**«Информационные технологии в экономике»**

ТЕМА 4. Основы сетевых информационных технологий

4.1. Модель взаимосвязи открытых систем

4.2. Организация взаимодействия устройств в сети

4.2.1. Методы передачи данных в сетях ЭВМ

4.2.2. Средства коммутации в компьютерных сетях

4.2.3. Организация сложных связей в глобальных сетях

4.3. Технология клиент-сервер

4.3.1. Модель файлового сервера. (FS)

4.3.2. Модель доступа к удаленным данным (RDA)

4.3.3. Модель сервера баз данных (DBS)

4.3.4. Модель сервера приложений (AS)

4.4. Технология работы в среде распределенной обработки данных

4.5. Базовые технологии обработки запросов в архитектурах

файл-сервера и клиент-сервера

## 4.1. Модель взаимосвязи открытых систем

Развитие средств вычислительной техники, а особенно появление персональных компьютеров привело к созданию нового типа информационно-вычислительных систем под *названием локальная вычислительная сеть* (ЛВС).

ЛВС нашли широкое применение в системах автоматизированного проектирования и технологической подготовки производства, системах управления производством и технологическими комплексами, в конторских системах, бортовых системах управления и т.д. ЛВС является эффективным способом построения сложных систем управления различными производственными подразделениями. ЛВС интенсивно внедряются в медицину, сельское хозяйство, образование, науку и др.

Локальная сеть - (LAN - Local Area Network), данное название соответствует объединению компьютеров, расположенных на сравнительно небольшой территории (одного предприятия, офиса, одной комнаты). Существующие стандарты для ЛВС обеспечивают связь между компьютерами на расстоянии от 2,5 км до 6 км (Ethernet и ARCNET, соответственно).

ЛВС - набор аппаратных средств и алгоритмов, обеспечивающих соединение компьютеров, других периферийных устройств (принтеров, дисковых контроллеров и т.п.) и позволяющих им совместно использовать общую дисковую память, периферийные устройства, обмениваться данными.

В настоящее время информационно-вычислительные системы принято делить на 3 основных типа:

- ***LAN*** (Lokal Area Network) - локальная сеть в пределах предприятия, учреждения, одной организации;

- ***MAN*** (Metropolitan Area Network) - городская или региональная сеть, т.е. сеть в пределах города, области и т.п.;

- ***WAN*** (Wide Area Network) - глобальная сеть, соединяющая абонентов страны, континента, всего мира.

Информационные системы, в которых средства передачи данных принадлежат одной компании и используются только для нужд этой компании , принято называть *Сеть Масштаба Предприятия* или *Корпоративная Сеть (Enterprise Network).* Для автоматизации работы производственных предприятий часто используются системы на базе протоколов *MAP/TOP:*

MAP (Manufacturing Automation Protocol) - сеть для производственных предприятий, заводов (выполняется автоматизация работы конструкторских отделов и производственных, технологических цехов). МАР позволяет создать единую технологическую цепочку от конструктора, разработавшего деталь, до оборудования, на котором изготавливают эту деталь.

TOP (Technical and Office Protocol) - протокол автоматизации технического и административного учреждения.

МАР/ТОР системы, полностью автоматизирующие работу производственного предприятия.

Основное назначение ЛВС - в распределении ресурсов ЭВМ: программ, совместимости периферийных устройств, терминалов, памяти. Следовательно, ЛВС должна иметь надежную и быструю систему передачи данных, стоимость которой должна быть меньше по сравнению со стоимостью подключаемых рабочих станций. Иными словами, стоимость передаваемой единицы информации должна быть значительно ниже стоимости обработки информации в рабочих станциях. Исходя из этого ЛВС, как система распределенных ресурсов, должна основываться на следующих принципах:

- единой передающей среды;

- единого метода управления;

- единых протоколов;

- гибкой модульной организации;

- информационной и программной совместимости.

Международная организация по стандартизации (ISO), основываясь на опыте многомашинных систем, который был накоплен в разных странах, выдвинула концепцию архитектуры открытых систем - эталонную модель, используемую при разработке международных стандартов.

На основе этой модели вычислительная сеть предстает как распределенная вычислительная среда, включающая в себя большое число разнообразных аппаратных и программных средств. *По вертикали* данная среда представляется рядом логических уровней, на каждый из которых возложена одна из задач сети. *По горизонтали* информационно-вычислительная среда делится на локальные части (открытые системы), отвечающие требованиям и стандартам структуры открытых систем.

Часть открытой системы, выполняющая некоторую функцию и входящая в состав того или иного уровня, называется *объектом*.

Правила, по которым осуществляется взаимодействие объектов одного и того же уровня, называются *протоколом* (методика связи).

Протоколы определяют порядок обмена информацией между сетевыми объектами. Они позволяют взаимодействующим рабочим станциям посылать друг другу вызовы, интерпретировать данные, обрабатывать ошибочные ситуации и выполнять множество других различных функций. Суть протоколов заключается в регламентированных обменах точно специфицированными командами и ответами на них (например, назначение физического уровня связи - передача блоков данных между двумя устройствами, подключенными к одной физической среде).

Каждый уровень подразделяется на две части:

- спецификацию услуг;

- спецификацию протокола.

Спецификация услуг определяет, *что делает уровень*, а спецификация протокола - *как он это делает*.

Причем, каждый конкретный уровень может иметь более одного протокола.

Большое число уровней, используемых в модели, обеспечивает декомпозицию информационно-вычислительного процесса на простые составляющие. В свою очередь, увеличение числа уровней вызывает необходимость включения дополнительных связей в соответствии с дополнительными протоколами и интерфейсами. Интерфейсы (макрокоманды, программы) зависят от возможностей используемой ОС.

Международная организация по стандартизации предложила *семиуровневую модель*, которой соответствует и программная структура (рис.4.1.).



Рассмотрим функции, выполняемые каждым уровнем программного обеспечения:

1.*Физический* - осуществляет как соединения с физическим каналом, так и расторжение, управление каналом, а также определяется скорость передачи данных и топология сети.

2.*Канальный* - осуществляет обрамление передаваемых массивов информации вспомогательными символами и контроль передаваемых данных. В ЛВС передаваемая информация разбивается на несколько пакетов или кадров. Каждый пакет содержит адреса источника и места назначения, а также средства обнаружения ошибок.

3.*Сетевой* - определяет маршрут передачи информации между сетями (ПЭВМ), обеспечивает обработку ошибок, а так же управление потоками данных.

Основная задача сетевого уровня - маршрутизация данных (передача данных между сетями). Специальные устройства - *Маршрутизаторы (Router)* определяют для какой сети предназначено то или другое сообщение, и направляет эту посылку в заданную сеть. Для определения абонента внутри сети используется *Адрес Узла (Node Address).* Для определения пути передачи данных между сетями на маршрутизаторах строятся *Таблицы Маршрутов (Routing Tables)*, содержащие последовательность передачи данных через маршрутизаторы. Каждый маршрут содержит адрес конечной сети, адрес следующего маршрутизатора и стоимость передачи данных по этому маршруту. При оценке стоимости могут учитываться количество промежуточных маршрутизаторов, время, необходимое на передачу данных, просто денежная стоимость передачи данных по линии связи. Для построения таблиц маршрутов наиболее часто используют либо *Метод Векторов* либо *Статический Метод*. При выборе оптимального маршрута применяют динамические или статические методы. На сетевом уровне возможно применение одной из двух процедур передачи пакетов:

- *датаграмм* - т.е., когда часть сообщения или пакет независимо доставляется адресату по различным маршрутам, определяемым сложившейся динамикой в сети. При этом каждый пакет включает в себя полный заголовок с адресом получателя. Процедуры управления передачей таких пакетов по сети называются датаграммной службой;

- *виртуальных соединений* - когда установление маршрута передачи всего сообщения от отправителя до получателя осуществляется с помощью специального служебного пакета - запроса на соединение. В таком случае для этого пакета выбирается маршрут и, при положительном ответе получателя на соединение закрепляется для всего последующего трафика (потока сообщений в сети передачи данных) и получает номер соответствующего виртуального канала (соединения) для дальнейшего использования его другими пакетами того же сообщения. Пакеты, которые передаются по одному виртуальному каналу, не являются независимыми и поэтому включают сокращенный заголовок, включающий порядковый номер пакета, принадлежащему одному сообщению.

*Недостатки:* значительная по сравнению с датаграммой сложность в реализации, увеличение накладных расходов, вызванных установлением и разъединением сообщений.

*ВЫВОД.* Датаграммный режим предпочтительнее использовать для сетей сложной конфигурации, где значительное число ЭВМ в сети, иерархическая структура сети, надежность, достоверность передачи данных по каналам связи, длина пакета более 512 байт.

4.*Транспортный* - связывает нижние уровни (физический, канальный, сетевой) с верхними уровнями, которые реализуются программными средствами. Этот уровень как бы разделяет средства формирования данных в сети от средств их передачи. Здесь осуществляется разделение информации по определенной длине и уточняется адрес назначения. Транспортный уровень позволяет мультиплексировать передаваемые сообщения или соединения. Мультиплексирование сообщений позволяет передавать сообщения одновременно по нескольким линиям связи, а мультиплексирование соединений - передает в одной посылке несколько сообщений для различных соединений.

5.*Сеансовый* - на данном уровне осуществляется управление сеансами связи между двумя взаимодействующими пользователями (определяет начало и окончание сеанса связи: нормальное или аварийное; определяет время, длительность и режим сеанса связи; определяет точки синхронизации для промежуточного контроля и восстановления при передаче данных; восстанавливает соединение после ошибок во время сеанса связи без потери данных.

6.*Представительский* - управляет представлением данных в необходимой для программы пользователя форме, генерацию и интерпретацию взаимодействия процессов, кодирование/декодирование данных, в том числе компрессию и декомпрессию данных. На рабочих станциях могут использоваться различные операционные системы : DOS, UNIX, OS/2. Каждая из них имеет свою файловую систему, свои форматы хранения и обработки данных. Задачей данного уровня является преобразование данных при передаче информации в формат, который используется в информационной системе. При приеме данных данный уровень представления данных выполняет обратное преобразование. Таким образом появляется возможность организовать обмен данными между станциями, на которых используются различные операционные системы.

Форматы представления данных могут различаться по следующим признакам:

- порядок следования битов и размерность символа в битах;

- порядок следования байтов;

- представление и кодировка символов;

- структура и синтаксис файлов.

Компрессия или упаковка данных сокращает время передачи данных. Кодирование передаваемой информации обеспечивает защиту ее от перехвата.

7.*Прикладной* - в его ведении находятся прикладные сетевые программы, обслуживающие файлы, а также выполняет вычислительные, информационно-поисковые работы, логические преобразования информации, передачу почтовых сообщений и т.п. Главная задача этого уровня - обеспечить удобный интерфейс для пользователя.

На разных уровнях обмен происходит различными единицами информации: биты, кадры, пакеты, сеансовые сообщения, пользовательские сообщения.

#### *Протоколы в ЛВС*

Организация ЛВС базируется на принципе многоуровневого управления процессами, включающими в себя иерархию протоколов и интерфейсов.

Протокол УФК определяет форму представления и порядок передачи данных через физический канал связи, фиксирует начало и конец кадра, который несет в себе данные, формирует и принимает сигнал со скоростью, присущей пропускной способности канала.

Второй уровень (канальный) можно разделить на два подуровня: управление доступом к каналу (УДК) и управление информационным каналом (УИК).

Протокол УДК устанавливает порядок передачи данных через канал, выборку данных.

Протокол УИК обеспечивает достоверность данных, т.е. формируются проверочные коды при передаче данных.

Во многих ЛВС отпадает необходимость в сетевом уровне. К нему прибегают при комплексировании нескольких ЛВС, содержащих моноканалы.

Протокол УП обеспечивает транспортный интерфейс, ликвидирующий различия между потребностями процессов в обмене данными и ограничениями информационного канала, организуемого нижними уровнями управления. Протоколы высоких уровней - УС, УПД, УПП - по своим функциям аналогичны соответствующим протоколам глобальных сетей, т.е. реализуется доступ терминалов к процессам, программ к удаленным файлам, передача файлов, удаленный ввод заданий, обмен графической информацией и др.

## 4.2. Организация взаимодействия устройств в сети

В зависимости от способа организации обработки данных и взаимодействия пользователей, который поддерживается конкретной сетевой операционной системой, выделяют два типа информационных систем:

- иерархические сети;

- сети клиент/сервер.

В *иерархических сетях* все задачи, связанные с хранением, обработкой данных, их представлением пользователям, выполняет центральный компьютер. Пользователь взаимодействует с центральным компьютером с помощью терминала. Операциями ввода/вывода информации на экран управляет центральный компьютер.

*Достоинства иерархических систем:*

- отработанная технология обеспечения отказоустойчивости, сохранности данных;

- надежная система защиты информации и обеспечения секретности.

*Недостатки:*

- высокая стоимость аппаратного и программного обеспечения, высокие эксплуатационные расходы;

- быстродействие и надежность сети зависят от центрального компьютера.

Примеры иерархических систем: SNA, IBM Corp., DNA, DEC.

В системах *клиент/сервер* обработка данных разделена между двумя объектами: клиентом и сервером. Клиент - это задача, рабочая станция, пользователь. Он может сформировать запрос для сервера: считать файл, осуществить поиск записи и т.п. Сервер - это устройство или компьютер, выполняющий обработку запроса. Он отвечает за хранение данных, организацию доступа к этим данным и передачу данных клиенту. В системах клиент/сервер нагрузка по обработке данных распределена между клиентом и сервером, поэтому требования к производительности компьютеров, используемых в качестве клиента и сервера, значительно ниже, чем в иерархических системах.

По организации взаимодействия принято выделять два типа систем, использующих метод клиент/сервер:

- равноправная сеть;

- сеть с выделенным сервером.

*Равноправная сеть* - это сеть, в которой нет единого центра управления взаимодействием рабочих станций, нет единого устройства хранения данных. Операционная система такой сети распределена по всем рабочим станциям, поэтому каждая рабочая станция одновременно может выполнять функции как сервера, так и клиента. Пользователю в такой сети доступны все устройства (принтеры, жесткие диски и т.п.), подключенные к другим рабочим станциям.

*Достоинства:* низкая стоимость (используются все компьютеры, подключенные к сети, и умеренные цены на прогр. обеспечение для работы сети); высокая надежность (при выходе из строя одной рабочей станции, доступ прекращается лишь к некоторой части информации).

*Недостатки:* работа сети эффективна только при количестве одновременно работающих станций не более 10; трудности организации эффективного управления взаимодействием рабочих станций и обеспечение секретности информации; трудности обновления и изменения ПО рабочих станций.

*Сеть с выделенным сервером* - здесь один из компьютеров выполняет функции хранения данных общего пользования, организации взаимодействия между рабочими станциями, выполнения сервисных услуг - *сервер* сети. На таком компьютере выполняется операционная система, и все разделяемые устройства (жесткие диски, принтеры, модемы и т.п.) подключаются к нему, выполняет хранение данных, печать заданий, удаленная обработка заданий. Рабочие станции взаимодействуют через сервер, поэтому логическую организацию такой сети можно представить топологией "звезда", где центральное устройство - сервер.

*Достоинства:* выше скорость обработки данных (определяется быстродействием центрального компьютера, и на сервер устанавливается специальная сетевая операционная система, рассчитанная на обработку и выполнение запросов, поступивших одновременно от нескольких пользователей); обладает надежной системой защиты информации и обеспечения секретности; проще в управлении по сравнению с равноправными.

*Недостатки:* такая сеть дороже из-за отдельного компьютера под сервер; менее гибкая по сравнению с равноправной.

Сети с выделенным сервером являются более распространенными. Примеры сетевых операционных систем такого типа: LAN Server, IBM Corp., VINES, Banyan System Inc., NetWare, Novell Inc.

### 4.2.1. Методы передачи данных в сетях ЭВМ

При обмене данными между узлами используются *три метода передачи данных:*

- симплексная (однонаправленная) передача (телевидение, радио);

- полудуплексная ( прием/передача информации осуществляется поочередно);

- дуплексная (двунаправленная), каждая станция одновременно передает и принимает данные.

Для передачи данных в информационных системах наиболее часто применяется последовательная передача. Широко используются следующие методы последовательной передачи: асинхронная и синхронная.

При *асинхронной* передаче каждый символ передается отдельной посылкой (рис.4.2).



При асинхронной передаче каждый символ передается отдельной посылкой. Стартовые биты предупреждают приемник о начале передачи. Затем передается символ. Для определения достоверности передачи используется бит четности (бит четности =1, если количество единиц в символе нечетно, и 0, в противном случае. Последний бит "стоп бит" сигнализирует об окончании передачи.

*Преимущества:* несложная отработанная система; недорогое (по сравнению с синхронным) интерфейсное оборудование.

*Недостатки:* третья часть пропускной способности теряется на передачу служебных битов (старт/стоповых и бита четности); невысокая скорость передачи по сравнению с синхронной; при множественной ошибке с помощью бита четности невозможно определить достоверность полученной информации.

Асинхронная передача используется в системах, где обмен данными происходит время от времени и не требуется высокая скорость передачи данных. Некоторые системы используют бит четности как символьный бит, а контроль информации выполняется на уровне протоколов обмена данными (Xmodem, Zmodem,MNP).

При использовании синхронного метода данные передаются блоками. Для синхронизации работы приемника и передатчика в начале блока передаются биты синхронизации. Затем передаются данные, код обнаружения ошибки и символ окончания передачи. При синхронной передаче данные могут передаваться и как символы, и как поток битов. В качестве кода обнаружения ошибки обычно используется *Циклический Избыточный Код Обнаружения Ошибок (CRC). Он вычисляется по содержимому поля данных и позволяет однозначно определить достоверность принятой информации.*

*Преимущества:* высокая эффективность передачи данных; высокие скорости передачи данных; надежный встроенный механизм обнаружения ошибок.

*Недостатки:* интерфейсное оборудование более сложное и, соответственно, более дорогое.

Протоколы SDLC и HDLC основываются на синхронной бит-ориентированной передаче данных.

### 4.2.2. Средства коммутации в компьютерных сетях

ЛВС можно создавать с любым из типов кабеля. Самым дешевым является кабель *Витая пара* со скрученной парой проводов, который используется в телефонии. Он может быть *Экранированным* и *Неэкранированным.* Экранированный более устойчив к электромагнитным помехам. Однако на практике чаще используется неэкранированный кабель, т.к. такой тип кабеля используется для разводки телефонных линий и, он дешевле экранированного. Наилучшим образом подходит для малых учреждений. Недостатками данного кабеля является высокий коэффициент затухания сигнала и высокая чувствительность к электромагнитным помехам, поэтому максимальное расстояние между активными устройствами в ЛВС при использовании витой пары до 100 метров.

*Коаксиальный кабель*. Этот кабель может использоваться в двух различных системах передачи данных: без модуляции сигнала и с модуляцией. В первом случае цифровой сигнал используется в таком виде, в каком он поступает из ПК и сразу же передается по кабелю на приемную станцию. Он имеет один канал передачи со скоростью до 10 Мбит/сек и максимальный радиус действия 4000 м. Во втором случае цифровой сигнал превращают в аналоговый и направляют его на приемную станцию, где он снова превращается в цифровой. Операция превращения сигнала выполняется модемом (модулятор/демодулятор); каждая станция должна иметь свой модем. Этот способ передачи является многоканальным (обеспечивает передачу по десяткам каналов, используя для этого всего лишь один кабель).Таким способом можно передавать звуки, видео сигналы, данные. Длина кабеля может достигать до 50 км.

Передача сигнала с модуляцией более дорогостоящая, чем без модуляции. Поэтому, наиболее эффективное его использование при передаче данных между крупными предприятиями.

*Оптоволоконный* кабель является новейшей технологией, используемой в ЛВС. Носителем информации является световой луч, который моделируется сетью и принимает форму сигнала. Такая система устойчива к внешним электрическим помехам и таким образом возможна очень быстрая и безошибочная передача данных (до 2 Гбит/с), и обеспечивает секретность передаваемой информации. Количество каналов в таких кабелях огромно. Передача данных выполняется только в симплексном режиме, поэтому для организации обмена данными устройства необходимо соединять двумя оптическими волокнами (на практике оптоволоконный кабель всегда имеет четное, парное кол-во волокон). К недостаткам можно отнести большую стоимость, а также сложность подсоединения.

Радиоволны в микроволновом диапазоне используются в качестве передающей среды в *Беспроводных Локальных Сетях*, либо между мостами или шлюзами для связи между ЛВС. В первом случае максимальное расстояние между станциями составляет 200-300 м, во втором - это расстояние прямой видимости. Скорость передачи данных - до 2 Мбит/с.

Беспроводные ЛС считаются перспективным направлением развития ЛС. Их преимущество - простота и мобильность. Исчезают проблемы, связанные с прокладкой и монтажом кабельных соединений. Достаточно установить интерфейсные платы на рабочие станции, и сеть готова к работе. Сдерживающим фактором широкого развития БЛС является отсутствие стандарта для таких сетей. Существующие БЛС, выполненные различными фирмами, как правило, полностью несовместимы между собой.

Поэтому необходимо дождаться принятия и опубликования стандарта IEEE 802.11 (Разработка стандартов в области локальных и региональных сетей).

Коммутационная сеть включает в себя множество серверов и ЭВМ, соединенных физическими (магистральными) каналами связи, использующие телефонные, коаксиальные кабели, спутниковые каналы связи. Вычислительные сети по способу передачи информации подразделяются на сети коммутации каналов, сети коммутации сообщений, сети коммутации пакетов и интегральные сети. Каждый из этих методов имеет свои плюсы и минусы. Достоинством сетей коммутации каналов является простота реализации (прямое соединение), а недостатком - низкий коэффициент использования каналов, высокая стоимость передачи данных, повышенное время ожидания других пользователей. При коммутации сообщений передача данных (сообщения) осуществляется после освобождения канала, пока оно не дойдет до адресата. Каждый сервер производит прием, проверку, сборку, маршрутизацию и передачу сообщения. Недостатком данного способа является низкая скорость передачи информации, невозможность ведения диалога между пользователями. К достоинствам можно отнести - уменьшение стоимости передачи, ускорение передачи. Пакетная коммутация подразумевает обмен небольшими пакетами (часть сообщения) фиксированной структуры, которые не дают возможности образования очередей в узлах коммутации. Достоинства: быстрое соединение, надежность, эффективность использования сети. При данном методе проблема передачи пакета решается способом фиксированной маршрутизации. Она предполагает наличие таблицы маршрутов, где закреплен маршрут от одного пользователя к другому. Сети, осуществляющие коммутацию каналов, сообщений и пакетов, называются интегрированными. К таким сетям относится разработанная в настоящее время новая сетевая технология АТМ.

АТМ - это коммуникационная технология, объединяющая принципы коммутации пакетов и каналов для передачи информации различного типа. АТМ - (ассинхронный режим передачи) , данная технология предусматривает интегрированную передачу речи, данных и видеоинформации в едином цифровом виде по одному и тому же каналу связи. Это позволяет отказаться от жестких ограничений по предоставляемой пользователю полосе пропускания канала связи, отказаться от разделения каналов по типам передаваемой информации и значительно расширить круг предоставляемых услуг. Основными достоинствами новой технологии является отсутствие ориентации на какой-либо тип передаваемой информации. Объединяемые в рамках АТМ информационные потоки от источников информации различной природы резко отличаются друг от друга требованиями к полосе пропускания. Если данные ЛВС в большинстве случаев не требуют гарантированного времени доставки пакетов и, соответственно, постоянства полосы пропускания канала связи, то системы кабельного телевидения и передача речи в интерактивном режиме без выполнения этого условия немыслимы. Поэтому процедура установления соединения в АТМ-сети предусматривает предварительное определение типа передаваемой информации, требуемой полосы пропускания и приоритет на занятие канала связи, что минимизирует загрузку межузловых каналов связи и обеспечивает предоставление услуг с заданным качеством.

Главным отличием АТМ от существующих технологий передачи информации является высокая скорость передачи - до 10 Гбит/ на канал связи. (На сегодняшний день - 2,5 Гбит/с). АТМ объективно совмещает функции, выполняемые локальными и глобальными сетями. Удаленным пользователям предоставляется "прозрачный" доступ к любым общим информационным ресурсам, а также обеспечивается всё многообразие услуг глобальных телекоммуникаций. Данная особенность технологии АТМ делает ее незаменимой при создании интегрированных распределенных корпоративных информационных сетей на базе волоконно-оптических каналов связи. Кроме того, эффективными уровнями применения АТМ являются высокоскоростные ЛВС со специфическими требованиями к трафику (содержащему видео- и CAD/CAM-файлы), а также магистральные и абонентские каналы передачи в региональных и внутригородских широкополосных сетях с интеграцией обслуживания.

Основным отличием АТМ от традиционных ЛВС-технологий является то, что АТМ по своей природе ориентирована на установление виртуальных соединений. *Виртуальное соединение* - это сконфигурированная определенным образом среда между двумя или более конечными устройствами для передачи информации. *Виртуальный канал* - фиксированный маршрут, состоящий из последовательности номеров портов коммутаторов, через которые проходят все ячейки при данном сеансе связи от одного пользователя к другому. Виртуальные каналы всегда однонаправленны, т.е. для передачи в обратном направлении между теми же пользователями используются уже другие номера идентификаторов. Понятие виртуального пути используется на каком-либо участке сети: несколько виртуальных каналов проходят по одному и тому же направлению, что дает возможность коммутатору переключать целые группы виртуальных каналов. Каждый физический канал может содержать несколько виртуальных путей и каналов. Так как конфигурация виртуальных соединений не связана с физическими каналами, то топология АТМ сети может быть любой. Коммутаторы при этом могут быть соединены в шину, кольцо или звезду, но чаще - это смесь всех возможных соединений. Это дает возможность реализовывать резервирование связей, что повышает надежность сети.

Обычные локальные сети (Ethernet, Token Ring) не проверяют доступность устройства назначения, а просто посылают туда пакет с информацией. Пакет должен иметь адрес назначения, который проверяется сетевыми устройствами на соответствие со своим собственным адресом. Перед передачей каких-либо сообщений в АТМ станция-источник проверяет доступность станции назначения и, только после этого устанавливается соединение. Только этим двум станциям виден поток информации.

АТМ реализует коммутацию коротких пакетов (ячеек), наложенную на коммутацию виртуальных каналов. В отличие от обычных информационных пакетов ячейки не содержат адресной информации и контрольной суммы. Коммутация происходит на основе идентификатора виртуального канала, определяющего одно из организованных соединений. Контрольная сумма считается ненужной из-за использования высококачественной кабельной системы с малой вероятностью ошибки. АТМ ориентировано на соединение протоколом. Перед передачей информации между пользователями организуется виртуальный или логический канал связи, остающийся в их распоряжении до окончания взаимодействия. Параметры этого этого канала могут быть различными, в зависимости от вида трафика и его интенсивности.

Для передачи звука определяется только потребная фиксированная полоса пропускания, а для файлового обмена между компьютерами даются параметры средней и максимальной интенсивности трафика. Так как ячейки имеют постоянную длину (53 байта), задержки прихода новой информации к потребителю всегда одинаковы. АТМ ячейки легко обрабатываемы при прохождении через коммутатор. При обработке пакета маршрутизатор вначале полностью его принимает в буфер, проверяет контрольную сумму, анализирует адресную информацию, содержание поля данных, и только после этого отправляет данный пакет. Программы современных маршрутизаторов содержат до нескольких миллионов строк кода, отсюда дороговизна таких устройств. В отличие от них коммутатор АТМ решает свои задачи аппаратным путем. Коммутатор, прочитав идентификатор в заголовке ячейки, переправляет ее из одного порта в другой, не задумываясь о ее содержании.

Исходя из вышесказанного можно сделать следующие выводы:

- сеть АТМ имеет всегда большую пропускную способность, чем сумма всех реализованных виртуальных каналов. При этом контроль осуществляется за счет ограничения подключения к сети новых пользователей логическими средствами самой сети;

- управление потоком данных осуществляет оконечное оборудование; сама АТМ сеть не имеет собственных средств для этого;

- на физическом уровне ошибки практически отсутствуют. АТМ сеть не имеет механизма проверки ошибок и их исправлений;

- процент потерянных ячеек очень невелик и предсказуем. АТМ не может функционировать на ненадежных каналах.

Существующие в настоящее время телекоммуникационные системы страдают рядом недостатков:

- зависимость от вида информации, которую они транспортируют;

- отсутствие гибкости, так как современные телекоммуникационные системы практически не обеспечивают адаптацию к изменениям требований со стороны систем управления к объемам передаваемой информации, к скорости передачи, времени доставки и достоверности;

- низкая эффективность использования ресурсов.

В настоящее время появилась возможность создания на базе технологии АТМ единой телекоммуникационной системы - широкополосной цифровой сети интегрального обслуживания (ШЦСИО), которая обеспечит выполнение следующих функций:

- транспортирование всех видов информации с помощью единого ассинхронного метода переноса (АТМ), при котором каждый пользователь получает от сети только тот ресурс, который ему необходим;

- поддержку интерактивных служб и служб распределения информации с выполнением требований как к вероятности блокировки, так и ко времени доставки информации;

- поддержку режимов с установлением и без установления соединения между абонентами;

- передачу как непрерывного, так и поблочного трафика, что за счет мультиплексирования позволяет более эффективно использовать единые сетевые ресурсы;

- преобразование сигналов и сообщений внутри сети на базе цифровой обработки сигналов;

- обеспечение пользователей такими услугами, как телеуправление и телеконтроль, видеотелефон, высокоскоростная передача данных, выдача данных и видеоинформации по требованию.

С каждым днем растет интерес к внедрению в телекоммуникационные сети технологии АТМ, что объясняется такими факторами, как:

- развитие систем удаленной обработки данных, требующих передачи достаточно больших объемов информации практически в реальном масштабе времени;

- непрерывный рост требований к высокоскоростным трактам, объединяющим ЛВС;

- рост потребности пользователей в предоставлении услуг по обмену подвижными и неподвижными изображениями.

В развитии вычислительных сетей наблюдается две тенденции:

- с одной стороны, существует тенденция объединения локальных сетей (LAN) в городские (MAN) и глобальные (WAN) сети с возможностью обеспечения высокоскоростного обмена;

* с другой стороны, в связи с быстрым ростом производительности рабочих станций и ПЭВМ, а также в связи с тем, что станции становятся мультимедиа-терминалами, существует тенденция резкого повышения скорости работы в самих локальных сетях.

### 4.2.3. Организация сложных связей в глобальных сетях

В глобальных сетях связь между ЛВС осуществляется посредством мостов.

**Мосты** - представляют собой программно аппаратные комплексы, которые соединяют ЛВС между собой, а также ЛВС и удаленные рабочие станции (РС), позволяя им взаимодействовать друг с другом для расширения возможностей сбора и обмена информацией.

Мост обычно определяется как соединение между двумя сетями, которые используют одинаковый протокол взаимодействия, одинаковый тип среды передачи и одинаковую структуру адресации.

Существует два базовых типа мостов NETWARE:

|  |  |
| --- | --- |
| 1. внутренний; 2. внешний.   Если мост располагается в файловом сервере - *внутренний мост.* |  |

Если мост располагается в рабочей станции - *внешний мост.*

Внешние мосты и их ПО устанавливаются в рабочей станции, которая функционирует не как файловый сервер. Поэтому внешний мост может передавать данные более эффективно, чем внутренний.

Существуют выделенные и совмещенные мосты.

**Выделенный** -это ПК, использующийся как мост, не может функционировать как рабочая станция.

**Совмещенный** - может функционировать и как мост и как рабочая станция - одновременно.

*Преимущество:* ограничиваются издержки на покупку дополнительного компьютера.

*Недостаток*: отсутствие потенциальных возможностей рабочей станции, размещенной в нем. Когда прикладная программа на РС зависает и вызывает остановку РС, функционирующей как мост программа моста также останавливает операции. Этот сбой прерывает разделение данных между сетями, а также прерывает сеансы работы РС, которые связаны через мост с файловым сервером.

Поскольку выделенный мост не используется как РС, то никакие ПП не вызовут такой сбой и не прервут работу.

Выбирая мост, необходимо сопоставить стоимость оборудования и риск возможности сбоя моста.

**Локальный мост** передает данные между сетями, которые расположены в пределах ограничений кабеля по расстоянию. Локальные мосты применяются в следующих случаях:

1. - для разделения больших сетей на две и более подсетей с целью увеличения быстродействия и уменьшения стоимости линий связи.

Например, в одной организации различные отделы разделяют одну и ту же сеть. Т.к. большие сети медленнее малых, то есть возможность выделить в небольшие подсети компактно расположенные отделы. Используя локальный мост Netware, отделы могут продолжать разделять данные таким образом, как если бы они работали в одной сети, приобретая при этом быстродействие и гибкость, присущие малой сети.



2. - с помощью локального моста можно расширить физические возможности сети. Если сеть Netware имеет максимально допустимое число узлов, поддержи­ваемое её аппаратной схемой адресации и есть необходимость в добавлении ещё нескольких узлов, то для расширения такой сети используется мост Netware. При этом включение в сеть дополнительного файлового сервера необязательно.



3.- объединение сетей в интерсеть.

Чтобы пользователи каждой сети могли получить доступ к информации других сетей, необходимо связать эти сети, образуя интерсеть.



**Удаленные мосты** применяются, когда расстояние **не** позволяет соединять сети посредством кабеля.

*Например:* соединение сети в г.Костроме с сетью г.Новгорода поставит перед необходимостью в использовании удаленного моста, так как ограничение по длине кабеля для локального моста будет превышено.

Удаленный мост использует промежуточную среду передачи (телефонные линии) для соединения с удаленной сетью или удаленными РС.

*При связи сети с удаленной сетью необходимо установить мост на каждом конце соединения, а при связи сети с удаленной РС - мост требуется только на сети.*

***Выбор модемов*** для организации удаленного взаимодействия должен определяться характеристиками и типом каналов связи, а также требованиями к возможностям модемов и их стоимости.

*Примечание:*

V - до 2400 бод - телеф. каналы связи (1бод=1бит/сек), используются с низко и средне - скоростными ассинхронными модемами (ассинхронный) ;

V - до 19,2 бод - в выделенных линиях,(синхронный); (обычно телефонная линия, имеющая максимальную скорость V=64 Кбит/с, либо коммутируемая телеф. линия со скоростью передачи данных V=9600 бит/с).

Удаленные мосты Netware поддерживают два вида методов последовательной передачи: ассинхронный и синхронный.



Основное различие между мостом в защищенном (protected - mode) режиме и мостом в реальном (real - mode) режиме заключается в количестве памяти, которое он может поддерживать.

Защищенный мост позволяет добавлять память, в то время как реальный мост предоставляет минимум памяти.

**Мост в защищенном режиме.** ПО моста в защищенном режиме поддерживает стандартный 1Мбайт памяти моста (640 Кб ОЗУ +доп. память) Оно (ПО) также поддерживает установку плат памяти в общем объеме до 8 Мб. Этот объем дополнительной памяти позволяет иметь мост, на котором могут выполняться доп. процессы (Valua Added Processes - VAP) в объеме памяти вплоть до 7 Мб.

Если планируется установить более чем один или два VAP - процесса, следует выбрать мост в защищенном режиме. При этом необходимо определить доп. количество плат памяти. Число дополняемых плат зависит от того, сколько VAP-процессов планируется выполнять. Если будет выполняться более чем два VAP-процесса, необходимо установить по крайней мере одну плату.

*Примечание.* Если требуется выполнять 4 VAP-процесса, например таких, как VAP печати и VAP обслуживания очереди, мост должен работать в защищенном режиме.

Прежде чем использовать мост в защищенном режиме, необходимо убедиться в соответствии типа компьютера возможности работы в совмещенном режиме.

**Мост в реальном режиме.** ПО моста в реальном режиме поддерживает стандартные 640 Кб основной памяти, в этом случае в мосте может выполняться один или два дополнительных ориентированных процесса (VAP). Мосты в реальном режиме могут быть как выделенными, так и совмещенными.

Вычислительная сеть позволяет пользователям сети использовать в своих работах сервис сетевой печати. Сетевыми печатающими устройствами (ПУ) могут быть принтеры, плоттеры или любые периферийные устройства.

ПУ является сетевым, если оно подключено извне к рабочей станции (РС) или сети, и может быть использовано в интересах различных пользователей или групп пользователей сети с различных участков сети. Последние модели современных ПУ имеют большие функциональные возможности, высокую производительность. Они достаточно дороги и применение их в виде локальных будет сопряжено с большими материальными затратами.

Сервис печати NETWARE позволяет сразу нескольким пользователям более эффективно использовать. Например, один лазерный принтер фирмы XEROX, подключенный в сеть даст возможность сэкономить средства, не приобретая другие.

Когда несетевая станция посылает запрос на печать на подключенный к ней принтер, этот запрос сразу же направляется на выполнение. Если пользователь будет работать с сетевыми принтерами, то информация, которую он выводит ПУ, будет направлена сначала в файловый или принт-сервер, а уже потом на принтер. Когда принтер готов выполнять очередной запрос, принт-сервер выбирает задание на печать из очереди и посылает его на принтер, соответствующий данной очереди. Принт-сервер является составной частью программной компоненты файл-сервера, которая выбирает задания на печать из очереди и направляет их в принтер. Принт-сервер может также присутствовать в сети в виде специализированной рабочей станции, которая призвана обслуживать процесс печати в сети или он может быть совмещен с ПО моста. В сети процесс сетевой печати может осуществляться и на принтерах, подключенным к обычным удаленным РС.

Принт-сервер NETWARE увеличивает возможности печати сети, он может обслуживать до 16 принтеров, подключенных к различным компьютерам, включенным в сеть и может быть инсталлирован (инсталляция - установка программного изделия на ПЭВМ) на файл-сервере, мосту или специализированной РС.

ПО принт-сервера обычно совмещено с ПО файлового сервера и использует VAP-процессы, загружаемые на файл-сервере. VAP-процессы принт-сервера используют в процессе работы на файл-сервере или мосту до 128 К памяти, включая DOS при загрузке на мосту. Для каждого притера добавляется еще по 10 К.

При использовании специализированного принт-сервера на РС для его работы требуется 200 К памяти, плюс по 10 К для каждого подключенного принтера. Эти цифры могут меняться в зависимости от загруженности принт-сервера.

Удаленный принтер требует на своей РС 9 К памяти. Эта цифра включает и объем буфера, необходимый для работы принтера. Удаленный принтер будет функционировать при отключенном файловом сервере, если принт-сервер оформлен в виде специализированной РС или он инсталлирован на мосту.

В системе NETWARE процесс печати реализован следующим образом: оболочка РС направляет файл по сети в файловый или принт-сервер, где он, согласно системному планированию, буферируется и ставится в очередь с параметрами задания для печати.

При одновременной посылке информации пользователями на печать, запрос, полученный первым, будет обработан в первую очередь. Все последующие запросы выстраиваются в очередь и будут в такой последовательности обработаны, если только они не получат высший приоритет.

Рабочим заданием на печать служат характеристики, определяющие, как должна производиться печать. К ним относятся: режим, формат, количество копий, а также указание конкретного принтера, который будет выполнять работу. Каждый пользователь создает задание на печать и направляет его в файл или принт-серверу, где оно уже ставится в очередь.

NETWARE версии 2.15 позволяет одному принтеру обслуживать несколько очередей, и одна очередь может обслуживаться несколькими принтерами. Например, при наличии нескольких запросов на печать, принтерам Printer0 и Printer1 может быть дано задание на выполнение очереди с более высоким приоритетом.

Можно также определить, каким пользователям разрешено помещать задания на печать в каждую очередь.



Любая очередь на печать должна быть спланирована с помощью специальных средств. Можно установить соответствие между очередями на печать и принтерами с помощью команд, которые вводятся с консоли файл-сервера, или из подготовленного файлa аutoexec.sys.

## 4.3. Технология клиент-сервер

В настоящее время весьма актуален переход от небольших локальных сетей персональных компьютеров к промышленным корпоративным информационным системам - UPSIZING. Большинство средних и крупных государственных и коммерческих организаций постепенно отказываются от использования только ПК, задачей сегодняшнего дня - создание открытых и распределенных информационных систем.

На сегодняшний день развитие информационных технологий - создание единых сетей предприятий и корпораций, объединяющих удаленные компьютеры и локальные сети, часто использующие разные платформы, в единую информационную систему. Т.е. необходимо объединить пользователей компьютеров в единое информационное пространство и предоставить им совместный доступ к ресурсам. Однако здесь возникает множество трудностей, связанных с решением задачи по организации каналов связи (кабель Ethernet не протянешь по городу, а тем более до другого конца планеты). При построении корпоративных сетей иногда используются телефонные каналы, но связь по таким коммутируемым линиям ненадежна, аренда выделенных линий связи дорога, а эффективность такого канала невысокая. Проблема возникает и при интегрировании в корпоративную сеть разнородных ЛВС, а также в подключении больших компьютеров, например, IBM mainframe или VAX. Сложности возникают и при объединении в одну локальную сеть компьютеров с разными ОС. Поэтому построение корпоративной сети задача не из легких.

Проблема первая - это каналы связи. Самым оптимальным вариантом является использование уже существующих глобальных сетей передачи данных общего пользования, чтобы коммуникационный протокол в корпоративной сети совпадал с принятым в существующих глобальных сетях. Наиболее рациональным выбором здесь следует считать протокол Х.25. Данный протокол позволяет работать даже на низкокачественных линиях связи, так как разрабатывался он для подключения удаленных терминалов к большим ЭВМ и соответственно включает в себя мощные средства коррекции ошибок, освобождая от этой работы пользователя.

Дальнейшее развитие Х.25 - Frame Relay, а также новые протоколы типа АТМ, хотя и обещают значительно большие скорости, требуют практически идеальных линий связи и, возможно, не скоро будут широко применяться в ближайшем будущем. Существующие в нашей стране глобальные сети общего доступа - SprintNet, Infotel, Pochet и прочие - построены на базе Х.25

Протокол Х.25 позволяет организовать в одной линии до 4096 виртуальных каналов связи. Если протянуть к офису одну выделенную линию. то ее можно использовать для объединения нескольких удаленных офисов, подключения корпоративных информационных ресурсов, доступа к системам электронной почты, базам данным - одновременно.

Выделенная линия - это обычная телефонная линия, с которой можно работать на скоростях 9600-28800 бит/с. Более скоростные линии (64 Кбит/с и >) стоят значительно дороже.

Обычно сети Х.25 строятся на двух типах оборудования - Switch или центр коммутации пакетов (ЦКП) и PAD (hfcket assembler/disassembler - сборщик/разработчик пакетов), называемый также пакетным адаптером данных (ПАД), или терминальным концентратором. ПАД служит для подключения к сети Х.25 оконечных устройств через порты. Примером использования ПАД в корпоративной сети - подключение банкоматов к центральному компьютеру банка.

**ЦКП** - его задача состоит в определении маршрута, т.е. в выборе физических линий и виртуальных каналов в них, по которым будет пересылаться информация.

Переход к многопользовательским СУБД - качественно технологический скачок, обеспечивающий деятельность организаций в будущем. Реализация перехода к новой информационной системе (ИС) зависит от используемой и перспективной моделей клиент-сервер.

**Модели клиент-сервер** - это технология взаимодействия компьютеров в сети. Каждый из компьютеров имеет свое назначение и выполняет свою определенную роль. Одни компьютеры в сети владеют и распоряжаются информационно-вычислительными ресурсами (процессоры, файловая система, почтовая служба, служба печати, база данных), другие имеют возможность обращаться к этим службам, пользуясь их услугами.

Компьютер, управляющий тем или иным ресурсом называют ***сервером*** этого ресурса, а компьютер, пользующийся им - ***клиентом***.

Каждый конкретный сервер определяется видом того ресурса, которым он владеет. Например, назначением сервера баз данных является обслуживание запросов клиентов, связанных с обработкой данных; файловый сервер, или *файл-сервер,* распоряжается файловой системой и т.д.

Этот принцип распространяется и на взаимодействие программ. Программа, выполняющая предоставление соответствующего набора услуг, рассматривается в качестве сервера, а программы пользующиеся этими услугами, принято называть *клиентами.* Программы имеют распределенный характер, т.е. одна часть функций прикладной программы реализуется в программе-клиенте, а другая - в программе-сервере, а для их взаимодействия определяется некоторый *протокол.*

Рассмотрим эти функции. Один из основных принципов технологии клиент-сервер заключается в разделении функций стандартного интерактивного приложения на четыре группы, имеющие различную природу.

*Первая группа*. Это функции ввода и отображения данных.

*Вторая группа* - объединяет чисто прикладные функции, характерные для данной предметной области (для банковской системы - открытие счета, перевод денег с одного счета на другой и т.д.).

*Третья группа* - фундаментальные функции хранения и управления информационно-вычислительными ресурсами (базами данных, файловыми системами и т.д.).

*Четвертая группа* - служебные функции, осуществляющие связь между функциями первых трех групп.

В соответствии с этим в любом приложении выделяются следующие логические компоненты:

- компонент представления (presentation), реализующий функции первой группы;

- прикладной компонент (business application), поддерживающий функции второй группы;

- компонент доступа к информационным ресурсам (resource manager), поддерживающий функции третьей группы, а также вводятся и уточняются соглашения о способах их взаимодействия (протокол взаимодействия).

*Различия в реализации технологии клиент-сервер* определяются следующими факторами:

- видами программного обеспечения, в которые интегрирован каждый из этих компонентов;

- механизмами программного обеспечения, используемыми для реализации функций всех трех групп;

- способом распределения логических компонентов между компьютерами в сети;

- механизмами, используемыми для связи компонентов между собой.

Выделяются четыре подхода, реализованные в следующих моделях:

1. модель файлового сервера (File Server - FS);

2. модель доступа к удаленным данным (Remote Data Access - RDA);

3. модель сервера баз данных (Data Base Server - DBS);

4. модель сервера приложений (Application Server - AS).

### 4.3.1. Модель файлового сервера. (FS) -

является базовой для локальных сетей ПК. До недавнего времени была популярна среди отечественных разработчиков, использовавших такие системы, как FoxPro, Clipper, Clarion, Paradox и т.д.

Одним из компьютеров в сети считается файловым сервером и предоставляет другим компьютерам услуги по обработке файлов. Файловый сервер работает под управлением сетевой операционной системы (Novell NetWare) и играет роль компонента доступа к информационным ресурсам (т.е. к файлам). На других ПК в сети функционирует приложение, в кодах которого совмещены компонент представления и прикладной компонент (рис.4.7.).

Клиент Сервер

*Запросы*

Компонент Прикладной Компонент доступа к

представления компонент ресурсам

*файлы*

Рис.4.7. Модель файлового сервера

Протокол обмена представляет собой набор вызовов, обеспечивающих приложению доступ к файловой системе на файл-сервере.

К недостаткам технологии данной модели относят низкий сетевой трафик (передача множества файлов, необходимых приложению), небольшое количество операций манипуляции с данными (файлами), отсутствие адекватных средств безопасности доступа к данным ( защита только на уровне файловой системы) и т.д.

### 4.3.2. Модель доступа к удаленным данным (RDA) –

существенно отличается от FS-модели методом доступа к информационным ресурсам. В RDA-модели коды компонента представления и прикладного компонента совмещены и выполняются на компьютере-клиенте. Доступ к информационным ресурсам обеспечивается операторами специального языка (SQL, если речь идет о базах данных) или вызовами функций специальной библиотеки (если имеется специальный интерфейс прикладного программирования - API).

Запросы к информационным ресурсам направляются по сети удаленному компьютеру, который обрабатывает и выполняет их, возвращая клиенту блоки данных (рис.4.8).

Клиент Сервер

SQL

Компонент Прикладной Компонент доступа к

представления компонент ресурсам

данные

##### Рис.4.8. Модель доступа к удаленным данным

Говоря об архитектуре клиент-сервер, подразумевают данную модель. Основное достоинство RDA-модели заключается в унификации интерфейса клиент-сервер в виде языка SQL и широком выборе средств разработки приложений. К недостаткам можно отнести существенную загрузку сети при взаимодействии клиента и сервера посредством SQL-запросов; невозможность администрирования приложений в RDA, т.к. в одной программе совмещаются различные по своей природе функции (представления и прикладные).

### 4.3.3. Модель сервера баз данных (DBS) -

реализована в некоторых реляционных СУБД (Informix, Ingres, Sybase, Oracle), (рис.4.9).

Ее основу составляет механизм хранимых процедур - средство программирования SQL-сервера. Процедуры хранятся в словаре баз данных, разделяются между несколькими клиентами и выполняются на том же компьютере, где функционирует SQL-сервер. В DBS-модели компонент представления выполняется на компьютере-клиенте, в то время как, прикладной компонент оформлен как набор хранимых процедур и функционирует на компьютере-сервере БД. Там же выполняется компонент доступа к данным, т.е. ядро СУБД.

Клиент *Вызов* Сервер

Компонент Прикладной Компонент доступа к

представления компонент SQL ресурсам

Рис.4.9. Модель сервера баз данных

Понятие информационного ресурса сужено до баз данных, поскольку механизм хранимых процедур - отличительная характеристика DBS-модели - имеется пока только в СУБД.

*Достоинства DBS-модели:*

- возможность централизованного администрирования прикладных функций;

- снижение трафика (вместо SQL-запросов по сети направляются вызовы хранимых процедур);

- возможность разделения процедуры между несколькими приложениями;

- экономия ресурсов компьютера за счет использования единожды созданного плана выполнения процедуры. К недостаткам относится:

- ограниченность средств написания хранимых процедур, представляющих собой разнообразные процедурные расширения SQL, которые уступают по изобразительным средствам и функциональным возможностям в сравнении с языками С или Pascal. Сфера их использована ограничена конкретной СУБД из-за отсутствия возможности отладки и тестирования разнообразных хранимых процедур.

На практике чаще используются смешанные модели, когда целостность базы данных и некоторые простейшие прикладные функции обеспечиваются хранимыми процедурами (DBS-модель), а более сложные функции реализуются непосредственно в прикладной программе, которая выполняется на компьютере-клиенте (RDA-модель).

### 4.3.4. Модель сервера приложений (AS) -

представляет собой процесс, выполняемый на компьютере-клиенте, отвечающий за интерфейс с пользователем (т.е. реализует функции первой группы). (рис.4.10).

Клиент Сервер Сервер

Компонент *API* Прикладной *SQL* Компонент доступа

представления компонент к ресурсам

Рис.4.10. Модель сервера приложений

Прикладной компонент реализован как группа процессов, выполняющих прикладные функции, и называется *сервером приложения* (Application Server - AS).

Доступ к информационным ресурсам осуществляет менеджер ресурсов (например, SQL-сервер). Из прикладных компонентов доступны такие ресурсы как, базы данных, очереди, почтовые службы и др. AS, размещенная на компьютере, где функционирует менеджер ресурсов, избавляет от необходимости направления SQL-запросов по сети, что повышает производительность системы.

Модели RDA и DBS опираются на двухзвенную схему разделения функций:

- в RDA-модели прикладные функции отданы программе-клиенту (прикладной компонент сливается с компонентом представления);

- в DBS-модели ответственность за их выполнение берет на себя ядро СУБД (прикладной компонент интегрируется в компонент доступа к информационным ресурсам).

В AS-модели реализована трехзвенная схема разделения функций. Здесь прикладной компонент выделен как важнейший изолированный элемент приложения. Сравнивая модели, AS обладает наибольшей гибкостью и имеет универсальный характер.

#### *Принципы перехода к новой информационной системе.*

При переходе к новой информационной системе (ИС) необходимо решить такие вопросы как выбор одной из четырех моделей, компоненты архитектуры ИС и инструментарий перехода.

Наиболее распространенной ИС является FS-модель (примем ее за исходную), а в качестве целевой - RDA-модель (наиболее распространена и отно­сительно проста). На практике наблюдаются и другие схемы перехода ( FS-->DBS, RDA--->DBS, RDA-->AS, FS-->AS). Наиболее типичный случай это FS-->RDA, это переход от локальных сетей ПК к архитектуре систем с сервером баз данных.

Следующий шаг - определение компонентов архитектуры системы, имеющей в своей основе RDA-модель - *компьютер-клиент и сервер баз данных*. Проблема заключается в выборе аппаратного и базового программного обеспечения этих компонентов.

На сегодняшний день используются ПК на базе процессоров 486 или Pentium под управлением ОС/2 MS Windows (распространенность, популярность, большое число приложений, широкий набор активно используемых русифицированных продуктов). Самое важное достоинство MS Windows - множество средств быстрой разработки приложений, работающих с SQL-ориентированными СУБД, и доступность этих средств для отечественных пользователей.

Говоря о сервере БД, необходимо упомянуть, что это должен быть мощный компьютер, снабженный высокоскоростными надежными механизмами дисковой памяти большой емкости и системой архивирования на магнитных лентах. Его работа должна осуществляться под управлением многозадачной многопользовательской ОС, поддерживающей промышленные стандарты.

Для RDA-модели характерны два ключевых компонента:

- ПК на базе процессоров 486/Pentium под управлением ОС MS Windows;

- высокопроизводительный RISC-компьютер (фирм Sun, Hewiett-Packard, IBM) под управлением соответствующей версии ОС UNIX.

## 4.4. Технология работы в среде распределенной обработки данных

Одной из важнейших сетевых технологий является распределенная обработка данных, позволяющая повысить эффективность удовлетворения информационной потребности пользователя и, обеспечить гибкость и оперативность принимаемых им решений.

Достоинствами распределенной обработки информации является:

- большое число взаимодействующих между собой пользователей;

- устранение пиковых нагрузок с централизованной базы данных за счет распределения обработки и хранения локальных баз данных на разных ЭВМ;

- возможность доступа пользователя к вычислительным ресурсам сети ЭВМ;

- обеспечение обмена данными между удаленными пользователями.

При распределенной обработке производится работа с базой, т.е. представление данных, их обработка, работа с базой на логическом уровне осуществляется на компьютере клиента, а поддержание базы в актуальном состоянии - на сервере. При наличии распределенной базы данных база размещается на нескольких серверах. В настоящее время созданы базы данных по всем направлениям человеческой деятельности: экономической, финансовой, кредитной, статистической, научно-технической, маркетинга, патентной информации, электронной документации и т.д.

Создание распределенных баз данных (РБД) было вызвано двумя тенденциями обработки данных, с одной стороны - интеграцией, а с другой - децентрализацией.

Интеграция подразумевает централизованное управление и ведение баз данных. Децентрализация обеспечивает хранение данных в местах их возникновения или обработки, при этом скорость обработки повышается, стоимость снижается, увеличивается степень надежности системы.

Распределенная база данных - база данных, части которой размещены на отдельных ЭВМ, входящих в сеть. При этом некоторые данные могут дублироваться.

При проектировании РБД осуществляется разбиение объекта на несколько частей (фрагментов) и размещение каждого фрагмента на один или несколько компьютеров. Размещение фрагментов может быть *избыточным* или *безызбыточным.*

При избыточном размещении необходимо определить степень дублирования фрагментов. Выгоды, получаемые от дублирования, пропорциональны соотношению объемов выборки данных и их обновления. Для поддержания целостности базы данных требуется корректировка всех копий. Преимущества дублирования уменьшаются с увеличением стоимости хранения фрагментов и, увеличиваются, так как повышается устойчивость системы против отказов. Эффективность работы пользователей с РБД зависит от обеспеченности их информацией о содержащихся в РБД данных, их структуре и размещении. Эту задачу решает сетевой словарь-справочник данных, находящийся в одной ЭВМ сети или дублирующийся на нескольких ЭВМ. При этом, словарь-справочник может иметь распределенную структуру, т.е. когда его отдельные фрагменты распределены по рабочим станциям сети.

К организации баз данных предъявляются такие общие требования как, обеспечение высокой скоростью обработки запросов, секретности, независимости (физической и логической) данных, безопасности и т.д. Кроме перечисленных требований, к РБД выдвигаются требования "прозрачности": распределенной структуры БД; совместного доступа к данным; распределенной обработки.

Распределенная структура БД предполагает независимость конечных пользователей и программ от способа размещения информации на рабочих станциях сети, т.е. формулирование запросов к РБД производится аналогично запросам к централизованной БД.

Совместный доступ к данным подразумевает модификацию одних и тех же данных несколькими пользователями не нарушая целостности РБД.

"Прозрачность" распределенной обработки означает независимость пользователей и программ от типа локальной вычислительной сети и применяемого сетевого программного обеспечения. Обработка запроса пользователя может производиться на нескольких ЭВМ.

Доступ пользователей к РБД и администрирование осуществляется с помощью системы управления распределенной базой данных (СУРБД), которая обеспечивает выполнение следующих функций:

- автоматическое определение ЭВМ, хранящей требуемые в запросе данные;

- декомпозицию распределенных запросов на частные подзапросы к БД отдельных ЭВМ;

- планирование обработки запросов;

- передачу частных подзапросов и их исполнение на удаленных ЭВМ;

- прием результатов выполнения частных подзапросов;

- поддержание в согласованном состоянии копий дублированных данных на различных ЭВМ сети;

- управление параллельным доступом пользователей к РБД;

* обеспечение целостности РБД.

## 4.5. Базовые технологии обработки запросов в архитектурах

## файл-сервера и клиент-сервера

Прикладные программы управления данными представляют собой необходимый инструмент для распределенной обработки.

Архитектура клиент-сервера сети позволяет различным прикладным программам одновременно использовать общую базу данных. Совершенно очевидно, что перенос программ управления данными с рабочих станций на сервер способствует высвобождению ресурсов рабочих станций, предоставляет возможность увеличить число частных, локально решаемых задач. Данная архитектура позволяет также централизовать ряд самых важных функций управления данными, такие, как защита информации баз данных, обеспечение целостности данных, управление совместным использованием ресурсов.

Одним из важных преимуществ архитектуры клиент-сервера в сетевой обработке данных является возможность сокращения времени реализации запроса. В подтверждение этому рассмотрим две *базовые* технологии обработки информации в архитектуре клиент-сервера сети и технологии использования традиционного файлового сервера.

Допустим, что прикладная программа базы данных загружена на рабочую станцию и пользователю необходимо получить все записи, удовлетворяющие некоторым поисковым условиям. В среде традиционного файлового сервера программа управления данными, которая выполняется на рабочей станции, должна осуществить запрос к серверу каждой записи базы данных (рис.4.11,а). Программа управления данными на рабочей станции может определить, удовлетворяет ли запись поисковым условиям, лишь после того, как она будет передана на рабочую станцию.

Очевидно, что данный технологический вариант обработки информации имеет наибольшее суммарное время передачи данных по каналам сети.

В среде клиент-сервера, напротив, рабочая станция посылает запрос высокого уровня серверу базы данных. Сервер базы данных осуществляет поиск записей на диске и анализирует их. Записи, удовлетворяющие условиям, могут быть накоплены на сервере. После того, как запрос целиком обработан, пользователю на рабочую станцию передаются все записи, которые удовлетворяют поисковым условиям (рис. 4.11,б).

Данная технология позволяет снизить сетевой трафик и повысить пропускную способность сети. Более того, за счет выполнения операции доступа к диску и обработки данных в одной системе сервер может осуществить поиск и обрабатывать запросы быстрее, чем если бы эти запросы обрабатывались на рабочей станции.

Прикладные программы баз данных клиент-сервера поддерживаются программными продуктами:

- NetWare Btrieve- программой управления записями с индексацией по ключу (выполняется на сервере);

- NetWare SQL - ядром реляционных баз данных, предназначенным для обеспечения системы защиты и целостности данных.

Службы баз данных NetWare Btrieve и NetWare SQL фирмы Novell позволяют разработчикам создавать надежные прикладные программы баз данных без необходимости написания собственных программ управления записями, что обеспечивает удобный перенос прикладных программ в среду клиент-сервера.

В настоящее время разработаны десятки тысяч прикладных автономных и многозадачных программ, ориентированных на клиента версий NetWare Btrieve, NetWare SQL, которые могут быть использованы организациями, создающими или имеющими сеть ЭВМ. Более того, версии NetWare Btrieve и NetWare SQL фирмы Novell для клиентов имеют согласованные API, что упрощает перенос программ из среды одного клиента в среду другого.



|  |  |
| --- | --- |
| Файл-сервер | Рабочая станция |

1. Типовая среда обработки запросов в сетях ЭВМ.



б) Распределенная среда обработки запросов в сетях ЭВМ.

Рис. 4.11. (а,б).Технологии обработки запросов по базовым вариантам

По степени изменчивости все базы данных (БД) можно разделить на два класса:

А - условно-постоянные (в основном для справочных систем);

Б - сильно динамичные (для банковских, биржевых систем и т.п.).

Для ведения баз данных первого и второго классов используются системы управления базами данных (СУБД), которые в значительной степени отличаются друг от друга как по функциональным возможностям, так и по эксплуатационным характеристикам.

*Например,*

- *для условно-постоянных БД* наиболее важными показателями являются показатели скорости отработки запросов и скорости формирования выходных отчетов по БД, а такие показатели, как скорость отработки транзакций и контроль целостности БД при отработке транзакций не столь критичны;

- *для сильно -динамичных БД*, на первый план выходят такие показатели, как скорость отработки транзакций, возможность контроля целостности, скорость формирования отчетов, согласованность по чтению и транзакциям. Менее критичны здесь скорости отработки запросов.

Поэтому любая СУБД не может одинаково успешно применяться при работе с БД разных классов. Такие системы, как CLIPPER, FOXPRO ориентированы на первый класс БД-(А), и здесь имеются неплохие результаты, а такие СУБД ,как Informix, Ingres, SyBase создавались для второго класса-(Б).

Исходя из вышесказанного напрашивается вывод: найти "золотую середину", которая удовлетворяла бы требованиям обоих классов (А) и (Б). Решением этой противоречивой задачи является использование дифференциальной организации файлов базы данных, или *дифференциальных файлов (ДФ).*

В последнее время разработчики СУБД ведущих фирм подошли к использованию идеи ДФ. Причинами явились следующие факты:

- значительно расширился класс решаемых на IBM PC задач, так, что термин "персональный компьютер" уже не соответствует действительности;

- широкое распространение локальных вычислительных систем (ЛВС);

- разработка многопользовательских и многозадачных систем;

- стремительное развитие технической базы ЭВМ (в большей степени дисковой памяти).

Остановимся на сути ДФ применительно к БД в ЛВС. Реализация идеи ЛВС в различных СУБД значительно отличается.

Идея ДФ включает три положения:

- основной файл БД остается неизменным при любых обновлениях базы данных, т.е., любые изменения БД последовательно накапливаются в специальном файле изменений (не путать с журналом транзакций) - ***ДФ***;

- никакие индексы для него не создаются и не поддерживаются.

Когда ДФ достигнет значительных размеров (примерно 25-40% от размеров БД), администратор вносит все изменения в основной файл БД в удобный момент времени в пакетном режиме.

В качестве примера возьмем сравнение книги, где наблюдаются опечатки в страницах и, базы данных с ДФ. Нет необходимости переиздания книги из-за нескольких опечаток или незначительных изменений. Если это количество имеет тенденцию к значительному росту и достигло предельного значения, то становятся оправданными затраты на переиздание книги, куда должны быть включены все накопленные изменения.

*Достоинства ДФ относятся к обеспечению высокой надежности, целостности БД и скорости отработки транзакций.*

Вопрос, какие скорости отработки транзакций можно обеспечить при использовании ДФ, является довольно важным. Очевидно, что скорость отработки транзакций при такой организации БД возрастет в десятки раз. При этом сервер базы данных практически напоминает обычный файл-сервер.

Что касается индексов, то проблемы их поддержания не существует (скорости добавления, удаления и модификации записей БД находятся на самом высоком уровне). Внесение добавлений в БД не отличается от добавлений в обычный последовательный файл. Время обновления записей БД не зависит ни от размеров БД, ни от длины ключей, ни от их числа. Временные затраты на блокировку (как одно из узких мест для БД и ЛВС) сведены к минимуму.

Для того, чтобы обеспечить согласованность данных по чтению нет необходимости блокировать целиком таблицу, что имеет место в ряде СУБД, т.е., когда запрос (или формируемый отчет) начинает выполняться, СУБД "запоминает" старший адрес в ДФ (моментальный снимок). При этом пользователь, инициализирующий свой запрос, не обязан ждать "своего момента". Он "не видит" никого из пользователей и получает снимок БД именно в этот момент времени. Далее, по мере выполнения запроса (даже очень быстрого) часть записей-целей могла быть или изменена или удалена. Это отразится только на старших адресах ДФ, а поэтому СУБД просто проигнорирует любые изменения данных, случившиеся после начала выполнения запроса. Гарантируется корректировка сложных и длительных запросов к БД, т.е. обеспечение согласованности по чтению и транзакциям.

Становится интересным вопрос, как в этом случае ведется поиск в БД. В этом случае по ассоциатору находится множество записей-целей: число и список их адресов в основной БД, после чего производится считывание "ассоциатора" ДФ и производится корректировка этого списка. За счет этой корректировки время поиска увеличивается, причем величина этого увеличения зависит от размеров ДФ. Своевременность обновления БД должна быть в компетенции администратора БД. Чтобы исключить существенные издержки, связанные с ДФ, можно накапливать изменения БД для их пакетной обработки и при поиске ДФ не учитывать. В ряде систем, таких как банковские, допускается потеря некоторой точности в период между циклами обновления - "контролируемое запаздывание".

Помимо всего прочего использование ДФ обеспечивает:

- возможность администратору восстанавливать случайно удаленные записи;

- возможность (при необходимости) хранить индексные файлы на самих рабочих станциях;

- возможность создания распределенных БД;

- одновременное выполнение транзакций.

Непротиворечивость данных может обеспечиваться механизмом захвата на уровне записи - откат транзакций любой доступной вложенности.