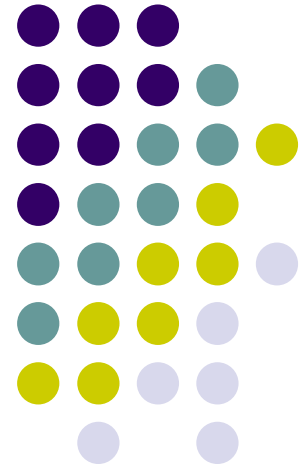


Распределенные системы

Лекция 1

Введение в курс.
Требования к распределенным
системам

к.т.н. Приходько Т.А.



Структура курса. Лекции



1. Распределенные системы: задачи, терминология принципы функционирования. Требования к распределенным системам.
2. Математическое представление распределенной системы. Сосредоточенные и распределенные системы. Распределенные задачи и алгоритмы. Надежность и безопасность распределенных систем
3. Модели и архитектуры распределенных систем. Архитектура клиент-сервер. Типовые задачи. Области применения. Многоярусная архитектура. Области применения. Технология RPC – Remote Procedure Call
4. Технологии middleware. Транзакционное взаимодействие. Объектно-ориентированный подход к распределенной обработке информации.
5. Компонентные Технологии middleware. Обзор технологий: CORBA, DCOM, .NET, WEB - сервисы. Язык определения интерфейсов IDL.
6. Распределенные системы BlockChain
7. Управление временем в распределенных системах. Синхронные и асинхронные распределенные системы. Алгоритмы синхронизации в распределенных системах
8. Управление временем в моделях распределенных систем. Типы времени в моделях распределенных систем. Виды имитационных моделей. Алгоритмы синхронизации в PCM. Балансировка нагрузки в PC.
9. Архитектуры распределенных систем
10. Микросервисы Spring.
11. Модели взаимодействия процессов
12. Агенты и мультиагентные системы. Взаимодействие агентов в системе. Архитектура и проектирование MAC. Современные международные стандарты создания агентов и платформы MAC. Агентные платформы. Области применения MAC.



Лекция 1

- Что такое распределенная система?
- Типы распределенных систем.
- Зачем нужны распределенные системы.
- Требования к РС (DS).
- Какие проблемы существуют при построении распределенных систем?

Распределенные системы: определения



- Эндрю С. Таненбаум (Andrew S. Tanenbaum) определяет **распределенную систему** как набор независимых компьютеров, представляющий их пользователям единой объединенной системой:

Распределенная система – набор независимых компьютеров, не имеющих общей совместно используемой памяти и **общего единого времени (таймера)** и взаимодействующих через коммуникационную сеть посредством передачи сообщений, где каждый компьютер использует свою собственную оперативную память и на котором выполняется отдельный экземпляр своей операционной системы, предоставляя свои службы друг другу для решения общей задачи.

Распределенные системы: определения

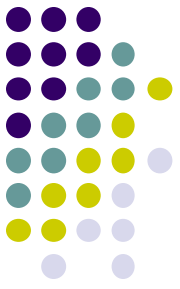


Термин "распределенная система" описывает широкий спектр систем от слабо связанных многомашинных комплексов, представляемых, например, набором персональных компьютеров, объединенных в сеть, до сильно связанных многопроцессорных систем.

Существуют системы:

- с разделяемой памятью
- с разделяемым временем

Распределенные системы: определения



Мы будем рассматривать распределенную систему

- с аппаратной точки зрения:
 - в виде совокупности взаимосвязанных автономных компьютеров или процессоров,
- с программной точки зрения:
 - в виде совокупности независимых процессов (исполняемых программных компонентов распределенной системы), взаимодействующих посредством передачи сообщений для обмена данными и координации своих действий.
- Компьютеры, процессоры или процессы будем называть **узлами распределенной системы**.

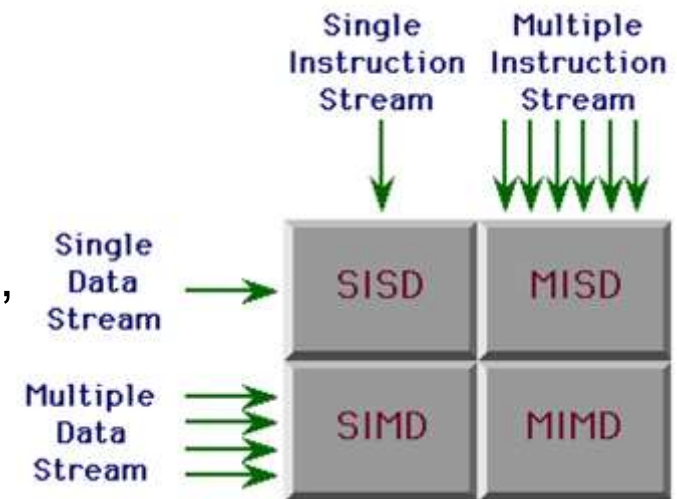
Варианты архитектур



Таксономия (Классификация) Флинна — общая классификация архитектур ЭВМ по признакам наличия параллелизма в потоках команд и данных.

Под **независимостью процессов** подразумевается тот факт, что каждый процесс имеет свое собственное состояние, представляемое набором данных, включающим текущие значения счетчика команд, регистров и переменных, к которым процесс может обращаться и которые может изменять. Состояние каждого процесса является полностью закрытым для других процессов: другие процессы не имеют к нему прямого доступа и не могут изменять его.

Классификация Флинна



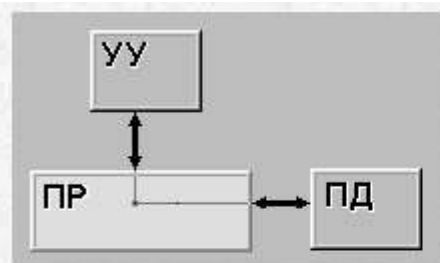
Варианты архитектур



Таксономия (Классификация) Флинна — общая классификация архитектур ЭВМ по признакам наличия параллелизма в потоках команд и данных.

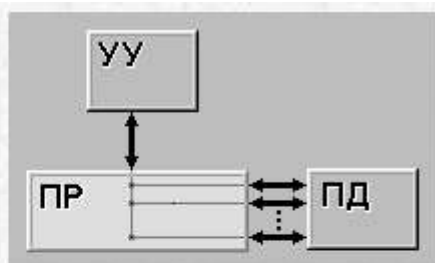
Классификация Флинна

SISD



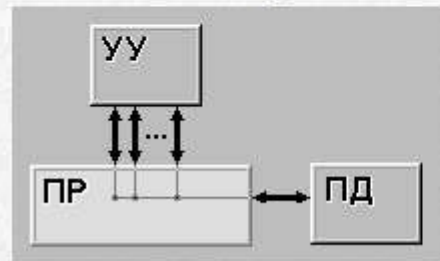
а)

SIMD



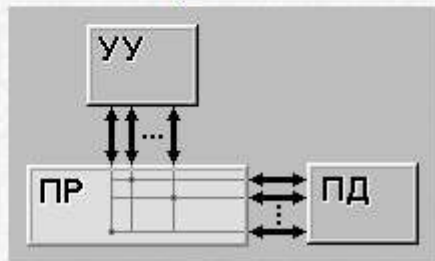
б)

MISD



в)

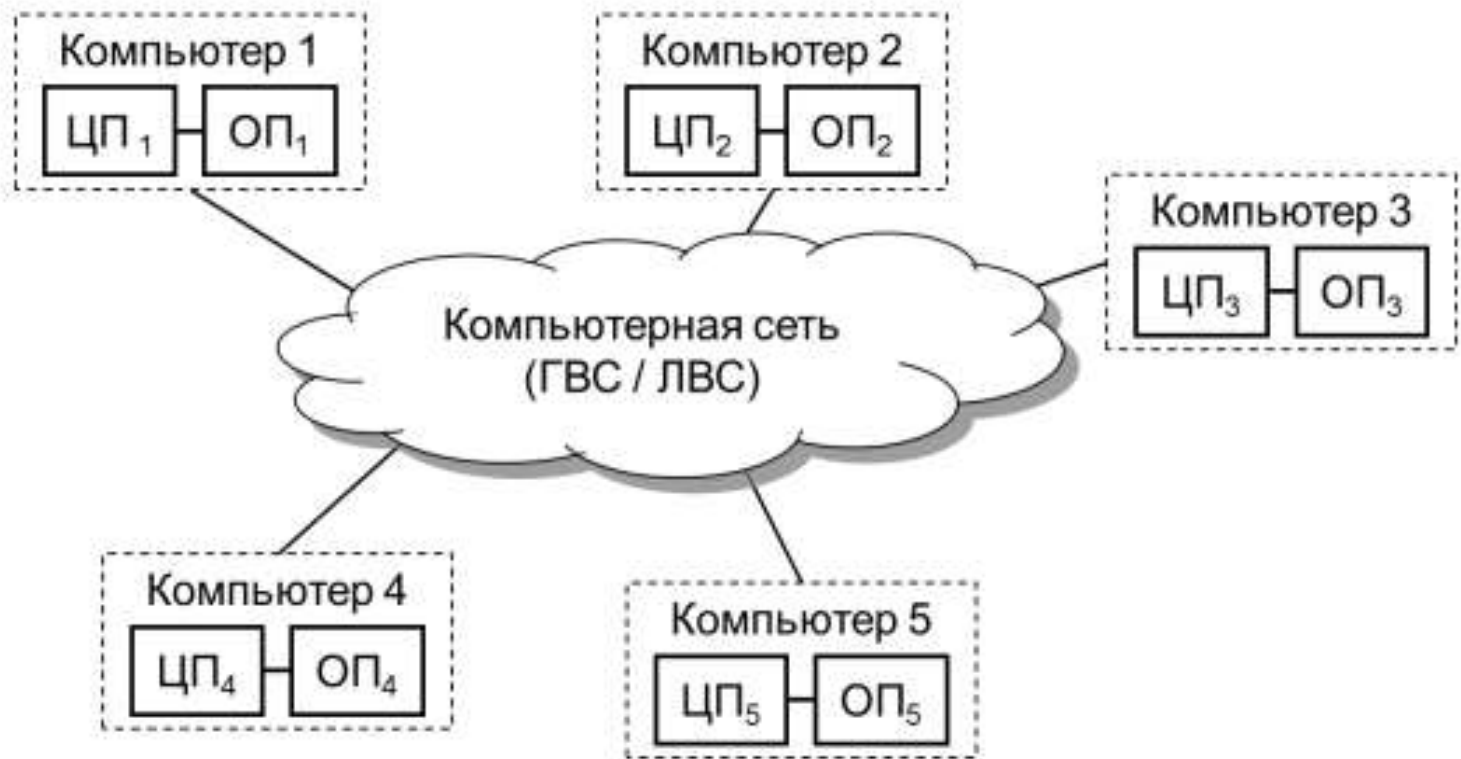
MIMD



г)

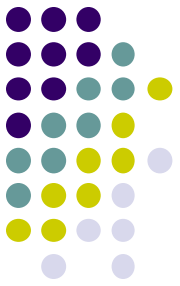
УУ — управляющее устройство (организует поток команд), ПР — процессор, ПД — поток данных

Типичная распределенная система



Структурная схема распределенной системы

Типы распределенных систем



- Сильно связанные системы (условно распределенные)
 - Общая память → Системы параллельной обработки (мультипроцессорные)



Типы распределенных систем



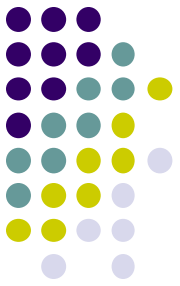
- Слабо связанные системы (реально распределенные)
 - Разделенная память → Системы распределенных вычислений (мультикомпьютерные)





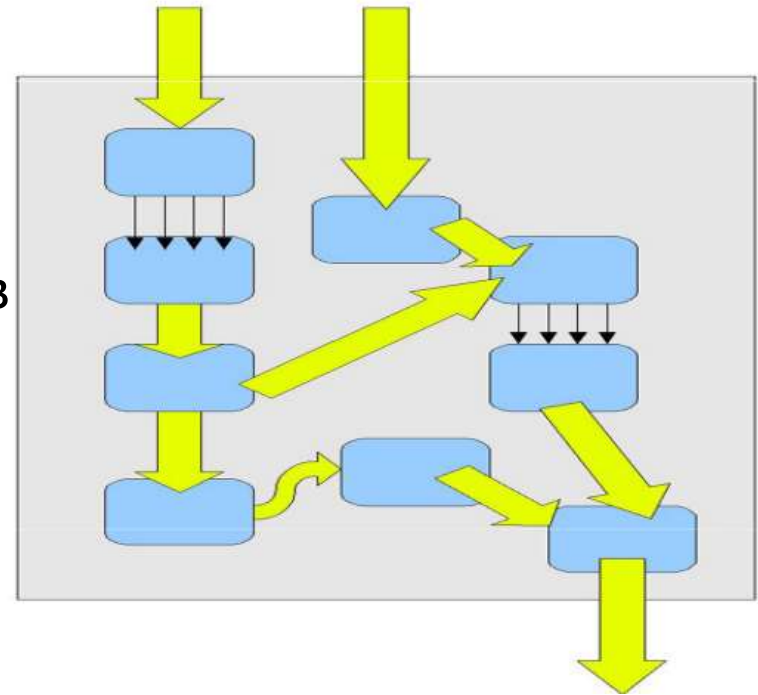
Отличительные признаки распределенных систем

Отличительные признаки распределенных систем (1/6)

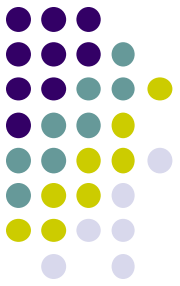


● Параллельность

- Независимые процессы
 - Синхронизация
- Необходимость разделения ресурсов
 - Данные
 - Сервисы
 - Устройства
- Типичные проблемы
 - Deadlocks
 - Ненадежные коммуникации (проблема освобождения ресурсов)



Отличительные признаки распределенных систем (2/6)



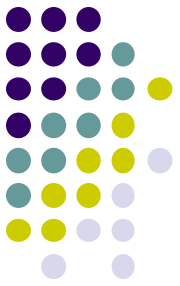
- Отсутствие “глобального” времени
 - Асинхронная передача сообщений
 - Ограниченная точность синхронизации часов
- Нет единого известного всем состояния системы
 - Нет ни одного процесса в распределенной системе, который бы знал текущее глобальное состояние системы
 - Следствие параллелизма и механизма передачи данных

Отличительные признаки распределенных систем (3/6)



- Отсутствие общей памяти
 - ключевая характеристика, из которой следует необходимость *обмена сообщениями* между программными компонентами распределенной системы для их взаимодействия и синхронизации.
 - некоторые распределенные системы могут предоставлять своим пользователям абстракцию единого адресного пространства для всех процессоров с помощью механизмов распределенной разделяемой памяти (англ. Distributed Shared Memory, DSM) – общая виртуальная память.

Отличительные признаки распределенных систем (4/6)

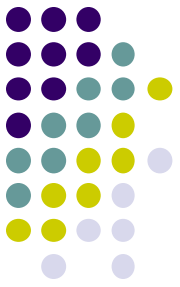


- Проблемы отказоустойчивости

Процессы выполняются автономно, изолированно, следовательно:

- Неудачи отдельных процессов могут остаться необнаруженными
- Отдельные процессы могут не подозревать об общесистемном сбое
- Сбои происходят чаще, чем в централизованной системе
- Появляются новые причины сбоев (которых не было в монолитных системах)
- Сетевые сбои фрагментируют систему

Отличительные признаки распределенных систем (5/6)

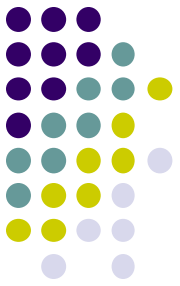


- Географическое распределение
 - Глобальная вычислительная сеть (ГВС).
 - Либо кластер из обыкновенных рабочих станций (англ. Cluster Of Workstation, COW), соединенных с помощью локальной вычислительной сети (ЛВС)

Подобные кластеры COW становятся все популярнее из-за относительно низкой стоимости входящих в нее компонентов с одной стороны и неплохой производительности – с другой. Например, ядро поисковой системы компании Google было изначально построено по архитектуре COW.



Отличительные признаки распределенных систем (6/6)



- Независимость и гетерогенность
 - различный состав, возможно и различные ОС
 - различная производительность
 - различное время выполнения идентичных задач
 - но при этом совместное функционирование с предоставлением своих служб друг другу для выполнения общей задачи.



Цели построения распределенных систем

Цели построения распределенных систем (1/4)



1. Географически распределенная вычислительная среда:

В качестве примера можно привести межбанковскую сеть. Каждый банк обслуживает счета своих клиентов и обрабатывает операции с ними. В случае же перевода денег из одного банка в другой требуется осуществление межбанковской транзакции и взаимодействие систем банков друг с другом.

- **Функциональное разделение:** узлы выполняют различные задачи
 - Клиент / сервер
 - Хост / Терминал
 - Сборка данных/ обработка данных
 - Решение - создание разделяемых сервисов
- **Естественное разделение** (определяемое задачей)
 - Система обслуживания сети супермаркетов
 - Сеть для поддержки коллективной работы

Цели построения распределенных систем (2/4)

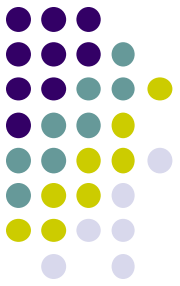


2. Требование увеличения производительности вычислений

Чтобы повысить производительность в сотни или тысячи раз и при этом обеспечивать хорошую масштабируемость решения необходимо свести воедино многочисленные процессоры и обеспечить их эффективное взаимодействие. Этот принцип реализуется в виде больших многопроцессорных систем и многомашинных комплексов (*кластер*).

- **Задача: Распределение нагрузки/балансировка:** назначение задачи на процессоры так, чтобы оптимизировать общую загрузку системы.
- **Результат: Усиление мощности:** различные узлы работают над одной задачей
 - Распределенные системы, содержащие набор микропроцессоров, по мощности могут приближаться к суперкомпьютеру
 - 10000 CPU, каждый 50 MIPS, вместе 500000 MIPS -> команда выполняется за 0.002 nsec, за это время свет проходит 0.6 mm -> любой существующий чип ведет вычисления дольше!

Цели построения распределенных систем (3/4)



3. Совместное использование ресурсов

- **Физическое разделение:** система строится в предположении, что узлы физически разделены (требования к надежности, устойчивости к сбоям).
- **Экономическая целесообразность:** набор дешевых чипов может обеспечить лучшие показатели отношения цена/производительность, чем мейнфрэйм
 - Мейнфрэйм: в 10 раз быстрее и в 1000 раз дороже.



Совместное использование

- дискового пространства
- вычислительной мощности
- программных компонентов

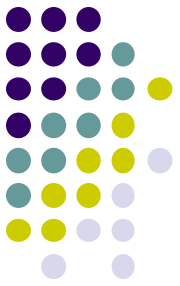
Цели построения распределенных систем (4/4)



4. Обеспечение отказоустойчивости

- **Распределение нагрузки/балансировка:** назначение задачи на процессоры так, чтобы оптимизировать общую загрузку системы, а значит, уберечь от сбоев.
- **Усиление мощности и надежности:** различные узлы работают над одной задачей
- **устойчивость к частичным отказам** за счет перераспределения нагрузки
 - распределенная система пытается скрывать факты отказов или ошибок в одних процессах от других процессов.

Требования к распределенным системам



1. Прозрачность (англ. transparency),
2. Открытость (англ. openness),
3. Безопасность (англ. security),
4. Масштабируемость (англ. scalability).

Несмотря на кажущуюся простоту и очевидность перечисленных свойств, их реализация на практике часто представляет собой непростую задачу.

Требования к распределенным системам



- Прозрачность

Способность скрывать свою распределенную природу, а именно, распределение процессов и ресурсов по множеству компьютеров, и представляться для пользователей и разработчиков приложений в виде единой централизованной компьютерной системы.

Стандарты эталонной модели для распределенной обработки в открытых системах Reference Model for Open Distributed Processing (RM-ODP) определяют *несколько типов прозрачности*.

Требования к распределенным системам



- Прозрачность

Доступа: доступ к локальным и удаленным ресурсам посредством одинаковых вызовов;

Расположения: доступ к ресурсам вне зависимости от их физического расположения;

Параллелизма: возможность нескольким процессам параллельно работать с ресурсами, не оказывая влияния друг на друга;

Репликации: возможность нескольким экземплярам одного ресурса использоваться без знания физических особенностей репликации;

Обработки ошибок: Защита программных компонентов от сбоев, произошедших в других программных компонентах. Восстановление после сбоев;

Прозрачность мобильности: Возможность переноса приложения между платформами, без его переделки;

Прозрачность производительности: возможность конфигурации системы с целью увеличения производительности при изменении состава платформы выполнения;

Прозрачность масштабируемости: возможность увеличения производительности без изменения структуры программной системы и используемых алгоритмов.

Требования к распределенным системам



- Прозрачность

Прозрачность	Описание
Доступ	Скрыть различия в представлении данных и то, как получен доступ к объекту
Местоположение	Скрыть, где находится объект
Перемещение	Скрыть, что объект может быть перемещен в другое место, когда используется
Миграция	Скрыть, что объект может переместиться в другое место
Репликация	Скрыть, что объект тиражируется или дублируется
Параллельность	Скрыть, что объект может быть разделен между несколькими независимыми пользователями
Отказ	Скрыть сбой и восстановление объекта

Различные формы прозрачности в распределенной системе.
Объект может быть ресурсом или процессом

Требования к распределенным системам



- Прозрачность

Очень важна для распределенных систем

- Прозрачность доступа и физического расположения имеет критическое значение для должного использования распределенных ресурсов.

Степень прозрачности. Важно отметить, что степень, до которой каждое из перечисленных свойств должно быть выполнено, может сильно варьироваться в зависимости от задач построения распределенной системы. Действительно, полностью скрыть распределение процессов и ресурсов вряд ли удастся. Из-за ограничения в скорости передачи сигнала, задержка на обращение к ресурсам, территориально удаленным от клиента, всегда будет больше, чем к ресурсам, расположенным поблизости.

Требования к распределенным системам



Открытость

- Гарантирует расширяемость
- Мобильность приложений
- Интероперабельность
- Мобильность пользователя
- Возможность повторного использования
- Важные факторы:
 - Наличие четких спецификаций
 - Наличие полной документации
 - Опубликованные интерфейсы
 - Тестирование и проверка на многих платформах

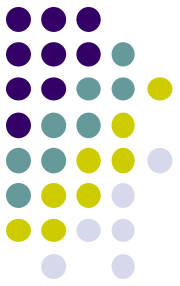
Требования к распределенным системам



Безопасность

- Три компонента:
 - Защищенность
 - Целостность
 - Доступность
- Задача: посылка значимой информации по сети безопасно и эффективно

Требования к распределенным системам



Безопасность

- Сценарий 1: Доступ к результатам тестирования студентов
 - Откуда мы знаем, что пользователь - преподаватель, имеющий доступ к данным?
 - Решение – *Авторизация*
- Сценарий 2: Посылка номера кредитной карты в интернет-магазин
 - Никто кроме получателя не должен прочитать данные
 - Решение – *Криптография*

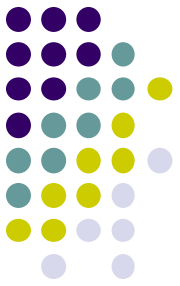
Требования к распределенным системам

Масштабируемость



- **Нагрузочная масштабируемость** - способность системы увеличивать свою производительность при увеличении нагрузки путем замены существующих аппаратных компонентов на более мощные или путем добавления новых аппаратных средств.
 - увеличение производительности каждого компонента системы с целью повышения общей производительности называют **вертикальным масштабированием**
 - увеличение количества сетевых компьютеров (серверов) распределенной системы – **горизонтальным масштабированием**.
- **Географическая масштабируемость** - способность системы сохранять свои основные характеристики, такие как производительность, простота и удобство использования, при территориальном разнесении ее компонентов от более локального взаиморасположения до более распределенного.
- **Административная масштабируемость** характеризует простоту управления системой при увеличении количества административно независимых организаций, обслуживающих части одной распредел. системы.

Примеры организации распределенных систем

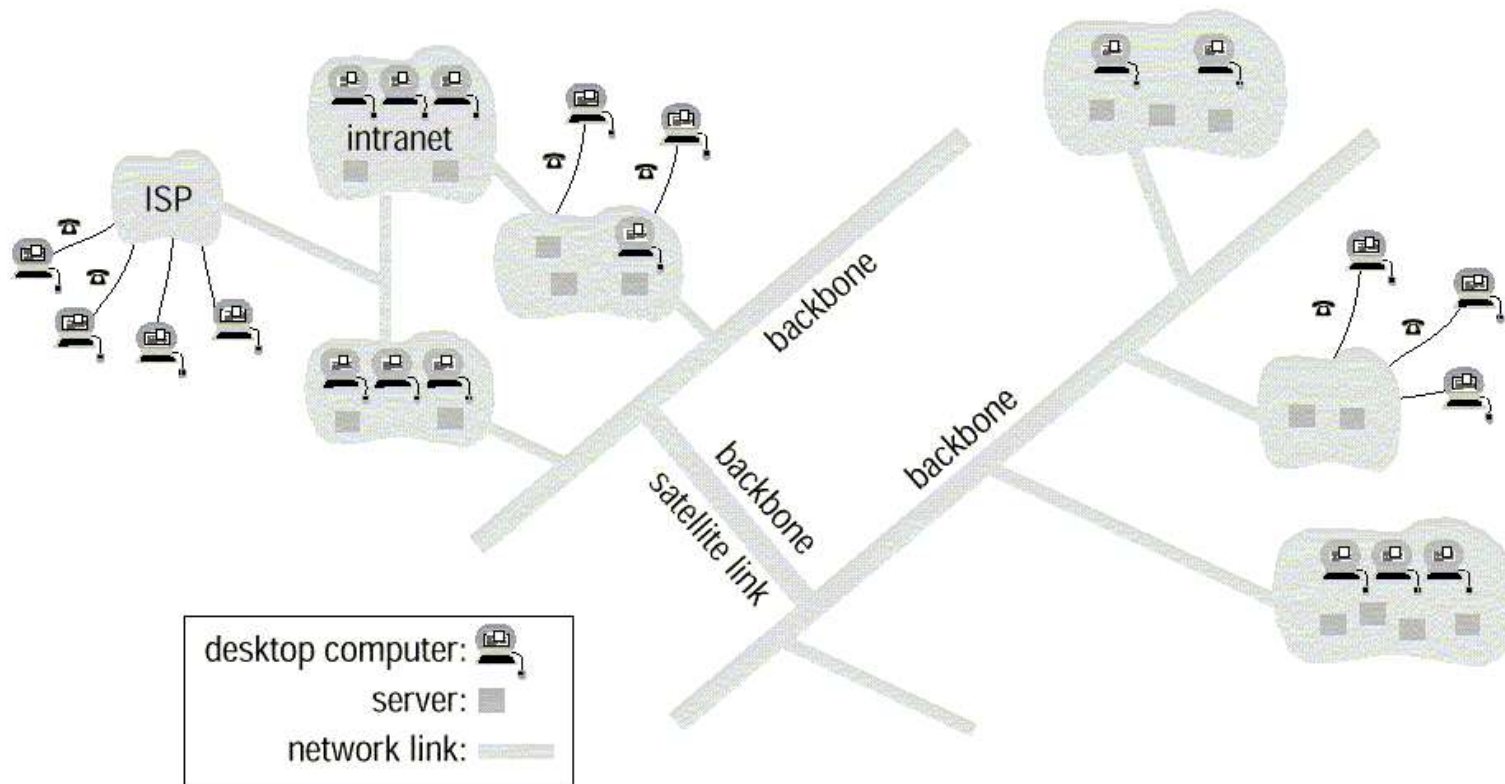


- Internet
- Intranet
- Вычислительные кластеры
- ...

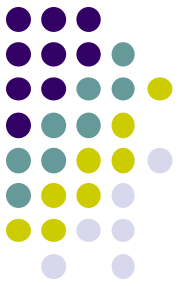
Пример 1: Internet



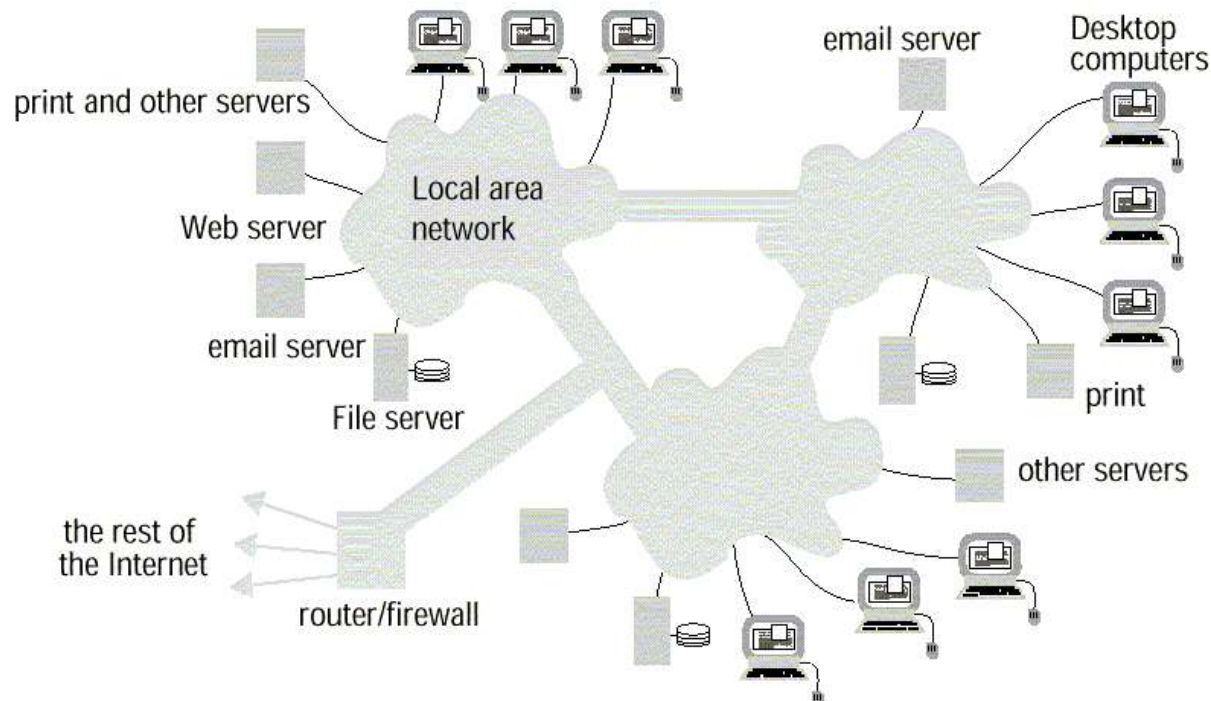
- Гетерогенная сеть компьютеров и приложений
- Реализация взаимодействия – стек TCP/IP



Пример 2: Intranet



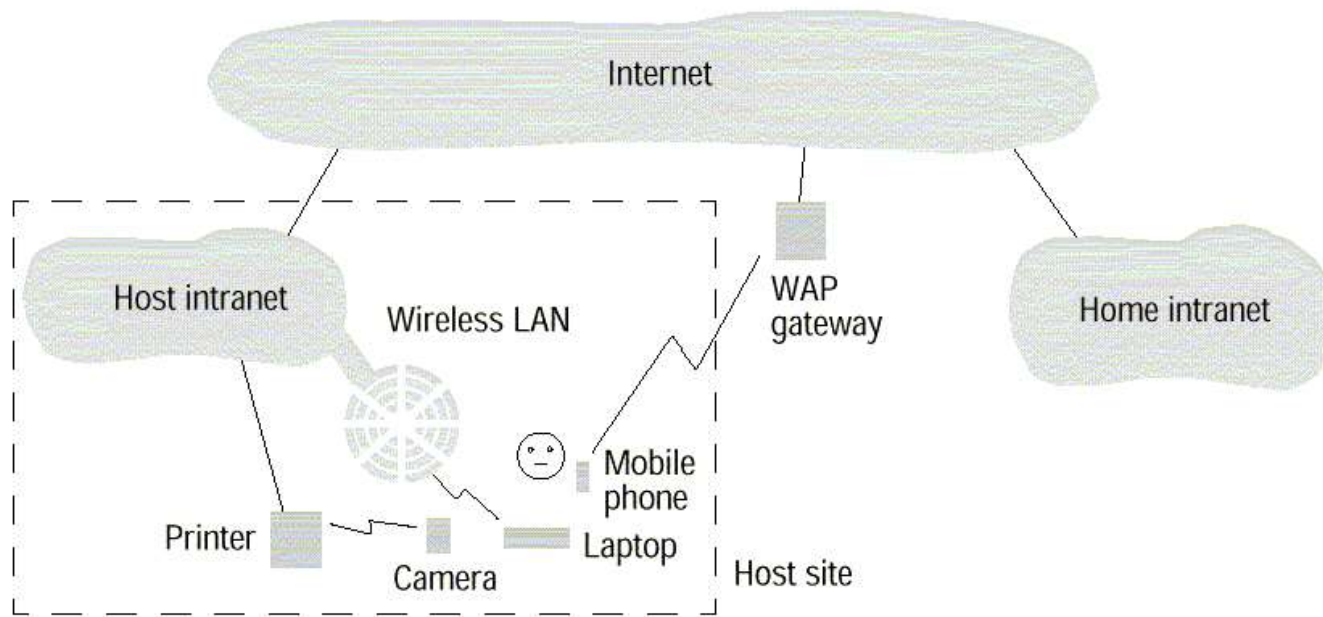
- Администрируется локально
- Взаимодействие с Internet
- Обеспечивает сервисами (внутренних и внешних пользователей)



Пример 3: Wireless Information Devices



- Система сотовой связи (GSM)
 - Ресурсы разделяемы (радио частота, время передачи на частоте,...)
- Laptop (подключаются к Wireless LAN)
- Handheld, PDAs etc.





Другие примеры

- Системы управления аэропортом
 - Интернет-система продажи билетов
 - Автомобильные управляющие системы
- Mercedes S класса сегодня имеет более 50 автономных встроенных процессоров соединенных общей шиной





Примеры...

- Телефонные системы
- Сложные сети предприятий
- Сетевые файловые системы
- WWW
- Сеть газопроводов, электроснабжения
- Логистические системы
- Банковская система
- Медицинская система
- Государственное и муниципальное управление
- Сети корпораций
- и многое другое...

Ошибки при проектировании



Сотрудник компании Sun Microsystems Питер Дейч в своей статье "Восемь заблуждений относительно распределенных вычислений" сформулировал основные ошибки, которые допускают при создании распределенных приложений.

1. Сеть является надежной.
2. Сеть является безопасной.
3. Сетевая топология неизменна.
4. Полоса пропускания не ограничена.
5. Задержки передачи сообщений равны нулю.
6. Систему обслуживает только один администратор.
7. Издержки на транспортную инфраструктуру равны нулю.
8. Сеть является однородной.

Проблемы при проектировании DS



1. Параллелизм
2. Прозрачность
3. Управляемость
4. Гетерогенность
5. Открытость
6. Безопасность
7. Масштабируемость
8. Обработка ошибок и восстановление после сбоев
9. Распространение приложения



Параллелизм

- Контроль параллелизма
 - Обращение нескольких потоков к ресурсу
 - Правильное планирование доступа в параллельных потоках (устранение взаимoisключений, транзакции)
 - Синхронизация (семафоры)
 - Безопасно, но уменьшают производительность
 - Разделяемые объекты (ресурсы) должны работать корректно в многопоточной среде



Управляемость

- Распределенные ресурсы не имеют центральной точки управления
- Локальная оптимизация не всегда означает глобальную оптимизацию
 - Нужен глобальный взгляд на проблему
 - Не всегда возможен (есть системы, никому конкретно не принадлежащие)



Гетерогенность

Гетерогенные = разные

- Различные
 - сетевые инфраструктуры,
 - hardware&software (пример Intel & Motorola, UNIX sockets & Winsock calls),
 - языки программирования (и представления данных!!!)
- Различия должны быть скрыты



Гетерогенность

- Интерфейсы и реализация могут быть разными
 - Базовые концепции обычно неизменны
- Вывод: необходимы стандарты



Гетерогенность

стандартизация

- **Middleware:** промежуточный программный слой
 - позволяет гетерогенным узлам взаимодействовать
 - Определяет однородную вычислительную модель
 - Поддерживает один или несколько языков программирования
 - Обеспечивает поддержку распределенных приложений
 - Вызов удаленных объектов
 - Удаленный вызов SQL
 - Распределенная обработка транзакций
- Примеры: CORBA, Java RMI, Microsoft DCOM



Гетерогенность

средства достижения общности

- Мобильный код: код разработан для миграции между узлами
 - Необходимо преодолевать аппаратные различия (разные наборы инструкций)
- Виртуальные машины
 - Компилятор «изготавливает» байт-код для VM
 - VM реализована для всех аппаратных платформ (Java)
- Методы грубой силы
 - Портируем код под каждую платформу...



Безопасность

Нерешенные проблемы:

- Атаки типа DoS (отказы в обслуживании)
- Небезопасность мобильного кода
 - Непредсказуемые эффекты
 - Может вести себя подобно троянскому коню...

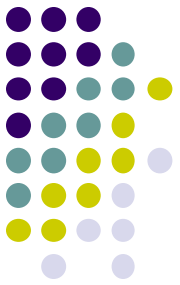
Масштабируемость



Распределенная система масштабируема, если она остается эффективной при увеличении числа обслуживаемых пользователей или ресурсов

- Проблемы:
 - Контроль стоимости ресурсов
 - Контроль потерь производительности
- Существуют естественные ограничения
 - Некоторые определяются легко
 - Другие труднее
- Обход узких мест
 - Децентрализация алгоритмов
 - Пример - Domain Name Service
 - Тиражирование и кэширование данных

Масштабируемость



Технологии масштабирования

- распространение (англ. distribution),
- репликация (англ. replication)
- кэширование (англ. caching).

Распространение предполагает разбиение множества поддерживаемых ресурсов на части с последующим разнесением этих частей по компонентам системы. Простым примером распространения может служить распределенная файловая система при условии, что каждый файловый сервер обслуживает свой набор файлов из общего адресного пространства.

Распространение и репликация позволяют распределить поступающие в систему запросы по нескольким ее компонентам, в то время как **кэширование** уменьшает количество повторных обращений к одному и тому же ресурсу.



Масштабируемость

- Стоимость физических ресурсов
 - Растет, при увеличении числа пользователей
 - Не должна расти быстрее, чем $O(n)$, где n = количество пользователей
- Потери производительности
 - Увеличиваются с ростом размера данных (и количества пользователей)
 - Время поиска не должно расти быстрее, чем $O(\log m)$, где m = размер данных



Обработка сбоев

- Сбои более частые, чем в централизованных системах, но обычно локальные
- Обработка сбоев включает
 - Определение факта сбоя (может быть невозможно)
 - Маскирование
 - Восстановление



Обработка сбоев

- Диагностика
 - Может быть возможна (ошибки передачи - контрольная сумма)
 - Может быть невозможна (удаленный сервер не работает или просто очень загружен?)
- Маскирование
 - Многие сбои могут быть скрыты
 - Может быть невозможно (все диски повреждены)
 - Не всегда хорошо

Распространение приложения



- Фрагментация
 - разделение приложения на модули для распространения
- Конфигурация
 - Связь модулей друг с другом (зависимости)
- Размещение
 - выгрузка модулей на целевую систему
 - Распределение вычислительных модулей между узлами (статическое или динамическое)

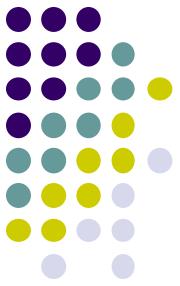
Итоги



- Распределенные системы это:
 - Автономные (но соединенные средой передачи данных) узлы
 - Взаимодействие осуществляется посредством передачи сообщений
- Много примеров того, что распределенные системы нужны и их нужно уметь строить
- Распределенные системы существуют и их нужно уметь развивать и поддерживать, учитывая перечисленные требования, сложности и стандарты.

Требования к распределенным системам

Вопросы к экзамену (зачету):



1. Дайте определение DS
2. Перечислите и охарактеризуйте типы распределенных систем.
3. Перечислить требования к DS
4. Какова роль программного обеспечения промежуточного уровня в распределенных системах?
5. Объясните, что такое прозрачность (распределения) и приведите примеры различных видов прозрачности.
6. Почему иногда так трудно скрыть наличие в распределенной системе сбоя и восстановление после него?
7. Почему реализация максимально возможной степени прозрачности — это не всегда хорошо?
8. Что такое открытая распределенная система и какие преимущества дает открытость?
9. Опишите точно, что такое масштабируемая система.
10. Масштабируемости можно добиться, используя различные методики (технологии). Что это за методики?