

Создание распределенной инфраструктуры для суперкомпьютерных приложений

Савин Геннадий Иванович
Корнеев Владимир Викторович
Шабанов Борис Михайлович
Телегин Павел Николаевич
Семенов Дмитрий Викторович
Киселев Андрей Валентинович
Кузнецов Алексей Владимирович
Вдовикин Олег Игоревич
Аладышев Олег Сергеевич
Овсянников Алексей Павлович

Основы Грид-технологии

Грид-технологии представляют собой программные решения для построения Грид-систем [1]. Характерной особенностью этих систем служит предоставление ресурсов по запросам пользователей без указания точного местоположения требуемых ресурсов (например, MAC-, IP-адресов). Воспринимаемый на уровне пользователей Грид-системы как единый вычислительный ресурс, этот ресурс физически состоит из распределенных в сетевой среде вычислительных ресурсов совокупности вычислительных установок: вычислительных систем, возможно различной архитектуры, персональных компьютеров, рабочих станций. Каждая из вычислительных установок, ресурсы которых доступны пользователям Грид-системы, принадлежит своим владельцам, возможно различным для разных установок. Появлению Грид-технологий предшествовало, с одной стороны, создание высокоскоростных компьютерных сетей, использующих открытые, универсальные протоколы и интерфейсы, а, с другой стороны, решение фундаментальных задач компьютерной безопасности таких, как аутентификация, авторизация ресурсов и контроль доступа к ресурсам.

Применение Грид-систем, в условиях непредсказуемого спроса на ресурсы со стороны пользователей, позволяет координировать потребление свободных ресурсов при отсутствии непосредственного управления их использованием со стороны владельцев этих ресурсов. Тем самым, Грид-технологии предоставляют альтернативу владению ресурсами для пользователей, имеющих большие ресурсные потребности, возникающие в непересекающихся временных окнах. Следует отметить, что объединение вычислительных установок в Грид-систему позволяет увеличить пропускную способность совокупности ресурсов за счет исключения простоя одних ресурсов при наличии перегруженности других ресурсов, а также увеличить отказоустойчивость и обеспечить требуемое качество обслуживания, характеризующееся такими параметрами, как время отклика, гарантированная пропускная способность, доступность.

Может возникнуть представление, что в Грид-систему рационально объединять только «недогруженные» ВС. Но это не так. Даже при достаточно высокой загрузке отдельных вычислительных систем (ВС) имеет смысл объединение их в Грид-систему. Во-первых, кроме увеличения пропускной способности, целью создания Грид-систем является также предоставление вычислительного ресурса, превышающего ресурсы возможности отдельных ВС, в том числе разных владельцев, для исполнения параллельных программ масштабных вычислительных задач. И, во-вторых, ВС разных владельцев имеют, даже при одной и той же аппаратной платформе, проблемную ориентацию установленного на них прикладного программного обеспечения, что расширяет возможности эффективной работы пользователей в рамках Грид-системы.

В настоящей работе представлены результаты создания Российской инфраструктуры для суперкомпьютерных приложений (РИСП) в МСЦ РАН. РИСП представляет собой Грид-систему для организации высокопроизводительных распределенных вычислений в сетевой среде, что обусловлено, с одной стороны, наличием необходимой вычислительной и коммуникационной инфраструктуры, а, с другой стороны, устойчивой тенденцией роста числа пользователей, требующих для решения своих задач значительных объемов вычислений.

Основы подхода к созданию РИСП

Грид-система, предназначенная для выполнения сложных научно-технических расчетов в МСЦ РАН, должна, во-первых, перераспределять задания пользователей между территориально разнесенными вычислительными системами, входящим в Грид-систему, и, во-вторых, согласованно выделять вычислительные модули (ВМ), возможно разных вычислительных систем, для исполнения параллельных программ. Будем называть Грид-систему с такими свойствами сетевой средой распределенных вычислений (ССРВ) [2, 3]. Структура ССРВ показана на рис. 1.

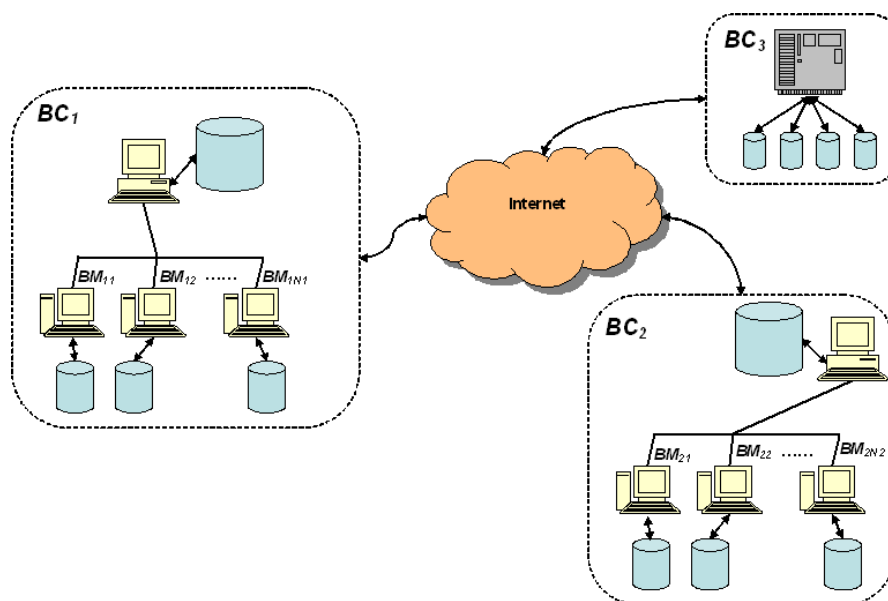


Рис.1. Структура ССРВ

Каждая ВС имеет управляющую машину (УМ), в качестве которой может выступать сама ВС, как, например ВС₃, показанная на рис.1. УМ всех ВС объединены сетью, которая позволяет передавать программы и данные между ними и выполнять удаленный запуск программ.

Вычислительные модули отдельных ВС или даже нескольких разных ВС могут быть объединены одной или несколькими высокоскоростными сетями, например Myrinet, Infiniband, Quadrics [4], для выполнения обменов данными при исполнении параллельных программ. При этом разные совокупности ВМ могут иметь разные наборы используемых сетей. Предполагается, что применяемые библиотеки для организации параллельных вычислений, например варианты реализации MPI, позволяют при запуске параллельной программы на выбранной подсистеме ВМ выполнять межмодульные обмены и операции синхронизации настолько эффективно, насколько позволяют доступные ВМ подсистемы сетевые ресурсы.

Каждая ВС, входящая в состав ССРВ, имеет собственную локальную систему пакетной обработки (СПО) заданий. Известны более двадцати СПО, из которых наиболее популярны свободно распространяемые PBS (Portable Batch System) и Condor, а также коммерческие LoadLeveler, PBS Professional и LSF (Load Sharing Facility). К значимым отечественным разработкам в этой области следует отнести СУПЗ МВС-1000, созданную ИПМ им. М.В. Келдыша РАН в кооперации с ИММ УрО РАН и НИИ «Квант», а также пакетную систему X-COM, разработанную в НИВЦ МГУ.

Применение СПО в ВС позволяет перейти от работы с индивидуальными компьютерами, распределенными в сети, к работе с единым пулом вычислительных модулей в режиме пакетной обработки заданий, в том числе параллельных. Все СПО имеют богатый настраиваемый набор средств управления процессом обработки заданий. Пользователь ВС может помещать задания в общую очередь СПО, используя единый интерфейс для запуска, модификации, снятия заданий и получения информации о заданиях. СПО автоматически распределяет задания по ВМ с учетом их загрузки, освобождает занятые ресурсы после завершения задания, доставляет результаты пользователю, а также обеспечивает разграничение прав пользователей и защиту вычислительных ресурсов от несанкционированного доступа.

Таким образом, для создания ССРВ без вмешательства в существующие программные средства и режим работы ВС разных владельцев необходимо построить дополнительный уровень программного обеспечения (ПО), реализующий функции системы управления (СУ) Грид-системы. Эта СУ ССРВ должна предоставлять возможность постановки заданий пользователей в общую очередь ССРВ и запуск заданий на свободных ресурсах любых одной или нескольких ВС, мониторинга ССРВ,

обеспечения отказоустойчивых вычислений. При этом каждая ВС получает задания как непосредственно от пользователей данной ВС, так и посредством ПО СУ ССРВ от пользователей других ВС. СУ переносит, в соответствии с принятыми в ней алгоритмами управления, задания из очереди ССРВ в очередь локальных СПО этих ВС.

Архитектура системы управления ССРВ

Построение системы управления ССРВ рационально базировать на программном пакете Globus Toolkit версии 2 (GTK2) [5], позиционированном разработчиками как программное обеспечение промежуточного уровня (middleware). Пакет GTK2 устанавливается на УМ всех ВС, входящих в ССРВ, и использует для взаимодействия между ВС ССРВ сетевые адреса УМ этих ВС. Средства пакета GTK2, обеспечивают запуск заданий по указываемым пользователем IP адресам и обмен файлами с указанием полных путей. Например, группа пользовательских команд для управления заданиями включает команды **globus-job-run**, **globus-job-submit**, соответствующие разным вариантам запуска задания **script** на удаленном узле с адресом **remote.host.ru**.

- **globus-job-run remote.host.ru –stage script**
- **globus-job-submit remote.host.ru script**

globus-job-submit осуществляет запуск исполняемого файла в пакетном режиме — с освобождением терминала. Это может быть, например, скрипт или файл типа .EXE. С помощью опций может быть заказана доставка на исполняющий компьютер стандартных файлов stdout, stderr, stdin с любых узлов ВС, в частности с узла, на котором выдана команда submit.

Во всех случаях в ответ на команду запуска возвращается идентификатор, который используется в командах определения статуса, получения вывода, завершения задания.

- **globus-job-submit remote.host.ru script**
https://remote.host.ru:1670/124540/783255567/

Команда **globus-job-status** позволяет узнать состояние задания, единственный параметр команды - это идентификатор задания, возвращаемый после успешного выполнения команды **globus-job-submit**.

- **globus-job-status**
- **https://remote.host.ru:1670/124540/783255567/**

Для получение стандартного вывода можно воспользоваться командой **globus-job-get-output**, входящей в состав группы пользовательских утилит для управления заданиями.

- **globus-job-get-output**
- **<https://remote.host.ru:1670/124540/783255567/>**

Для снятия задания с выполнения необходимо воспользоваться командой ***globus-job-cancel***, в качестве параметра передаётся идентификатор задания, возвращаемый после успешного выполнения команды ***globus-job-submit***.

- **globus-job-cancel**
- **[https:// remote.host.ru:1670/124540/783255567/](https://remote.host.ru:1670/124540/783255567/)**

СУ ССРВ строится добавлением уровня ПО, ведущего очередь пользовательских заданий, поступающих в ССРВ, и автоматически, без участия пользователей, определяющего сетевые адреса ресурсов, выделяемых заданиям из очереди ССРВ.

Использование GTK2 предоставляет разработчику СУ ССРВ высокоуровневый интерфейс, скрывающий детали взаимодействия различных компонентов Грид-системы нижнего уровня: используемые сетевые протоколы, архитектуру ВС, входящих в состав Грид-системы, тип хранилищ данных и т.д. Набор инструментов Globus Toolkit состоит из набора модулей. Для каждого модуля определён интерфейс, используя возможности которого высокоуровневое приложение сможет вызывать функции этого модуля. Реализованы модули с использованием операций низкого уровня, соответствующих той или иной платформе. Такой подход предоставляет следующие выгоды для создания СУ ССРВ:

- разработчики программных средств СУ должны использовать только инструментарий, предоставляемый пакетом Globus Toolkit, вместо того, что бы изучать весь набор архитектуру- и протоколовзависимых интерфейсов;
- программные средства СУ, написанные с использованием инструментария Globus Toolkit, являются переносимыми: при появлении новых архитектур ВС или новых технологий достаточно реализовать инструментарий Globus Toolkit v.2 для данной архитектуры ВС или технологии;
- разработчик программных средств СУ не должен заботиться об оптимизации работы механизмов, используемых на нижнем уровне Грид-системы, поскольку Globus Toolkit для каждой операции выбирает самый эффективный метод реализации;
- внесение изменений в инструментарий пакета Globus Toolkit с целью оптимизации, исправления ошибок, добавления новых возможностей и т.д. не требует внесения изменений в СУ.

Ключевым элементом Globus Toolkit, обеспечивающим безопасный доступ к любому ресурсу, служит модуль управления ресурсами GRAM (Globus Resource Allocation Manager). Будем называть клиентом по отношению к ресурсу любой процесс, требующий доступа к рассматриваемому ресурсу посредством обращения к модулю GRAM.

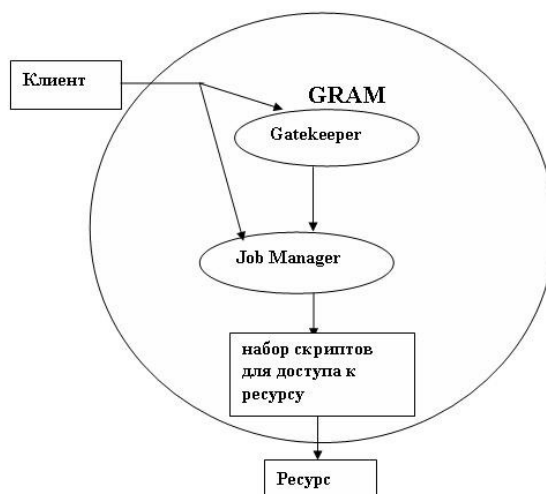


Рис. 2. Доступ «клиента» к «ресурсу» посредством GRAM

GRAM состоит из двух компонентов Gatekeeper и Job Manager. Процесс Gatekeeper запускается с правами суперпользователя. При обращении к процессу Gatekeeper, выполняется взаимная аутентификация клиента и вычислительного ресурса с использованием сертификатов стандарта X-509, что обеспечивает надлежащее разграничение доступа к ресурсу и исключение возможности несанкционированного доступа. Взаимная аутентификация клиента и процесса Gatekeeper осуществляется в соответствии с протоколом SSL. Собственно в этом заключается существенная особенность Globus Toolkit: каждый ресурс имеет свой сертификат и список сертификатов клиентов, которым разрешен доступ к рассматриваемому ресурсу. Сертификат считается достоверным (а аутентификация успешной), если принимающей стороне известен подписавший его центр сертификации, и сертификат подписан правильно, с учетом срока его действия.

Устанавливаемая по умолчанию конфигурация GRAM осуществляет доступ к ресурсу путем порождения нового процесса Unix. Предусмотрена возможность расширения функциональности процесса JobManager для реализации требуемого механизма доступа к ресурсу. Эти расширения реализуются в виде набора скриптов. Создатели GTK2 предусмотрели наборы скриптов для ряда локальных СПО. При запуске задания служба GRAM определяет используемую систему планирования и в дальнейшем использует соответствующий скрипт.

Однако возможность создать набор скриптов широко используется разработчиками Грид-систем, например, для контроля выполнения политик доступа к ресурсу, формулируемых владельцем ресурса и виртуальными организациями, к которым принадлежит пользователь [6], а также биллинга ресурсов и применения экономических моделей предоставления ресурсов на базе соглашений между поставщиками и потребителями ресурсов [7] и достижения других целей.

В случае если в качестве ресурса выступает локальная СПО, то при успешной аутентификации происходит авторизация пользователя. Основная схема авторизации в Globus заключается в том, что пользователь авторизуется в ВС, если в файле авторизации (*grid-mapfile*) извлекаемому из сертификата Gatekeeper идентификатору пользователя Грид-системы поставлено в соответствие имя локального пользователя данной ВС. Если идентификатор пользователя Грид-системы в файле авторизации отсутствует, то пользователь признается не авторизованным, и запрос отвергается.

В результате авторизации определяется локальная учетная запись, от имени которой пользователь Грид-системы получает доступ к ресурсу. После авторизации пользователя, процесс Gatekeeper ограничивает свои права правами этого (непривилегированного) пользователя и запускает экземпляр процесса Job Manager Instance (JMI), осуществляющий дальнейшее управление заданием все время его выполнения, реализуя запросы на остановку задания, получения статуса задания и другие. JMI получает от клиента параметры запуска задания (исполняемый файл, входные, выходные файлы, аргументы) и передает его на выполнение. Для каждого задания запускается собственный экземпляр JMI.

Локальные СПО всех ВС выступают в качестве ресурса, доступ к которому контролируется соответствующими экземплярами GRAM. Для запуска задания с использованием команд *globus-job-run* и *globus-job-submit* в локальной СПО вычислительной системы, адрес которой указан в параметрах исполняемой команды, применяются специальные расширения службы GRAM пакета Globus Toolkit, позволяющие запускать, удалять и контролировать выполнение задания в локальной системе планирования. В пакет Globus Toolkit включена поддержка нескольких распространенных систем планирования заданий: LSF, PBS, PRUN, CONDOR и др. В ходе создания СУ ССРВ был добавлен скрипт, выполняющий запуск заданий под управлением СУПЗ МВС, широко используемой на отечественных суперкомпьютерах семейства МВС-1000 [8]. Данная СПО, в том числе, обеспечивает возможность синхронного запуска ветвей одной параллельной программы на вычислительных модулях разных ВС ССРВ.

Интерфейс между ПО СУ ССРВ и средствами Globus Toolkit v.2 реализуется посредством API для разработки собственных программных

средств, использующих возможности модуля управления ресурсами GRAM (Globus Resource Allocation Manager).

Безопасность ССРВ

Реализация взаимной аутентификации сторон при доступе к ресурсу входит составной частью системы безопасности Globus Toolkit - Globus Security Infrastructure (GSI), созданной на основе стандарта Generic Security Services (GSS). Этот стандарт определяет API, используемый во многих системах защиты информации для взаимной аутентификации клиента с сервером при использовании Kerberos и криптографии с открытым ключом. Система безопасности ССРВ характеризуется следующим:

- точкой доступа пользователя к ресурсам ССРВ может являться УМ любой ВС, входящей в ее состав, для аутентификации в ССРВ пользователь должен иметь в точке доступа свой закрытый ключ и сертификат стандарта X.509, подписанный любым доверенным центром сертификации (ЦС) этой ВС;
- аутентификация пользователей и процессов осуществляется по протоколу SSL с использованием подписанного ЦС сертификата X.509, включающего уникальный идентификатор предъявителя и его открытый ключ;
- в ССРВ может существовать несколько центров сертификации, каждый из которых подписывает сертификаты пользователей для своего сегмента ССРВ, головной центр сертификации ССРВ подписывает сертификаты центров сертификации отдельных сегментов ССРВ;
- при регистрации пользователя в ССРВ создается процесс-посредник (ргоху), который обладает подмножеством прав пользователя и имеет сертификат с ограниченным сроком действия (ргоху-сертификат), который и используется при последующих операциях аутентификации, избавляя пользователя от необходимости вводить пароль для доступа к ресурсам различных ВС;
- при запросе пользователем ресурсов каждой ВС осуществляется авторизация пользователя в этой ВС путем отображения глобального идентификатора пользователя в локальную учетную запись: пользователь ССРВ может выполнять свои задания на тех ВС из состава ССРВ, на которых он зарегистрирован.

Для получения пользователем сертификата он должен сгенерировать пару ключей (открытый и закрытый). Закрытый ключ сохраняется у пользователя, для чего хранится в отдельном файле и для повышения секретности может быть дополнительно зашифрован с помощью пароля, который требуется вводить вручную при каждом предъявлении сертификата.

Открытый ключ включается в заявку на сертификат (неподписанный сертификат) и пересылается выбранному центру сертификации. Центр

сертификации подписывает (если считает это возможным) сертификат и пересылает его обратно с соблюдением правил инфраструктуры открытых ключей, гарантирующих отсутствие возможности подмены открытых ключей. Сертификат имеет достаточно продолжительный срок действия, устанавливаемый администратором центра сертификации.

При любом безопасном контакте (например, при запуске задания или при получении удаленного доступа к файлу) обе стороны (или только одна, если взаимная аутентификация не требуется) обязаны предъявить свои сертификаты и доказать, что они обладают соответствующими закрытыми ключами. Сертификат считается достоверным (а аутентификация успешной), если принимающей стороне известен подписавший его ЦС, и он подписан правильно. Для передачи секретной информации каждая сторона контакта шифрует ее с помощью открытого ключа, извлеченного из сертификата другой стороны (шифрование в системе Globus используется опционально).

Как отмечалось ранее, при осуществлении взаимодействий от имени пользователя, ргоху-процесс предъявляет прокси-сертификат, подписанный владельцем сертификата, который включает, в частности, исходный сертификат, что позволяет убедиться в его достоверности. Прокси-сертификаты необходимы при делегировании прав владельца сертификата для выполнения некоторых действий от его имени, с тем, чтобы сохранить в секрете (не передавать по сети) закрытый ключ пользователя. При делегировании можно ограничить круг полномочий прокси-сертификата, запретив использовать его для запуска процессов. Владелец прокси-сертификата в свою очередь может делегировать права (подписать прокси-сертификат следующего уровня) и т.д.

Прокси-сертификат удобнее в использовании, но обладает меньшей секретностью, поскольку его закрытый ключ хранится вместе с ним в одном файле и не защищен паролем. Поэтому прокси-сертификат выписывается, как правило, на ограниченный срок.

Предъявление прокси-сертификата сводится к помещению его в файл с предопределенным именем или к занесению имени файла сертификата в переменную среды X509_USER_PROXY.

Администратор ВС может управлять набором центров сертификации, которым доверяет подчиненный ему вычислительный ресурс. Список известных системе ЦС, необходимый для удостоверения подлинности сертификатов, хранится в файле с предопределенным именем.

Администратор ССРВ имеет возможность создать собственный центр сертификации, используя дополнительный пакет SimpleCA, входящий в Globus Toolkit 2.2. При установке центра сертификации также создается пара ключей и сертификат. Основным отличием сертификата центра сертификации от сертификатов других субъектов является то, что он подписан собственным закрытым ключом. Закрытый ключ центра сертификации шифруется на некотором пароле. Пользователь, обладающий

этим паролем, может подписывать сертификаты направленные данному центру сертификации.

Все вышеописанные процедуры поддерживаются реализованным в Globus интерфейсом GSS API, который используют все программы, нуждающиеся в аутентификации. Для получения сертификата и ручного выписывания прокси-сертификата имеются интерактивные утилиты.

Текущее состояние РИСП

С целью развития вычислительных технологий и оснащения современными ВС научных организаций РАН существовавшая как единая ВС МВС-15000ВМ была разделена на 5 частей. Четыре части ВС МВС-15000ВМ были перевезены в центры РАН в различных городах России. Каждая из частей МВС-15000ВМ стала представлять собой ВС, состоящую из файлового сервера, УМ и решающего поля, находящийся под управлением собственной СПО СУПЗ.

Задача по созданию и эффективному использованию ССРВ на базе образованных из МВС-15000ВМ вычислительных систем упрощается тем, что все ВС, которые включаются в состав ССРВ, имеют одинаковую архитектуру, одинаковый набор аппаратного и программного обеспечения, так как являются частями одной и той же ВС. Тем самым достигается полная совместимость на уровне исполняемых модулей, т.е. параллельная программа, собранная и отлаженная на одной из ВС, будет запускаться и функционировать на любой другой ВС из состава ССРВ.

На рис. 3 приведена архитектура системы МВС-15000ВМ.

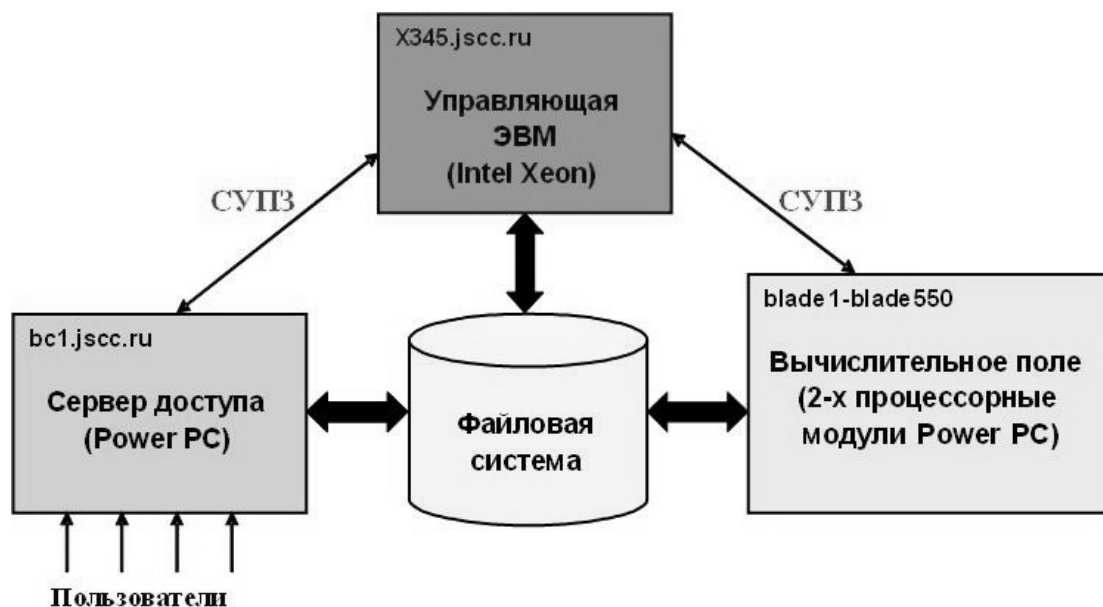


Рис. 3. Архитектура системы МВС-15000ВМ

В МВС-15000ВМ реализованы специальные решения по обеспечению безопасности, надёжности и отказоустойчивости. Управляющая компонента системы сосредоточена на отдельной управляющей ЭВМ, к которой закрыт

доступ для всех пользователей, кроме администраторов. На этой УМ выполняются серверные службы и серверная часть СУПЗ. Для доступа к системе из числа модулей вычислительного поля выделяется один модуль для выполнения специальных функций - так называемая инструментальная ЭВМ: сервер доступа. На инструментальной ЭВМ устанавливаются различные клиентские приложения и клиентская часть СУПЗ. Зарегистрированные на ВС МВС-15000ВМ пользователи имеют доступ только к этой ЭВМ. Так как инструментальная ЭВМ выделяется из состава решающего поля, то параллельные программы, собранные и отлаженные на ней, будут запускаться и правильно функционировать на всех ВМ решающего поля.

При создании ССРВ из нескольких кластеров, каждый из которых имеет ту же архитектуру, что и МВС 15000 ВМ, на инструментальной ЭВМ каждого кластера, являющейся для пользователей точкой доступа к системе, устанавливается клиентское ПО ССРВ: компоненты, обеспечивающие пользовательский интерфейс, ПО для запуска брокеров заданий и клиентская часть ПО Globus Toolkit.

На УМ всех ВС устанавливается связующее программное обеспечение GTK2, на базе которого функционирует программное обеспечение управления ресурсами и заданиями ССРВ [9]. При этом на УМ каждого кластера устанавливаются: серверная часть ПО Globus Toolkit, система менеджеров и компоненты, обеспечивающие интерфейс с СПО СУПЗ. На рис. 4 показана структура установленного ПО на УМ ВС (x345) и на инструментальной ЭВМ (bc).

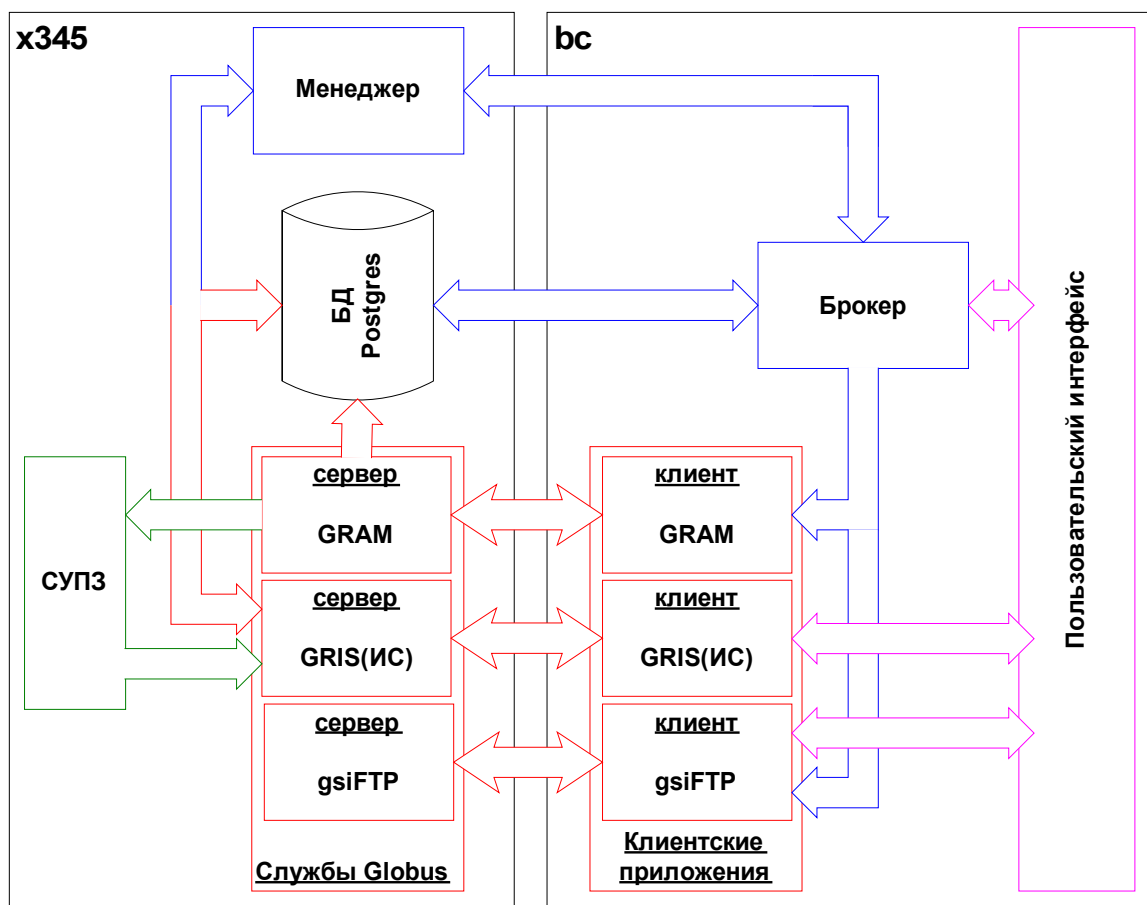


Рис. 4. Структура установленного ПО

Взаимодействие между клиентской и серверной частью СУПЗ происходит по специальному протоколу.

В результате проведённых работ удалось решить следующие задачи:

- развернута инфраструктура ССРВ на базе распределенных вычислительных ресурсов МСЦ РАН, представленных тремя ВС одной и той же архитектуры, размещенными в разных городах и связанных сетью Internet;
- обеспечена доступность для пользовательских заданий всех территориально распределенных вычислительных ресурсов системы МВС-15000ВМ;
- удовлетворены все требования к безопасности, надёжности и отказоустойчивости, предъявляемые к организации вычислений на вычислительных ресурсах МСЦ РАН;

- сохранены привычная модель организации вычислений и привычный пользовательский интерфейс.

Литература

1. I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids. Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.
2. ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ по проекту «СЕТЕВАЯ СРЕДА РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ НА ОСНОВЕ ТЕХНОЛОГИЙ GRID» в рамках программы фундаментальных исследований Президиума РАН № 21 «Разработка фундаментальных основ создания распределенной информационно-вычислительной среды на основе технологий грид»
3. Корнеев В.В., Киселёв А.В., Семёнов Д.В., Сахаров И.Е. Управление метакомпьютерными системами. // М. «Открытые системы» №2, 2005г.
4. Корнеев В.В. Вычислительные системы. М.: «Гелиос АРВ». 2004 г.
5. Globus Toolkit 2.4, <http://www.globus.org>
6. Russell Lock, Ian Sommerville. Grid Security and its use of X.509 Certificates. Department of Computer Science Lancaster University. Funded by EPSRC project studentship associated with the UK EPSRC DIRC project grant GR/N13999
7. C. Anglano et al. – *An accounting system for the DataGrid Project v3.0* - DataGrid-01-TED-0115-3_0 – http://www.to.infn.it/grid/accounting/Current_prop.pdf
8. Руководство программиста. Суперкомпьютер МВС 15000ВМ. www.jscc.ru
9. Руководство программиста. Грид. www.jscc.ru

Сведения об авторах

Савин Геннадий Иванович – академик РАН, директор Межведомственного суперкомпьютерного центра РАН, заведующий кафедрой высокопроизводительных вычислительных систем МФТИ

Шабанов Борис Михайлович – кандидат технических наук, первый заместитель директора Межведомственного суперкомпьютерного центра РАН, заведующий базовой кафедрой высокопроизводительных вычислительных систем МИЭТ

Корнеев Владимир Викторович – доктор технических наук, сотрудник НИИ «КВАНТ»

Телегин Павел Николаевич – кандидат технических наук, заведующий отделом Межведомственного суперкомпьютерного центра РАН. Адрес электронной почты: telegin@jscc.ru

Семенов Дмитрий Викторович - сотрудник НИИ «КВАНТ»

Киселев Андрей Валентинович - сотрудник НИИ «КВАНТ»

Кузнецов Алексей Владимирович - сотрудник НИИ «КВАНТ»

Вдовикин Олег Игоревич - заведующий лабораторией Межведомственного суперкомпьютерного центра РАН

Аладышев Олег Сергеевич - заведующий отделом Межведомственного суперкомпьютерного центра РАН

Овсянников Алексей Павлович - заведующий отделом Межведомственного суперкомпьютерного центра РАН