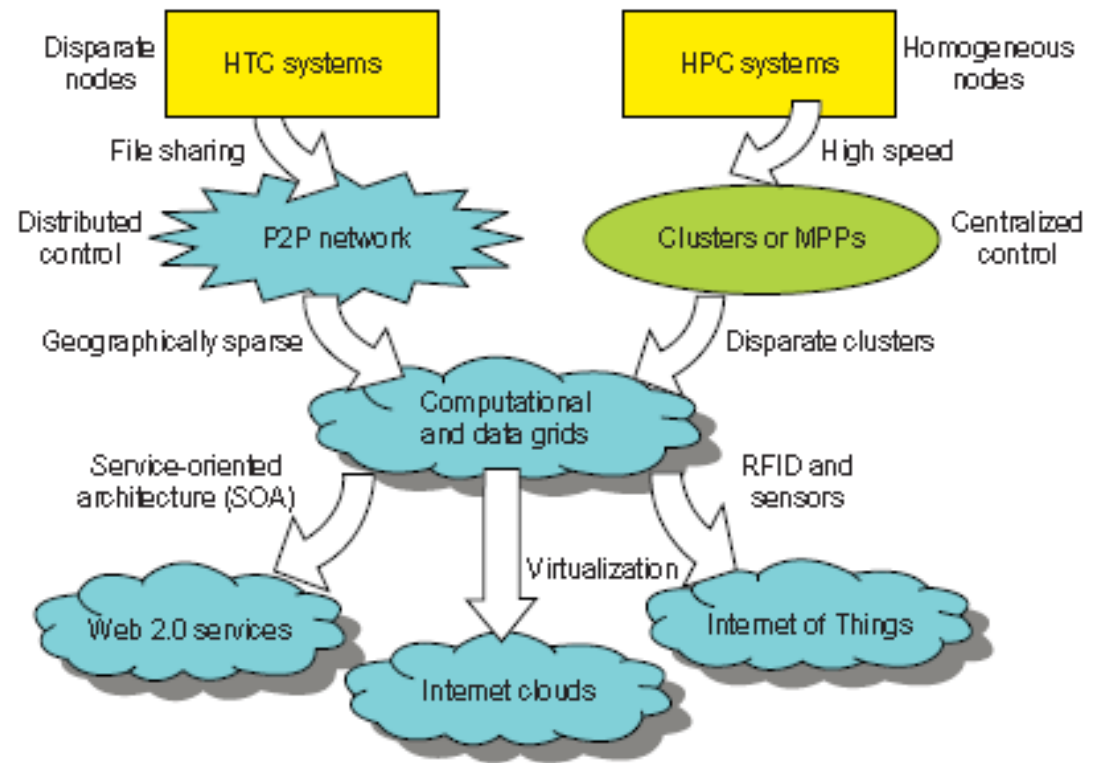


Введение в распределенные ИС

Типы распределенных систем

Эволюция распределенных систем



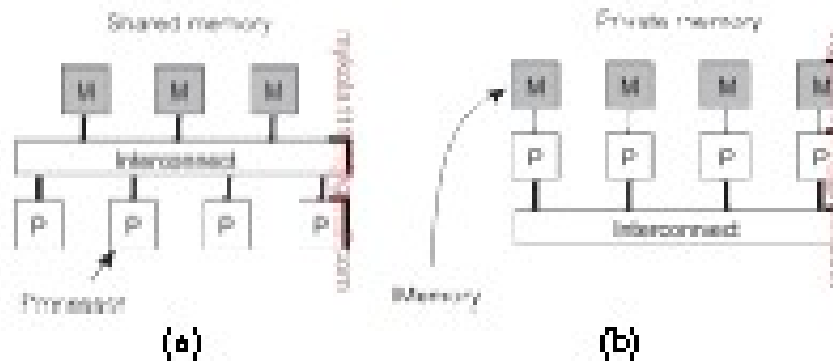
- Типы современных РС сформировались в процессе эволюции двух классов систем:
- Систем высокопроизводительных вычислений (HPC – High Performance Computing);
- Систем высокой пропускной способности (HTC – High Throughput Computing);

Типы распределенных систем

- Системы высокопроизводительных распределенных вычислений (High performance distributed computing).
- Распределенные информационные системы (Distributed information systems).
- Повсеместные - всеобъемлющие системы (Pervasive systems). Это мобильные и встраиваемые системы, сюда же относят системы слежения обладающие сетью датчиков и множеством исполнительных механизмов и др. РС.

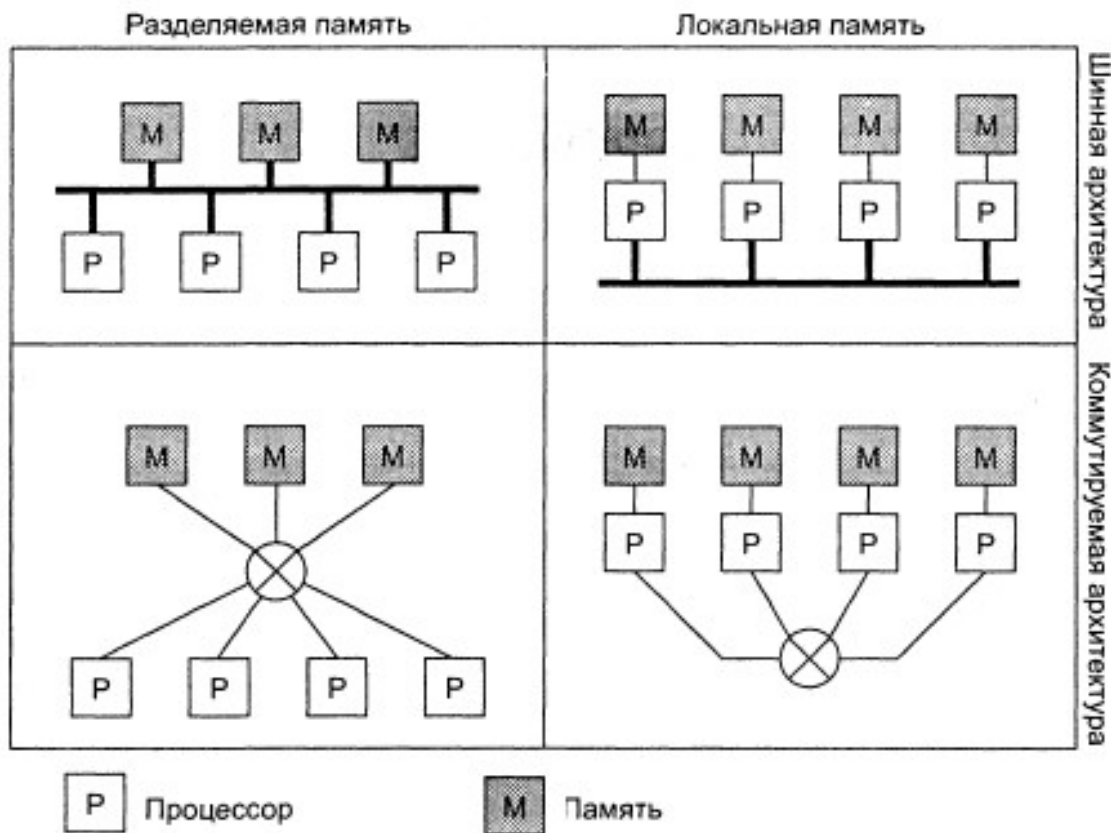
Системы высокопроизводительных распределенных вычислений (High performance distributed computing)

- Высокопроизводительные вычисления возникли с появлением мультимикропроцессорных и мультимашинных систем.



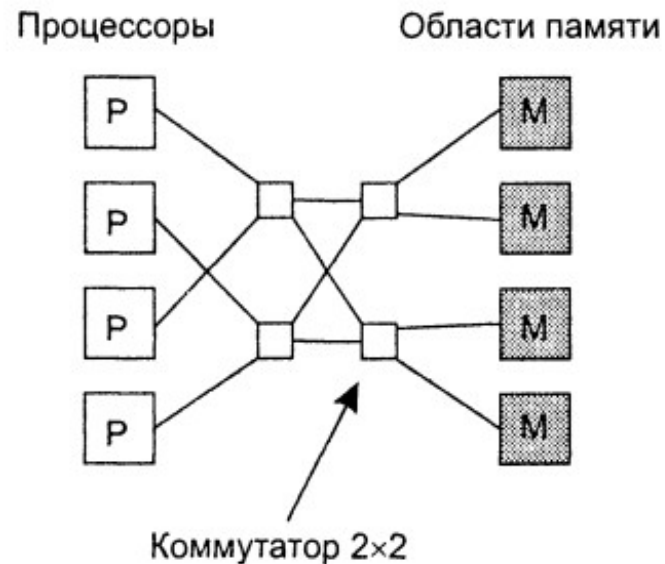
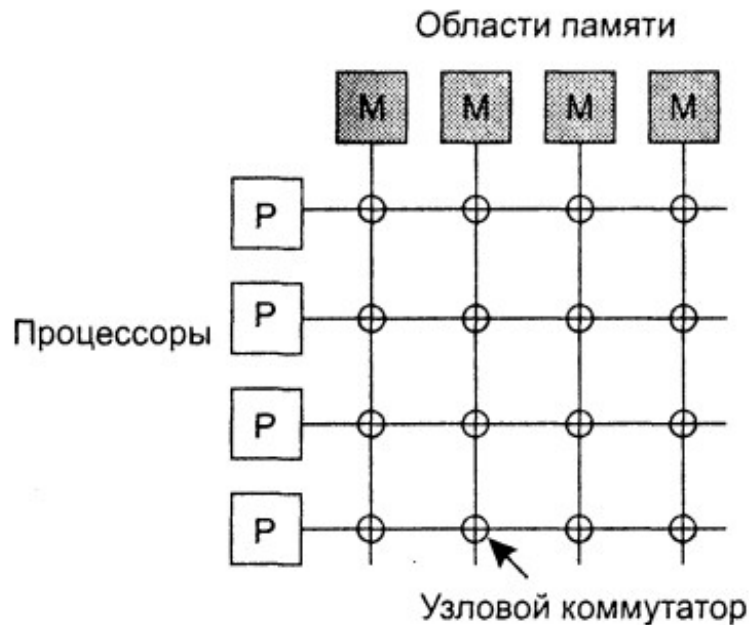
Концепции аппаратных решений для РИС

- Р ИС это системы построенные из набора независимых



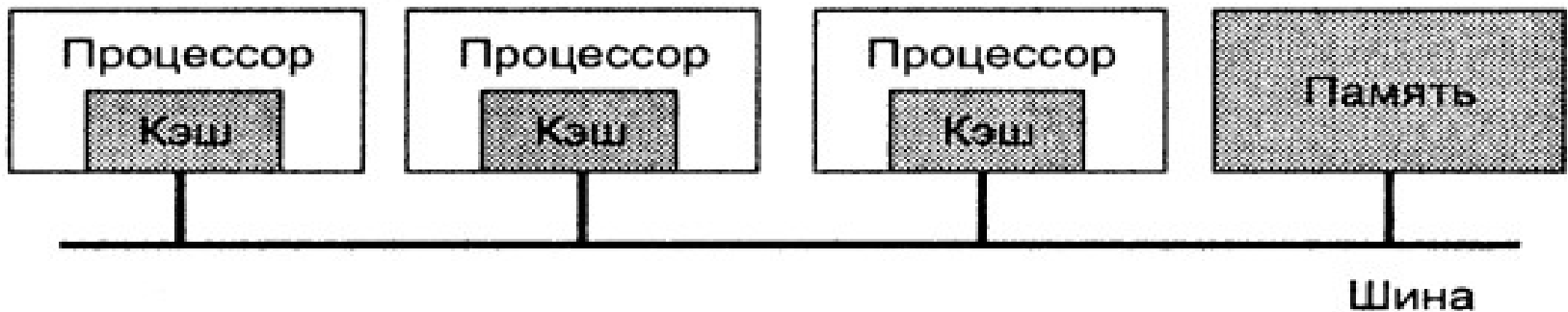
Все компьютеры можно разделить на две группы. Системы, в которых компьютеры используют память совместно, обычно называются *мультипроцессорами* (*multiprocessors*), а работающие каждый со своей памятью — *мультикомпьютерами* (*multicomputers*). Сегодня многомашинные системы состоят из многопроцессорных компьютеров

Решение проблемы масштабируемости шинной архитектуры



Мультипроцессоры

- Мультипроцессорные системы обладают одной характерной особенностью: все процессоры имеют прямой доступ к общей памяти.
- Введение кэша создает серьезные проблемы само по себе, а общая шина создает проблемы при большом числе процессоров



Многомашинные системы

- Это вычислительные системы обмен данными между которыми реализуется через устройства ввода вывода.
- Сегодня такими устройствами являются сетевые карты.
- Современные многомашинные системы представляют собой множество дешевых серверов объединяемых компьютерной сетью.

Классификация ВС Флинна

- Среди всех рассматриваемых систем классификации ВС наибольшее признание получила классификация, предложенная в 1966 году М. Флинном.
- В ее основу положено понятие потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором.
- В зависимости от количества потоков команд и потоков данных Флинн выделяет четыре класса архитектур:
 - SISD (Single Instruction stream/Single Data stream) - системы с одиночным потоком команд и одиночным потоком данных (ОКОД) ;
 - MISD (Multiple Instruction stream/Single Data stream) - системы с множественным потоком команд и одиночным потоком данных (МКОД);
 - SIMD (Single Instruction stream/Multiple Data stream) - системы с одиночным потоком команд и множественным потоком данных (ОКМД);
 - MIMD (Multiple Instruction stream/Multiple Data stream) - системы с множественным потоком команд и множественным потоком данных (МКМД)

◦ .

SISD

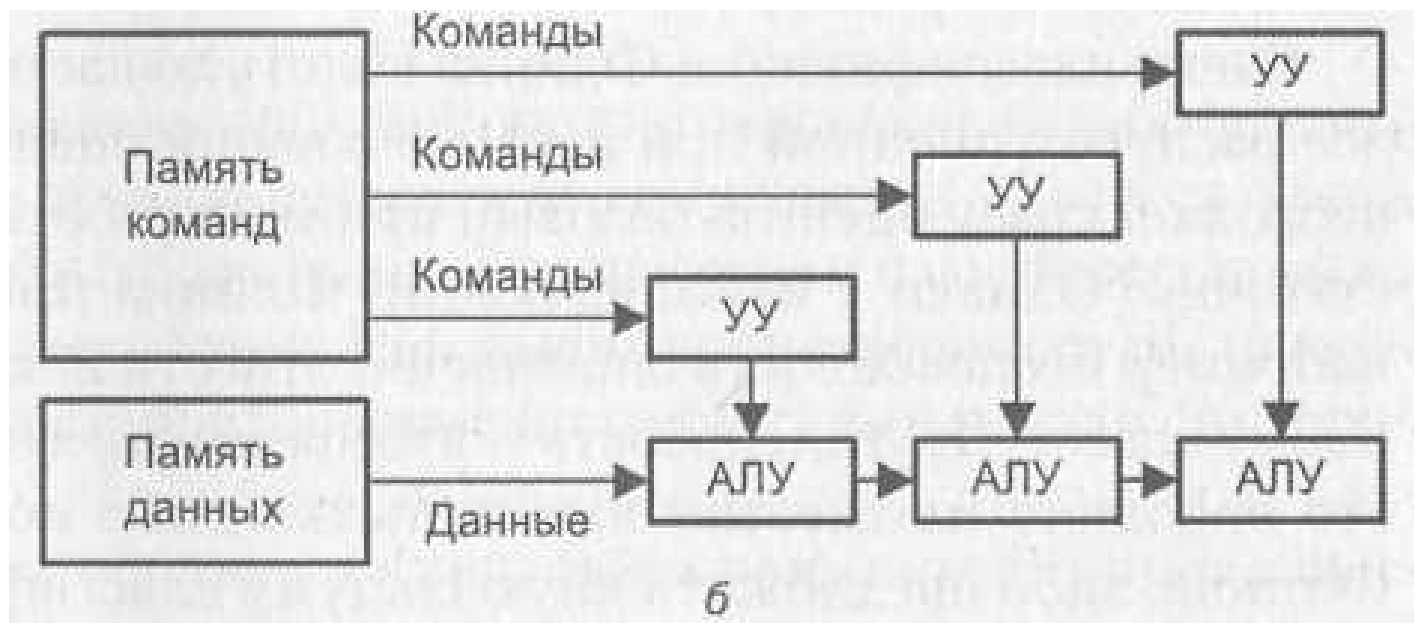


- *SISD* (Single Instruction Stream/Single Data Stream) — одиночный поток команд и одиночный поток данных.
- Представителями этого класса являются, прежде всего, классические фон-неймановские ВМ, где имеется только один поток команд, команды обрабатываются последовательно и каждая команда инициирует одну операцию с одним потоком данных.
- То, что для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться конвейерная обработка, не имеет значения, поэтому в класс SISD одновременно попадают как ЭВМ CDC 6600 со скалярными функциональными устройствами, так и CDC 7600 с конвейерными.

Вычислительные системы класса SISD

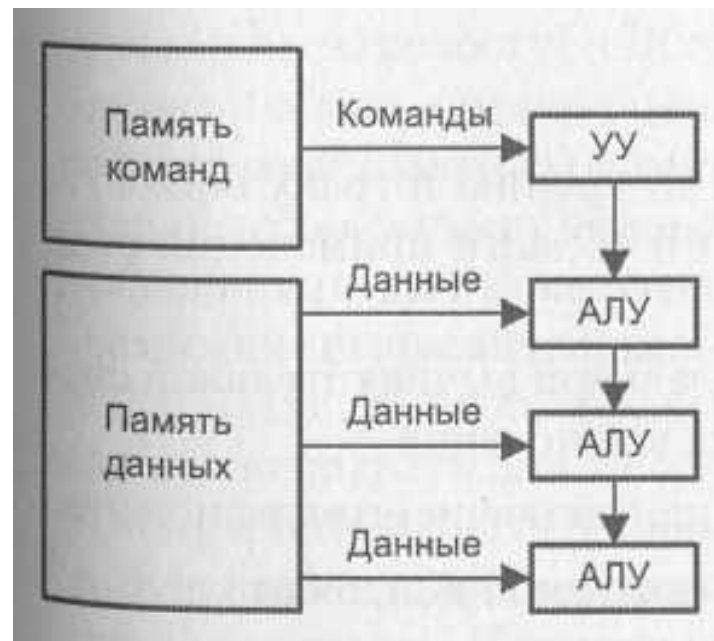
- Системы этого класса – обычные однопроцессорные ЭВМ включающие в себя запоминающее устройство (ЗУ) для команд и данных (оно чаще всего бывает общим) и один процессор, содержащий арифметическо-логическое устройство (АЛУ) и устройство управления (УУ).
- В современных системах этого класса наиболее широко используется первый путь организации параллельной обработки – совмещение во времени различных этапов решения разных задач, при котором в системе одновременно работают различные устройства: ввода, вывода и собственно обработки информации.

MISD



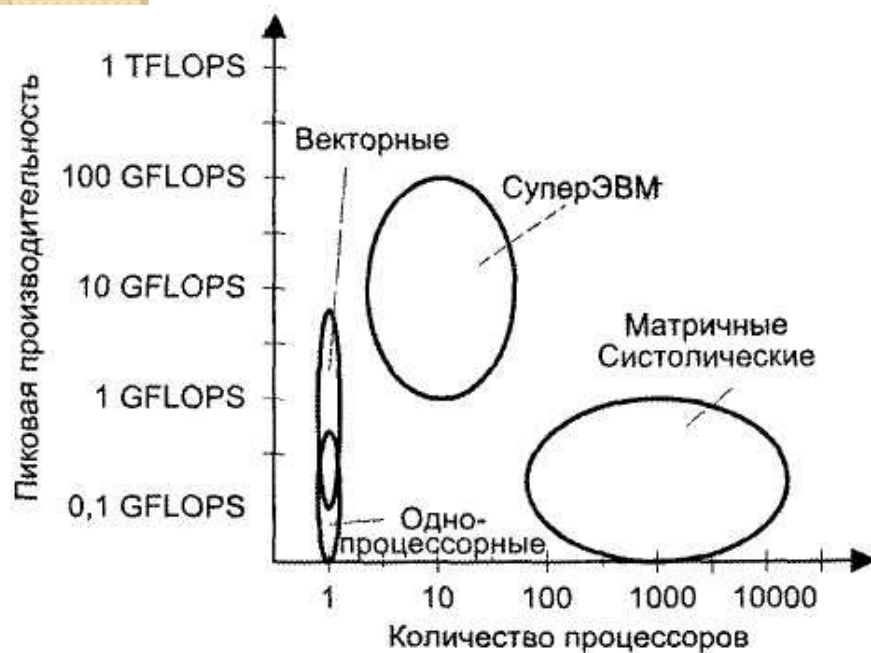
- *MISD* (Multiple Instruction Stream/Single Data Stream) — множественный поток команд и одиночный поток данных.
- Из определения следует, что в архитектуре ВС присутствует множество процессоров, обрабатывающих один и тот же поток данных.
- Ни Флинн, ни другие специалисты в области архитектуры компьютеров до сих пор не сумели представить убедительный пример реально существующей вычислительной системы, построенной на данном принципе.

SIMD



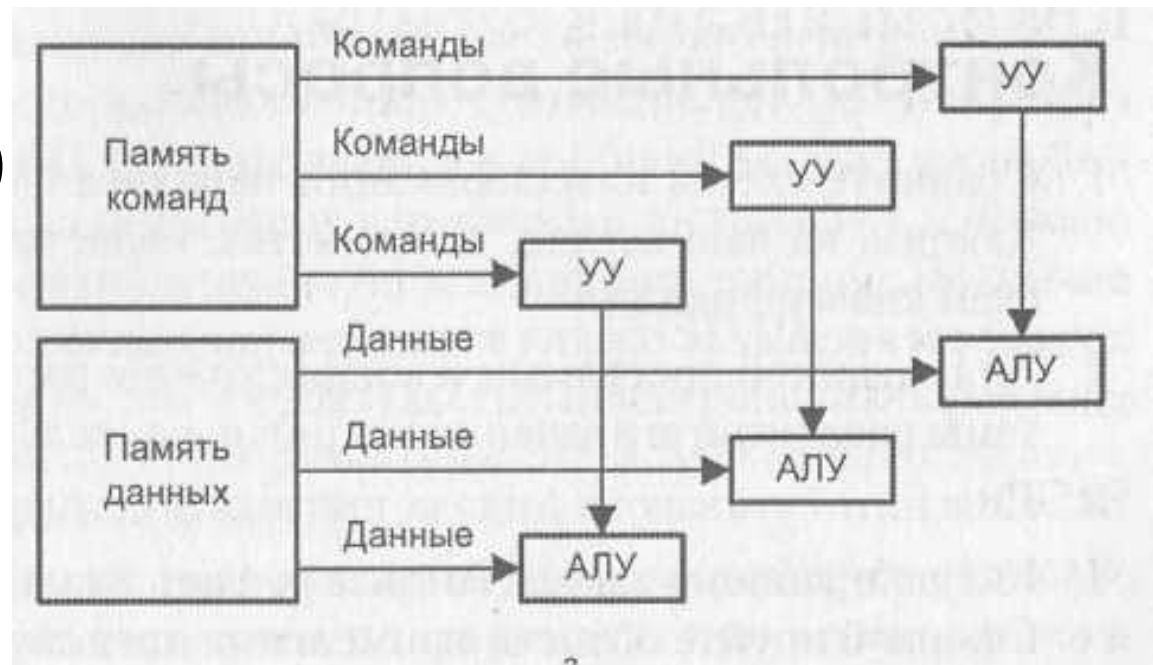
- Single Instruction Stream/Multiple Data Stream) — одиночный поток команд и множественный поток данных ЭВМ данной архитектуры позволяют выполнять одну арифметическую операцию сразу над многими данными — элементами вектора.
- Бесспорными представителями класса SIMD считаются матрицы процессоров, где единое управляющее устройство контролирует множество процессорных элементов. Все процессорные элементы получают от устройства управления одинаковую команду и выполняют ее над своими локальными данными. В принципе в этот класс можно включить и векторноконвейерные ВС, если каждый элемент вектора рассматривать как отдельный элемент потока данных.

Вычислительные системы класса SIMD



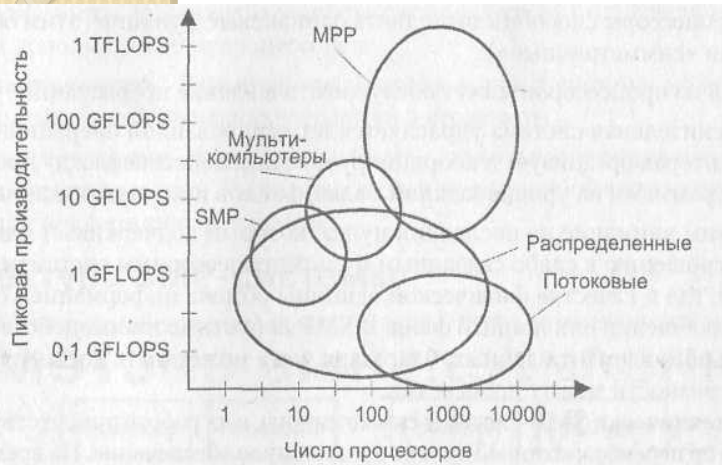
- SIMD-системы были первыми вычислительными системами, состоящими из большого числа процессоров, и среди первых систем, где была достигнута производительность порядка GFLOPS.
- Согласно классификации Флинна, к классу SIMD относятся ВС, где множество элементов данных подвергается параллельной, но однотипной обработке.
- SIMD-системы во многом похожи на классические фон-неймановские ВМ: в них также имеется одно устройство управления, обеспечивающее последовательное выполнение команд программы.
- Различие касается стадии выполнения, когда общая команда транслируется множеству процессоров (в простейшем случае — АЛУ), каждый из которых обрабатывает свои данные.
- класс SIMD составляют векторные (векторно-конвейерные), матричные, ассоциативные, систолические и VLIW-вычислительные системы.

MIMD



- *MIMD* (Multiple Instruction Stream/Multiple Data Stream) — множественный поток команд и множественный поток данных . Класс предполагает наличие в вычислительной системе множества устройств обработки команд, объединенных в единый комплекс и работающих каждое со своим потоком команд и данных. Класс MIMD чрезвычайно широк, поскольку включает в себя всевозможные мультипроцессорные системы.
- Класс MIMD чрезвычайно широк, поскольку включает в себя всевозможные мультипроцессорные системы.

Вычислительные системы класса MIMD



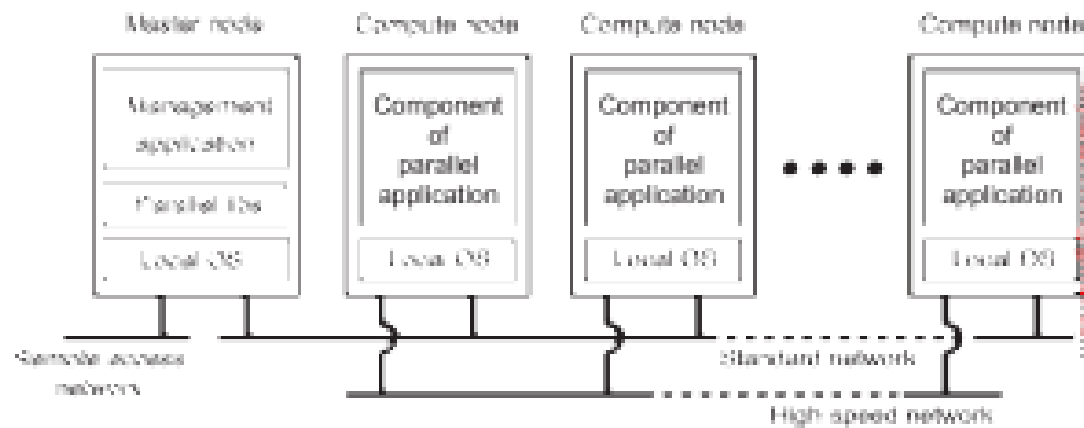
- В настоящее время тем не менее наметился устойчивый интерес к архитектурам класса MIMD.
- MIMD-системы обладают большей гибкостью, в частности могут работать и как высокопроизводительные однопользовательские системы, и как многопрограммные ВС, выполняющие множество задач параллельно.
- Кроме того, архитектура MIMD позволяет наиболее эффективно распорядиться всеми преимуществами современной микропроцессорной технологии.
- ВС класса MIMD образуют два больших класса систем:
 - Многопроцессорные
 - Многомашинные

Недостатки классификации Флинна

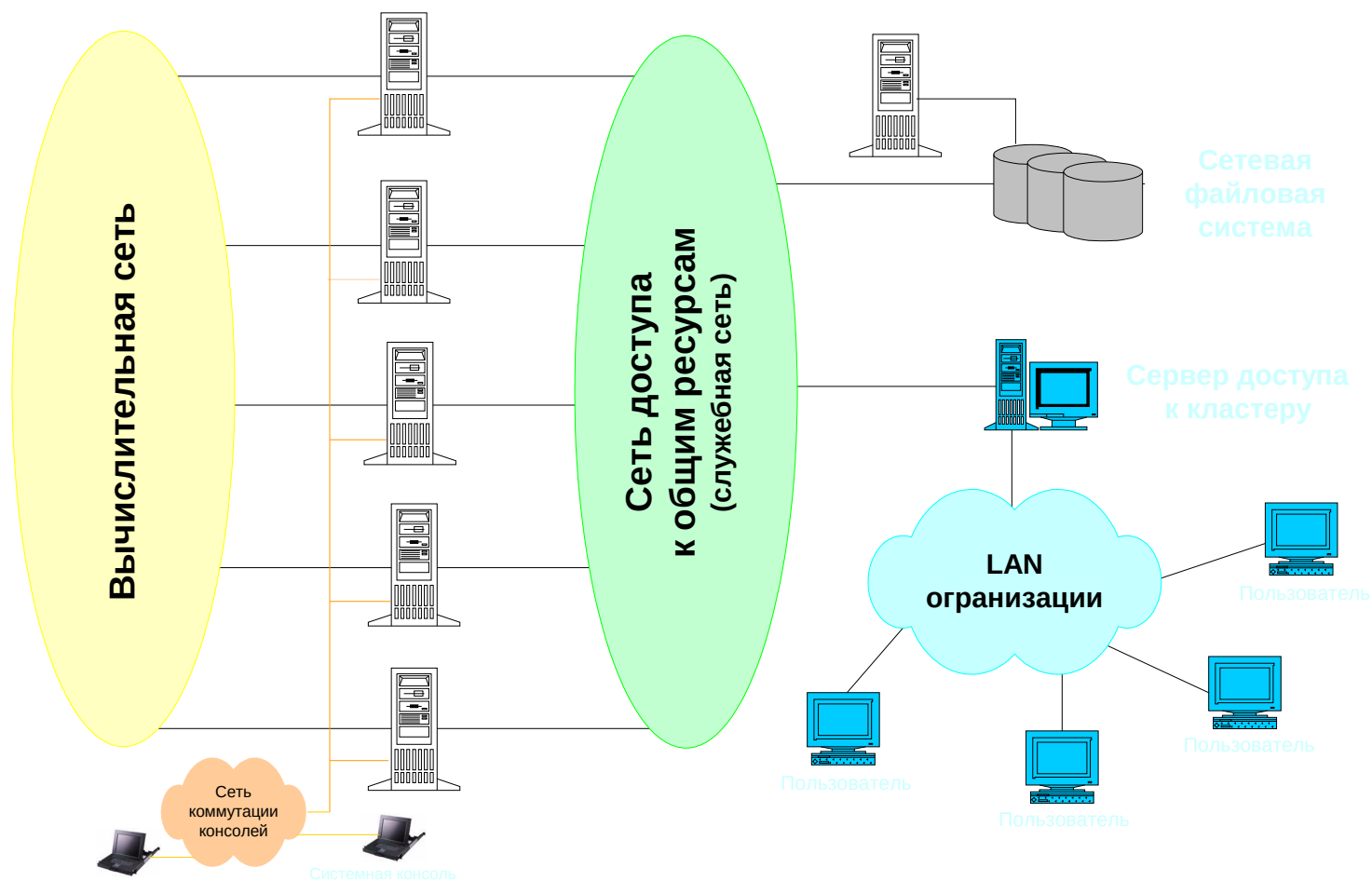
- Схема классификации Флинна до сих пор является наиболее распространенной.
- Она применяется при первоначальной оценке той или иной ВС, поскольку позволяет сразу оценить базовый принцип работы системы.
- Однако у классификации Флинна имеются и очевидные недостатки:
 - неспособность однозначно отнести некоторые архитектуры к тому или иному классу;
 - другая слабость — это чрезмерная насыщенность класса MIMD.
- Все это породило множественные попытки либо модифицировать классификацию Флинна, либо предложить иную систему классификации.

Кластерные системы

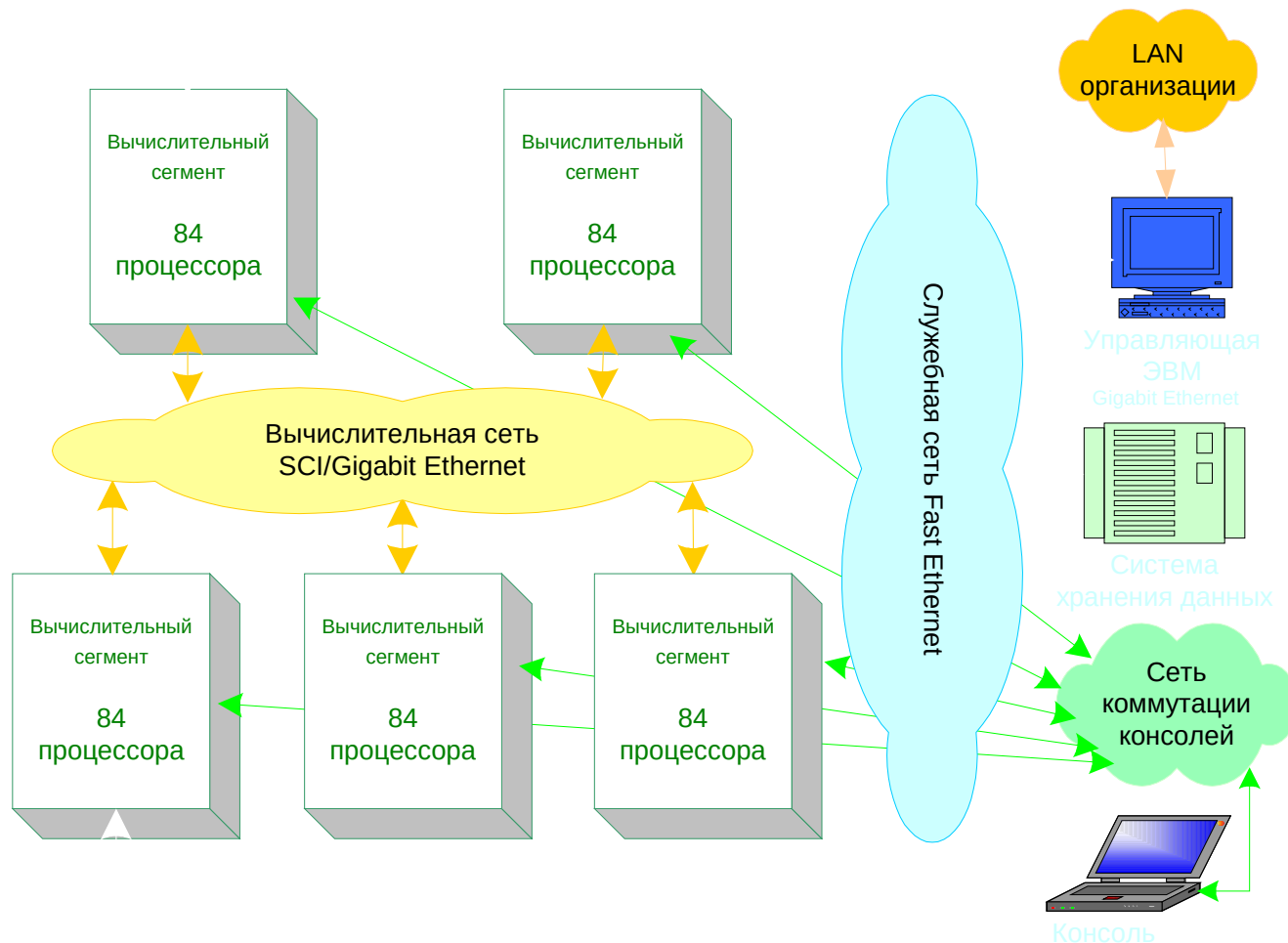
- Одним из примеров кластерных систем являются кластера Beowulf.
- В них различают управляющий и вычислительные узлы, объединяемые в многомашинную систему с помощью компьютерной сети.



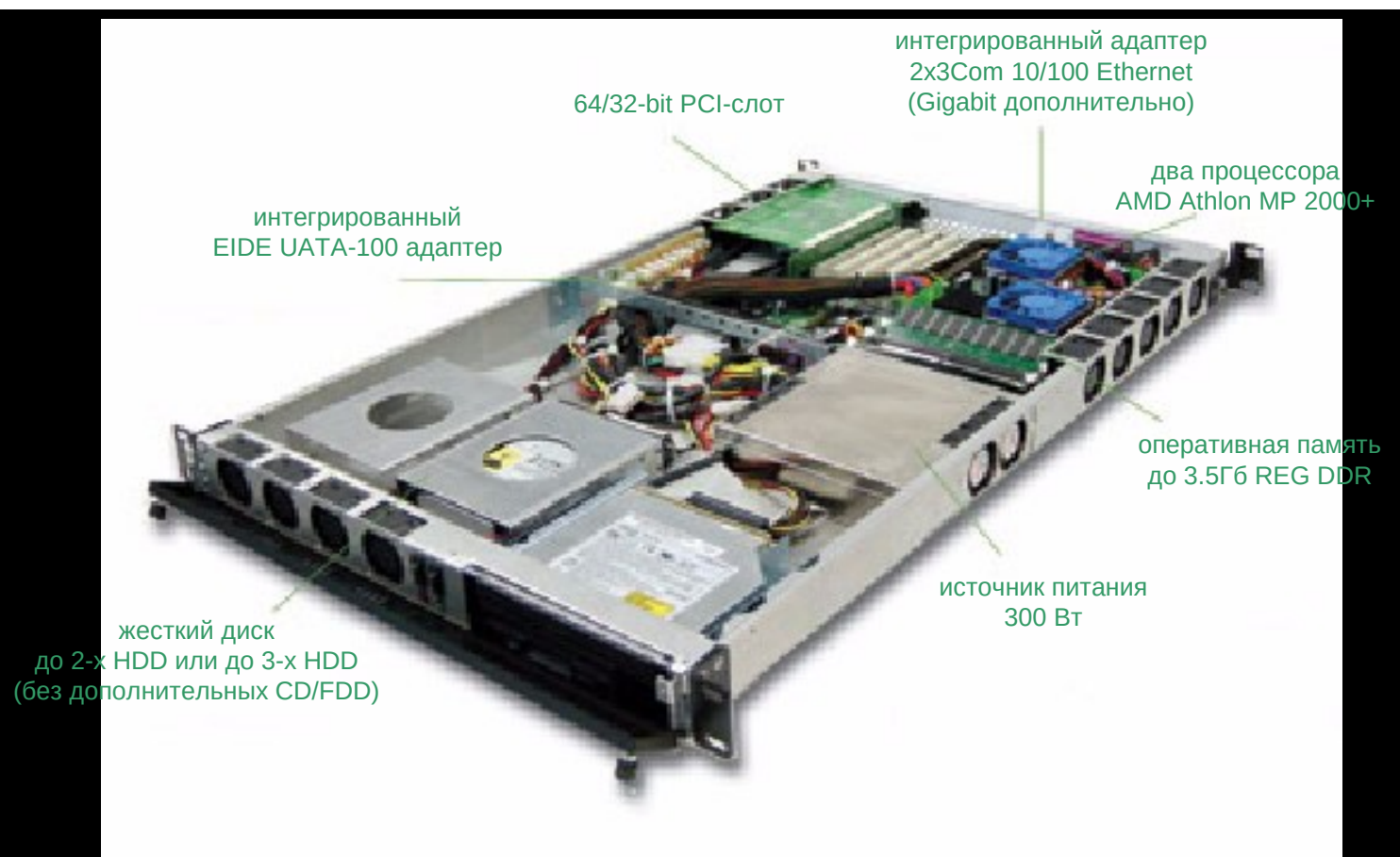
Архитектура высокопроизводительной кластерной системы



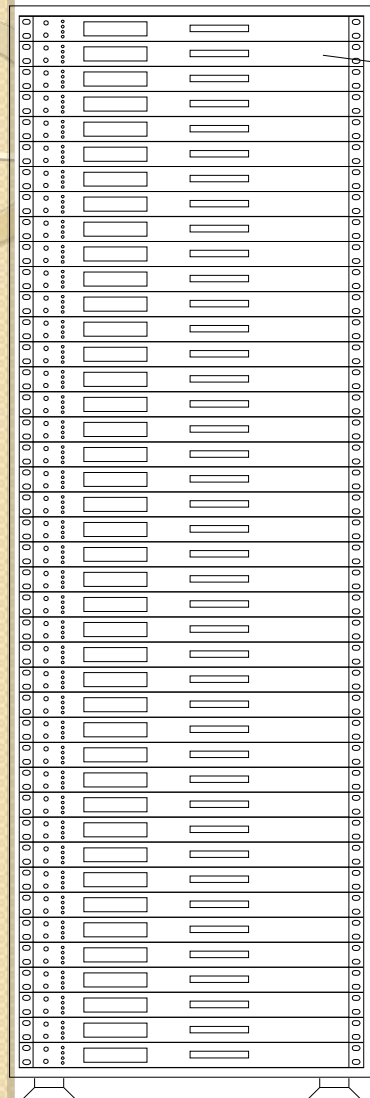
Общая структура ВВС



Вычислительный узел



Конструкция ВВС



Конструктивное исполнение
вычислительной стойки

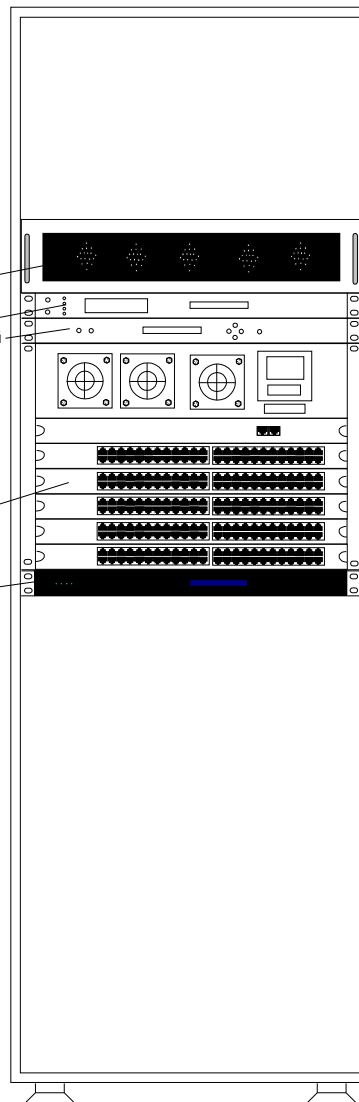
Вычислительный
узел

Система хранения
данных

Управляющая ЭВМ
Коммутатор консолей

Коммутационное
оборудование
служебной сети

Консоль



Конструктивное
исполнение стойки TC-SCI

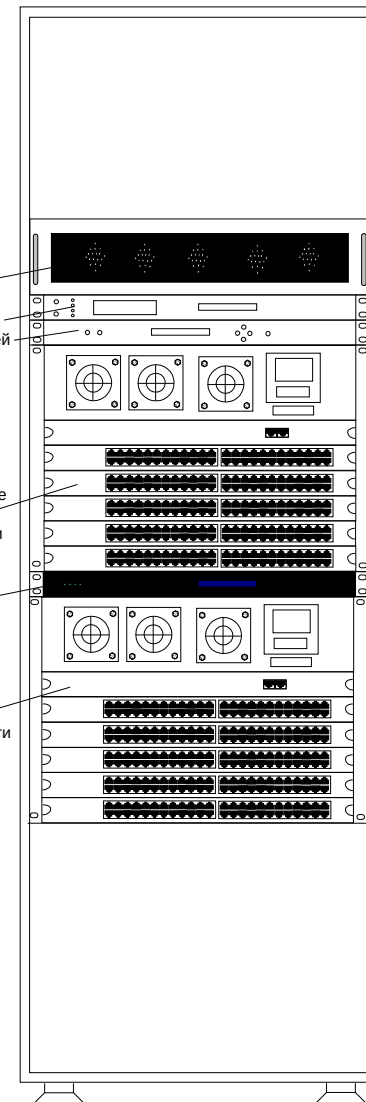
Система хранения
данных

Управляющая ЭВМ
Коммутатор консолей

Коммутационное
оборудование
служебной сети

Консоль

Коммутационное
оборудование
вычислительной сети



Конструктивное исполнение
стойки TC-GE

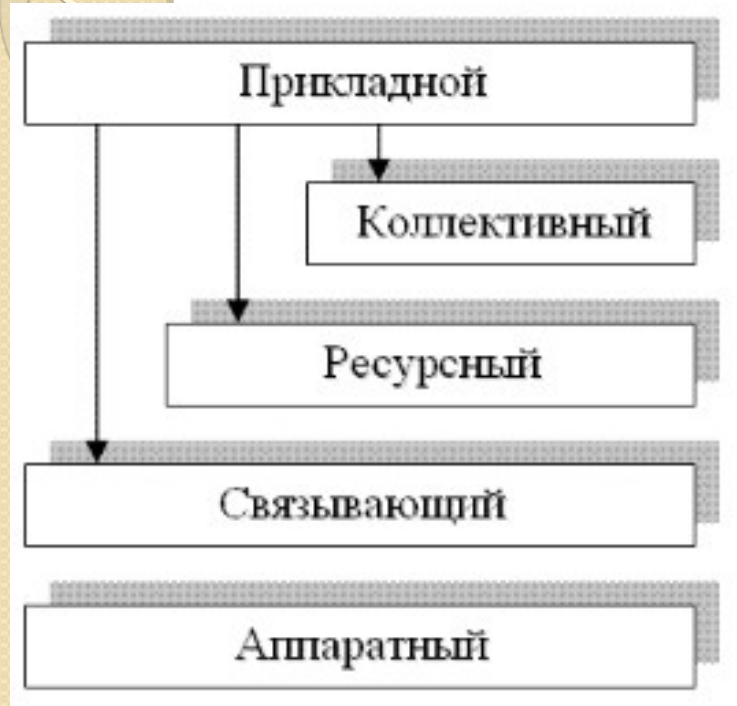
Grid системы

- **Грид – это система, которая координирует распределенные ре-сурсы** посредством стандартных, открытых, универсальных протоколов и интерфейсов для обеспечения нетривиального ка-чества обслуживания (QoS – Quality of Service).

Ян Фостер книга «Грид. Новая инфраструктура вычислений» 1998 г.

- Концепция грид-вычислений идея подобна концепции электросети (англ. Power Grid): нам не важно, откуда к нам в розетку приходит электричество. Независимо от этого мы можем подключить к электросети утюг, компьютер или стиральную машину.
- Аналогично и в идеологии грид: мы можем запустить любую задачу с любого компьютера или мобильного устройств на вычисление, ресурсы же для этого вычисления должны быть автоматически предоставлены на удаленных высокопроизводительных серверах, независимо от типа нашей задачи.

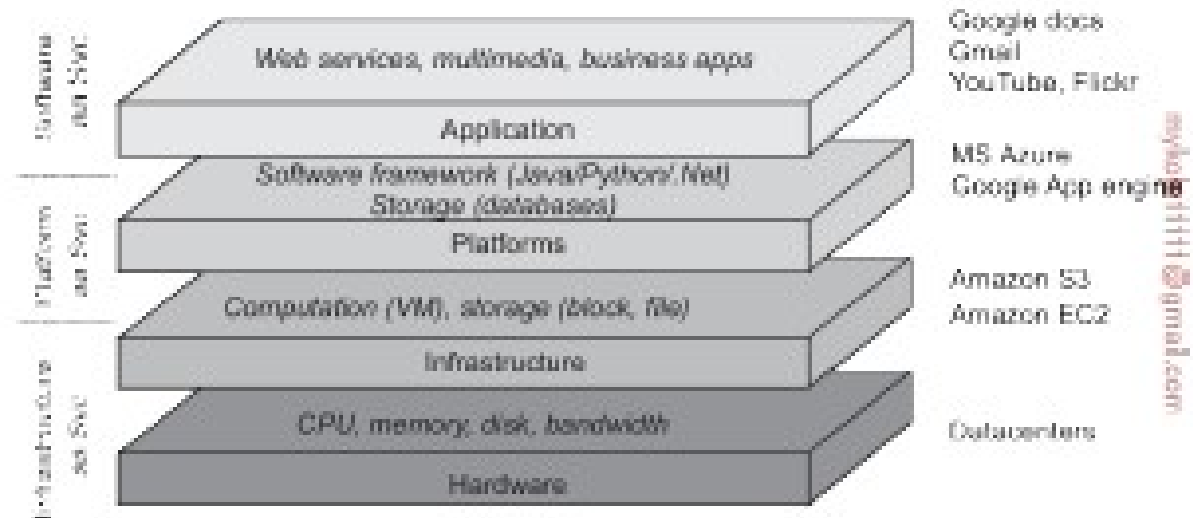
Архитектура Grid систем



- Ключевой проблемой grid является объединение ресурсов из различных организаций для совместного использования. В результате формируется федерация систем, которая влечет за собой образование виртуальных организаций.
- Для построения Grid была разработана и принята в качестве стандарта *OGSA (Open Grid Services Architecture – Открытая архитектура грид-сервисов)*, основанная на архитектуре SOA – Service Oriented Architecture.

Облачные вычисления (Cloud Computing)

- Является развитием концепции Utility Computing – (утилитарный, практичный) предоставление обслуживания запросов по заявкам пользователей с оплатой основанной на учете использованных ресурсов.
- Облачные вычисления характеризуются предоставлением пользователю легко доступных вирту.



Распределенные информационные системы

- Это прежде всего информационные системы организаций – корпоративные ИС.
- В этих системах используются:
 - ❖ различные клиент-серверные модели построения приложений;
 - ❖ различные методы интеграции данных;
 - ❖ различные способы распределенных транзакций и репликаций;
 - ❖ и др. способы организации взаимодействий в рамках РИС.

РИС – распределенная обработка транзакций

- Это одна из задач решаемых в РИС, связанная с проблемой вложенных транзакций (Nested Transaction) и проблемой обработки очереди транзакций к множеству различных баз данных.

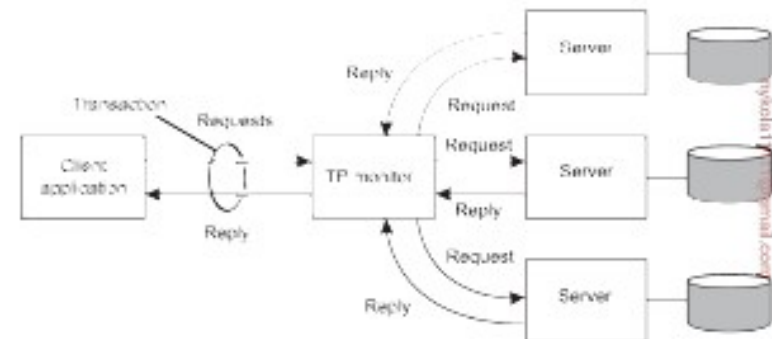
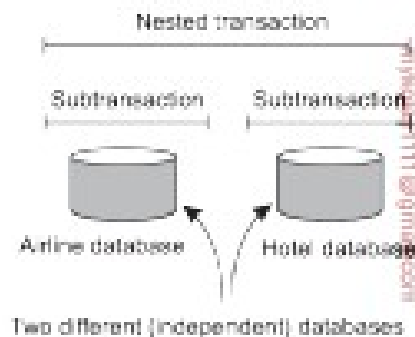
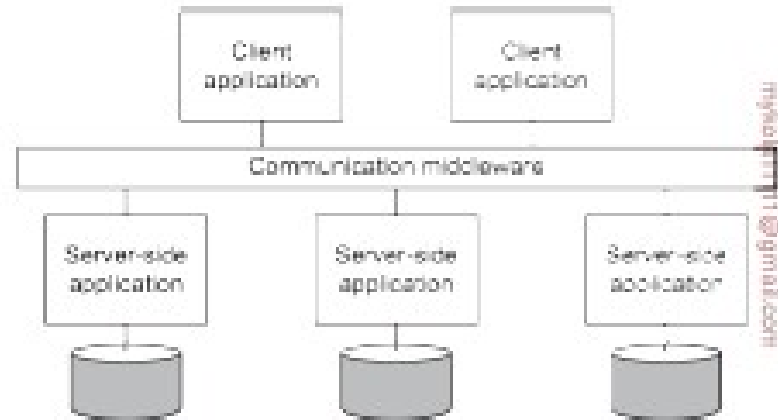


РИС – интеграция корпоративных приложений

❖ Используются следующие технологии коммуникаций между приложениями:

- RPC
- RMI
- MOM
- Publish/subscribe
- SOA



Всеобъемлющие (всепроникающие) системы (Pervasive systems)

- К этому типу РИС относятся:
- Вездесущие (Ubiquitous systems) системы – пользователи даже не подозревают, что взаимодействуют с ними. Например, системы организации движения в городах, системы видеонаблюдения и т.п.
- Мобильные компьютерные системы.
- Системы сенсоров (датчиков).
- Интернет вещей (Internet of Things – IoT).

Вездесущие (Ubiquitous systems) системы

- Основные особенности этих систем:
 - Распределенность в пространстве, узлы системы объединены сетью, между ними обеспечивается прозрачный доступ.
 - Интерактивный обмен между пользователями и устройствами системы – носит ненавязчивый характер.
 - Автономность. Устройства действуют автономно без взаимодействия с людьми.
 - Интеллектуальность. Система в целом может управлять широким составом действий и взаимодействий.
 - Пример: системы управления автомобилями и др.

КОНЦЕПЦИИ ПРОГРАММНЫХ РЕШЕНИЙ РИС

Программные системные решения для РИС

- Э.Танненбаум описал следующие виды программных решений РИС

Система	Описание	Основное назначение
Распределенные операционные системы	Сильно связанные операционные системы для мультипроцессоров и гомогенных мультимикомпьютерных систем	Соккрытие и управление аппаратным обеспечением
Сетевые операционные системы	Слабо связанные операционные системы для гетерогенных мультимикомпьютерных систем (локальных или глобальных сетей)	Предоставление локальных служб удаленным клиентам
Средства промежуточного уровня	Дополнительный уровень поверх сетевых операционных систем, реализующий службы общего назначения	Обеспечение прозрачности распределения

Распределенная ОС

- Это сильносвязанная ОС которая используется для управления мультипроцессорными и гомогенными многомашинными системами.

Сетевые ОС

- Это слабосвязанные ОС предназначенные для управления гетерогенными вычислительными системами.
- Эти системы должны обеспечивать удаленный доступ к локальным службам.

Средства промежуточного уровня

- Их задача обеспечение прозрачности распределения.

Соотношение программных решений РИС и модели OSI

Прикладной	Приложение	Приложение
Представительский	Распределенная операционная система	Сетевая или серверная операционная система
Сеансовый		
Транспортный		
Сетевой		
Канальный	Аппаратура	Аппаратура
Физический		

Мультикомпьютерные ОС



- Мультикомпьютерные операционные системы, не предоставляющие средств для совместного использования памяти, могут предложить приложениям только средства для обмена сообщениями.

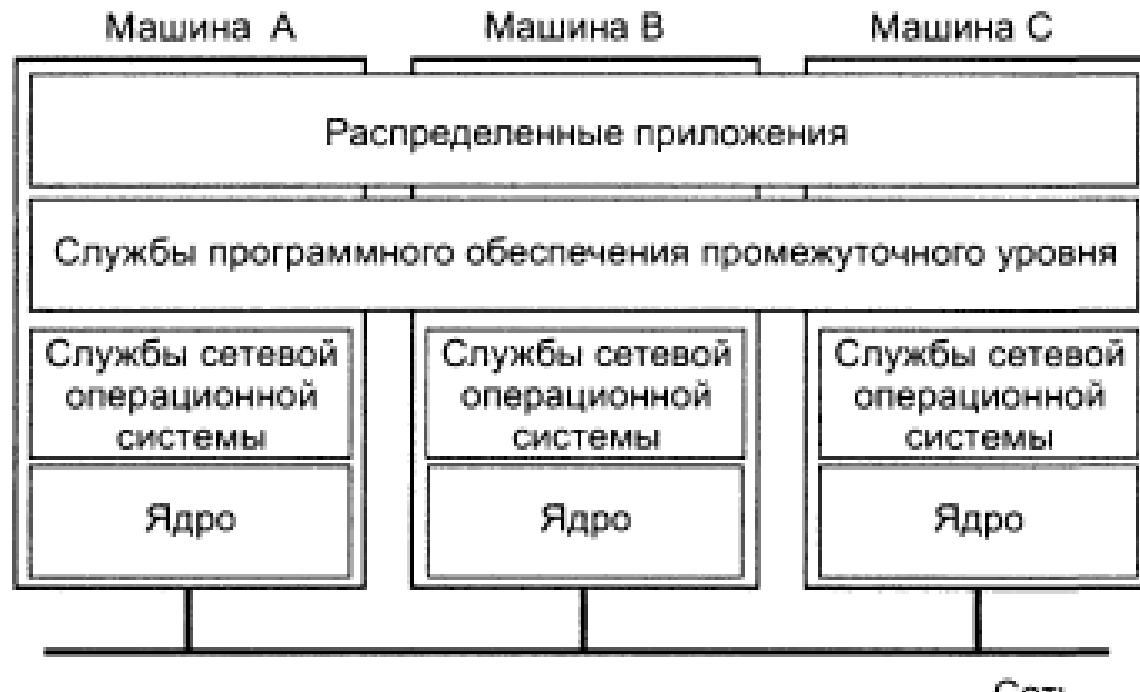
Сетевые операционные системы

- В противоположность распределенным операционным системам сетевые операционные системы не нуждаются в том, чтобы аппаратное обеспечение, на котором они функционируют, было гомогенно и управлялось как единая система.
- Напротив, обычно они строятся для набора однопроцессорных систем, каждая из которых имеет собственную операционную систему.



Программное обеспечение промежуточного уровня

- Каждая локальная система, составляющая часть базовой сетевой операционной системы, предоставляет управление локальными ресурсами и простейшие коммуникационные средства для связи с другими компьютерами.
- задача промежуточного уровня — скрыть разнообразие базовых платформ от приложений.



Модели промежуточного уровня

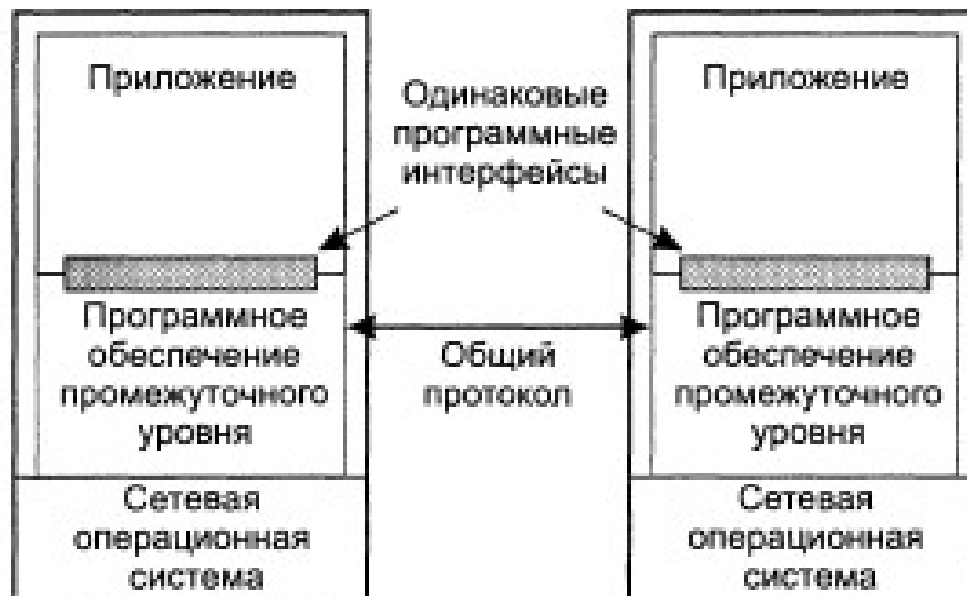
- Программное обеспечение промежуточного уровня базируется на некоторой модели, или *парадигме, определяющей распределение и связь*.
- Например, такой моделью является представление **всех** наблюдаемых **объектов** в виде **файлов**. Этот подход был изначально введен в UNIX.
- Другая важная ранняя модель программного обеспечения промежуточного уровня основана на *удаленных вызовах процедур (Remote Procedure Calls), RPC*
- Далее появились системы промежуточного уровня, реализующих представление о *распределенных объектах {distributed objects}*
- Успех среды Web в основном определяется тем, что она построена на базе потрясающе простой, но высокоэффективной модели *распределенных документов {distributed documents}*.
- Затем возникла модель основанная на использовании **Веб сервисов**

Службы промежуточного уровня

- Для систем промежуточного уровня существует некоторое количество стандартных служб. Это:
 - Служба коммуникаций реализующая прозрачность доступа.
 - Служба именования. Проблема состоит в том, что для эффективного поиска имени в большой системе местоположение разыскиваемой сущности должно считаться фиксированным.
 - Служба распределенных транзакций.
 - Служба обеспечения безопасности.

Промежуточный уровень и открытость

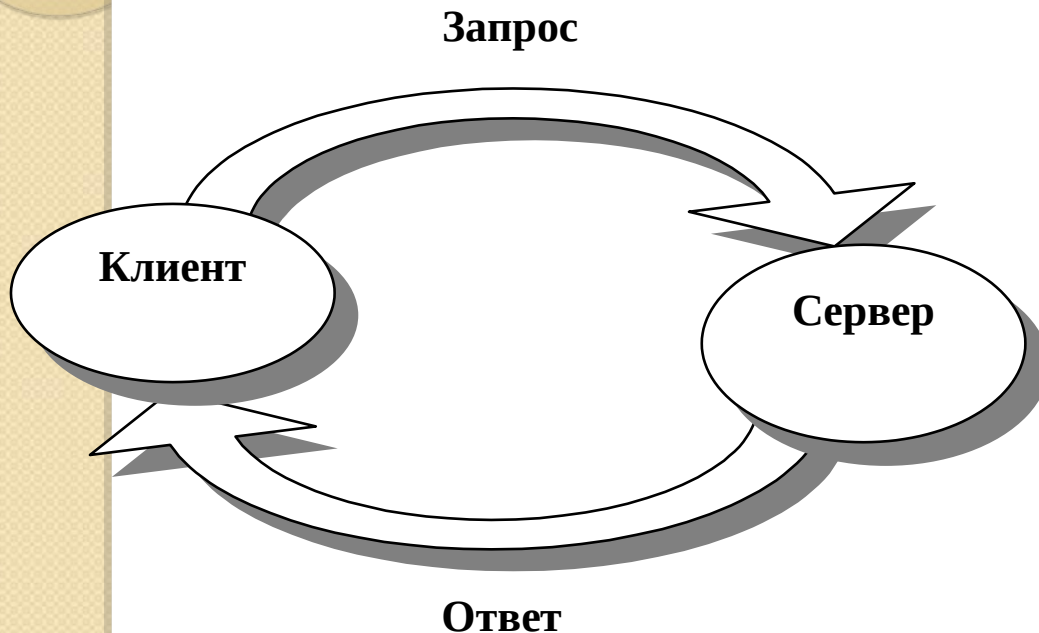
- Для гарантии совместной работы различных реализаций необходимо, чтобы к сущностям разных систем можно было



Сравнение программных решений для распределенных систем

Характеристика	Мультимикропроцессорная ОС	Мультимикрокомпьютерная ОС	Сетевая ОС	ПО промежуточного уровня
Прозрачность	Очень высокая	Высокая	Низкая	Высокая
Идентичность ОС на всех узлах	Поддерживается	Поддерживается	Не поддерживается	Не поддерживается
Число копий ОС	1	N	N	N
Коммуникации на основе	Общей памяти	Сообщений	Файлов	Зависит от модели
Управление ресурсами	Глобальное, централизованное	Глобальное, распределенное	Отдельно на узле	Отдельно на узле
Масштабируемость	Отсутствует	Умеренная	Да	Различная
Открытость	Закрытая	Закрытая	Открытая	Открытая

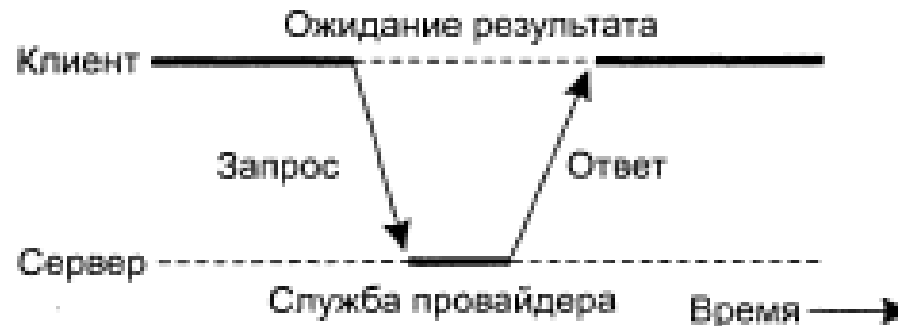
Архитектура клиент-сервер



- В классической архитектуре “клиент-сервер” любая информационная система должна иметь минимум три основные функциональные части:
 - модуль хранения данных;
 - модуль обработки;
 - модуль интерфейса с пользователем.

Модель клиент-сервер

- В базовой модели клиент-сервер все процессы в распределенных системах делятся на две возможно перекрывающиеся группы. Процессы, реализующие некоторую службу, например службу файловой системы или базы данных, называются *серверами {servers}*. Процессы, запрашивающие службы у серверов путем отправки запроса и последующего ожидания ответа от сервера, называются *клиентами {clients}*.

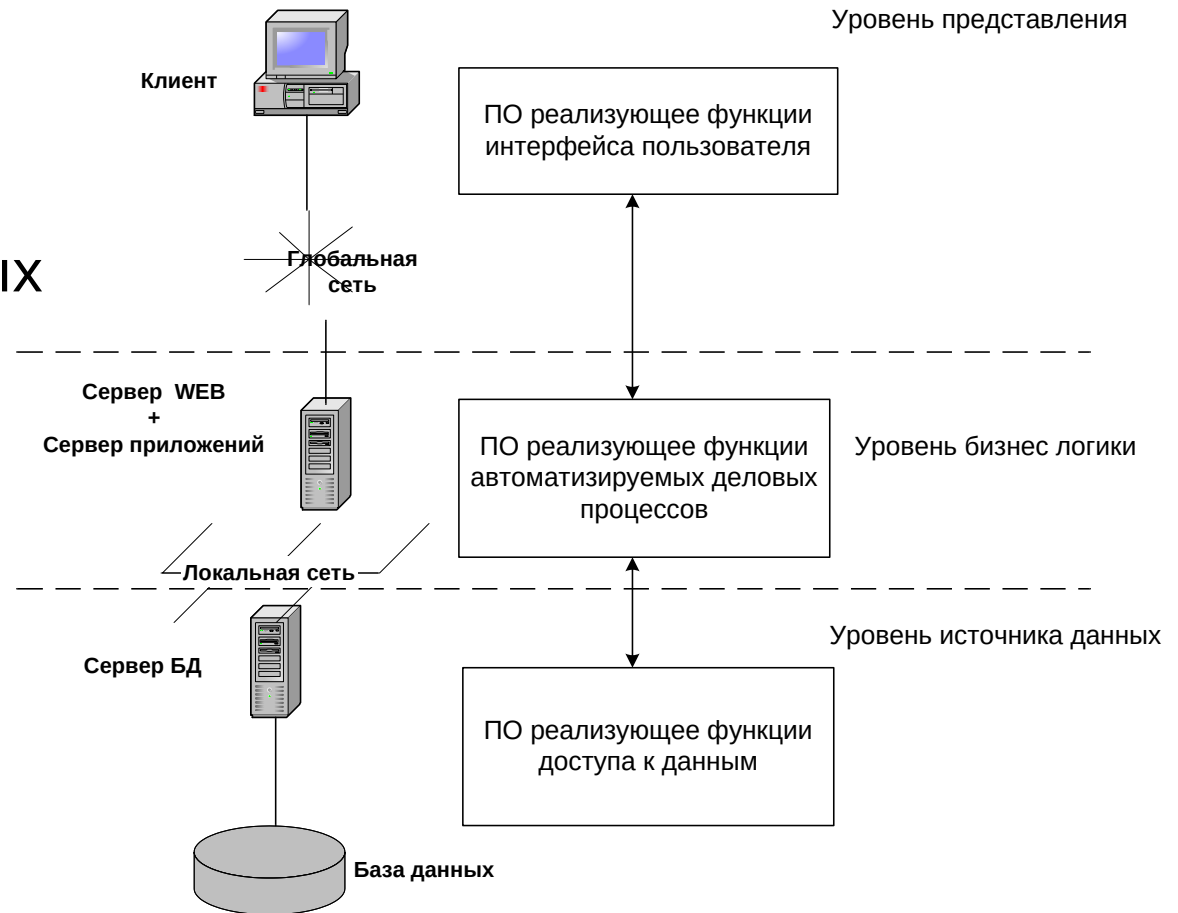


Разделение приложений по уровням

- Рассматривая множество приложений типа клиент-сервер, предназначенных для организации доступа пользователей к базам данных, многие рекомендовали разделять их на три уровня.
 - уровень пользовательского интерфейса;
 - уровень обработки;
 - уровень данных.

Три основных уровня корпоративных приложений

- Представление
- Бизнес-логика приложения
- Источник данных



Варианты архитектуры клиент-сервер

- Простейшая организация предполагает наличие всего двух типов машин.
 - Клиентские машины, на которых имеются программы, реализующие только пользовательский интерфейс или его часть.
 - Серверы, реализующие все остальное, то есть уровни обработки и данных.
- На самом деле такая система не является распределенной: все происходит на сервере, а клиент представляет собой не что иное, как простой терминал.

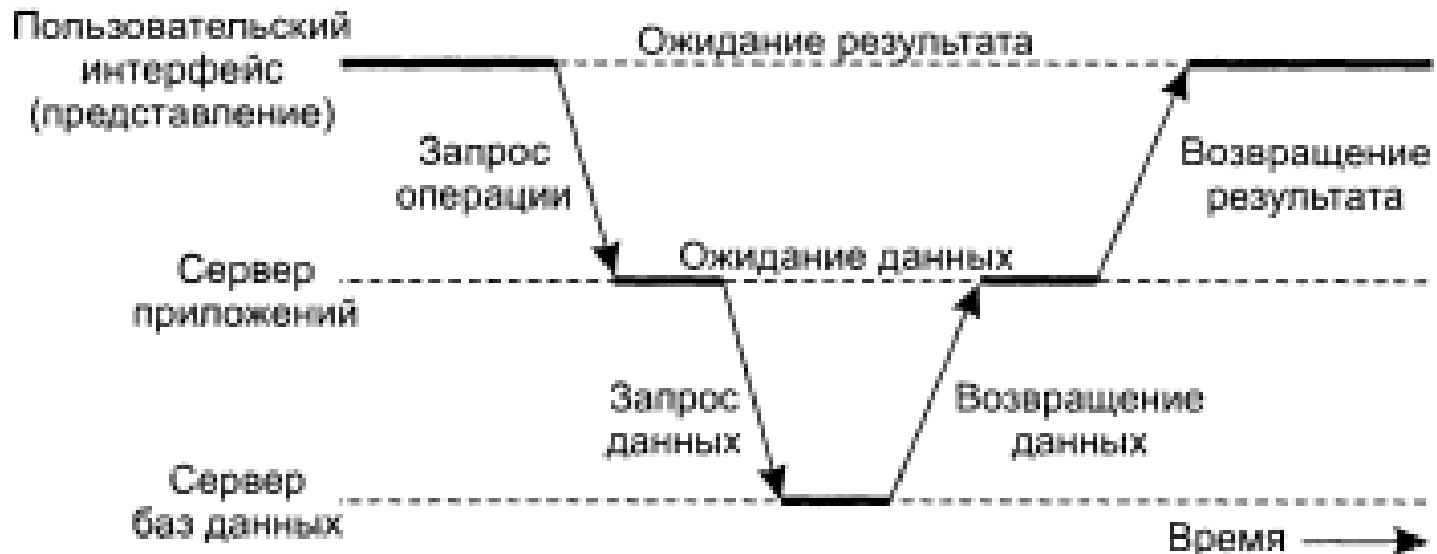
Физически двузвенные архитектуры

- Один из подходов к организации клиентов и серверов — это распределение программ, находящихся на уровне приложений, по различным машинам.

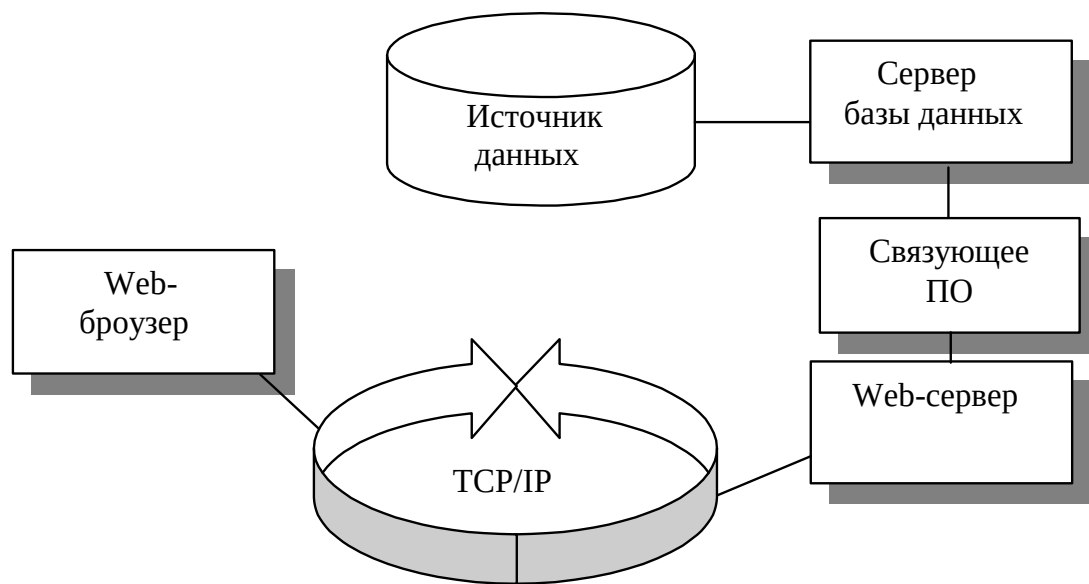


Физически трехзвенная архитектура

- Рассматривая только клиенты и серверы, мы упускаем тот момент, что серверу иногда может понадобиться работать в качестве клиента.



Архитектура унифицированного доступа к информационным ресурсам



Архитектура многоуровневых клиент-серверных систем достаточно хорошо согласуется с применением современной Web-технологии построения информационных систем. В этом случае в качестве клиентской части используется (одна или с расширениями) унифицированная, единая для всех клиентов программа - просмотрщик (Web-браузер), а сервер приложений дополняется Web-сервером и программами вызова процедур сервера.

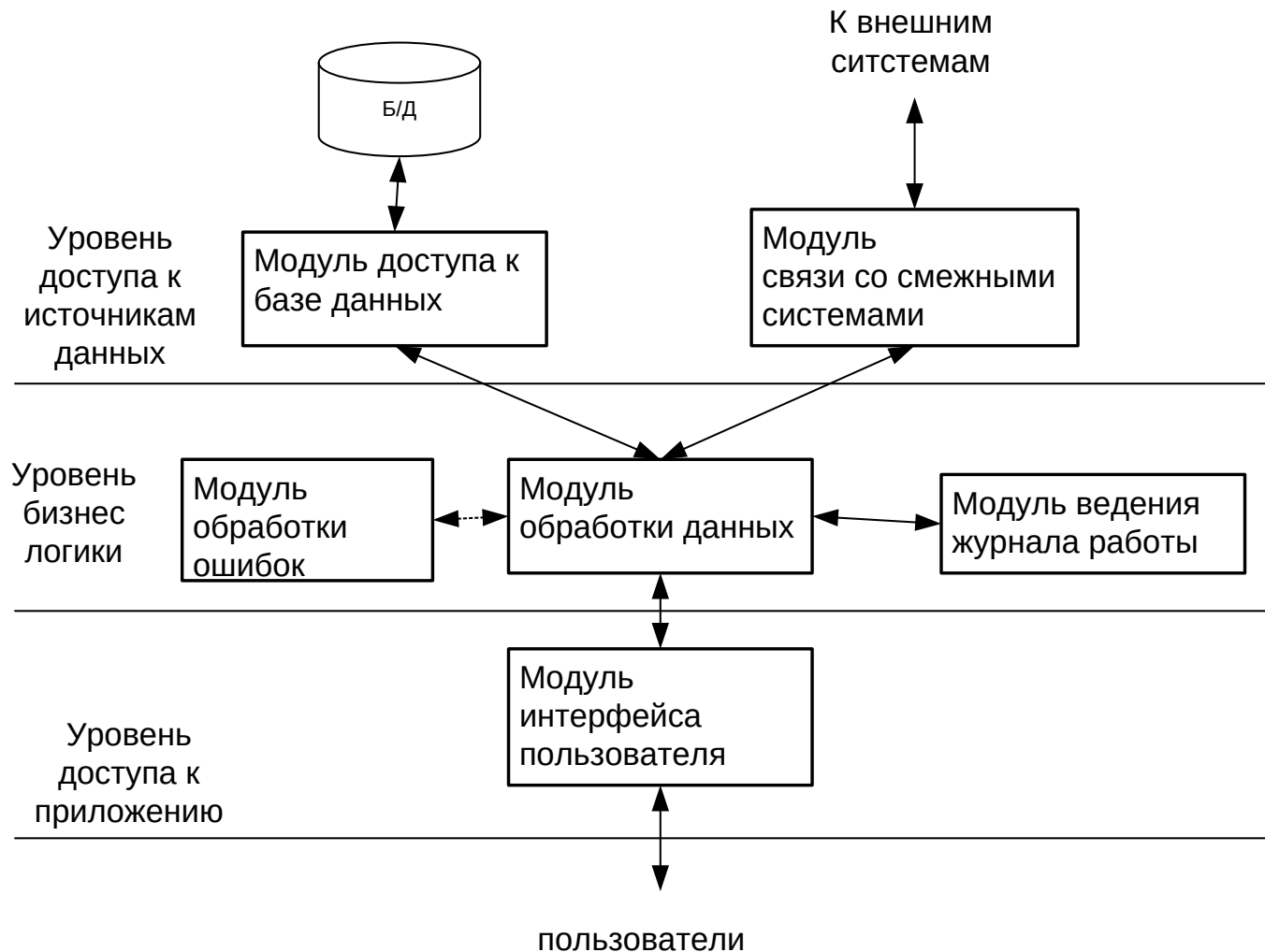


ВИДЫ РАСПРЕДЕЛЕННОСТИ РИС

Системы с вертикальным распределением

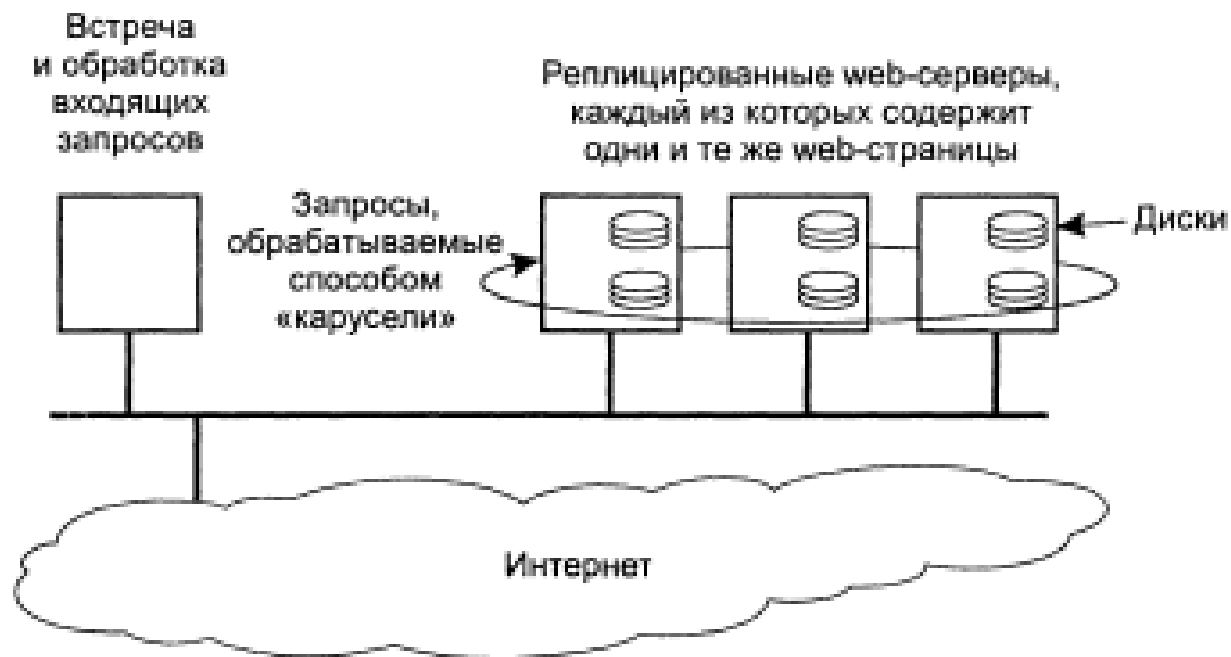
- Характеристической особенностью вертикального распределения является то, что оно достигается размещением логически различных компонентов на разных машинах.

Архитектура приложения (расслоение приложения)



Горизонтальное распределение

- При таком типе распределения клиент или сервер может содержать физически разделенные части логически однородного модуля, причем работа с каждой из частей может происходить независимо. Это делается для выравнивания загрузки.
- Пример, web-сервер, реплицированный на несколько машин локальной сети.



Вопросы ?



**СПАСИБО ЗА
ВНИМАНИЕ !**