

# Архитектура вычислительных систем. Лекция 5

- ❑ Метакомпьютинг
- ❑ Проекты Интернет-компьютинга
- ❑ Концепция GRID
- ❑ Концепция облачных вычислений
- ❑ Поисковая система GOOGLE как пример распределенной вычислительной системы

# Почему метакомпьютинг?

- Общество, наука и производство становятся все более информационными, высокопроизводительные вычисления – все более востребованными.
  - При этом суперкомпьютеры со сверхвысокой производительностью всегда были крайне дефицитным и дорогим ресурсом, потребность в котором, как показывает практика, со временем только возрастает.
  - Одна из актуальных задач – обеспечение эффективности использования уже установленной компьютерной техники. Число компьютеров в различных организациях только лишь научно-образовательного сегмента исчисляется сотнями тысяч. Однако эффективность использования не превышает и нескольких процентов.
  - Еще одна серьезная проблема – обеспечение преемственности в развитии средств вычислительной техники. Обновление компьютерного парка происходит раз в два-три года, что на практике приводит к выводу из обращения вполне работоспособных компьютерных систем из-за неумения совместить в едином комплексе компьютеры с различными характеристиками. Основная проблема здесь – отсутствие эффективных технологий, применимых к реальным неоднородным вычислительным средам, состоящим из десятков, сотен и тысяч параллельно работающих различных компьютеров.
- ⇒ Нужны технологии, позволяющие сделать суперкомпьютерный ресурс доступным. Один из вариантов решения – метакомпьютерные системы.

# Что такое метакомпьютинг?



Распределённые вычисления — способ выполнения сложных расчётов путём их разделения между множеством компьютеров

В том числе, Интернет дает уникальные возможности для распределенных вычислений, его можно рассматривать как самый большой компьютер в мире.

Никакая самая мощная вычислительная система не может сравниться по суммарной производительности и объемам памяти с совокупностью ресурсов, которыми обладают компьютеры, подключенные к Интернету и составляющие, по сути, **компьютер из компьютеров = метакомпьютер**.

Отсюда происходит специальное название для организации процесса вычислений на такой вычислительной системе – **метакомпьютинг**.

Совершенно необязательно рассматривать именно Интернет в качестве коммуникационной среды для метакомпьютера, эту роль может выполнять любая другая сетевая технология.

**Базовый принцип метакомпьютинга: инфраструктура, объединяющая в единую вычислительную систему уже имеющиеся в наличии компьютеры с использованием уже имеющихся коммуникаций.**

Вместе с тем, к Интернету с этой точки зрения проявляется особый интерес, поскольку на данный момент никакая другая вычислительная система по своей мощности не может сравниться с потенциальными возможностями глобальной сети.

# Как появился метакомпьютинг

- История метакомпьютинга начиналась с научных приложений. В начале 90х возникла идея создать из многочисленных суперкомпьютерных центров США большой метакомпьютер, чтобы пользователи могли получать неограниченные ресурсы для вычислений и хранения данных.
- Первые прототипы реальных систем метакомпьютинга стали доступными с середины 90х.
- Некоторые системы претендовали на универсальность, другие сразу были ориентированы на решение конкретных задач. Где-то ставка делалась на использование высокопроизводительных сетей и специальных протоколов, а где-то за основу брались обычные каналы и протокол HTTP.
- Реализация идеи метакомпьютинга продолжается до сих пор. А разработки и технологии – оказались применимы не только для организации высокопроизводительных вычислений в науке и высоких технологиях, но и для других областей жизни – банки, торговые и производственные предприятия имеют по своей природе распределенную структуру и нуждаются в инфраструктуре, позволяющей организовать межкорпоративное взаимодействие на основе распределенных программных приложений.
- Метакомпьютинг (и, в частности, ГРИД как одно из его воплощений) опирается и развивает традиционные технологии Интернет, и в литературе о нем часто говорят как об Интернете нового поколения.

# Особенности метакомпьютерных систем:

- ❑ **Огромные ресурсы по всем параметрам** – число процессоров, память, количество приложений, пользователей. Все это во много раз превышает показатели одного даже мощного суперкомпьютера.
- ❑ **Распределенная система по своей природе.** Компоненты метакомпьютера могут быть удалены друг от друга на сотни и даже тысячи км, что неизбежно вызовет большую латентность и, следовательно, скажется на оперативности их взаимодействия.
- ❑ Система может **динамически менять конфигурацию**. Какие-то компьютеры подключаются, какие-то становятся недоступными. Соответственно, задача поддержки работы метакомпьютера состоит в поиска подходящих ресурсов, проверке их работоспособности, распределении поступающих задач вне зависимости от текущей конфигурации в целом.
- ❑ **Неоднородность метакомпьютера.** Разные ОС поддерживают разные системы команд, системы представления данных и т.д. Работа разных составляющих метакомпьютера может иметь разную загрузку, каналы связи – разную пропускную способность. В состав метакомпьютера могут входить системы с принципиально разной архитектурой – от домашних ПК, до мощнейших суперкомпьютеров из списка Top500.
- ❑ **Метакомпьютер объединяет ресурсы разных организаций**, политика безопасности и доступа к ресурсам может сильно различаться. Отсюда необходимость стандартизации всех его служб и сервисов.
- ❑ В первую очередь идет не об аппаратуре, а об **инфраструктуре**.

# Классификация систем метакомпьютинга (1)

Один из первых проектов – объединение в единый метакомпьютер двух 512-процессорных суперкомпьютеров CrayT3e, находящихся в университете Штутгарта (Германия) и Питтсбургском суперкомпьютерном центре (США). Обмен данными производился через Интернет на базе стека протоколов TCP/IP.

Другой пример из числа первых реализаций идей метакомпьютинга – проект **CONDOR**: распределение задач по существующей корпоративной сети рабочих станций, которые ночами отдают свое время для внешних вычислений. В настоящее время CONDOR – один из лидеров по разработке ПО для организации вычислений и управления ресурсами в распределенных системах.

- ❑ Научные гриды («классический» грид) — инфраструктура, формируемая на базе концепции «виртуальных организаций» для реализации высокоресурсных научных приложений. Строится на основе стеков как сетевых, так и специально созданных протоколов (Globus Toolkit);
- ❑ Добровольный грид (Интернет-компьютинг) — распределенные вычисления на основе использования добровольно предоставляемого свободного ресурса персональных компьютеров;
- ❑ Коммерческий грид на основе выделения вычислительных ресурсов по требованию — обычные коммерческие приложения работают на виртуальном компьютере, который, в свою очередь, состоит из нескольких физических компьютеров, объединённых грид-технологиями.
- ❑ Облачные инфраструктуры, предоставляющие сервис для вычислений

# Классификация систем мета-компьютинга (2)

Пропускная способность каналов связи неуклонно растет, латентность взаимодействия компьютеров снижается, коммуникационные характеристики локальных и глобальных сетей уже превзошли параметры первых кластерных систем.

При этом распределенные вычислительные среды обладают набором специфических свойств, которых не было ни у одной из ранее существовавших компьютерных систем. Это создает специфические проблемы при организации распределенных вычислений.

В первую очередь это:

- гетерогенность во всех смыслах (производительность, ОС, представление данных и т.п.)
- динамичность (состав вычислительной системы может меняться со временем)
- географическая удаленность вычислительных сегментов друг от друга

Поиск методов эффективного использования вычислительных сред является одной из основных задач, стоящих перед современным компьютерным сообществом.

Инструментарий должен быть универсальным, позволяя решать широкий класс задач. Он должен быть прост в освоении и дать возможность быстро развернуть расчет на всех основных типах компьютерных систем с использованием полномочий обычных пользователей.



# Классификация систем мета-компьютинга (3)

Как уже сказано, работы по использованию распределенных вычислительных сред идут в разных направлениях, в том числе:

Первое направление - создание универсальных сред ([Грид](#)). Как правило, при создании таких сред за основу берут стандарты [Globus Toolkit](#), формируются глобальные полигоны, объединяющие в рамках высокоскоростных сетей значительные распределенные ресурсы.

Второе направление – создание специализированных вычислительных порталов с жесткой ориентацией на конкретную задачу. Речь идет о создании специализированных сред для решения заранее predetermined набора многократно используемых «тяжелых» вычислительных задач. Примеры: [SETI@home](#) с жесткой ориентацией на конкретную задачу. Речь идет о создании специализированных сред для решения заранее predetermined набора многократно используемых «тяжелых» вычислительных задач. Примеры: SETI@home, [Folding@home](#)

Третий путь – инструментарий для быстрого создания под конкретную задачу временной распределенной вычислительной среды, объединяющей максимум доступных вычислительных ресурсов. Сюда можно отнести как коммерческие (пример - диспетчер виртуальных машин [VMware](#)), так и исследовательские (пример - **проект X-Com**) проекты. Требования:

- эффективность, легкость в установке, освоении и использовании,
- возможность задействовать все основные типы компьютеров,
- работа по открытым каналам связи (в частности, через Интернет),



# Проекты Интернет-компьютинга (1)



SETI@home – один из популярных научных некоммерческих проектов. Предлагается желающим принять участие в поиске внеземных цивилизаций. Космический шум, записываемый радиотелескопом в Аресибо, делится на небольшие блоки и рассылается на компьютеры участников для поиска сигналов с особыми характеристиками, которые могут иметь искусственное внеземное происхождение. Предполагалось, что если такой сигнал будет обнаружен, участников, чьи компьютеры занимались обработкой этих юнитов, занесут в список соавторов всех последующих научных публикаций.

Аббревиатура SETI означает Search for Extra-Terrestrial Intelligence at Home — поиск внеземного разума на дому. Проект инициирован исследовательским центром SETI при Калифорнийском университете в Беркли, впервые представлен широкой общественности 17 мая 1999. Один из крупнейших проектов такого типа по исследованию межзвездного пространства. С 31 марта 2020 проект заморожен, задания не рассылаются.

Исходными целями SETI@home являлись:

1. проделать полезную научную работу, поддерживая наблюдательный анализ поиска разумной внеземной жизни. Проект не дал окончательных результатов: не было собрано никаких доказательств сигналов внеземного разума.
2. доказать жизнеспособность и практичность концепции добровольных вычислений. Считается, что эта цель достигнута.

# Проекты Интернет-компьютинга (2)



Другие известные проекты связанные с космосом:

[Einstein@home](#) занимается составлением для всего неба излучаемых звёздами-пульсарами гравитационных полей.

Это делается с целью проверки общей теории относительности Эйнштейна в целом и проверки одной из гипотез Эйнштейна о существовании гравитационных волн, возникающих при столкновениях черных дыр и взрывах звезд, а также получения ответов на вопросы «Распространяются ли гравитационные волны со скоростью света?» и «Чем они отличаются от электромагнитных волн?»

Проект стартовал в 2004, в разные годы велась обработка данных с разных телескопов, был обнаружен целый ряд неизвестных ранее радиопульсаров. Проект активен до сих пор.

[MilkyWay@Home](#) — проект добровольных распределённых вычислений в области астрофизики.

Цель проекта — попытка создания высокоточной трёхмерной динамической модели звёздных потоков в нашей Галактике — Млечный Путь (англ. Milky Way). В качестве второстепенной цели проект также занимается разработкой и оптимизацией алгоритмов распределённых вычислений. Проект был запущен в декабре 2007г, активен в настоящее время.

# Проекты Интернет-компьютинга (3)



Примеры проектов, связанные с медициной и биологией:

Проект [Folding@home](#) – проект распределённых вычислений для проведения компьютерного моделирования свёртывания молекул белка для получения более точного представления о болезнях, вызываемых дефектными белками. Проект запущен 1 октября 2000 года учёными из Стэнфордского университета.

Цель проекта — с помощью моделирования процессов свёртывания и развёртывания молекул белка получить лучшее понимание причин возникновения таких болезней, как Альцгеймера, Паркинсона, диабет 2 типа, болезнь Крейтцфельда — Якоба (коровье бешенство), склероз и различные формы онкологических заболеваний.

К настоящему времени проект Folding@home успешно смоделировал процесс свёртывания белковых молекул на протяжении 5—10 мкс — что в тысячи раз больше предыдущих попыток моделирования.

По результатам эксперимента вышло более 212 научных работ. Результаты выкладываются в открытый доступ и могут использоваться учёными.

[Rosetta@home](#) – проект, направленный на решение одной из самых больших проблем в молекулярной биологии — моделирование структуры белков из их аминокислотных последовательностей, что дает основу для проектирования новых белковых структур, в том числе для лекарственных форм.

# Проекты Интернет-компьютинга (4)



[LHC@home](#) занимается моделированием процессов, происходящих при экспериментах на LHC (Large Hadron Collider) — самом большом в мире ускорителе частиц.

Проект инициирован сотрудниками CERN для проведения расчётов, необходимых при постройке и эксплуатации Большого адронного коллайдера. Вычисления стартовали в сентябре 2004. Сначала число участников проекта было ограничено и составляло 1000 человек, в дальнейшем ограничение отменено. В марте 2011 запущен проект LHC@Home2.0, целью которого уже является моделирование столкновений пучков протонов.

Участники проекта [DIMES](#) помогают в изучении структуры и топологии Интернета — строят его карту. По словам руководителей проекта, иначе как с использованием распределенных, вычислений, такую карту создать невозможно. Проект стартовал в 2005 году.

Проект [Climate Prediction](#) для прогноза изменений климата Земли в ближайшие 50 лет. Проект должен показать, насколько точны существующие методы долговременного предсказания изменений климата, и насколько сильно на их точность влияют вариации и неточности в исходных данных. Проект осуществляется с помощью запуска добровольцами сотен тысяч немного отличающихся друг от друга исходными данными компьютерных моделей земного климата.

# Проекты Интернет-компьютинга (5)



Примеры проектов, связанных с математикой:

Проект Seventeen or Bust («Семнадцать или провал») направлен на решение одной из фундаментальных задач теории чисел — задачи Серпинского.

$k$  – число Серпинского, если для любого  $n$  число  $k \times 2^n + 1$  – составное. Пример:  $k=3$  не является числом Серпинского, т.к. среди чисел  $3 \times 2^n + 1$  попадаются простые. В 1962 найдено было 78557, известно также 271129 и ряд других. Вопрос – является ли 78557 наименьшим?

Проект стартовал в марте 2002 года, в апреле 2016 года после потери сервера данных был поглощен проектом PrimeGrid и стал его подпроектом. К концу 2017 года из семнадцати значений  $k$ , которые нужно проверить, осталось пять: 21181, 22699, 24737, 55459 и 67607.

Проект GIMPS направлен на поиск простых чисел с использованием формулы Марсенна.

Известно: если  $p$  простое число, то  $2^p - 1$  тоже простое. Премия \$100000 была выдана в 2009 организацией Electronic Frontier Foundation за найденное простое число Марсенна с 10млн цифр ( $p=43112609$ ). Десятки тысяч компьютеров объединены на добровольной основе; проект действует до сих пор (<http://mersenne.org>), сейчас обещаны премии \$150000 и 250000 соответственно за 100млн и 1млрд цифр.



# Концепция ГРИД (1)

Грид – 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.

Распределённые (или грид-) вычисления в целом являются разновидностью 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. При этом отдельная ячейка вычислительной системы может быть приобретена как обычный неспециализированный компьютер. Таким образом можно получить практически те же вычислительные мощности, что и на обычных суперкомпьютерах, но с гораздо меньшей стоимостью.

## Грид-компьютинг. Отличие от Интернет-компьютинга:

- 1) нацеленность на инновационные приложения, связанные с необходимостью разделения крупномасштабных ресурсов (включая возможность высокопроизводительных вычислений).
- 2) концепция виртуальных организаций (ВО), которые договариваются о выделении части своих ресурсов в «общий котел» в рамках ГРИД
- 3) Децентрализованная инфраструктура. Безопасность, управление, доступ к ресурсам регулируются стандартами (стек протоколов)



# Концепция ГРИД (2)

В 1998 Кессельман и Фостер дали определение в книге "The Grid: Blueprint for a New Computing Infrastructure": Грид – аппаратно-программная инфраструктура, которая обеспечивает надежный, устойчивый, повсеместный и недорогой доступ к высокопроизводительным компьютерным ресурсам.

В 2000 уточнил определение: Грид-компьютинг – это скоординированное разделение ресурсов и решение задач в динамически меняющихся виртуальных организациях (ВО) со многими участниками.

✓ Отдельные пользователи и/или институты, подчиняющиеся таким правилам, и образуют **виртуальную организацию**.

✓ Средством обеспечения взаимодействия и общности инфраструктуры Грид являются **стандартные протоколы**.

## Таким образом, ГРИД:

- координирует использование ресурсов при отсутствии централизованного управления этими ресурсами.
- использует стандартные, открытые, универсальные протоколы и интерфейсы.
- должна обеспечивать высококачественное обслуживание.

**Первое поколение Грид.** CONDOR – корпоративные рабочие станции ночью отдают свое время для внешних вычислений. I-WAY – объединение в виртуальную сеть трех реальных разнородных коммутируемых сетей (на базе АТМ-технологии).

# Концепция ГРИД (3)

**Второе поколение** – новые требования к инструментам и сервису.

- Административная иерархия – способ взаимодействия пользователя с системой.
- Коммуникационные службы – надежность и безопасность связи.
- Информационные службы – доступность ресурсов (копии баз данных).
- Единое пространство имен.
- Распределенные файловые системы.

GLOBUS – в 1998 создана компания Апплайд метакомпьютинг, 2001 – Аваки, одно из лидирующих мест в области разработки Грид-технологий – фактически стандарт. Что разрабатывают:

- Резервирование ресурсов на основе HTTP-протокола
- Grid-FTP для передачи файлов
- Аутентификация и безопасность
- Удаленный доступ к данным

**Третье поколение.** Примеры: Superjanet England, Geantnet – Europe; APAn – Asia; TERAGRID – USA; CERN (коллайдер LHC), NanoGRID

Принципиально новые инструменты и возможности.

- Научные порталы для вызова пакетов прикладных программ.
- Распределенные вычисления на десятках тыс. компьютеров.
- Анализ сверхбольших объемов данных благодаря объединению систем хранения информации средствами Grid.

# Концепция ГРИД (4)

## Структура грид-системы ЦЕРНа LHC Computing GRID

Грид-система, предназначенная для обработки данных, получаемых с LHC, имеет иерархическую структуру.

- Самая верхняя точка иерархии, нулевой уровень — [CERN](#) (получение информации с детекторов, сбор «сырых» научных данных, которые будут храниться до конца работы эксперимента). За первый год работы планируется собрать до 15 [петабайт](#) (тысяч терабайт) данных первой копии.
- Первый уровень, Tier1 — хранение второй копии этих данных в разных уголках мира. Один центр первого уровня — CMS Tier1 — в ЦЕРНе. 11 центров - в Италии, Франции, Великобритании, США. Тайване. Сейчас центр 1го уровня организуется у нас). 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](#)), три центра в Москве ([НИИЯФ](#), [МГУ](#), [ФИАН](#), [ИТЭФ](#)), Троицке ([ИЯИ](#), [ИТЭФ](#)), Троицке ([ИЯИ](#)), Протвино ([ИФВЭ](#)), Санкт-Петербурге ([СПбГУ](#))<sup>[5]</sup> и Гатчине ([ПИЯФ](#)). Кроме того, в единую сеть с этими центрами связаны и центры других стран-участниц ОИЯИ — в [Харькове](#)). Кроме того, в единую сеть с этими центрами связаны и центры других стран-участниц ОИЯИ — в Харькове, [Минске](#)). Кроме того, в единую сеть с этими центрами связаны и центры других стран-участниц ОИЯИ — в Харькове, Минске, [Ереване](#)). Кроме того, в единую сеть с этими центрами связаны и

# Концепция ГРИД (5)

## ИТАК:

- ❑ Грид-технологии обеспечивают **гибкий, безопасный и скоординированный общий доступ к ресурсам**, под которыми понимаются не только процессорные ресурсы или ресурсы хранения информации, но и сетевые ресурсы, а также системное или прикладное программное обеспечение.
- ❑ Технологии грид **включают в себя лишь наиболее общие и универсальные аспекты**, одинаковые для любой системы (архитектура, протоколы, интерфейсы, сервисы). Используя эти технологии и наполняя их конкретным содержанием, можно реализовать ту или иную грид-инфраструктуру, предназначенную для решения того или иного класса прикладных задач.
- ❑ Грид-технологии **НЕ являются технологиями параллельных вычислений** – задачей технологий грид является лишь **координация** использования ресурсов (хотя в рамках конкретной грид-системы возможно организовать параллельные вычисления с использованием существующих технологий параллельных вычислений).
- ❑ Для построения полностью функциональной грид-системы необходимо **программное обеспечение промежуточного уровня (middleware)**, построенное на базе существующих инструментальных средств и предоставляющее высокоуровневые сервисы задачам и пользователям. Создание и реализация грид-технологий является сложной научной и практической проблемой, находящейся на стыке большого количества научно-технических направлений

# Облачные вычисления

**Облачные вычисления** — модель обеспечения удобного сетевого доступа к некоторому общему фонду конфигурируемых вычислительных и информационных ресурсов (включая возможность получать из сети нужные вычислительные мощности).

## Плюсы:

- Затраты
- Скорость
- Глобальный масштаб
- Производительность
- Надежность



## Задачи:

- создание новых приложений и служб;
- хранение, резервное копирование и восстановление данных;
- размещение веб-сайтов и блогов;
- потоковое аудио и видео;
- доставка программного обеспечения по запросу;
- анализ данных с целью поиска шаблонов и прогнозирования.

**Облако** — сервер (или группа серверов), на котором пользователь, используя Интернет, хранит информацию или производит вычисления

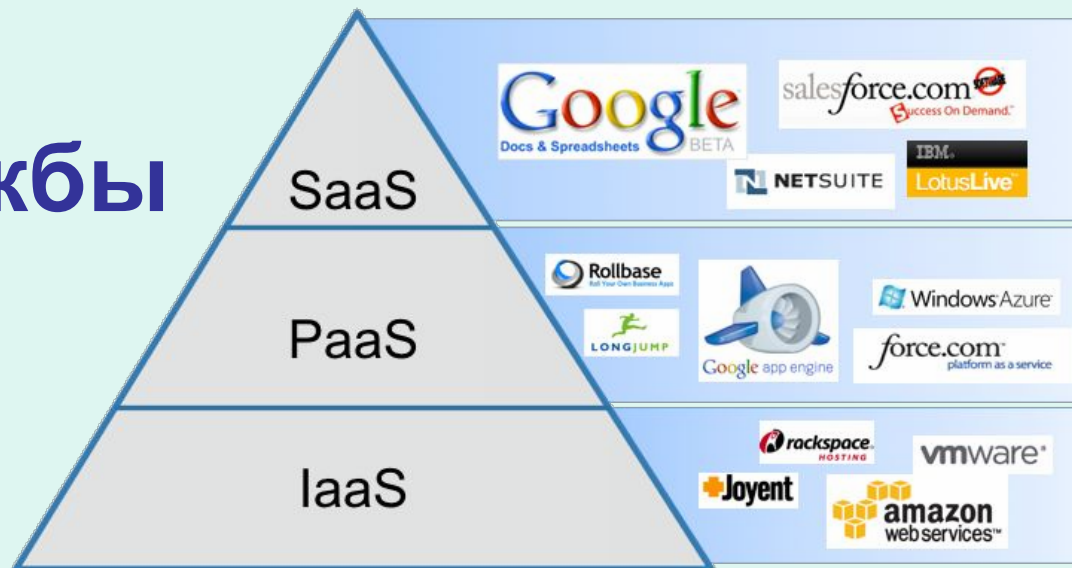
- **Общедоступное облако** - компания предоставляет облако сторонним заказчикам
- **Частное облако** - компания создает «облако» для себя
- **Гибридное облако**

Пример: Яндекс-диск — одно из многих облачных хранилищ





# Облачные службы



- ❑ **Инфраструктура как услуга (IaaS).** Предоставляется ИТ-инфраструктура: серверы, виртуальные машины, хранилище, сети и операционные системы
- ❑ **Платформа как услуга (PaaS).** Облачные вычислительные службы, которые поставляют среду для разработки, тестирования, доставки приложений программного обеспечения и управления ими.
- ❑ **Программное обеспечение как услуга (SaaS).** Облачные поставщики размещают программное обеспечение и базовую инфраструктуру и управляют ими, а также занимаются всем обслуживанием, включая обновление программного обеспечения и установку исправлений безопасности.

# Поисковая система GOOGLE как пример распределенной вычислительной системы (1)

Google – всем известная система поиска информации в Интернете. Популярность – благодаря простоте интерфейса и малому времени отклика. Рассмотрим - как все это организовано.

Требования:

- проиндексировать и сохранить всю информацию в Интернете (более 40 млрд. страниц),
- находить нужную страницу за 0.5 сек,
- круглосуточно обслуживать десятки тысяч запросов в секунду,
- никогда не отключаться

Функционирование Google обеспечивает множество информационных центров по всему миру. Тем самым, обеспечивается защита от сбоев и децентрализованная обработка запросов на ближайших к месту расположения IP-адреса центрах.

Механизм обработки поступившего запроса:

- Запрос поступает (1) и направляется службой распределения нагрузки к одному из обработчиков запросов (2).
- Параллельно – в систему проверки правописания и языка (3)
- Параллельно – в сервер контекстной рекламы (4)
- Параллельно – поиск запрошенного слова в индексных серверах (5)

# Поисковая система GOOGLE как пример распределенной вычислительной системы (2)

Для повышения производительности поиск по всем словам запроса ведется параллельно.

В ответ на запрос:

- индексные серверы возвращают наборы идентификаторов документов по каждому поисковому сегменту (6)
- Далее идентификаторы комбинируются, чтобы содержать все слова запроса (7)

Для ускорения:

- Копии многих страниц хранятся на всех серверах.
- Сами документы также разбиваются на сегменты и обрабатываются параллельно.

Далее результаты возвращаются к обработчику запроса, где:

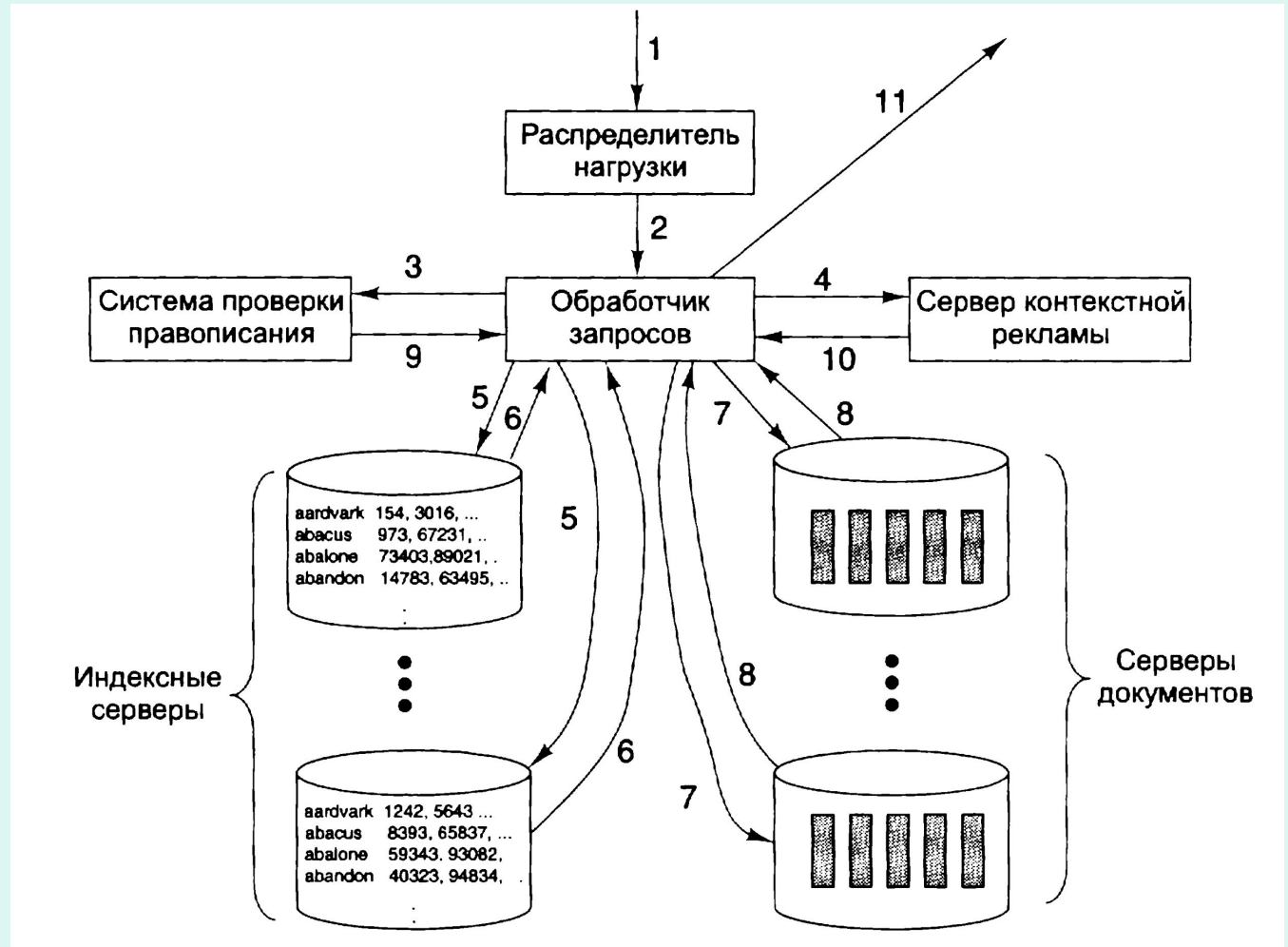
- сортируются в соответствии с рейтингом страниц (8)
- информация о возможных ошибках правописания (9)
- Контекстная реклама (10)

Таким образом, понятно, что эффективная обработка запросов базируется на широком использовании параллелизма на всех этапах работы.



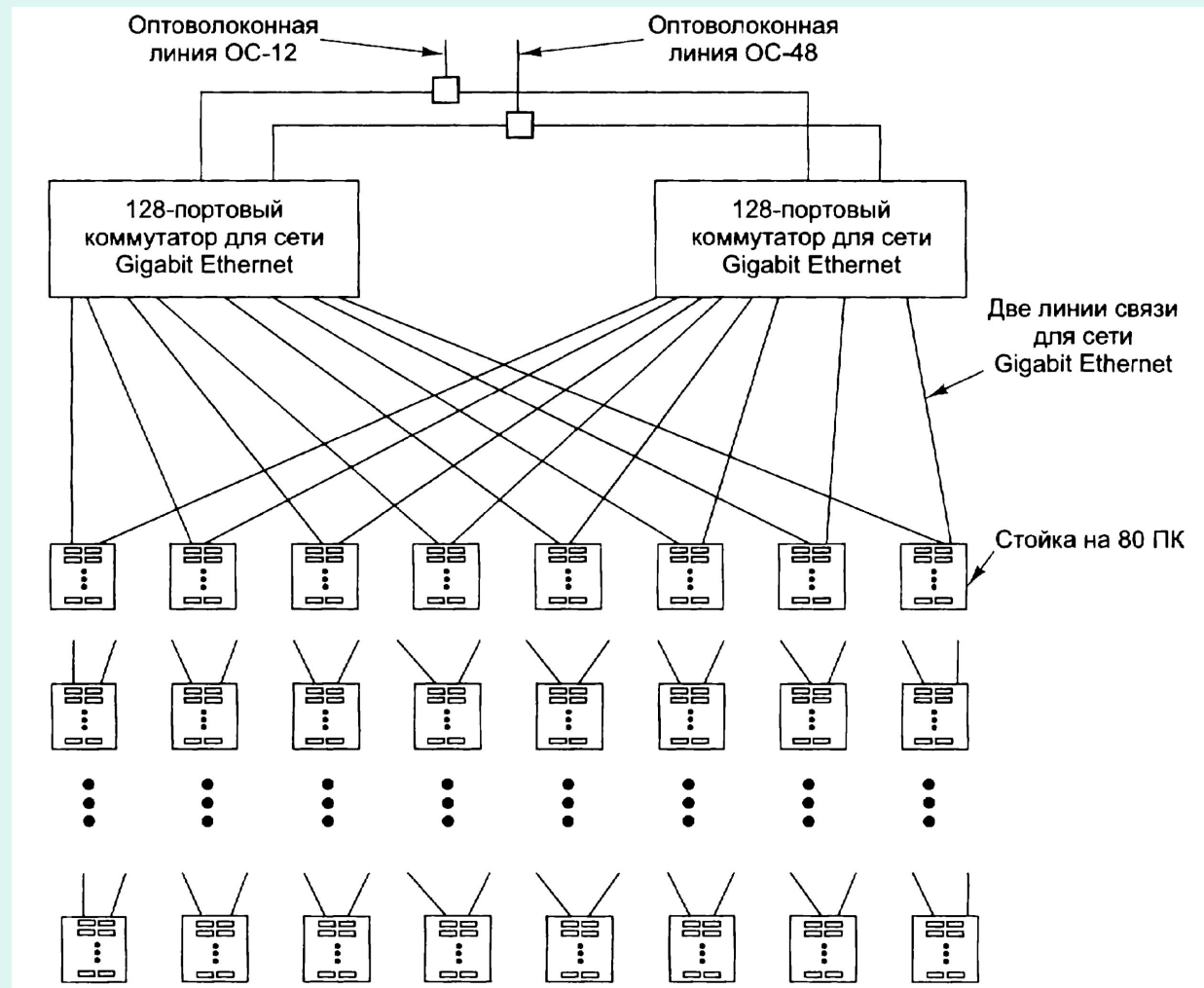
# Поисковая система GOOGLE как пример распределенной вычислительной системы (3)

## Схема обработка поискового запроса



# Поисковая система GOOGLE как пример распределенной вычислительной системы (4)

## Схема типичного Google-кластера



# Поисковая система GOOGLE как пример распределенной вычислительной системы (4)

- ❑ Разработчики Google при проектировании серверов отказались от самого быстрого оборудования в пользу достаточно дешевых ПК, работающих параллельно. Тем самым, оптимизировано соотношение «цена\производительность»
- ❑ Отказоустойчивое ПО компенсирует возможные сбои сетевого и компьютерного оборудования. В ответ на сбой обработки запроса в первую очередь запускается перезагрузка, в большинстве случаев это решает проблему.
- ❑ Дублирование оборудования, в том числе подключение резервных оптоволоконных коммуникаций – повышает надежность и пропускную способность.
- ❑ Специальные системы охлаждения в связи с высоким энергопотреблением (до 3000Вт/куб.м)

