

Лекция 5 Обзор и архитектура вычислительных сетей. Одноранговая архитектура Архитектура клиент – сервер

Разновидности архитектуры компьютерных сетей

Концепция вычислительных сетей является логическим результатом эволюции компьютерной технологии. По мере эволюции вычислительных систем сформировались следующие разновидности архитектуры компьютерных сетей:

- одноранговая архитектура;
- классическая архитектура «клиент – сервер»;
- архитектура «клиент – сервер» на основе Web-технологии.

Правильно выбранная архитектура компьютерной сети позволяет достичнуть выдвинутых требований по общей производительности, надежности защиты сетевых ресурсов, гибкости настройки сети, а также минимизации денежных затрат на ее построение и администрирование.

Одноранговая сеть – это сеть, в которой отсутствует выделенный сервер, а клиентские компьютеры могут использовать ресурсы друг друга. В одноранговой сети все компьютеры равноправны: нет иерархии среди компьютеров и нет выделенного сервера. Каждый компьютер функционирует и как клиент, и как сервер, нет отдельного компьютера, ответственного за администрирование всей сети. Все пользователи самостоятельно решают, что на своем компьютере можно сделать общедоступным по сети.

Централизованно управлять защитой в одноранговой сети сложно, так как каждый пользователь устанавливает ее самостоятельно, да и «общие» ресурсы могут находиться на всех компьютерах, а не только на центральном сервере. Такая ситуация представляет серьезную угрозу для всей сети.

Явные недостатки, свойственные одноранговой архитектуре и развитие инструментальных средств привели к появлению вычислительных систем с архитектурой «клиент – сервер». **Клиент – серверная технология** – это стиль работы приложений, где клиентский процесс запрашивает обслуживание у процесса сервера. **Сервер** – это программа, предоставляющая доступ к каким-либо услугам, например к электронной почте, файлам, ftp, Web, или данным (в качестве сервера баз данных). **Клиент** – это приложение, которое соединяется с сервером, чтобы воспользоваться предоставляемыми им услугами.

Компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер...). Клиент и сервер какого-либо ресурса могут находиться как в рамках одной вычислительной системы, так и на различных компьютерах, связанных сетью.

Модели архитектуры «клиент – сервер»

Основной принцип технологии «клиент–сервер» заключается в разделении функций приложения на три группы:

- ввод и отображение данных (взаимодействие с пользователем);
- прикладные функции, характерные для данной предметной области;
- функции управления ресурсами (файловой системой, базой данных и т.д.).

Поэтому, в любом приложении выделяются следующие компоненты:

- компонент представления данных;
- прикладной компонент;
- компонент управления ресурсом.

На основе распределения перечисленных компонентов между рабочей станцией и сервером сети выделяют следующие модели архитектуры «клиент –сервер»:

- модель доступа к удаленным данным;
- модель сервера управления данными;
- модель комплексного сервера;
- трехзвенная архитектура «клиент – сервер».

Модель доступа к удаленным данным, при которой на сервере расположены только данные, имеет следующие особенности:

- невысокая производительность, так как вся информация обрабатывается на рабочих станциях;
- снижение общей скорости обмена при передаче больших объемов информации для обработки с сервера на рабочие станции.

При использовании **модели сервера управления данными** кроме самой информации на сервере располагается менеджер информационных ресурсов (например, система управления базами данных). Компонент представления и прикладной компонент совмещены и выполняются на компьютере-клиенте, который поддерживает как функции ввода и отображения данных, так и чисто прикладные функции. Доступ к информационным ресурсам обеспечивается либо операторами специального языка (например, SQL в случае использования базы данных), либо вызовами функций специализированных программных библиотек. Запросы к информационным ресурсам направляются по сети менеджеру ресурсов (например, серверу базы данных), который обрабатывает запросы и возвращает клиенту блоки данных. Наиболее существенные особенности данной модели:

- уменьшение объемов информации, передаваемых по сети, так как выборка необходимых информационных элементов осуществляется на сервере, а не на рабочих станциях;
- унификация и широкий выбор средств создания приложений;
- отсутствие четкого разграничения между компонентом представления и прикладным компонентом, что затрудняет совершенствование вычислительной системы.

Модель сервера управления данными целесообразно использовать в случае обработки умеренных, не увеличивающихся со временем объемов информации. При этом сложность прикладного компонента должна быть невысокой.

Модель комплексного сервера строится в предположении, что процесс, выполняемый на компьютере-клиенте, ограничивается функциями представления, а собственно прикладные функции и функции доступа к данным выполняются сервером.

Преимущества модели комплексного сервера:

- высокая производительность;
- централизованное администрирование;
- экономия ресурсов сети.

Модель комплексного сервера является оптимальной для крупных сетей, ориентированных на обработку больших и увеличивающихся со временем объемов информации.

Архитектура «клиент—сервер», при которой прикладной компонент расположен на рабочей станции вместе с компонентом представления (модели доступа к удаленным данным и сервера управления данными) или на сервере вместе с менеджером ресурсов и данными (модель комплексного сервера), называют двухзвенной архитектурой.

При существенном усложнении и увеличении ресурсоемкости прикладного компонента для него может быть выделен отдельный сервер, называемый сервером приложений. В этом случае говорят о **трехзвенной архитектуре «клиент-сервер»**. Первое звено – компьютер – клиент, второе – сервер приложений, третье – сервер управления

данными. В рамках сервера приложений могут быть реализованы несколько прикладных функций, каждая из которых оформляется как отдельная служба, предоставляющая некоторые услуги всем программам. Серверов приложения может быть несколько, каждый из них ориентирован на предоставление некоторого набора услуг.

. Принцип работы архитектуры клиент-сервер», основанной на Webтехнологии

В настоящее время наиболее перспективной является **архитектура «клиент-сервер», основанная на Web-технологии**. Обмен информацией по Web-технологии не отличается от информационного обмена, реализуемого по принципу «клиент-сервер», когда программа-сервер осуществляет обработку запросов, поступающих от программы-клиента.

В соответствии с Web-технологией на сервере размещаются так называемые Web-документы, которые визуализируются и интерпретируются программой навигации (Web-навигатор, Web-браузер), функционирующей на рабочей станции. В Web-технологии существует система гиперссылок, включающая ссылки на следующие объекты:

- другую часть Web-документа;
- другой Web-документ или документ другого формата (например, документ Word или Excel), размещаемый на любом компьютере сети;
- мультимедийный объект (рисунок, звук, видео);
- программу, которая при переходе на нее по ссылке, будет передана с сервера на рабочую станцию для интерпретации или запуска на выполнение навигатором;
- любой другой сервис – электронную почту, копирование файлов с другого компьютера сети, поиск информации и т.д.
- Передачу с сервера на рабочую станцию документов и других объектов по запросам, поступающим от навигатора, обеспечивает функционирующая на сервере программа, называемая Web-сервером. Когда Web-навигатору необходимо получить документы или другие объекты от Web-сервера, он отправляет серверу соответствующий запрос. При достаточных правах доступа между сервером и навигатором устанавливается логическое соединение. Далее сервер обрабатывает запрос, передает Webнавигатору результаты обработки и разрывает установленное соединение. Таким образом, Web-сервер выступает в качестве информационного концентратора, который доставляет информацию из разных источников, а потом в однородном виде предоставляет ее пользователю.

Прикладные сервисы Internet

Internet представляет собой всемирное объединение взаимосвязанных компьютерных сетей. Использование общих протоколов семейства TCP/IP и единого адресного пространства позволяет говорить об Internet как о единой глобальной «метасети», или «сети сетей». При работе на компьютере, имеющем подключение к Internet, можно установить связь с любым другим подключенным к Сети компьютером и реализовать обмен информацией с использованием следующих прикладных сервисов Internet:

Удаленный доступ (**telnet**) Remote Login – работа на удаленном компьютере в режиме, когда ваш компьютер эмулирует терминал удаленного компьютера, т. е. вы можете делать все то же (или почти все), что можно делать с обычного терминала той машины. Трафик, относящийся к этому виду работы в сети, в среднем составляет около 19% всего сетевого трафика. Начать сеанс удаленного доступа можно в UNIX, подав команду telnet и указав имя машины, с которой вы хотите работать. Если номер порта опустить, то ваш компьютер по умолчанию эмулирует терминал той машины и вы входите в систему как обычно. Указание номера порта позволяет связываться с нестандартными серверами, интерфейсами. Telnet – протокол эмуляции терминала, который обеспечивает поддержку удаленного доступа в Internet . **Telnet** – так же называется программа в UNIX, которая

обслуживает эти сеансы работы; telnet имеет и свой собственный набор команд, которые управляют собственно этой программой, т.е. сеансом связи, его параметрами, открытием новых, закрытием и т.д.; эти команды подаются из командного режима telnet, в который можно перейти, нажав так называемую escapesequence клавиш, которая вам сообщается при достижении удаленной машины.

FTP – File Transfer Protocol - протокол передачи файлов, определяющий правила передачи файлов с одного компьютера на другой.

Протокол ftp используют для пересылки файлов.

Для работы с FTP нужно иметь доступ на ту удаленную машину, с которой вы хотите перекачать себе файлы, т.е. иметь входное имя и знать соответствующий пароль. Доступ должен быть как минимум типа dial-up (по вызову). Для использования FTP, нужно подать команду FTP с указанием имени рабочей машины, на которой вы хотите провести сеанс. FTP также позволяет (у него свой набор команд) производить поиск файла на удаленной машине, то есть переходить из директории в директорию, просматривать содержимое этих директорий, файлов. Имеется возможность сжимать данные при пересылке и после их разжимать в прежний вид.

Подключение к Интернет

Для работы в Интернете необходимо:

- физически подключить компьютер к одному из узлов Всемирной сети;
- получить *IP-адрес* на постоянной или временной основе;
- установить и настроить программное обеспечение – программы-клиенты тех служб Интернета, услугами которых предполагается пользоваться.

Организации, предоставляющие возможность подключения к своему узлу и выделяющие IP-адреса, называются **поставщиками услуг Интернета** (используется также термин *сервис-провайдер*). Они оказывают подобную услугу на договорной основе.

Физическое подключение может быть **выделенным** или **коммутируемым**. Для выделенного соединения необходимо проложить новую или арендовать готовую физическую линию связи (кабельную, оптоволоконную, радиоканал, спутниковый канал и т. п.). Такое подключение используют организации и предприятия, нуждающиеся в передаче больших объемов данных. От типа линии связи зависит ее **пропускная способность** (измеряется в единицах бит в секунду). В настоящее время пропускная способность мощных линий связи (оптоволоконных и спутниковых) составляет сотни мегабит в секунду (Мбит/с).

В противоположность выделенному соединению коммутируемое соединение – временное. Оно не требует специальной линии связи и может быть осуществлено, например, по телефонной линии. Коммутацию (подключение) выполняет автоматическая телефонная станция (АТС) по сигналам, выданным в момент набора телефонного номера.

Для телефонных линий связи характерна низкая пропускная способность. В зависимости от того, какое оборудование использовано на станциях АТС по пути следования сигнала, различают *аналоговые* и *цифровые* телефонные линии. Основную часть телефонных линий в городах России составляют устаревшие аналоговые линии. Их предельная пропускная способность немногим более 30 Кбит/с (одна-две страницы текста в секунду или одна-две фотографии стандартного размера в минуту). Пропускная способность цифровых телефонных линий составляет 60-120 Кбит/с, то есть в 2-4 раза выше. По аналоговым телефонным линиям связи можно передавать и видеинформацию (что используется в видеоконференциях), но размер окна, в котором отображаются видеоданные, обычно невелик (порядка 150x150 точек) и частота смены кадров мала для получения качественного видеоряда (1-2 кадра в секунду). Для сравнения: в обычном телевидении частота кадров — 25 кадров в секунду.

Телефонные линии связи никогда не предназначались для передачи цифровых сигналов – их характеристики подходят только для передачи голоса, причем в достаточно узком диапазоне частот – 300-3 000 Гц. Поэтому для передачи цифровой информации несущие сигналы звуковой частоты *модулируют* по амплитуде, фазе и частоте. Такое преобразование выполняет специальное устройство – **модем**.

Организация электронной почты. Типы протоколов, используемые при почтовом обмене

Электронная почта один из важнейших информационных ресурсов Internet.

Основные понятия, на которых построена электронная почта, параллельны основным концепциям построения обычной почты. Вы посыпаете людям письма по их конкретным адресам. Они, в свою очередь, пишут вам на ваш почтовый адрес.

При настройке программы работы с электронной почтой независимо от ее интерфейса необходима следующая информация от провайдера: имя сервера исходящей почты, имя сервера входящей почты, имя пользователя и пароль, а также типы протоколов, используемые при почтовом обмене.

Протокол **Simple Mail Transfer Protocol** (SMTP). Для работы электронной почты в Интернет специально разработан этот протокол, который является протоколом прикладного уровня и использует транспортный протокол TCP. Однако совместно с этим протоколом используется и UUCP (Unix – to – Unix-Copy – Program) протокол. UUCP хорошо подходит для использования телефонных линий связи. Разница между SMTP и UUCP заключается в том, что при использовании первого протокола почтового обмена программа, функционирующая на сервере, пытается найти машину получателя почты и установить с ней взаимодействие в режиме on-line для того, чтобы передать почту в ее почтовый ящик. В случае использования **SMTP** почта достигает почтового ящика получателя за считанные минуты и время получения сообщения зависит только от того, как часто получатель просматривает свой почтовый ящик. При использовании **UUCP** почта передается по принципу «stop – go», т.е. почтовое сообщение передается по цепочке почтовых серверов от одной машины к другой, пока не достигнет машины-получателя или не будет отвергнута по причине отсутствия абонента-получателя. С одной стороны, **UUCP** позволяет доставлять почту по плохим телефонным каналам, так как не требуется поддерживать линию все время доставки от отправителя к получателю, а с другой стороны, время доступа к адресату значительно возрастает. В целом же общие рекомендации таковы: если имеется возможность надежно работать в режиме on-line и это является нормой, то следует настраивать почту для работы по протоколу **SMTP**, если линии связи плохие или on-line используется чрезвычайно редко, то лучше использовать UUCP.

Основой любой почтовой службы является система адресов. Без точного адреса невозможно доставить почту адресату. В Интернет принята система адресов, которая базируется на доменном адресе машины. Например, для пользователя tala машины с адресом citmgu.ru почтовый адрес будет выглядеть так: tala@citmgu.ru

Таким образом, адрес состоит из двух частей: идентификатора пользователя, который записывается перед знаком «комерческого эй» – «@», и доменного адреса машины, который записывается после знака «@».

Протокол **SMTP** был разработан для обмена почтовыми сообщениями в сети Интернет, он не зависит от транспортной среды и может использоваться для доставки почты в сетях с протоколами, отличными от TCP/IP. Взаимодействие в рамках **SMTP** строится по принципу двусторонней связи, которая устанавливается между отправителем и получателем почтового сообщения. При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель на эти запросы отвечает. Фактически, отправитель выступает в роли клиента, а получатель – сервера.

Канал связи устанавливается непосредственно между отправителем и получателем сообщения. При таком взаимодействии почта достигает абонента в течение нескольких секунд после отправки.

Протокол обмена почтовой информацией **POP3** предназначен для разбора почты из почтовых ящиков пользователей на их рабочие места при помощи программ-клиентов. Если по протоколу **SMTP** пользователи отправляют корреспонденцию через Интернет, то по протоколу **POP3** они получают корреспонденцию из своих почтовых ящиков на почтовом сервере в локальные файлы.

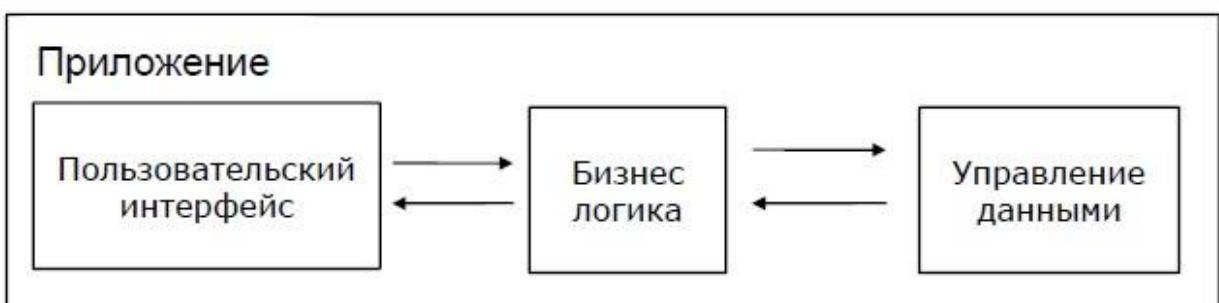
Такая схема предполагает, что пользователь имеет почтовый ящик на машине-сервере, которая не выключается круглосуточно. Все почтовые сообщения складываются в этот почтовый ящик. По мере необходимости пользователь из своего почтового клиента обращается к почтовому ящику и забирает из него пришедшую на его имя почту. При отправке программа-клиент обращается непосредственно к серверу рассылки почты и передает отправляемые сообщения на этот сервер для дальнейшей рассылки.

Современные информационные и телекоммуникационные технологии (ИТТ) с их стремительно растущим потенциалом и быстро снижающимися издержками открывают большие возможности для новых форм организации труда и занятости в рамках как отдельных корпораций, так и общества в целом. Спектр таких возможностей значительно расширяется – нововведения воздействуют на все сферы жизни людей, семью, образование, работу, географические границы человеческих общностей и т. д. Сегодня ИТТ могут внести решающий вклад в укрепление взаимосвязи между ростом производительности труда, объемов производства, инвестиций и занятости. Новые виды услуг, распространяющиеся по сетям, в состоянии создать немало рабочих мест, что подтверждает практика последних лет.

Базовые функции информационных систем.

Архитектура информационной системы - концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

С точки зрения **программно-аппаратной реализации** можно выделить ряд типовых архитектур ИС.



Компоненты информационной системы по выполняемым функциям можно разделить на три слоя: слой представления, слой бизнес-логики и слой доступа к данным.

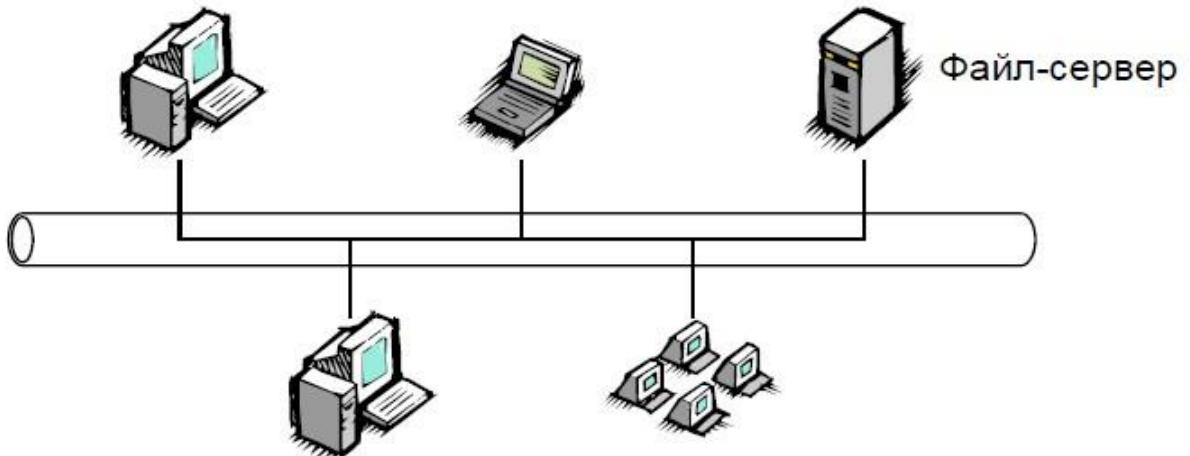
Слой представления - все, что связано с взаимодействием с пользователем: нажатие кнопок, движение мыши, отрисовка изображения, вывод результатов поиска и т.д.

Бизнес логика - правила, алгоритмы реакции приложения на действия пользователя или на внутренние события, правила обработки данных.

Слой доступа к данным - хранение, выборка, модификация и удаление данных, связанных с решаемой приложением прикладной задачей.

Традиционные архитектуры информационных систем. Файл-серверная архитектура

Появились локальные сети. Файлы начали передаваться по сети. Сначала были одноранговые сети - все компьютеры равноправны.



Потом возникла идея хранения всех общедоступных файлов на выделенном компьютере в сети - файл-сервере.



Файл-серверные приложения — приложения, схожие по своей структуре с локальными приложениями и использующие сетевой ресурс для хранения программы и данных. Функции сервера: хранения данных и кода программы. Функции клиента: обработка данных происходит исключительно на стороне клиента.

Количество клиентов ограничено десятками.

Плюсы:

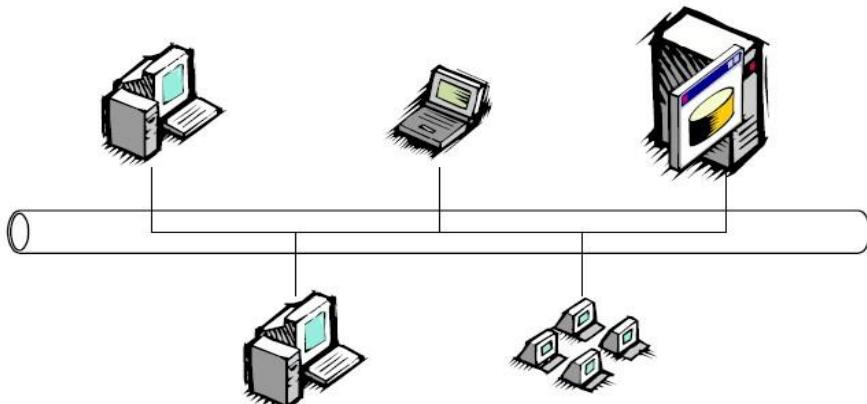
1. Многопользовательский режим работы с данными;
2. Удобство централизованного управления доступом;
3. Низкая стоимость разработки; Минусы:
 1. Низкая производительность;
 2. Низкая надежность;
 3. Слабые возможности расширения;

Недостатки архитектуры с файловым сервером очевидны и вытекают главным образом из того, что данные хранятся в одном месте, а обрабатываются в другом. Это означает, что их нужно передавать по сети, что приводит к очень высоким нагрузкам на сеть и, вследствие этого, резкому снижению производительности приложения при увеличении числа одновременно работающих клиентов. Вторым важным недостатком такой архитектуры является децентрализованное решение проблем целостности и согласованности данных и одновременного доступа к данным. Такое решение снижает надежность приложения.

Клиент-серверная архитектура

Ключевым отличием архитектуры клиент-сервер от архитектуры файл-сервер является абстрагирование от внутреннего представления данных (физической схемы данных). Теперь клиентские программы манипулируют данными на уровне логической схемы.

Итак, использование архитектуры клиент-сервер позволило создавать надежные (в смысле целостности данных) многопользовательские ИС с централизованной базой данных, независимые от аппаратной (а часто и программной) части сервера БД и поддерживающие графический интерфейс пользователя (ГИП) на клиентских станциях, связанных локальной сетью. Причем издержки на разработку приложений существенно сокращались.



Основные особенности:

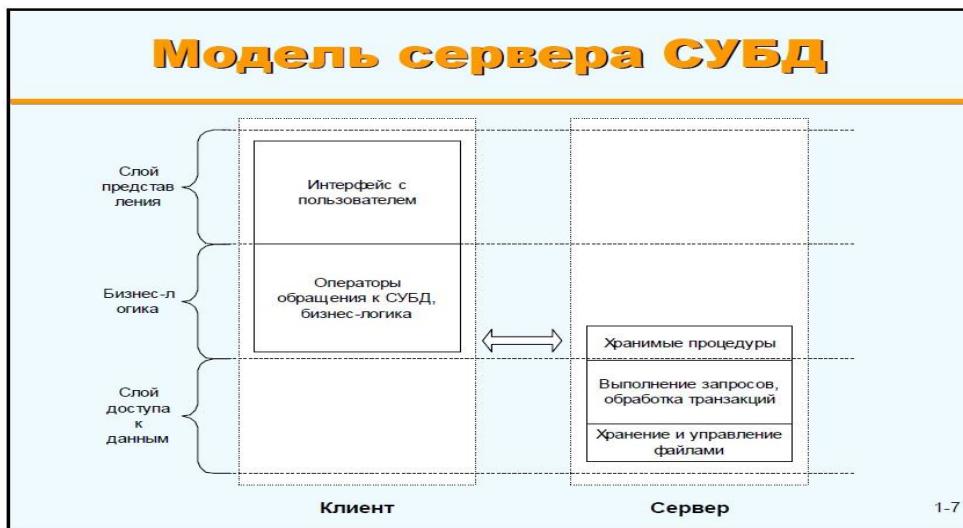
- Клиентская программа работает с данными через запросы к серверному ПО.
- Базовые функции приложения разделены между клиентом и сервером.

Плюсы:

- Полная поддержка многопользовательской работы
- Гарантия целостности данных

Минусы:

- Бизнес логика приложений осталась в клиентском ПО. При любом изменении алгоритмов, надо обновлять пользовательское ПО на каждом клиенте.
- Высокие требования к пропускной способности коммуникационных каналов с сервером, что препятствует использование клиентских станций иначе как в локальной сети.
- Слабая защита данных от взлома, в особенности от недобросовестных пользователей системы.
- Высокая сложность администрирования и настройки рабочих мест пользователей системы.
- Необходимость использовать мощные ПК на клиентских местах.
- Высокая сложность разработки системы из-за необходимости выполнять бизнеслогику и обеспечивать пользовательский интерфейс в одной программе.



Нетрудно заметить, что большинство недостатков классической или 2-х слойной архитектуры клиент-сервер происходят от использования клиентской станции в качестве исполнителя бизнес-логики ИС. Поэтому очевидным шагом дальнейшей эволюции архитектур ИС явилась идея "тонкого клиента", то есть разбиения алгоритмов обработки данных на части связанные с выполнением бизнес-функций и связанные с отображением информации в удобном для человека представлении. При этом на клиентской машине оставляют лишь вторую часть, связанную с первичной проверкой и отображением информации, перенося всю реальную функциональность системы на серверную часть.

Переходная к трехслойной архитектуре (2.5 слоя)

Использование хранимых процедур и вычисление данных на стороне сервера сокращают трафик, увеличиваю безопасность. Клиент все равно реализует часть бизнеслогики.

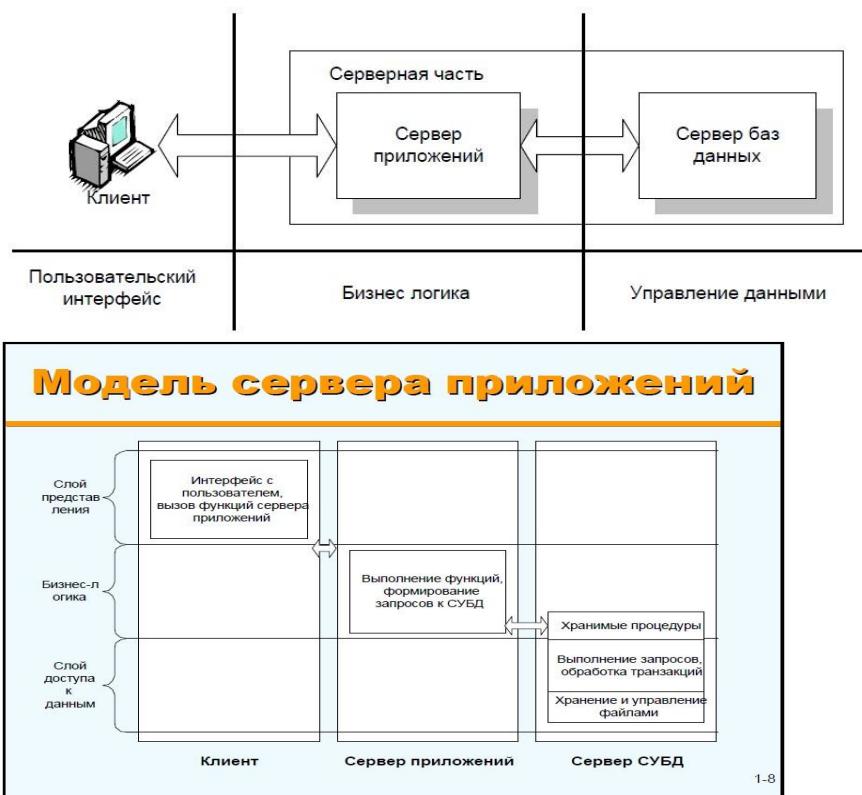
Как видно, такая организация системы весьма напоминает организацию первых унитарных систем с той лишь разницей, что на пользовательском месте стоит не терминал (с пресловутым зеленым экраном), а персональный компьютер, обеспечивающий ГИП, например, в последнее время в качестве клиентских программ часто применяют стандартные www-броузеры. Конечно, такой возврат к почти унитарным системам произошел уже на ином технологическом уровне. Обязательным стало использование СУБД со всеми их преимуществами. Программы для серверной части пишут, в основном, на специализированных языках, пользуясь механизмом хранимых процедур сервера БД. Таким образом, на уровне логической организации, ИС в архитектуре клиент-сервер с тонким клиентом расщепляется на три слоя - слой данных, слой бизнес-функций (хранимые процедуры) и слой представления. К сожалению, обычно, в такой схеме построения ИС не удается написать всю бизнес-логику приложения на не предназначенные для этого встроенных языках СУБД. Поэтому, очень часто часть бизнесфункций реализуется в клиентской части систем, которая от этого неотвратимо "толстеть". Отчасти поэтому, отчасти потому, что физически такие ИС состоят из двух компонентов, эту архитектуру часто называют 2.5-слойный клиент-сервер.

В отличие от 2-х слойной архитектуры 2.5-слойная архитектура обычно не требует наличия высокоскоростных каналов связи между клиентской и серверной частями системы, так как по сети передаются уже готовые результаты вычислений - почти все вычисления производятся на серверной стороне. Существенно улучшается также и защита информации - пользователям даются права на доступ к функциям системы, а не на доступ к ее данным и т.д. Однако вместе с преимуществами унитарного подхода архитектура 2.5 перенимает и

все его недостатки, как-то: ограниченную масштабируемость, зависимость от программной платформы, ограниченное использование сетевых вычислительных ресурсов. Кроме того программы для серверной части системы пишутся на встроенных в СУБД языках описания хранимых процедур, предназначенных для валидации данных и построения несложных отчетов, а вовсе не для написания ИС масштаба предприятия. Все это снижает быстродействие системы, повышает трудоемкость создания с модификации ИС и самым негативным образом оказывается на стоимости аппаратных средств, необходимых для ее функционирования.

Трехуровневая клиент-серверная архитектура

Для решения этих проблем и была предложена так называемая 3-х слойная архитектура клиент-сервер. Основным ее отличием от архитектуры 2.5 является **физическое** разделение программ, отвечающих за хранение данных (СУБД) от программ эти данные обрабатывающих (сервер приложения (СП), application server (AS)). Такое разделение программных компонент позволяет оптимизировать нагрузки как на сетевое, так и на вычислительное оборудование комплекса.



Компоненты трёхзвенной архитектуры, с точки зрения программного обеспечения реализуют определенные сервера БД, web-сервера и браузеры. Место любого из этих компонентов может занять программное обеспечение любого производителя. Ниже представлено описание взаимодействия компонентов трехуровневой архитектуры клиентсерверного приложения. Сервер БД представлен MySQL-сервером; сервер приложений технологиями: ADO.NET, ASP.NET и web-сервером IIS; роль клиента выполняет любой web-браузер.

Браузер клиента 1-> Сервер IIS 2-> Исполняющая среда ASP.NET 2.0 3-> Провайдер данных ADO.NET 2.0 4-> Сервер MySQL 5-> Провайдер данных ADO.NET 2.0 6-> Исполняющая среда ASP.NET 2.0 7-> Сервер IIS 8-> Браузер клиента

- 1 — браузер клиента отправляет HTTP-запрос;
- 2 — на стороне сервера служба Web Internet Information Server (web-сервер IIS) определяет тип запрашиваемого ресурса, и для случая запроса *.aspx (расширение файлов страниц ASP.NET) загружает соответствующее ему (запросу) расширение Internet Server Application Programming Interface (ISAPI). Для страниц aspx это расширение isapi_aspnet.dll. IIS также осуществляет идентификацию и авторизацию пользователя от которого поступил запрос. В свою очередь расширение isapi_aspnet.dll загружает фабрику обработчиков ASP.NET. Далее, фабрика обработчиков создает объектную модель запрашиваемой страницы и обрабатывает действия пользователя.
- 3 — в ходе генерации ответа приложению ASP.NET может потребоваться обращение к БД, в этом случае используя библиотеки классов провайдера данных ADO.NET 2.0, выполняющая среда обращается к серверу БД;
- 4 — провайдер данных ADO.NET 2.0 передает запрос на операцию с БД серверу MySQL;
- 5 — сервер MySQL осуществляет обработку запроса, выполняя соответствующие операции с БД ;
- 6 — провайдер данных ADO.NET 2.0 передает результаты запроса объекту страницы;
- 7 — объект страницы с учетом полученных данных осуществляет рендеринг графического интерфейса страницы и направляет результаты в выходной поток; ▪ 8 — сервер IIS отправляет содержимое сгенерированной страницы клиентскому браузеру.

Плюсы:

1. Тонкий клиент.
2. Между клиентской программой и сервером приложения передается лишь минимально необходимый поток данных - аргументы вызываемых функций и возвращаемые от них значения. Это теоретический предел эффективности использования линий связи, даже работа с ANSI-терминалами (не говоря уже об использование протокола http) требует большей нагрузки на сеть.
3. Сервер приложения ИС может быть запущен в одном или нескольких экземплярах на одном или нескольких компьютерах, что позволяет использовать вычислительные мощности организации столь эффективно и безопасно как этого пожелает администратор ИС.
4. Дешевый трафик между сервером приложений и СУБД. Трафик между сервером приложений и СУБД может быть большим, однако это всегда трафик локальной сети, а их пропускная способность достаточно велика и дешева. В крайнем случае, всегда можно запустить СП и СУБД на одной машине, что автоматически сведет сетевой трафик к нулю.
5. Снижение нагрузки на сервер данных по сравнению с 2.5-слойной схемой, а значит и повышение скорости работы системы в целом.
6. Дешевле наращивать функциональность и обновлять ПО.

Минусы:

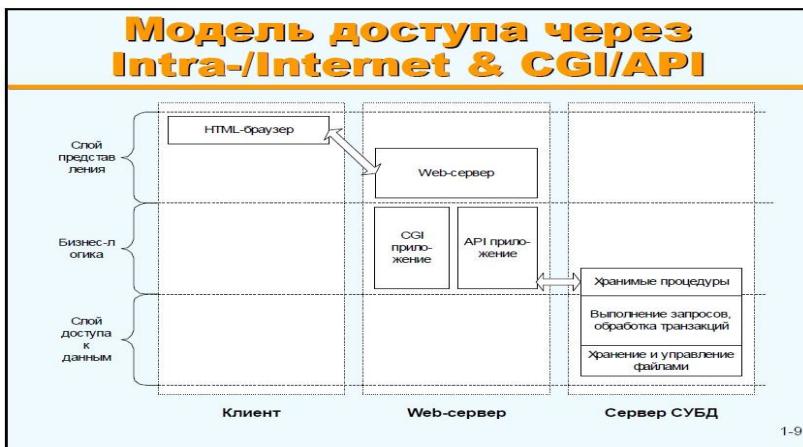
1. Выше расходы на администрирование и обслуживание серверной части.

Масштабируемость систем выполненных в 3-х слойной архитектуре впечатляет. Одна и та же система может работать как на одном отдельно стоящем компьютере, выполняя на нем программы СУБД, СП и клиентской части, так и в сети, состоящей из сотен и тысяч

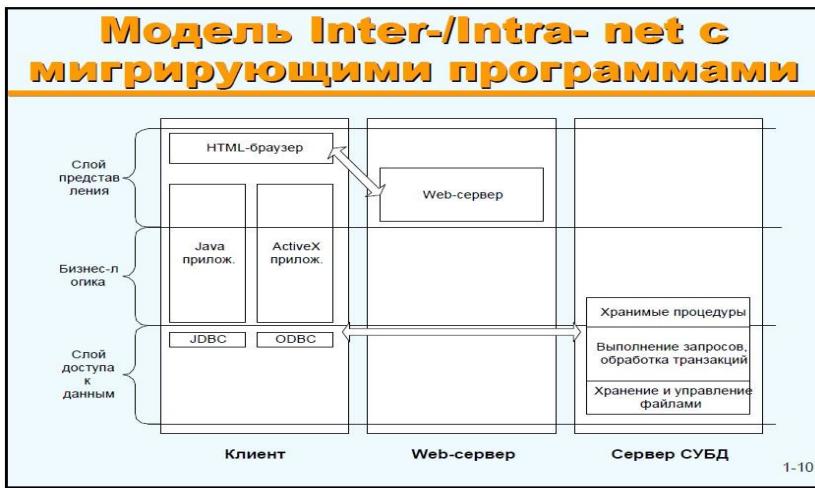
машин. Как уже было отмечено, единственным фактором, препятствующим бесконечной масштабируемости, является лишь требование ведения единой базы данных.

Помимо требования увеличения производительности системы с ростом масштабов деятельности важным фактором является и расширение ее функциональной наполненности. И здесь 3-х слойная схема выигрывает у своих предшественников. Для расширения функциональности не обязательно менять **всю** систему как в случае 2.5слойной схемы - достаточно установить новый сервер приложения с требуемой функцией. Отпадают и многие проблемы связанные с переустановкой клиентских частей программы на множестве компьютеров, быть может весьма удаленных, столь актуальные для 2слойной схемы - парадигма "тонкого" клиента предоставляет для этого целый ряд возможностей.

Internet/Intranet – технологии



Архитектура на основе Internet/Intranet с мигрирующими программами



Распределенные информационные системы.

В литературе можно найти различные определения распределенных систем, причем ни одно из них не является удовлетворительным и не согласуется с остальными. Для наших задач хватит достаточно вольной характеристики.

Распределенная система — это набор независимых вычислительных машин, представляющийся их пользователям единой объединенной системой.

В этом определении оговариваются два момента. Первый относится к аппаратуре: все машины автономны.

Второй касается программного обеспечения: пользователи думают, что имеют дело с единой системой. Важны оба момента. Позже в этой главе мы к ним вернемся, но сначала рассмотрим некоторые базовые вопросы, касающиеся как аппаратного, так и программного обеспечения.

Характеристики распределенных систем:

1. От пользователей скрыты различия между компьютерами и способы связи между ними. То же самое относится и к внешней организации распределенных систем.
2. Пользователи и приложения единообразно работают в распределенных системах, независимо от того, где и когда происходит их взаимодействие.

Распределенные системы должны также относительно легко поддаваться расширению, или масштабированию. Эта характеристика является прямым следствием наличия независимых компьютеров, но в то же время не указывает, каким образом эти компьютеры на самом деле объединяются в единую систему.

Распределенные системы обычно существуют постоянно, однако некоторые их части могут временно выходить из строя. Пользователи и приложения не должны уведомляться о том, что части системы заменены или починены или, что добавлены новые для поддержки дополнительных пользователей.

Для того чтобы поддержать представление системы в едином виде, организация распределенных систем часто включает в себя дополнительный уровень программного обеспечения, находящийся между верхним уровнем, на котором находятся пользователи и приложения, и нижним уровнем, состоящим из операционных систем.



Рис. 1.1. Распределенная система организована в виде службы промежуточного уровня.

Соответственно, такая распределенная система обычно называется *системой промежуточного уровня* (*middleware*). Отметим, что промежуточный уровень распределен среди множества компьютеров.

Особенности распределенных ИС

- Ссылки
- Задержки выполнения запросов
- Активация/деактивация
- Постоянное хранение
- Параллельное исполнение
- Отказы

- Безопасность

Ссылки

Ссылки на объекты в программных модулях на ОО языках программирования (например, C++) являются указателями в памяти.

1. Ссылки на объекты в распределенных системах в противоположность являются более комплексными:

1.1.Содержат информацию о размещении

1.2.Информацию о безопасности

1.4.Ссылки на объектные типы

2. Ссылки на распределенные объекты значительно больше (40 байт для Orbix)

Задержки выполнения запросов

Локальные вызовы требуют порядка пары сотен наносекунд

Запрос к объекту требует от 0.1 до 10 миллисекунд

Интерфейсы в распределенной системе должны быть спроектированы так, чтобы снизить время выполнения запросов:

1. Снизить частоту обращения;

2. Укрупнить выполняемые функции.

Активация/Деактивация

Объекты в ОО языках находятся в виртуальной памяти от создания до уничтожения
В распределенных системах

1. Больше объектов

2. Объекты могут не использоваться на протяжении долгого времени

Реализации распределенных объектов

1. Переносятся в память при активации

2. Удаляются из памяти при деактивации

Постоянное хранение

Объекты могут иметь или не иметь состояние.

Объекты имеющие состояние должны сохранять его на постоянный носитель между:

1. Деактивацией объекта

2. Активацией объекта Может быть достигнуто:

1. Записью в файловую систему

2. Отражением на реляционные БД

3. С помощью объектных БД

Параллельное исполнение

В нераспределенных системах исполнение в основном последовательное, иногда конкурентное в разных нитях процессов.

Распределенные компоненты выполняются параллельно, что приводит к необходимости согласования выполнения.

Отказы

Запросы в распределенных системах имеют большую вероятность отказов

Клиенты обязаны проверять факт выполнения запросов сервером

Безопасность

Безопасность в ОО приложениях может выполняться на основе контроля сеансов.

При работе распределенных систем возникают вопросы безопасности:

1. Кто запрашивает выполнение операции?
2. Как мы можем удостовериться, что субъект является именно тем за кого он себя выдает?
3. Как мы примем решение предоставлять или нет субъекту право на выполнение сервиса?
4. Как мы можем неопровергимо доказать, что сервис был предоставлен?

Контрольные вопросы

1. Дать определение сети.
2. Чем отличается коммуникационная сеть от информационной сети?
3. Как разделяются сети по территориальному признаку?
4. Что такая информационная система?
5. Что такое каналы связи?
6. Дать определение физического канала связи.
7. Дать определение логического канала связи.
8. Как называется совокупность правил обмена информацией между двумя или несколькими устройствами?
9. Как называется объект, способный осуществлять хранение, обработку или передачу данных, в состав которого входят компьютер, программное обеспечение, пользователи и др. составляющие, предназначенные для процесса обработки и передачи данных?
10. Каким параметром характеризуется загрузка сети?
11. Что такое метод доступа?
12. Что такая совокупность правил, устанавливающих процедуры и формат обмена информацией?
13. Чем отличается рабочая станция в сети от обычного персонального компьютера?
14. Какие элементы входят в состав сети?
15. Как называется описание физических соединений в сети?
16. Что такое архитектура сети?
17. Как назвать способ определения, какая из рабочих станций сможет следующей использовать канал связи?
18. Перечислить преимущества использования сетей.
19. Чем отличается одноранговая архитектура от клиент серверной архитектуры?
20. Каковы преимущества крупномасштабной сети с выделенным сервером?
21. Какие сервисы предоставляет клиент серверная архитектура?
22. Преимущества и недостатки архитектуры терминал – главный компьютер.
23. В каком случае используется одноранговая архитектура?
24. Что характерно для сетей с выделенным сервером?

Задание

КАРТОЧКА № 1

ТЕМА: Базовые сетевые топологии и комбинированные топологические решения

ЗАДАНИЕ: Ответьте на вопросы, заполнив пропуски

- 1) топология –
.....;
.....;
- 2) – объект, предоставляющий сервис другим объектам сети по их запросам;
- 3) клиент –
.....;
.....;
- 4) недостатки конфигурации «кольцо»:
а)
б)
- 5) К каким топологиям относятся следующие локальные сети?
а) ЛВС в торговом центре без центрального управления, но с большим количеством периферийных устройств (≈ 1200)
б) Крупная организация со сложными связями, нуждающаяся в центральном управлении и использовании глобальной сети.

КАРТОЧКА № 2

ТЕМА: Базовые сетевые топологии и комбинированные топологические решения

ЗАДАНИЕ: Ответьте на вопросы, заполнив пропуски

- 1) сервер –
.....;
.....;
- 2) – рабочие станции, которые используют ресурсы сервера и предоставляют удобные интерфейсы пользователя;
- 3) архитектура –
.....;
.....;
- 4) недостатки конфигурации «шина»:
а).....
б)
- 5) К каким топологиям относятся следующие локальные сети?
а) ЛВС отдела снабжения, связывающая три компьютера для пользования общими периферийными устройствами
б) ЛВС, соединяющая бухгалтерию и склад и требующая устойчивость к высоким перегрузкам