

Модели архитектуры «клиент-сервер»

«*Клиент-сервер*» — модель взаимодействия компьютеров в сети. Как правило, компьютеры не являются равноправными. Некоторые компьютеры в сети (серверы) владеют и распоряжаются информационно-вычислительными ресурсами, такими как процессоры, файловая система, почтовая служба, служба печати, база данных. Другие же компьютеры (клиенты) имеют возможность обращаться к этим службам, пользуясь услугами первых.

Конкретный сервер определяется видом ресурса, которым он владеет:

- если ресурсом являются базы данных, то речь идет о сервере баз данных, назначение которого — обслуживать запросы клиентов, связанные с обработкой данных;
- если ресурс — это файловая система, то говорят о файловом сервере, или файл-сервере, и т. д.

В сети один и тот же компьютер может выполнять роль как клиента, так и сервера. Например, в информационной системе, включающей персональные компьютеры, большую ЭВМ и мини-компьютер под управлением UNIX, последний может выступать как в качестве сервера базы данных, обслуживая запросы от клиентов — персональных компьютеров, так и в качестве клиента, направляя запросы большой ЭВМ.

Этот же принцип распространяется и на взаимодействие программ.

Если одна из них выполняет некоторые функции, предоставляя другим соответствующий набор услуг, то такая программа выступает в качестве сервера. Программы, которые пользуются этими услугами, принято называть клиентами. Так, ядро реляционной SQL-ориентированной СУБД часто называют сервером базы данных, или SQL-сервером, а программу, обращающуюся к нему за услугами по обработке данных — SQL-клиентом.

Как поддерживающая интерфейс с пользователем программа, СУБД, в общем случае, реализует следующие основные функции:

- **управление** данными, находящимися в базе;
- **обработка** информации с помощью прикладных программ;
- **представление** информации в удобном для пользователя виде.

Если система размещается на одной ЭВМ, то все функции собраны в одной программе. При размещении СУБД в сети возможны различные варианты распределения функций по узлам. В зависимости от числа узлов сети, между которыми выполняется распределение функций СУБД, можно выделить двухзвенные модели, трехзвенные модели и т.д. Место разрыва функций соединяется коммуникационными функциями (средой передачи информации в сети).

Двухзвенные модели распределения функций

Двухзвенные модели соответствуют распределению функций СУБД между двумя узлами сети. Компьютер (узел сети), на котором обязательно присутствует функция управления данными, назовем компьютером-сервером. Компьютер, близкий к пользователю и обязательно занимающийся вопросами представления информации, назовем компьютером-клиентом.

Наиболее типичными вариантами разделения функций между компьютером-сервером и компьютером-клиентом являются следующие:

- распределенное представление;
- удаленное представление;
- удаленный доступ к данным;
- распределенная БД.

Спектр моделей архитектуры клиент-сервер

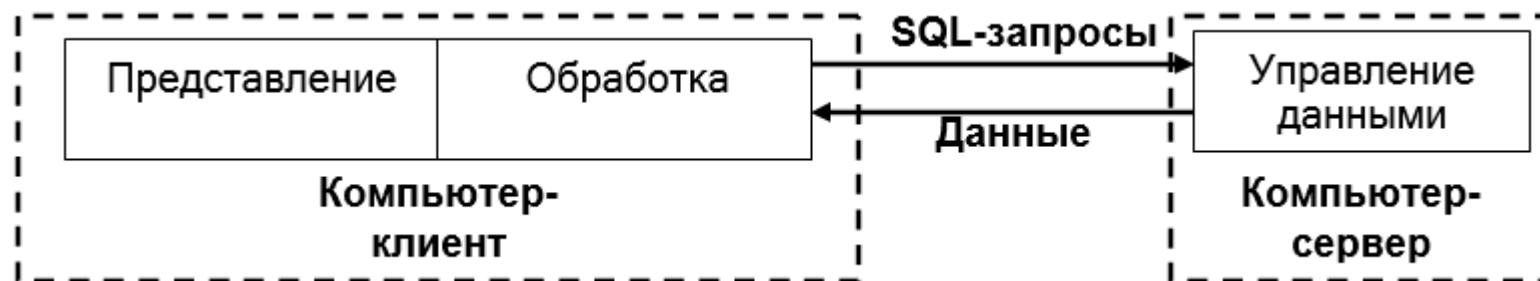


- распределенное представление;
- удаленное представление;
- удаленный доступ к данным;
- распределенная БД.

В моделях удаленного доступа к данным и удаленного представления производится строгое распределение функций между компьютером-клиентом и компьютером-сервером. В других моделях имеет место выполнение одной из двух функций одновременно на двух компьютерах: управления данными (модель распределенной БД) или представления информации (модель распределенного представления).

Модель удаленного доступа к данным (Remote Data Access — RDA):

- программы, реализующие функции представления информации и логику прикладной обработки, совмещены и выполняются на компьютере-клиенте,
- обращение за сервисом управления данными происходит через среду передачи с помощью операторов языка SQL или вызовом функций специальной библиотеки API (Application Programming Interface — интерфейса прикладного программирования) как показано на рисунке.



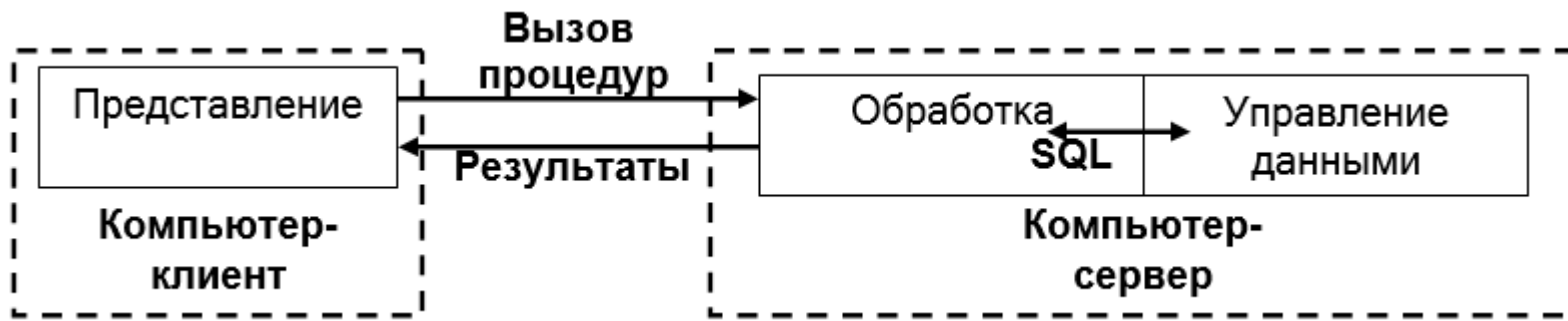
Основное достоинство RDA-модели: существует множество готовых СУБД, имеющих SQL-интерфейсы, и инструментальных средств, обеспечивающих быстрое создание программ клиентской части. Средства разработки чаще всего поддерживают графический интерфейс пользователя в MS Windows, стандарт интерфейса ODBC и средства автоматической генерации кода.

Недостатки RDA-модели:

- Довольно высокая загрузка системы передачи данных вследствие того, что вся логика сосредоточена в приложении, а обрабатываемые данные расположены на удаленном узле.
- Системы, построенные на основе модели RDA, неудобны с точки зрения разработки, модификации и сопровождения. Основная причина состоит в том, что в получаемых приложениях прикладные функции и функции представления тесно взаимосвязаны. Поэтому даже при незначительном изменении функций системы требуется переделка всей прикладной ее части, усложняющая разработку и модификацию системы.

Модель удаленного представления (сервера БД):

- (DataBase Server — DBS) отличается от предыдущей модели тем, что функции компьютера-клиента ограничиваются функциями представления информации, в то время как прикладные функции обеспечиваются приложением, находящимся на компьютере-сервере.



Перед сервером БД стоят следующие задачи:

1. Необходимо, чтобы БД в каждый момент времени отражала текущее состояние предметной области, которое определяется не только собственными данными, но и связями между объектами данных. То есть данные, которые хранятся в БД, в каждый момент времени должны быть непротиворечивыми.

2. БД должна отражать некоторые правила предметной области, законы, по которым она функционирует.

3. Необходим постоянный контроль за состоянием БД, отслеживание всех изменений и адекватная реакция на них.

4. Необходимо, чтобы возникновение некоторой ситуации в БД четко и оперативно влияло на ход выполнения прикладной задачи.

5. Одна из важных проблем СУБД — контроль типов данных. В настоящий момент СУБД может проверять только стандартно-допустимые типы данных, т.е. такие, которые определены в DDL (Data Definition Language) — языке описания данных, являющемся частью SQL. Однако в реальных предметных областях могут потребоваться нестандартные типы данных (плоскостные и пространственные координаты, единицы различных метрик, пятидневная неделя, т.е. рабочая неделя, в которой сразу после пятницы следует понедельник, дроби, не говоря уже о графических изображениях).

Централизованный контроль в модели сервера баз данных выполняется с использованием механизма триггеров. Триггеры являются частью БД и представляют собой некоторый тумблер, который срабатывает при возникновении определенного события в БД. СУБД проводит мониторинг всех событий, которые вызывают созданные и описанные триггеры в БД, и при возникновении соответствующего события сервер запускает соответствующий триггер. Триггеры могут вызывать хранимые процедуры.

Механизм использования триггеров предполагает, что при срабатывании одного триггера могут возникнуть события, которые вызовут срабатывание других триггеров. Этот мощный инструмент требует тонкого и согласованного применения, чтобы не получился бесконечный цикл срабатывания триггеров.

И хранимые процедуры, и триггеры хранятся в словаре БД, они могут быть использованы несколькими клиентами, что существенно уменьшает дублирование алгоритмов обработки данных в разных клиентских приложениях.

Достоинства DBS-модели:

- возможность хорошего централизованного администрирования приложений на этапах разработки, сопровождения и модификации,
- эффективное использование вычислительных и коммуникационных ресурсов.

Недостатки DBS-модели:

- сильная привязка операторов хранимых процедур к конкретной СУБД,
- низкая эффективность использования вычислительных ресурсов ЭВМ, поскольку не удастся организовать управление входным потоком запросов к программам компьютера-сервера, а также обеспечить перемещение процедур на другие компьютеры-серверы.

Модель распределенного представления:

В данной модели имеется мощный компьютер- сервер, а клиентская часть системы практически вырождена. Функцией клиентской части является просто отображение информации на экране монитора и связь с основным компьютером через локальную сеть.

СУБД подобного рода могут иметь место в сетях, поддерживающих работу так называемых X-терминалов. В них основной компьютер (хост-машина) должен иметь достаточную мощность, чтобы обслуживать несколько X-терминалов. X-терминал тоже должен обладать достаточно быстрым процессором и иметь достаточный объем оперативной памяти (дисковые накопители отсутствуют).

Все программное обеспечение находится на хост-машине. Программное обеспечение X-терминала, выполняющее функции управления представлением и сетевые функции, загружается по сети с сервера при включении X-терминала.

Модель распределенного представления реализует централизованную схему управления вычислительными ресурсами, отсюда следуют ее основные достоинства.

Достоинства модели распределенного представления:

- простота обслуживания и управления доступом к системе,
- относительная дешевизна (ввиду невысокой стоимости оконечных терминалов).

Недостатки модели распределенного представления:

- уязвимость системы при невысокой надежности центрального узла,
- высокие требования к серверу по производительности при большом числе клиентов.

Модель распределенной БД:

Данная модель предполагает использование мощного компьютера-клиента, причем данные хранятся на компьютере-клиенте и на компьютере-сервере. Взаимосвязь обеих баз данных может быть двух разновидностей:

- а) в локальной и удаленной базах хранятся отдельные части единой БД;
- б) локальная и удаленная БД являются синхронизируемыми друг с другом копиями.

Достоинства модели распределенной БД:

- гибкость создаваемых на ее основе ИС, позволяющих компьютеру-клиенту обрабатывать локальные и удаленные БД;
- при наличии механизмов координации соответствия копий система в целом, кроме того, обладает высокой живучестью, так как разрыв соединения клиента и сервера не приводит к краху системы, а ее работа может быть восстановлена с возобновлением соединения.

Недостатки модели распределенного представления:

- высокие затраты при выполнении большого числа одинаковых приложений на компьютерах-клиентах

Трехзвенная модель распределения функций

Трехзвенная модель распределения функций представляет собой типовой вариант, при котором каждая из трех функций приложения реализуется на отдельном компьютере. Варианты распределения функций приложения на большее число компьютеров могут также иметь место.

Рассматриваемая нами модель имеет название модель сервера приложений, или AS-модель (Application Server) и показана на рисунке.



Достоинство AS-модели:

- гибкость и универсальность вследствие разделения функций приложения на три независимые составляющие. Во многих случаях эта модель оказывается более эффективной по сравнению с двухзвенными.

Недостаток AS-модели:

- более высокие затраты ресурсов компьютеров на обмен информацией между компонентами приложения по сравнению с двухзвенными моделями

Стандарт ODBC

При построении информационных систем типа клиент-сервер возникает проблема доступа со стороны СУБД или приложений, разработанных в одной среде, к данным, порожденным другой СУБД. В среде Windows эта проблема решается с помощью стандартного интерфейса ODBC (Open Database Connectivity — совместимость открытых баз данных) фирмы Microsoft. Основное его назначение заключается в обеспечении унифицированного доступа к локальным и удаленным БД различных производителей.

Интерфейс ODBC обеспечивает взаимную совместимость серверных и клиентских компонентов доступа к данным. Для реализации унифицированного доступа к различным СУБД, было введено понятие драйвера ODBC (представляющего собой динамически загружаемую библиотеку).

При построении ODBC-архитектура содержит четыре компонента:

- приложение;
- менеджер драйверов;
- драйверы;
- источники данных.

Роли среди них распределены следующим образом. Приложение вызывает функции ODBC для выполнения SQL-инструкций, получает и интерпретирует результаты; менеджер драйверов загружает ODBC-драйверы, когда этого требует приложение; ODBC-драйверы обрабатывают вызовы функций ODBC, передают операторы SQL СУБД и возвращают результат в приложение; источник данных (data source) — объект, скрывающий СУБД, детали сетевого интерфейса, расположение и полное имя базы данных и т.д.