

ТЕМА: ТЕХНОЛОГИЯ GRID

ПЛАН:

- ВВЕДЕНИЕ
- 1. ТЕХНОЛОГИЯ GRID
- 2. ТЕХНОЛОГИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ
- 3. ПРОМЕЖУТОЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В
РАСПРЕДЕЛЁННЫХ СИСТЕМАХ
- СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ВВЕДЕНИЕ

Область распределенных вычислительных систем в настоящее время характеризуется быстрыми темпами изменения идеологий и подходов. За короткую историю существования систем такого типа появилось множество различных парадигм реализации распределенных вычислений, набравших большой вес и общее признание, но практически исчезнувших впоследствии под давлением более новых и модных подходов. Однако когда технология исчезает из виду, очень часто она появляется вновь под новым именем. В результате происходит непрерывное перемешивание базовых концепций с новейшими подходами к разработке.

В середине 1990-х существовало два основных подхода к разработке распределенных вычислительных систем. С одной стороны, концепция Веб представляла собой ориентированное на человека распределенное информационное пространство. С другой стороны, технологии распределенных объектов, такие как CORBA и DCOM были в первую очередь ориентированы на создание распределенных сред, которые эмулировали процесс разработки и исполнения локальных приложений, обеспечивая преимущества доступа к сетевым ресурсам. Но, несмотря на первоначальную идею Веб как пространства, которое позволяло многим людям обмениваться информацией, большинство пользователей просто потребляли информацию, не публикуя ничего взамен. Между тем системы распределенных объектов росли с точки зрения предоставляемых возможностей, но становились все более тяжелыми в плане разработки и использования.

Сразу после начала нового тысячелетия произошел взрыв развития новых методов и промежуточного программного обеспечения для распределенных вычислительных систем, включая технологии одноранговых сетей (peer-to-peer или P2P) и грид-технологии. Применение P2P позволило множеству

пользователей, которые раньше были простыми потребителями информации, поучаствовать в предоставлении контента. С другой стороны, применение технологии грид позволило интегрировать крупные комплексы обработки и хранения данных, обеспечивая их доступность для различных правительственных и научных пользователей. Концепция грид-вычислений была ориентирована на построение инфраструктуры, обеспечивающей «вычисления по требованию», аналогично тому, как мы сейчас получаем доступ к коммунальным услугам, например, к электричеству.

В то же время, бизнес-сообщество занялось разработкой следующего поколения спецификаций, призванных решить проблемы ранних стандартов распределенных объектных технологий посредством Веб-сервисов и сервис-ориентированной архитектуры. Слияние бизнес подхода к предоставлению вычислительных ресурсов в виде сервисов и концепции грид-вычислений привело к появлению в конце 2010-х новой концепции получившей название Облачных вычислений.

ТЕХНОЛОГИЯ GRID

Термин «грид» был введен в обращение Яном Фостером в начале 1998 года публикацией книги «Грид. Новая инфраструктура вычислений»:

Грид - это система, которая координирует распределенные ресурсы посредством стандартных, открытых, универсальных протоколов и интерфейсов для обеспечения нетривиального качества обслуживания (QoS - Quality of Service).

Хотя в последнее десятилетие базовая идея грид не претерпела существенных изменений, всеобъемлющего определения грид не существует до сих пор.

Основной идеей, заложенной в концепции грид-вычислений, является централизованное удаленное предоставление ресурсов, необходимых для решения различного рода вычислительных задач. В каком-то смысле, концепция грид-вычислений идея рифмуется с концепцией электросети (англ. Power Grid): нам не важно, откуда к нам в розетку приходит электричество. Независимо от этого мы можем подключить к электросети утюг, компьютер или стиральную машину. Также и в идеологии грид: мы можем запустить любую задачу с любого компьютера или мобильного устройств на вычисление, ресурсы же для этого вычисления должны быть автоматически предоставлены на удаленных высокопроизводительных серверах, независимо от типа нашей задачи.

С более практической точки зрения, основная задача, лежащая в основе концепции грид, это согласованное распределение ресурсов и решение задач в условиях динамических, многопрофильных виртуальных организаций. Распределение ресурсов, в котором заинтересованы разработчики грид, это не обмен файлами, а прямой доступ к компьютерам, программному обеспечению, данным и другим ресурсам, которые требуются для совместного решения задач

и стратегий управления ресурсами, возникающих в промышленности, науке и технике. Виртуальной организацией (ВО) называют ряд отдельных людей или учреждений, объединенных едиными правилами коллективного доступа к распределенным вычислительным ресурсам. Исследования и разработки в сообществе грид привели к разработке протоколов, сервисов и инструментария, направленного именно на те проблемы, которые возникают при попытке создания масштабируемых ВО. Эти технологии включают в себя:

- . решения по безопасности, поддерживающие управление сертификацией и политиками безопасности, когда вычисления производятся несколькими организациями;

- . протоколы управления ресурсами и сервисами, поддерживающие безопасный удаленный доступ к вычислительным ресурсам и ресурсам данных, а также перераспределение различных ресурсов;

- . протоколы запроса информации и сервисы, обеспечивающие настройку и мониторинг состояния ресурсов, организаций и сервисов;

- . сервисы обработки данных, обеспечивающие поиск и передачу наборов данных между системами хранения данных и приложениями.

Выделяют следующие уровни архитектуры грид:

- . Базовый уровень (Fabric) - содержит различные ресурсы, такие как компьютеры, устройства хранения, сети, сенсоры и др. Здесь определяются службы, обеспечивающие непосредственный доступ к процессорным мощностям, ресурсам памяти, информационным ресурсам, сетевым ресурсам, использование которых распределено посредством протоколов Грид;

- . Связывающий уровень (Connectivity) - определяет коммуникационные протоколы и протоколы аутентификации, обеспечивая передачу данных между ресурсами базового уровня. Связывающий уровень грид основан на стеке протоколов TCP/IP: Интернет (IP, ICMP), транспортные протоколы (TCP, UDP), прикладные протоколы (DNS, OSRF...);

. Ресурсный уровень (Resource) - реализует протоколы взаимодействия с ресурсами PBC и их управления. Данные протоколы, обеспечивающие выполнение следующих функций:

согласование политик безопасности использования ресурса;

процедура инициации ресурса;

мониторинг состояния ресурса;

контроль над ресурсом;

учет использования ресурса;

. Коллективный уровень (Collective) - отвечает за глобальную интеграцию различных наборов ресурсов и может включать в себя службы каталогов; службы совместного выделения, планирования и распределения ресурсов; службы мониторинга и диагностики ресурсов; службы репликации данных;

. Прикладной уровень (Applications) - инструментарий для работы с грид и пользовательские приложения, исполняемые в среде ВО. Они могут использовать ресурсы, находящиеся на любых нижних слоях архитектуры Грид.

ТЕХНОЛОГИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

С одной стороны, у термина «Облачные вычисления» нет устоявшегося стандартного определения. С другой стороны множество различных корпораций, ученых и аналитиков дают собственные определения этому термину. Определение облачных вычислений вызвало дебаты и в научном сообществе. В отличие от определений, которые можно найти в коммерческих изданиях, научные определения ориентируются не только на то, что будет предоставлено пользователю, но и на архитектурные особенности предлагаемой технологии.

Ян Фостер определяет облачные вычисления как «парадигму крупномасштабных распределенных вычислений, основанную на эффекте

масштаба, в рамках которой пул абстрактных, виртуализованных, динамически масштабируемых вычислительных ресурсов, ресурсов хранения, платформ и сервисов предоставляется по запросу внешним пользователям через Интернет».

Другое определение, возможно, чуть более целостное, чем предыдущее:

«Облако - это большой пул легко используемых и легкодоступных виртуализованных ресурсов (таких как аппаратные комплексы, сервисы и др.). Эти ресурсы могут быть динамически перераспределены (масштабированы) для подстройки под динамически изменяющуюся нагрузку, обеспечивая оптимальное использование ресурсов. Этот пул ресурсов обычно предоставляется по принципу «оплата по мере использования». При этом владелец облака гарантирует качество обслуживания на основе определенных соглашений с пользователем».

Все определения иллюстрируют одну простую мысль: феномен облачных вычислений объединяет несколько различных концепций информационных технологий и представляет собой новую парадигму предоставления информационных ресурсов (аппаратных и программных комплексов). Со стороны владельца вычислительных ресурсов облачные вычисления ориентированы на предоставление информационных ресурсов внешним пользователям. Со стороны пользователя, облачные вычисления - это получение информационных ресурсов в виде услуги у внешнего поставщика, оплата за которую производится в зависимости от объема потребленных ресурсов согласно установленному тарифу. Ключевыми характеристиками облачных вычислений являются масштабируемость и виртуализация.

Масштабируемость представляет собой возможность динамической настройки информационных ресурсов к изменяющейся нагрузке, например к увеличению или уменьшению количества пользователей, изменению необходимой емкости хранилищ данных или вычислительной мощности. Виртуализация, которая также рассматривается как важнейшая технология всех

облачных систем, в основном используется для обеспечения абстракции и инкапсуляции. Абстракция позволяет унифицировать «сырые» вычислительные, коммуникационные ресурсы и хранилища информации в виде пула ресурсов и выстроить унифицированный слой ресурсов, который содержит те же ресурсы, но в абстрагированном виде. Они представляются пользователям и верхним слоям облачных систем как виртуализованные серверы, кластеры серверов, файловые системы и СУБД. Инкапсуляция приложений повышает безопасность, управляемость и изолированность. Еще одной важной особенностью облачных платформ является интеграция аппаратных ресурсов и системного ПО с приложениями, которые предоставляются конечному пользователю в виде сервисов.

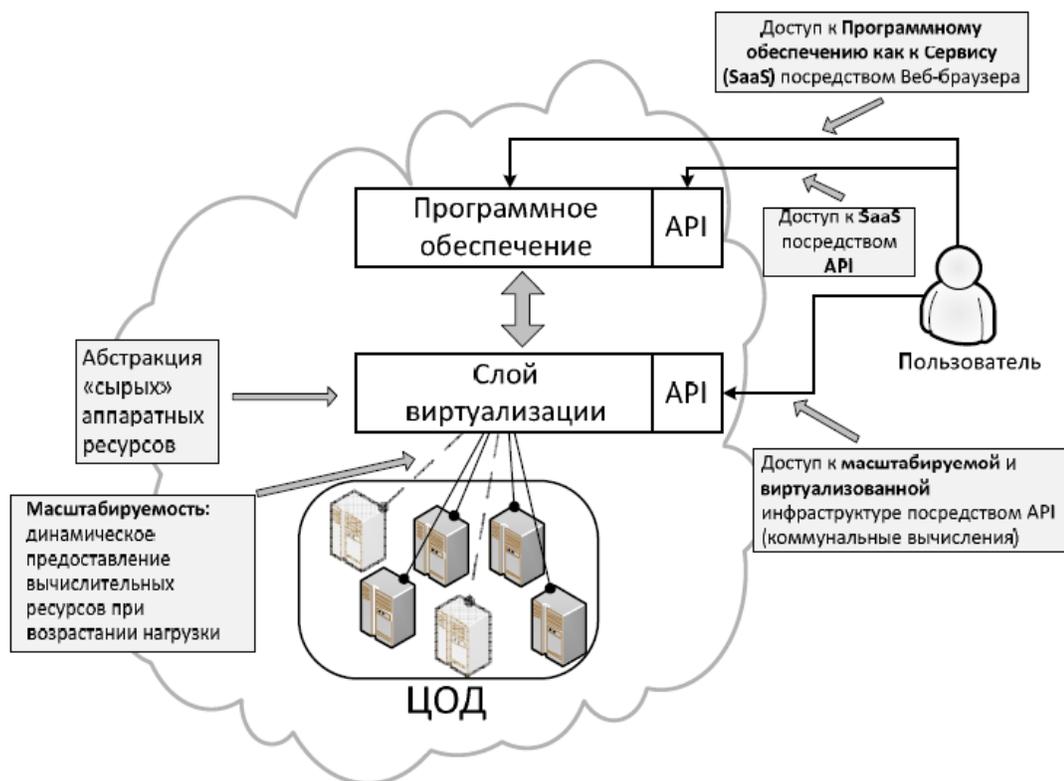


Рисунок 1 - Основные особенности облачных вычислений

В соответствии со всем вышесказанным, можно выделить следующие

основные черты облачных вычислений:

облачные вычисления представляют собой новую парадигму предоставления вычислительных ресурсов;

базовые инфраструктурные ресурсы (аппаратные ресурсы, системы хранения данных, системное ПО) и приложения предоставляются в виде сервисов. Данные сервисы могут предоставляться независимым поставщиком для внешних пользователей по принципу «оплата по мере использования»;

основными особенностями облачных вычислений являются виртуализация и динамическая масштабируемость;

облачные сервисы могут предоставляться конечному пользователю через веб-браузер или посредством определенного программного интерфейса.

ПРОМЕЖУТОЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В РАСПРЕДЕЛЁННЫХ СИСТЕМАХ

Распределенная вычислительная система представляет собой программно аппаратный комплекс, ориентированный на решение определенных задач. С одной стороны, каждый вычислительный узел является автономным элементом. С другой стороны, программная составляющая РВС должна обеспечивать пользователям видимость работы с единой вычислительной системой. В связи с этим выделяют следующие важные характеристики РВС:

возможность работы с различными типами устройств:

с различными поставщиками устройств;

с различными операционными системами,

с различными аппаратными платформами.

Вычислительные среды, состоящие из множества вычислительных систем на базе разных программно-аппаратных платформ, называются гетерогенными;

возможность простого расширения и масштабирования;

перманентная (постоянная) доступность ресурсов (даже если некоторые элементы РВС некоторое время могут находиться вне доступа);

сокрытие особенностей коммуникации от пользователей.

Для обеспечения работы гетерогенного оборудования РВС в виде единого целого, стек программного обеспечения (ПО) обычно разбивают на два слоя. На верхнем слое располагаются распределенные приложения, отвечающие за решение определенных прикладных задач средствами РВС. Их функциональные возможности базируются на нижнем слое - промежуточном программном обеспечении (ППО). ППО взаимодействует с системным ПО и сетевым уровнем, для обеспечения прозрачности работы приложений в РВС (см. рисунок 2). Для того чтобы РВС могла быть представлена пользователю как единая система, применяют следующие типы прозрачности в РВС:

прозрачный доступ к ресурсам - от пользователей должна быть скрыта разница в представлении данных и в способах доступа к ресурсам РВС;

прозрачное местоположение ресурсов - место физического расположения требуемого ресурса должно быть несущественно для пользователя;

репликация - сокрытие от пользователя того, что в реальности существует более одной копии используемых ресурсов;

параллельный доступ - возможность совместного (одновременного) использования одного и того же ресурса различными пользователями независимо друг от друга. При этом факт совместного использования ресурса должен оставаться скрытым от пользователя;

прозрачность отказов - отказ (отключение) каких-либо ресурсов РВС не должен оказывать влияния на работу пользователя и его приложения.

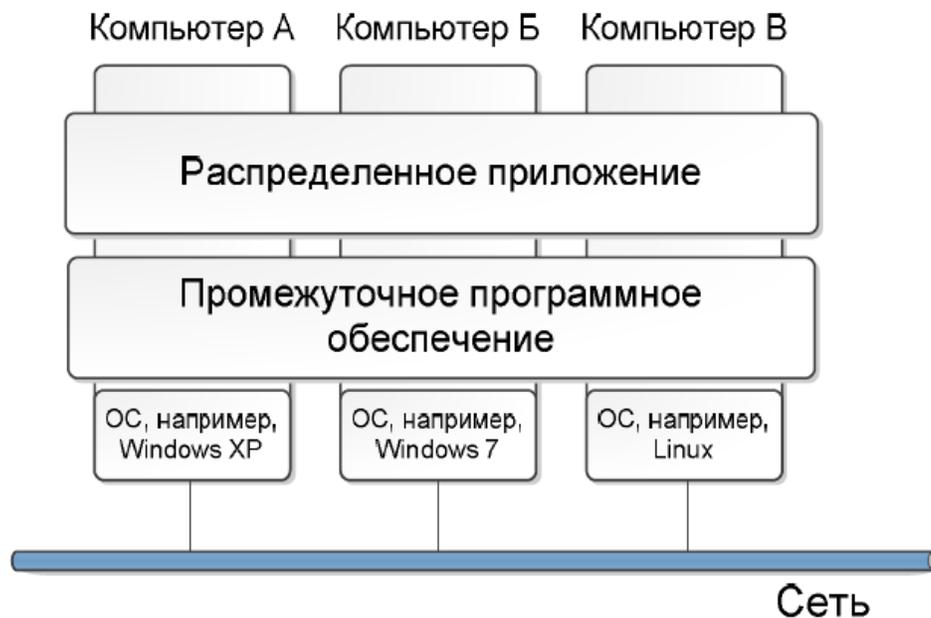


Рисунок 2 - Слои программного обеспечения в РВС

.1 Стандарты Грид

Ключевым моментом в разработке грид приложений является стандартизация, позволяющая организовать поиск, использование, размещение и мониторинг различных компонентов, составляющих единую виртуальную систему, даже если они предоставляются различными поставщиками услуг или управляются различными организациями. К началу 2001 года в различных проектах были представлены различные методы реализации грид-вычислений. Но все они сходились в одном: для гибкого, прозрачного и надежного предоставления доступа к вычислительным ресурсам была предложена сервисно-ориентированная модель.

В 2001 году в качестве базы для создания стандарта архитектуры грид приложений была выбрана технология веб-сервисов. Разработанный стандарт архитектуры грид получил название OGSA (Open Grid Services Architecture - Открытая архитектура грид-сервисов). Он основывается на понятии грид-

сервиса. Грид-сервисом называется сервис, поддерживающий предоставление полной информации о текущем состоянии (потенциально временного) экземпляра сервиса, а также поддерживающий возможность надежного и безопасного исполнения, управления временем жизни, рассылки уведомлений об изменении состояния экземпляра сервиса, управления политикой доступа к ресурсам, управления сертификатами доступа и виртуализации. Грид-сервис поддерживает следующие стандартные интерфейсы.

- . Поиск. Грид приложениям необходимы механизмы для поиска доступных сервисов и определения их характеристик.

- . Динамическое создание сервисов. Возможность динамического создания и управления сервисами - это один из базовых принципов OGSA, требующий наличия сервисов создания новых сервисов.

- . Управление временем жизни. Распределенная система должна обеспечивать возможность уничтожения экземпляра грид-сервиса.

- . Уведомление. Для обеспечения работы грид приложения наборы грид сервисов должны иметь возможность асинхронно уведомлять друг друга о изменениях в их состоянии.

Совместные усилия сообщества грид и организаций по разработке стандартов веб-сервисов привело к определению стандартов, соответствующих требованиям грид. В предложенном стандарте WSRF (Web Service Resource Framework) специфицированы универсальные механизмы для определения, просмотра и управления состоянием удаленного ресурса, что является критически-важным с точки зрения грид.

В настоящее время, существуют две системы, обеспечивающие инфраструктуру разработки грид-систем в соответствии со стандартами OGSA, реализованными посредством WSRF: Globus Toolkit и UNICORE.

3.2 Система Globus

- это проект по разработке и предоставлению инфраструктуры для грид-вычислений. Становление данного проекта приходится на 1997 год, а его разработка продолжается и сегодня.

Цель его создания - предоставление возможности приложениям работать с распределенными разнородными вычислительными ресурсами как с единой виртуальной машиной. Основная направленность данного проекта - вычислительные грид-системы. Под вычислительной грид-системой подразумевается инфраструктура аппаратных и программных ресурсов, реализующая надежный и полномасштабный доступ к высокопроизводительным вычислительным системам, независимо от географического расположения пользователей или ресурсов.

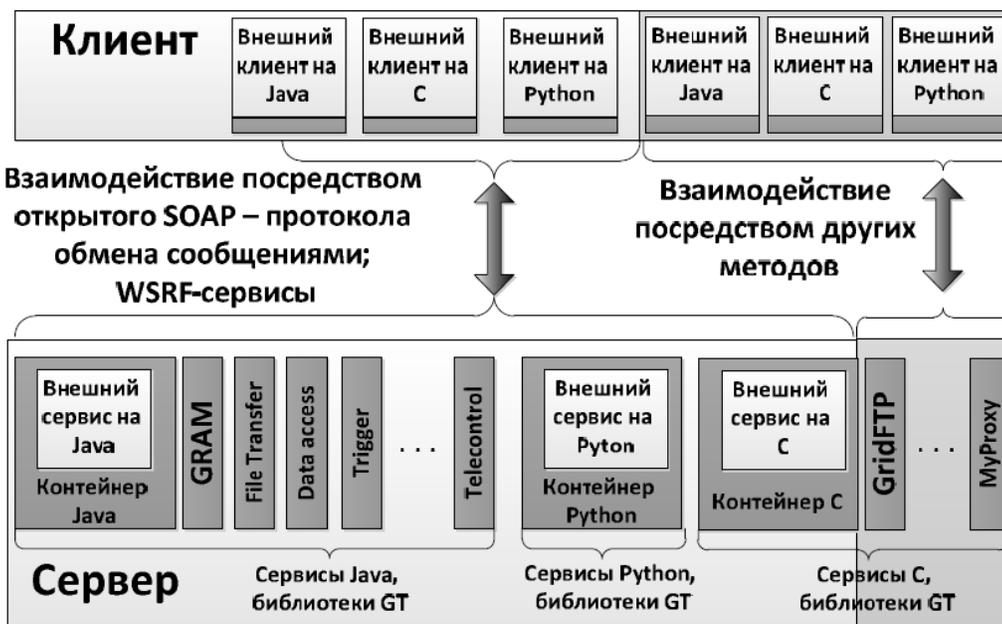


Рисунок 3 - Общая схема взаимодействия компонентов Globus Toolkit 4.0

Базовым элементом системы Globus выступает Globus Toolkit (инструментарий Globus), описывающий базовые сервисы и возможности,

необходимые для создания вычислительных грид-систем. Система Globus предоставляет высокоуровневым приложениям доступ к сервисам, каждый из которых приложение или разработчик может использовать для достижения собственных целей. Такой метод работы может быть реализован только при высокой степени изолированности отдельных сервисов и четко определенном программном интерфейсе каждого предоставляемого сервиса.

Рассмотрим базовые сервисы, предоставляемые системой Globus на сегодняшний день (см. рисунок 3).

- . Протокол GRAM (“Globus Toolkit Resource Allocation Manager” - Менеджер Распределения Ресурсов Globus Toolkit) используется для распределения вычислительных ресурсов и для контроля вычислений, с использованием данных ресурсов.

- . Расширенная версия протокола передачи файлов GridFTP используется для организации доступа к данным, включая вопросы безопасности и параллелизма высокоскоростной передачи данных.

- . Контейнеры для пользовательских сервисов, поддерживающие аутентификацию, управление состоянием, поиск и т.п. обеспечивающие поддержку стандартов WSRF, WS-Security, WS-Notification.

- . Сервисы аутентификации и безопасности соединений GSI (“Grid Security Infrastructure” - Инфраструктура Безопасности Грид).

- . Распределенный доступ к информации о структуре и состоянии системы распределенных вычислений.

- . Удаленный доступ к данным посредством последовательных и параллельных интерфейсов.

- . Создание, кэширование и поиск исполняемых ресурсов.

- . Библиотеки, для обеспечения взаимодействия сторонних приложений с GTK 4.0 и/или пользовательскими сервисами.

3.3 Система UNICORE

Проект UNICORE (Uniform Interface to Computing Resources - единый интерфейс к вычислительным ресурсам) зародился в 1997 году, и к настоящему моменту представляет собой комплексное решение, ориентированное на обеспечение прозрачного безопасного доступа к ресурсам грид.

Архитектура UNICORE 6 формируется из клиентского, сервисного и системного слоев (см. рисунок 4). Верхним слоем в архитектуре является клиентский слой. В нем располагаются различные клиенты, обеспечивающие взаимодействие пользователей с грид средой:

UCC (Unicore Command Line Client - клиент командной строки для UNICORE): клиент, обеспечивающий интерфейс командной строки для постановки задач и получения результатов;

URC (Unicore Rich Client - многофункциональный клиент UNICORE): клиент, основанный на базе интерфейса среды Eclipse, предоставляет в графическом виде полный набор всех функциональных возможностей системы UNICORE;

HiLA (High Level API for Grid Applications - высокоуровневый программный интерфейс для приложений грид): обеспечивает разработку клиентов к системе UNICORE;

Порталы: доступ пользователей к грид-ресурсам через интернет, посредством интеграции UNICORE и систем интернет-порталов.

Промежуточный сервисный слой содержит все сервисы и компоненты системы UNICORE, основанные на стандартах WSRF и SOAP. Шлюз - это компонент, обеспечивающий доступ к узлу UNICORE посредством аутентификации всех входящих сообщений. Компонент XNJS обеспечивает управление задачами и исполнение ядра UNICORE 6. Регистр сервисов обеспечивает регистрацию и поиск ресурсов, доступных в грид-среде. Также, на

уровне сервисного слоя обеспечивается поддержка безопасных соединений, авторизации и аутентификации пользователей.

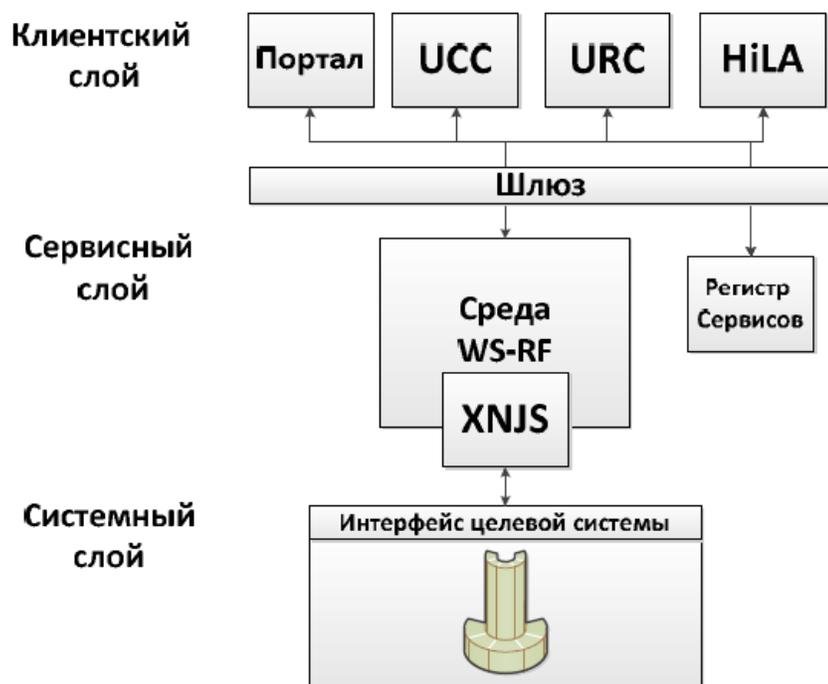


Рисунок 4 - Архитектура UNICORE 6

В основании архитектуры UNICORE лежит системный слой. Интерфейс целевой системы (TSI - Target System Interface) обеспечивает взаимодействие между UNICORE и отдельным ресурсом грид-сети. Он обеспечивает трансляцию команд, поступающих из грид-среды локальной системе.

Основным достоинством использования системы UNICORE 6 для разработки распределенных вычислительных систем можно считать наличие богатого арсенала различных клиентов, обеспечивающих взаимодействие пользователя с ресурсами вычислительной сети, а также развитых средств обеспечения безопасности при разработке грид-приложений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Радченко Г.И. Распределённые вычислительные системы. Учебное пособие. Челябинск: Фотохудожник, 2012. 184с.
2. Клементьев И.П., Устинов В.А. Введение в Облачные вычисления. Екатеринбург: УрГУ, 2009. 233с.
3. В. Furht, A. Escalante (eds.), Handbook of Cloud Computing, Springer Science+Business Media, LLC 2010.
4. <http://ru.wikipedia.org/wiki/Облачные_вычисления>
5. <<https://ru.wikipedia.org/wiki/Грид>>