.

Ресурсный уровень включает два основных класса протоколов:

 информационные протоколы, предназначенные для получения информации о структуре и состоянии ресурса, его конфигурации;

 протоколы управления, обеспечивающие согласованность доступа к разделяемому ресурсу и определяющие необходимые операции, которые ресурс должен выполнить.

1.4.3.1.4 Коллективный уровень: координация ресурсов

Протоколы коллективного уровня отвечают за взаимодействие всех элементов пула ресурсов, что и отражено в самом названии.

1.4.3.1.5 Прикладной уровень: запуск приложений в грид-среду

Этот высший уровень грид-архитектуры включает пользовательские приложения, которые исполняются в среде объединенных ресурсов.

1.4.3.2 Архитектура сервисов распределенных систем и технологии ее реализации

Сегодня наиболее предпочтительными подходами к построению распределенных систем считаются сервисно-ориентированная архитектура (Service Oriented Architecture, SOA), технология веб-сервисов, и грид-стандарты (в первую очередь - Открытая архитектура грид-сервисов (Open Grid Services Architecture, OGSA)).

1.4.3.2 Архитектура сервисов распределенных систем и технологии ее реализации

1.4.3.2.1 Сервисно-ориентированная архитектура (SOA) распределенных систем

Сервис (служба) - программный компонент, к которому можно удаленно

обратиться посредством компьютерной сети, и предоставляющая некоторые

функциональные возможности запрашивающей стороне.

 Сервисно-ориентированная архитектура (service-oriented architecture, SOA) - является основой построения надежных распределенных систем, которые в качестве услуг предоставляют функциональные возможности, с дополнительным акцентом на слабые связи между взаимодействующими сервисами.

Вообще говоря, сервисы имеют следующие характеристики:

 Сервисы могут быть полезными каждый сам по себе, или они могут быть

объединены, чтобы предоставить единый высокоуровневый сервис.

 фактически,

сервисы и различаются/идентифицируются в соответствии с совокупностью

сообщений, которые они могут принять и ответов, которые они могут направить

запрашивающей стороне.

 Сервис может участвовать в таких процессах обработки запросов, при которых

порядок, в котором сообщения посылаются и принимаются, влияет на результат

операций, выполненных сервисом.

 Сервис может быть полностью независимым, или может зависеть от

существования других сервисов, или каких-либо ресурсов, например, баз данных.

 Сервисы предоставляют информацию о своих возможностях, интерфейсах,

политике и поддерживаемых протоколах связи.



Рис.3 Взаимодействие сервисов в SOA-среде

Потенциальный клиент, который может быть другим сервисом (или человеком), делает

запрос в сервис регистрации (2), чтобы найти сервис, который удовлетворяет его

потребностям. Регистрационный сервис возвращает (возможно пустой) список

подходящих сервисов; клиент выбирает один из них и передает ему запрос, используя

любой взаимно распознаваемый протокол (3). В этом примере, сервис отвечает (4),

передавая или результат требуемой операции или сообщение об ошибке.

1.4.3.2.2 Принцип слабой связи

Этот термин подразумевает, что взаимодействующие программные компоненты имеет минимальное знание друг о друге: они находят информацию, которая им нужна для взаимодействия непосредственно перед взаимодействием.

Достоинствами слабой связи являются:

 Гибкость: сервис может быть расположен на любом сервере, а при необходимости ‑ перемещен.

 Масшабируемость: функциональные возможности сервиса могут быть расширены или сужены.

 Возможность модификации реализации.

 Отказоустойчивость: Если сервис становится недоступным по любой причине, клиенты могут сделать запрос к службе регистрации для обнаружения другого сервиса, который предоставляет требуемые услуги.

1.4.3.2.4 Веб-сервисы

Веб-сервисы (или веб-службы) – это распределенные программные компоненты, идентифицируемые своим сетевым адресом, интерфейс которых описан на специальном ≪диалекте≫ языка XML (eXtensible Markup Language), а именно WSDL (Web Service Description Language). Другие программные системы могут взаимодействовать с веб-сервисами согласно этому описанию посредством сообщений, основанных на другом ≪диалекте≫ XML - SOAP, и передаваемых с помощью интернет-протоколов.

1.4.3.2.5 Веб-сервисы и SOA

SOA и веб-сервисы являются ≪ортогональными≫ понятиями: сервисная ориентация – это архитектурный стиль, а веб-сервисы - технология выполнения. Они, конечно, могут использоваться совместно – как это часто и случается, но они взаимно независимы.

Веб-сервисы хорошо подходят в качестве строительных блоков SOA-среды, но в их определении нет ничего, что обязательно требует воплощение принципов SOA.

SOA – модульный подход к разработке программнного обеспечения, основанный на использовании распределенных, слабо связанных заменяемых компонентов, оснащенных стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.Программные комплексы, разработанные в соответствии с СОА, обычно реализуются как набор веб-служб взаимодействующих по протоколу СОАП.

1.4.3.2.7 Сервисно-ориентированный грид

Быстрый прогресс в технологии веб-сервисов и разработке соответствующих стандартов обеспечили эволюционный путь от жесткой и узко-направленной архитектуры грид-систем первого поколения к стандартизированным, сервис-ориентированным гридам.



Рис. 6 Упрощенная схема сервисно-ориентированного грида

На схеме показана единая консоль и для запуска заданий в грид-среду, и для управления грид-ресурсами. Программное обеспечение интерфейса пользователя (консоли) обращается к сервису регистрации, чтобы получить информацию о существующих грид-ресурсах. Затем пользователь посредством консоли входит в контакт с сервисами, ≪представляющими≫ (виртуализующими) каждый ресурс, чтобы запросить периодическое получение данных о работе ресурсов и получение извещений о существенных изменениях в их состоянии (например, если ресурс становится недоступным или сильно загруженным).

Пользователь направляет запрос на запуск задания в службу запуска, которая передает запрос службе распределения заданий (часто называемой планировщиком). Служба распределения контактирует со службой, представляющей приложение, и запрашивает информацию о требованиях к ресурсам для выполнения задания. Затем служба распределения запрашивает у службы регистрации информацию о всех подходящих ресурсах в гриде и напрямую контактирует с ними, чтобы убедиться в их доступности.

Если подходящие ресурсы доступны, планировщик выбирает наилучшую доступную совокупность ресурсов и передает информацию о их сервису приложения с запросом на начало выполнения. В противном случае планировщик ставит задание в очередь и выполняет его, когда необходимые ресурсы становятся доступными. Когда выполнение задания заканчивается, сервис приложения сообщает о результате планировщику, который извещает об этом сервис запуска заданий. Сервис запуска заданий, в свою очередь, уведомляет пользователя.

1.4.3.2.8 Open Grid Services Architecture (OGSA)

Конвергенция SOA и вычислительных грид-систем воплощена Глобальным грид-форумом (Global Grid Forum, GGF) в Открытой архитектуре грид-сервисов (Open Grid services Architecture, OGSA). Основной документ [34], фиксирующий архитектуру OGSA, описывает общие принципы построения сервисно-ориентированного грида в терминах необходимых функциональных возможностей, например, выполнение заданий, управление ресурсами и данными, обеспечение безопасности. Основной целью этого и

других документов, посвященных OGSA (см. [23]), является стандартизация интерфейсов и поведения основного (базового) набора грид-сервисов, чтобы они могли взаимодействовать как друг с другом, так и с грид-приложениями независимо от конкретных реализаций таких грид-сервисов.