Архитектура параллельных вычислительных систем

к.т.н., доцент
Костичев Сергей Валентинович

Кластерные и Grid-технологии

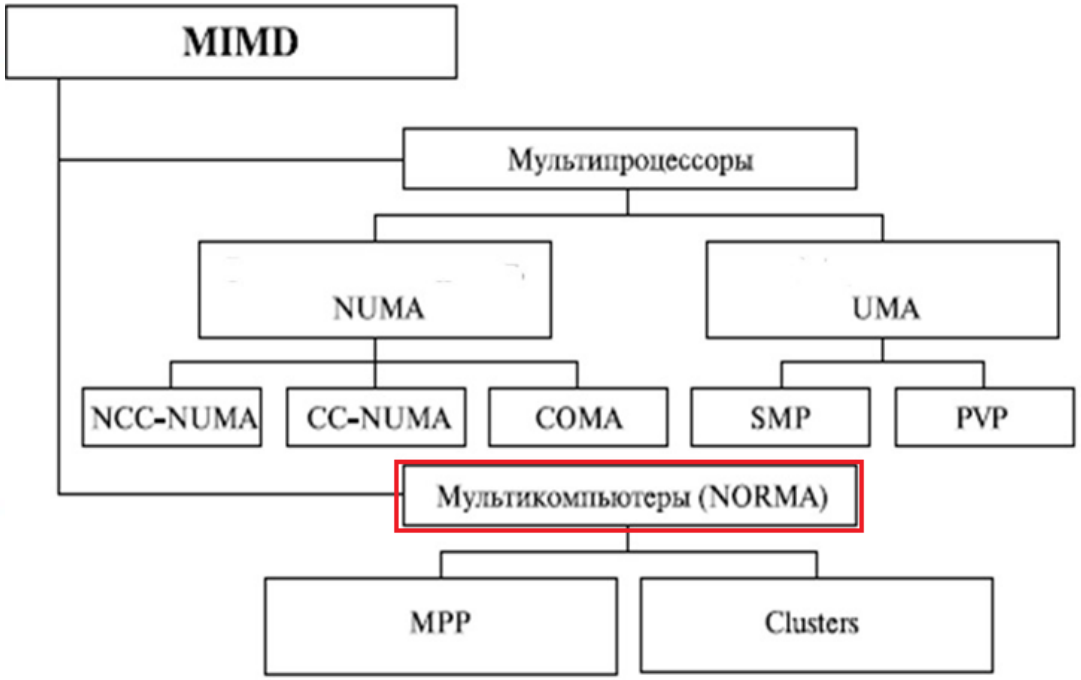
snenv@mail.ru

Учебные вопросы:

1. Вычислительные кластеры
2. Метакомпьютинг
3. Концепция, архитектура и технологии GRID
4. Ресурсы GRID
5. Уровни GRID
6. Инструментальные средства Grid
7. Примеры реализаций GRID

1. Вычислительные кластеры

Мультикомпьютер



- Мультикомпьютер = Многопроцессорная ВС с распределенной памятью
- Каждый процессор обладает собственной (локальной) памятью и способен адресоваться только к ней.
- Доступ к удаленной памяти возможен только путем обмена сообщениями с процессором, которому принадлежит адресуемая память

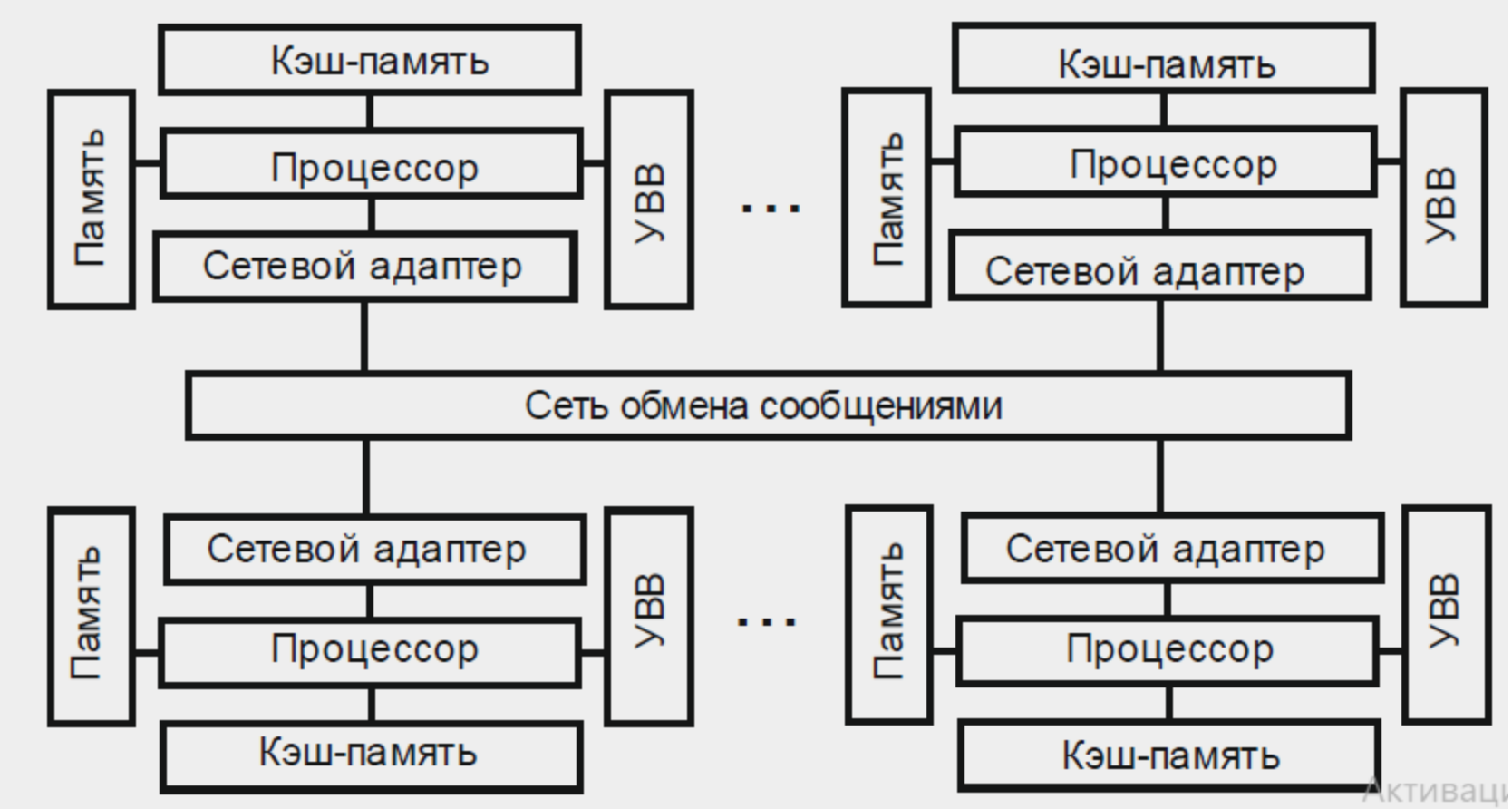
Достоинства:

- при доступе к данным нет конкуренции за шину или коммутаторы;
- нет общей шины ⇒ нет ограничений на число процессоров. Размер системы ограничивает только сеть, объединяющая процессоры;
- нет проблемы когерентности кэш-памяти.

Недостатки:

- сложность обмена информацией между процессорами.
- отсутствие общей памяти снижает скорость межпроцессорного обмена.
- каждый процессор может использовать только ограниченный объем локального банка памяти.
- сложность написания эффективных программ

Системы с массовой параллельной обработкой (MPP)



Обобщенная структура MPP-системы

Главные особенности, по которым ВС причисляют к классу MPP:

- стандартные микропроцессоры;
- физически распределенная память;
- сеть соединений с высокой пропускной способностью и малыми задержками;
- хорошая масштабируемость (до тысяч процессоров);
- асинхронная MIMD-система с пересылкой сообщений;
- программа представляет собой множество процессов, имеющих отдельные адресные пространства.

- Характерная черта MPP-систем — **наличие единственного управляющего устройства** (процессора), распределяющего задания между множеством подчиненных ему устройств, чаще всего одинаковых (взаимозаменяемых), принадлежащих одному или нескольким классам.

- В некотором приближении имеет смысл считать, что **на ЦП выполняется ядро ОС (планировщик заданий)**, а на подчиненных ему — приложения.

Кластерные вычислительные системы

- Одно из самых современных направлений в области создания ВС — это кластеризация.

- По производительности и коэффициенту готовности кластеризация представляет собой альтернативу MPP системам.

- **Появление кластерных ВС**:
  - прогресс в области сетевых технологий
  - серийные однокристальные МП высокой мощности

- **Кластер** - несколько компьютеров (узлов), объединяемых при помощи сетевых технологий на базе шинной архитектуры или коммутатора и предстающих перед пользователем как единый информационно-вычислительный ресурс.

- Узлы - рабочие станции, серверы, ПК, однопроцессорная ВМ, ВС типа SMP или MPP. На узле собственная копия ОС.

- Каждый узел в состоянии функционировать самостоятельно и отдельно от кластера.

Достоинства:

- надежность (при сбое одного узла другой берет на себя нагрузку)
- масштабируемость
- дешевизна

Недостаток: дешевизна оборачивается большими накладными расходами на взаимодействие узлов

Согласно Aberdeen Group, **кластер** – многомашинный комплекс, который

- выглядит с точки зрения пользователя как единая система
- обеспечивает высокую надежность (готовность)
- имеет общую файловую систему
- обладает свойством масштабируемости
- гибко перестраивается
- управляется/администрируется как единая система

Изначально перед кластерами ставились две задачи:

- достичь большой вычислительной мощности
- обеспечить повышенную надежность ВС.

1-й коммерческий кластер **(**Компания **DEC,** 1983):
VAX-кластер - слабосвязанная многомашинная система с общей внешней памятью, обеспечивающая единый механизм управления и администрирования.

Свойства VAX-кластера:

- *Разделение ресурсов*. Компьютеры VAX в кластере могут разделять доступ к внешней памяти.

- *Высокая готовность*. Если происходит отказ одного из VAX-компьютеров, задания его пользователей автоматически могут быть перенесены на другой компьютер кластера.

- *Высокая пропускная способность*. Ряд прикладных систем могут пользоваться возможностью параллельного выполнения заданий на нескольких компьютерах кластера.

- *Удобство обслуживания системы*. Общие базы данных могут обслуживаться с единственного места. Прикладные программы могут инсталлироваться только однажды на общих дисках кластера и разделяться между всеми компьютерами кластера.

- *Расширяемость*. Увеличение вычислительной мощности кластера достигается подключением к нему дополнительных VAX-компьютеров.

Работа VAX-кластера определялась **двумя главными компонентами**:

- высокоскоростной механизм связи,
- системное ПО (обеспечивает клиентам прозрачный доступ к системному сервису).

Проект beowulf (1994)

Оказал огромное влияние на развитие кластерных технологий NASA, 16 процессоров Intel 100МГц по 16Mb ОП на каждом, объединены с использованием обычной сети Ethernet (10 мбит\сек).

Основа общего подхода «a la Beowulf» **к построению параллельных кластерных компьютеров,** состоящих из

- одного серверного узла (головного)
- нескольких подчиненных узлов, соединенных посредством стандартной компьютерной сети.

Кластер строится на основе

- стандартных компонент (ПК под LINUX),
- стандартных сетевых адаптеров.
- нет специального программного пакета beowulf.
- есть несколько пакетов ПО, которые пользователи нашли пригодными в рамках кластеров данной концепции: LINUX, MPI, системы управления ресурсами и другие стандартные продукты.

Кластерные вычислительные системы

- В качестве узлов кластеров могут использоваться
  - одинаковые ВС (**гомогенные** кластеры)
  - разные (**гетерогенные** кластеры).

- По своей архитектуре кластерная ВС является **слабо связанной** системой.

- **Преимущества**, достигаемые с помощью кластеризации:
  - абсолютная масштабируемость;
  - наращиваемая масштабируемость;
  - высокий коэффициент готовности;
  - превосходное соотношение цена/производительность

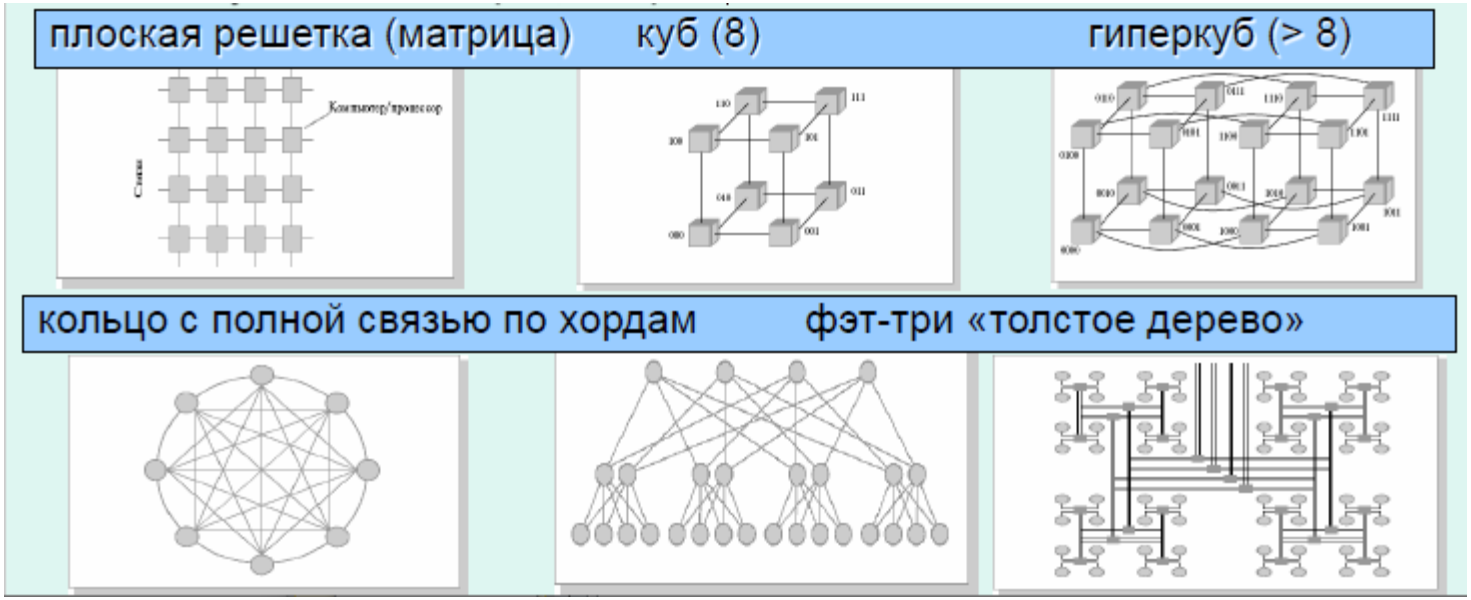
- На уровне аппаратного обеспечения кластер — это совокупность независимых ВС, объединенных сетью.

- При соединении машин в кластер **объединение** производится **при помощи сетевых технологий** на базе шинной архитектуры или коммутатора.

- Неотъемлемая часть кластера — **специализированное ПО**, на которое возлагается задача обеспечения бесперебойной работы

- **Типы кластеров** (предложены Язеком Радаевским и Дугласом Эдлайном)
  - ***Класс I***. Класс машин строится целиком из стандартных деталей, которые продают многие продавцы компьютерных компонент
  - ***Класс II***. Система имеет эксклюзивные или не широко распространенные детали. Этим можно достичь очень хорошей производительности, но при более высокой стоимости.

- **Характеристики**, определяющие производительность кластеров
  - тип используемых процессоров
  - латентность (время начальной задержки при передаче сообщений);
  - пропускная способность сети (скорость передачи информации). При коротких сообщениях – латентность играет главную роль, при длинных – скорость.
  - топология соединения процессоров



Выводы

- **Кластеры строятся на основе** стандартных процессорных узлов, стандартного сетевого оборудования, стандартных ОС и систем управления ресурсами.

- **Критерии выбора комплектующих**: надежность, распространенность, предсказуемость производительности, проверка нагрузочными тестами.

- **Развитие кластеров идет по**
  - наращиванию числа процессоров,
  - использованию все более мощных процессоров (Intel, Alpha, AMD),
  - развитию коммуникационных технологий – Gigabit Ethernet

Российские кластерные проекты

МВС-100К с пиковой производительностью 140 TFlops (реальная 95 Тфлопс)

- Установлен в Суперкомпьютерном Центре РАН.
- Состав: 1460 вычислительных модуля, каждый с двумя 4-ядерными процессорами Intel Xeon, 3ГГц (11680 ядер).
- Для объединения узлов кластера используется технология Infiniband DDR. Infiniband – один из самых высокоскоростных современных стандартов: пиковая пропускная способность каналов 10, 20, 30, 40 Gb/sec, латентность 1.2 мксек. На MPI-тесте скорость двунаправленных обменов данными между двумя ВМ с использованием библиотек MPI находится на уровне 1.4 Гбайт/сек. Латентность между двумя соседними узлами – 3.2 мкс, самыми дальними 4.5 мкс.
- В редакции Top500 2012г –148е

Суперкомпьютер «Ломоносов»

- Установлен в НИВЦ МГУ в рамках проекта СКИФ по разработке семейства высокопроизводительных систем.
- В редакции Top500 в 2014г - 42е, в 2016 – 41е.

- 1-й гибридный суперкомпьютер такого масштаба в РФ и Восточной Европе. Используются 3 вида вычислительных узлов и процессоры с различной архитектурой.

- Основные узлы (обеспечивают свыше 90% производительности системы)- модули T-Blade2 на базе 4- и 6-ядерных микропроцессоров Intel Xeon 2.93 GHz и ускорители Nvidia 2070 GPU.

- Сеть: Infiniband QDR. Реальная производительность на тесте LINPACK: 397 Тфлопс, пиковая – 510 Тфлопс, отношение реальной и пиковой (эффективность): 78%.

- Число ВУ 5130, процессоров 10260, ядер 44000.
- В состав центра НИВЦ МГУ входят также кластеры «Чебышев» и «Менделеев» (в рамках проекта СКИФ).

2. Почему метакомпьютинг?

- Компьютеры со сверхвысокой производительностью - дефицитные, **дорогие и востребованные вычислительные ресурсы**, Нужны технологии, позволяющие сделать суперкомпьютерный ресурс доступным.

- Обеспечение эффективности **использования уже установленной компьютерной техники**.

- Неумения совместить в едином комплексе компьютеры с различными характеристиками. Основная проблема - **отсутствие эффективных технологий параллельного программирования, применимых к реальным неоднородным вычислительным средам**, состоящим из десятков, сотен и тысяч параллельно работающих различных компьютеров.

Принцип метакомпьютинга - создание инфраструктуры, объединяющей в единую вычислительную систему уже имеющиеся в наличии компьютера с использованием уже имеющихся коммуникаций.
В качестве коммуникационной среды для метакомпьютера может выполнять любая другая сетевая технология (необязательно Интернет)

Классификация систем мета-компьютинга

- **Научный Grid** («классический» Grid) — хорошо распараллеливаемые приложения программируются специальным образом (например, с использованием Globus Toolkit);

- **Добровольный Grid** (Интернет-компьютинг) — распределенные вычисления на основе использования добровольно предоставляемого свободного ресурса персональных компьютеров;

- **Коммерческий Grid** на основе выделения вычислительных ресурсов по требованию — обычные коммерческие приложения работают на виртуальном компьютере, который, в свою очередь, состоит из нескольких физических компьютеров, объединённых с помощью Grid-технологий.

Термин **Grid** используется как аналог электрической сети - включение в **Grid** пользователей должно быть столь же легким, как и включение бытовых приборов.

Направления работ по использованию мета-компьютинга

- **создание универсальных сред** (Grid). Как правило, при создании таких сред за основу берут *Globus Toolkit*. Направление перспективное, однако, реальные системы достаточно тяжелы в установке, администрировании и сопровождении

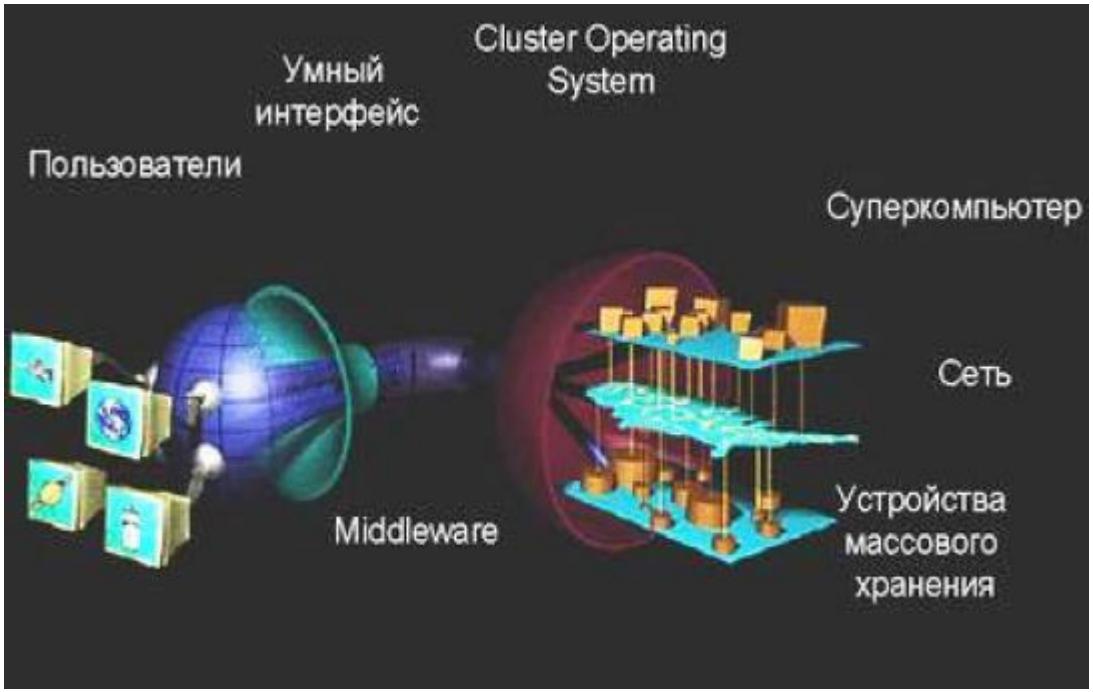
- здесь универсальность среды заменяет четкая **ориентация на конкретные задачи**. Это направление (Интернет-компьютинг) проще в реализации, однако в каждом случае среда жестко ориентирована на решение только одной конкретной задачи.

- **разработка инструментария** для быстрого создания распределенной вычислительной среды, объединяющей максимум доступных вычислительных ресурсов (коммерческий проект-диспетчер виртуальных машин *Vmware*, исследовательский проект - *X-Com*.

3. Концепция, архитектура и технологии GRID

Концепция Grid

- **Grid** – географически распределённая инфраструктура, объединяющая множество ресурсов разных типов (процессоры, долговременная и оперативная память, хранилища и базы данных, сети), доступ к которым пользователь может получить из любой точки, независимо от места их расположения.



Участники Grid- вычислений

- **Распределённые** (или Grid-) **вычисления** - это разновидность параллельных вычислений, которые проводятся на обычных компьютерах, подключенных к сети (локальной или глобальной) при помощи обычных протоколов, например Ethernet.

- В этом их отличие от «традиционных» параллельных вычислений на суперкомпьютерах, которые содержат множество процессоров, подключенных к локальной высокоскоростной шине. Таким образом **можно получить практически те же вычислительные мощности, что и на обычных суперкомпьютерах, но с гораздо меньшей стоимостью**.

Архитектура метакомпьютерной (GRID) среды

- Архитектура ПО глобальной вычислительной среды предполагает ***два*** уровня
  - **локальный** - Система управления пакетной обработкой (в кластере или без него)
  - **глобальный** (метакомпьютерный, GRID) - Выход за пределы локальной сети

- Основные **отличия уровней**:
  - Аппаратная среда глобального уровня ненадежна и недетерминирована
  - Некоторые средства доступа к распределенным ресурсам в глобальных сетях не работают
  - Многие **механизмы базовых ОС отказывают при работе в глобальной среде**. Например, контроль прав доступа к ресурсам и способ доступа к ресурсам

- Несмотря на перечисленные отличия, **без кластерного уровня обойтись нельзя** - они образуют важные структурные единицы метакомпьютера.

- Кластерные и метакомпьютерные (GRID) системы находятся на разных стадиях развития.

- **Кластерный** подход предлагает замкнутые рассчитанные на пользователя решения.

- В соответствии с этим **GRID** формируется и **развивается в двух направлениях**:
  - разработка средства взаимодействия кластеров – *Condor*
  - создание аппарата, учитывающего специфику распределенной открытой глобальной сети – *Legion, Globus Toolkit*

Технологии GRID

- В 2000 в статье "The Anatomy of the Grid" Фостер уточнил определение Grid: **Grid-компьютинг** – это **скоординированное** разделение ресурсов и решение задач в динамически меняющихся **виртуальных организациях** со многими участниками.
- Такое разделение должно быть под строгим контролем, определенном правилами. Отдельные пользователи и/или институты, подчиняющиеся таким правилам образуют **виртуальную организацию (ВО)**.

- **ВО** - совокупность людей и организаций, решающих совместно ту или иную общую задачу и предоставляющих друг другу свои ресурсы. Это **объединение специалистов из некоторой прикладной области, которые объединяются для достижения общей цели**.
- Реальная организация может участвовать в одной или нескольких ВО, разделяя некоторые или все ресурсы, контролируемые ей. ВО может образовываться динамически и иметь ограниченное время существования.

- Средством обеспечения взаимодействия и общности инфраструктуры Grid являются стандартные протоколы.

Итак, GRID-технологии :

- **координируют использование ресурсов при отсутствии централизованного управления этими ресурсами.**
- **используют стандартные, открытые, универсальные протоколы и интерфейсы.**
- **должны обеспечивать высококачественное обслуживание.**
- **НЕ являются технологиями параллельных вычислений – задачей технологий GRID является лишь координация использования ресурсов** (хотя в рамках конкретной Grid-системы возможно организовать параллельные вычисления с использованием существующих технологий параллельных вычислений)

Системное ПО

- Интерес представляют ведущиеся в рамках GRID работы по
  - глобальным файловым системам
  - системам сертификации и авторизации пользователей
  - оптимизации сетевой передачи данных
  - управлению ресурсами
  - планированию и диспетчеризации процессов

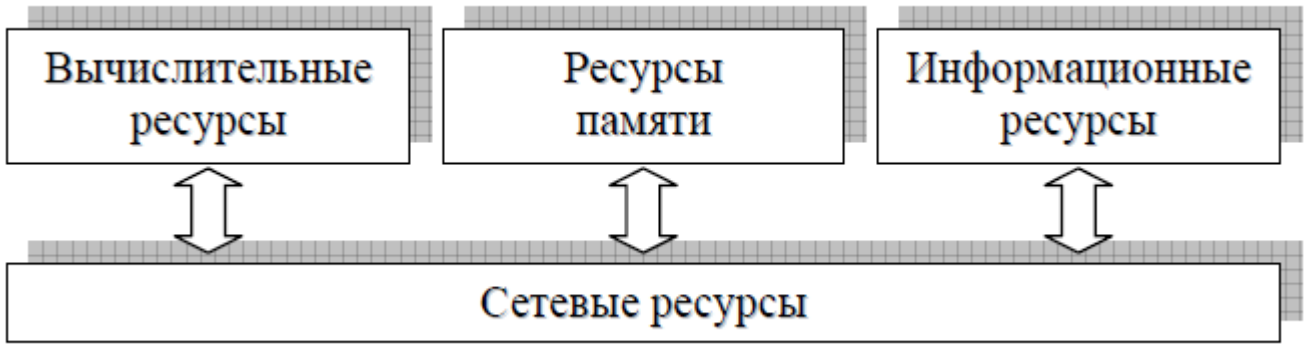
- Для построения полностью функциональной Grid-системы необходимо ПО промежуточного уровня (**middleware**), построенное на базе существующих инструментальных средств и предоставляющее высокоуровневые сервисы задачам и пользователям.
  Это ПО не часть ОС и не прикладное ПО
  Это – **межплатформенное, связующее ПО**

4. Ресурсы GRID

Ресурс - одно из основных понятий архитектуры Grid.

Основные типы ресурсов:

- вычислительные ресурсы;
- ресурсы хранения данных;
- информационные ресурсы, каталоги;
- сетевые ресурсы.



Вычислительные ресурсы предоставляют задаче пользователя Grid-системы процессорные мощности.

- Вычислительными ресурсами могут быть как кластеры, так и рабочие станции.

- Любая ВС -это потенциальный вычислительный ресурс Grid-системы. Необходимое условие для этого - наличие специального ПО, называемого **ПО промежуточного уровня (middleware)**, реализующего стандартный внешний интерфейс с ресурсом и позволяющего сделать ресурс доступным для Grid -системы.

- Основная характеристика вычислительного ресурса - **производительность**

Ресурсы памяти - пространство для хранения данных.

- Для доступа к ресурсам памяти также **используется ПО промежуточного уровня**, реализующее унифицированный интерфейс управления и передачи данных.

- Как и в случае вычислительных ресурсов, физическая архитектура ресурса памяти не принципиальна для Grid-системы.

- Основная характеристика ресурса памяти - его **объем**.

Информационные ресурсы и каталоги являются особым видом ресурсов памяти.

- Они служат для хранения и предоставления метаданных и информации о других ресурсах Grid -системы.

- Информационные ресурсы позволяют структурированно хранить огромный объем информации о текущем состоянии Grid -системы и эффективно выполнять задачи поиска.

Сетевой ресурс является связующим звеном между распределенными ресурсами Grid -системы.

- Географически распределенные системы на основе рассматриваемой технологии способны объединять тысячи ресурсов разного типа, независимо от их географического положения.

- Основная характеристика сетевого ресурса - **скорость** передачи данных.

Распределение ресурсов

- Эффективное распределение ресурсов и их координация являются основными задачами системы Grid, и для их решения используется **планировщик** (брокер ресурсов).

- Планировщик определяет наиболее подходящие ресурсы для каждой конкретной задачи и резервирует их для ее выполнения.

- Во время выполнения задача может запросить у планировщика дополнительные ресурсы или освободить избыточные.

- Пользователю не нужно знать о физическом расположении ресурсов, отведенных его задаче. Вся работа по использованию ресурсов ложится на планировщик.

5. Уровни GRID

Архитектура Grid имеет несколько уровней

Базовый уровень (Fabric Layer) описывает службы, непосредственно работающие с ресурсами.

Ресурсы должны поддерживать

- механизм запросов (enquiry);
- механизм управления ресурсами (resource management)

Уровень связи (Connectivity Layer) определяет коммуникационные протоколы и протоколы аутентификации.

- **Коммуникационные протоколы** обеспечивают обмен данными между компонентами базового уровня.

- **Протоколы аутентификации**, основываясь на коммуникационных протоколах, предоставляют криптографические механизмы для идентификации и проверки подлинности пользователей и ресурсов.

- Протоколы уровня связи должны обеспечивать надежный транспорт и маршрутизацию сообщений, а также присвоение имен объектам сети.
- Несмотря на существующие альтернативы, **сейчас протоколы уровня связи в Grid -системах предполагают использование только стека протоколов TCP/IP**, в частности: на сетевом уровне – IP и ICMP, транспортном уровне – TCP, UDP, на прикладном уровне – HTTP, FTP, DNS, RSVP.

Ресурсный уровень (Resource Layer)
реализует протоколы, обеспечивающие выполнение следующих функций:

- согласование политик безопасности использования ресурса;
- процедура инициации ресурса;
- мониторинг состояния ресурса;
- контроль над ресурсом;
- учет использования ресурса.

Различают **два основных класса протоколов ресурсного уровня:**

- **информационные протоколы**, которые получают информацию о структуре и состоянии ресурса;

- **протоколы управления**, которые используются для согласования доступа к разделяемым ресурсам. Протоколы управления должны проверять соответствие запрашиваемых действий политике разделения ресурса, включая учет и возможную оплату.

Список требований к функциональности протоколов ресурсного уровня близок к списку для **базового уровня архитектуры** Grid. Добавилось требование единой семантики для различных операций с поддержкой системы оповещения об ошибках.

Коллективный уровень (Collective Layer) отвечает за глобальную интеграцию различных наборов ресурсов, в отличие от ресурсного уровня, сфокусированного на работе с отдельно взятыми ресурсами.

- В коллективном уровне различают общие и специфические (для приложений) протоколы.

- Компоненты уровня реализуют большое множество вариантов разделения ресурсов:
  - службы директорий
  - сервисы совместного размещения, планировки и посредничества
  - сервисы мониторинга и диагностики
  - сервисы репликации данных
  - системы программирования (например, версии интерфейса передачи сообщений MPI для Grid)
  - сервисы обнаружения программного обеспечения
  - общие сервисы авторизации
  - средства сотрудничества

Функции коллективного уровня могут быть в виде

- постоянных сервисов (с привязанными к ним протоколами);
- SDK (Software Development Kit) (с соответствующими API), спроектированные для последующей связи с приложениями.

➢ Коллективные компоненты могут быть подогнаны под нужды обособленных групп пользователей, виртуальных организаций или областей приложений

Прикладной уровень (Application Layer) описывает пользовательские приложения, работающие в среде виртуальной организации. Приложения функционируют, используя сервисы, определенные на нижележащих уровнях.

Для облегчения работы с прикладными программными интерфейсами пользователям предоставляются наборы инструментальных средств для разработки программного обеспечения (SDK).

Сбои

Grid - сложная информационная среда, создаваемая человеком. Для такой системы важна **проблема обеспечения надежного функционирования**.

В Grid -системах используется сложная **система обнаружения и классификации ошибок**.

- Если ошибка произошла по вине задачи, то задача будет остановлена, а соответствующая диагностика направлена ее владельцу (пользователю).
- Если причиной сбоя послужил ресурс, то планировщик произведет перераспределение ресурсов для данной задачи и перезапустит ее.

- Из-за огромного количества задач и постоянно меняющейся сложной конфигурации системы **важно своевременно определять перегруженные и свободные ресурсы**, что и делает планировщик.

- Жизненно важным свойством является **отсутствие** так называемой **единственной точки сбоя**. Это означает, что отказ любого ресурса не должен приводить к сбою в работе всей системы. Именно поэтому планировщик, система мониторинга и другие сервисы Grid -системы распределены и продублированы.

6. Инструментальные средства Grid (Globus Toolkit)

- Одно из направлений развития GRID - создание аппарата, учитывающего специфику распределенной открытой глобальной сети

- В рамках проекта Globus Project создан набор инструментальных средств **Globus Toolkit**. Они позволяют построить полнофункциональную Grid - систему.

- Globus Toolkit - совокупность программных компонент, реализующих необходимые части архитектуры.

- Globus Toolkit не содержит брокера ресурсов, оставляя задачу его реализации разработчикам

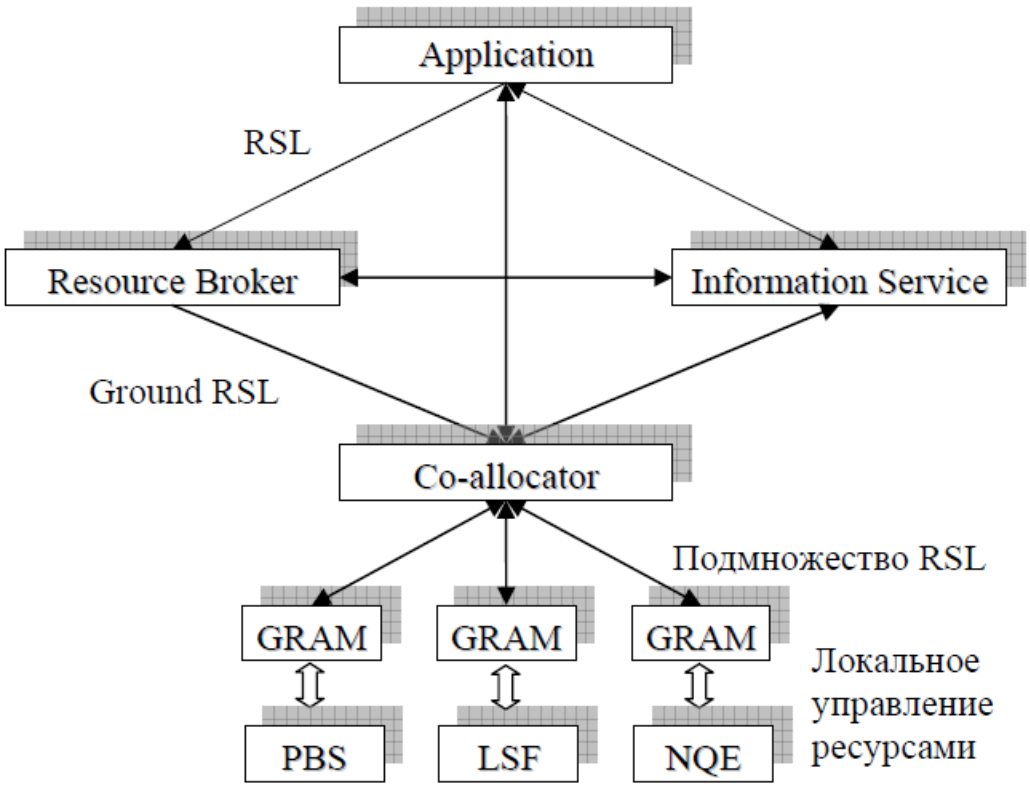
Globus Toolkit состоит из следующих основных компонент:

- **GRAM** (Globus Resource Allocation Manager), ответственный за создание/удаление процессов. Этот компонент устанавливается на вычислительном узле Грид-системы .

- **GSI** (Globus Security Infrastructure) обеспечивает защиту, включающую шифрование данных, а также аутентификацию и авторизацию

- **GASS** (Global Access to Secondary Storage) определяет различные стратегии размещения данных.

Управление ресурсами в Globus Toolkit



RSL (Resource Specification Language)
Системы управления ресурсами и загрузкой кластеров:
-***PBS*** (Portable Batch System)
-***LSF*** (Load Sharing Facility)
-***NQE*** (Network Queuing Environment)
Resource Broker - брокер ресурсов
Co-allocator- высокоуровневый менеджер ресурсов

- Запросы пользовательских приложений выражаются на RSL.
- ***Resource Broker*** отвечает за высокоуровневую балансировку загрузки в определенном домене. Он выбирает наиболее оптимальный вариант, генерит новый RSL-запрос (ground RSL) и передает его Co-allocator.
- ***Co-allocator*** производит декомпозицию запросов ground RSL на множество более простых RSL-запросов и передает эти запросы GRAM. Основные функции Co-allocator:
    - коллективное выделение ресурсов;
    - добавление/удаление ресурсов к ранее выделенным;
    - получение информации о состоянии задач;
    - передача начальных параметров задачам.
- При отсутствии сообщений об ошибках от ***GRAM***, задача пользователя запускается на исполнение. В случае, если один из GRAM возвращает ошибку, задача либо снимается с выполнения, либо попытка запуска производится повторно.

Безопасность в Globus Toolkit

- Инфраструктура безопасности Grid - Grid Security Infrastructure (**GSI**) - обеспечивает безопасную работу в незащищенных сетях общего доступа (Интернет), предоставляя такие сервисы, как аутентификация, конфиденциальность передачи информации и единый вход в Grid -систему.

- Под единым входом подразумевается, что пользователю нужно лишь один раз пройти процедуру аутентификации, а далее система сама будет его аутентифицировать на всех ресурсах, которыми он пользуется.

- GSI основана на надежной и широко используемой инфраструктуре криптографии с открытым ключом (Public Key Infrastructure – PKI).

Дальнейшее развитие инструментальных средств

- Основной **недостаток** Globus Toolkit - отсутствие унифицированных средств разработки приложений, способных взаимодействовать между собой и предоставлять друг другу различные сервисы).

- Была предложена Открытая архитектура сервисов Грид (Open Grid Services Architecture – **OGSA**). Стандарт OGSA **определяет основной набор услуг**, которые предоставляют Грид-системы, и описывает их архитектуру.

- Стандарт OGSA предлагает конструировать Грид-системы по принципу сервис-ориентированной архитектуры (**SOA**) и опирается на семейство технологий веб-сервисов (Web-services).

- **Технологии GRID** включают в себя лишь наиболее **общие аспекты**, одинаковые для любой системы (архитектура, протоколы, интерфейсы, сервисы).
- Наполняя их конкретным содержанием, можно реализовать **GRID-инфраструктуру,** предназначенную для решения того или иного класса прикладных задач.

- **Применение GRID** может дать новое качество решения следующих классов задач:

  • массовая обработка потоков данных большого объема;
  • многопараметрический анализ данных;
  • моделирование на удаленных суперкомпьютерах;
  • реалистичная визуализация больших наборов данных;
  • сложные бизнес-приложения с большими объемами вычислений
  • вычисления «по требованию» (On-Demand Computing), крупные разовые расчеты;

Перспективы использования Grid

Некоторые точки зрения на технологию Grid

- ***Grid – это следующее поколение Internet***. Grid не является альтернативой Internet: это набор дополнительных протоколов и сервисов, которые основаны на протоколах и сервисах Internet

- ***Grid - источник бесплатных вычислительных ресурсов***. Такие вычисления не подразумевают неограниченного доступа к ресурсам, но используют контролируемое разделение ресурсов.

- ***Grid требует распределенной ОС.*** Реально необходима установка определенных сервисов на системы, входящие в сообщество (т.е. создание виртуальной машины).

- ***Grid требует использования новых программных моделей***. Существуют определенные трудности по сравнению с последовательной (параллельной) машиной, однако это не центральные вопросы и основная концепция программирования не изменилась.

- ***Grid делает высокопроизводительные компьютеры ненужными.*** Технология Grid, возможно, только увеличит потребности в таких машинах из-за облегчения схемы доступа.

.

7. Примеры реализаций GRID

7.1 GRID-система LHC Computing GRID

Предназначена для обработки данных, получаемых с LHC (Large Hadron Collider - Большóй адрóнный коллáйдер).
Имеет иерархическую структуру.

На ней реализовано ПО LCG (LHC Computing Grid), разрабатываемое в Европейском центре ядерных исследований (CERN).

Пакет LCG состоит из нескольких частей, называемых элементами. Каждый элемент является самостоятельным набором программ, реализующих некоторый сервис, и предназначен для установки на компьютер под управлением ОС Scientific Linux.

• **Tier0**— CERN (получение информации с детекторов, сбор «сырых» научных данных, которые будут храниться до конца работы эксперимента). За первый год работы собрано 15 петабайт (тысяч терабайт) данных первой копии.

• **Tier1** — хранение второй копии этих данных в разных уголках мира. Один центр первого уровня — CMS Tier1 — в CERN . 11 центров - в Италии, Франции, Великобритании, США. Тайване, России . Центры обладают значительными ресурсами для хранения данных.

• **Tier2** — следующие в иерархии, многочисленные центры второго уровня. Наличие крупных ресурсов для хранения данных не обязательно; обладают хорошими вычислительными ресурсами. Российские центры: в Дубне (ОИЯИ), в Москве (НИИЯФ МГУ, ФИАН, ИТЭФ), Троицке (ИЯИ), Протвино (ИФВЭ), Санкт-Петербурге (СПбГУ) и Гатчине (ПИЯФ).
Кроме того, в единую сеть с этими центрами связаны и центры других стран-участниц ОИЯИ — в Харькове, Минске, Ереване, Софии, Баку и Тбилиси.

Более 85 % всех вычислительных задач LHC выполняется вне CERN , из них более 50% — на центрах второго уровня.

7.2 Проект X-COM

X-Com – семейство программ прототипов систем организации масштабных расчетов в распределенных средах. Разработка НИВЦ РАН.

Идеология. Активизируя на своих компьютерах специальный сервис, организации включаются в среду, предписывая режим использования своих ресурсов.
Сервис обеспечивает безопасность подключения компьютера к среде, причем полный контроль над компьютерами остается за хозяевами компьютеров, и только они решают, когда и в каком объеме их компьютеры могут быть использованы под общие нужды.
В определенные моменты времени (ночью, в выходные и праздники) без каких-либо затрат и вложений можно сформировать и предоставить для использования компьютерные системы с колоссальной производительностью.

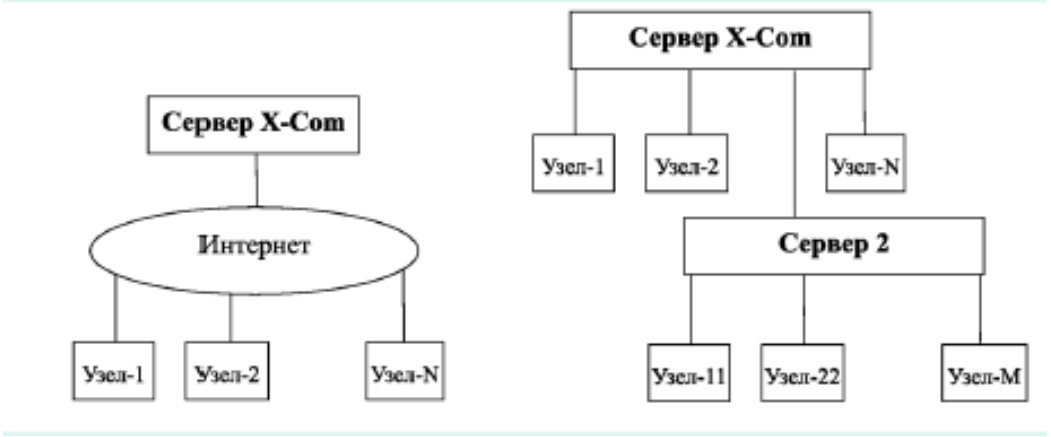
Семейство X-Com включает в себя две версии системы метакомпьютинга:
- Java-версия X-Com
- версия нового поколения на базе Perl.

Иерархия узлов и серверов

Система X-Com реализована по принципам клиент-серверной архитектуры, имеет два основных компонента:

- Сервер X-Com (центральная часть системы)
- Узел (любая вычислительная единица, на которой происходит расчет прикладной программы

Система организована в виде произвольного иерархического дерева. Листья-вычислительные узлы, в корне дерева останется центральный сервер X-Com. Промежуточный сервера с точки зрения центрального сервера - обычный вычислительный узел, а с точки зрения нижележащих вычислительных узлов - центральный сервер.



Сервер X-Com – содержит серверную часть программы пользователя, отвечает за:

- разделение исходной задачи на блоки
- распределение заданий
- координацию работ всех узлов
- контроль целостности результата
- сбор результата расчета в единое целое

Узел - любая вычислительная единица, отвечает за:

- расчет блоков прикладной задачи
- запрос заданий для расчета от сервера
- передачу результатов расчета на сервер

Все коммуникации между узлами и сервером в X-Com происходят через сеть Интернет. Используется только протокол HTTP, что позволяет подключать к системе практически любые вычислительные мощности, имеющие доступ в Интернет. Система не требует настройки для работы через прокси-сервер, firewall и другие системы защиты.

Алгоритм работы прикладной программы в системе X-Com

- Прикладная программа разбивается на две части клиентскую (вычислительную) и серверную.
- Серверная часть управляет формированием заданий для расчета на узлах. Вычислительная часть прикладной программы представляет собой основной расчетный модуль.
- Обе части могут быть реализованы с помощью любой технологии программирования

Во время работы сервера X-Com существует возможность просматривать общее состояние хода вычислений с помощью веб-интерфейса.

Возможны пять запросов от узла к серверу.

- Дай версию программы для моей аппаратно-программной платформы
- Дай задание
- Получи результат и дай следующее задание
- Получи результат
- Сообщение о статусе расчета на данном узле.

Проверка корректности результата. Возможны 4 основных способа.

- Прямая проверка корректности серверной частью прикладной программы. От выдавшего ошибку узла соединения больше не принимаются

- Прямая проверка корректности другим узлом. Другому узлу выдается задание в виде «обратной задачи»: по результатам, подлежащим проверке, надо восстановить исходные данные. Если данные совпали с теми, что были выданы в пакете задания, значит все в порядке. Иначе задание обрабатывается повторно, номера узлов, занимающихся проверкой и выполнением, заносятся в черный список.

- Метод многократного перерасчета. Одно и то же задание выдается нескольким узлам. В случае расхождения результатов правильный результат определяется методом голосования.

- Отсутствие проверки корректности.

ДОСТОИНСТВА:

- быстрая адаптация прикладных программ,
- отказоустойчивость (за счет повторной выдачи заданий при сбоях),
- учет динамической масштабируемости системы,
- учет неоднородности аппаратно-программных платформ.
- простота настройки и использования: не требуется специализированных компьютеров и высокоскоростных сетей; не требуется специальных навыков для настройки.

Система обеспечивает возможность эффективного решения больших задач с использованием уже существующих вычислительных ресурсов.

Вопросы?