Модель «клиент-сервер»

**“Клиент-сервер”** – это модель взаимодействия компьютеров и программ  в сети. Некоторые компьютеры в сети владеют и распоряжаются информационно-вычислительными ресурсами, такими, как процессоры, файловая система, почтовая служба, служба печати, база данных. Другие компьютеры имеют возможность обращаться к этим ресурсам. Компьютер, управляющий тем или иным ресурсом, принято называть сервером ресурса, а компьютер, желающий им воспользоваться – клиентом. Конкретный сервер определяется видом ресурса, которым он владеет. Так, если ресурсом является база данных, то речь идет о сервере баз данных; если ресурс – 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.

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.

Первоначально СУБД имели централизованную архитектуру. В ней  СУБД функционировала на центральном компьютере (большая ЭВМ или мини-компьютер). Там же располагалась база данных. К центральному компьютеру были подключены терминалы, выступавшие в качестве рабочих мест пользователей. Все процессы, связанные с обработкой данных, а именно: поддержка ввода пользователя, формирование, оптимизация и выполнение запросов, обмен с устройствами внешней памяти и т. д., выполнялись на центральном компьютере, что предъявляло жесткие требования к его производительности.

В современной СУБД, имеющей модель «клиент-сервер», прикладные программы носят распределенный характер. Часть функций прикладной программы   ( приложения )  реализуется  в  программе-клиенте,  другая – в

программе-сервере, причем для их взаимодействия определяется некоторый протокол.

В модели «клиент-сервер» функции стандартного интерактивного приложения разделяются на четыре группы. Первая группа – это функции ввода и отображения данных. Вторая группа объединяет чисто прикладные функции, характерные для данной предметной области (например, для банковской системы – открытие счета, перевод денег с одного счета на другой и

т. д.). К третьей группе относятся функции доступа к  информационным ресурсам (базам данных, файловым системам и т. д.). Наконец, функции четвертой группы – это служебные функции, играющие роль связок между функциями первых трех групп.

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

– компонент представления, реализующий ввод и отображение данных;

– прикладной компонент, поддерживающий чисто прикладные функции;

– компонент доступа к информационным ресурсам.

Кроме того вводится соглашение о способах их взаимодействия (протокол взаимодействия).

Различия в реализациях модели "клиент-сервер" определяются четырьмя факторами. Во-первых, тем, в какие виды программного обеспечения интегрирован каждый из этих компонентов. Во-вторых, тем, какие механизмы программного обеспечения используются для реализации функций всех трех групп. В-третьих, как логические компоненты распределяются между компьютерами в сети. В-четвертых, какие механизмы используются для связи компонентов между собой.

Имеются четыре варианта воплощения модели "клиент-сервер":

– модель файлового сервера (File Server – FS);

– модель доступа к удаленным данным (Remote Data Access – RDA);

– модель сервера базы данных (DataBase Server – DBS);

– модель сервера приложений (Application Server – AS).

**FS-модель** является базовой для локальных сетей персональных компьютеров. Недавно она была исключительно популярной среди  разработчиков, использовавших такие системы, как FoxPRO, Clipper, Paradox. Суть модели в следующем. Один из компьютеров в сети считается файловым сервером и предоставляет услуги по обработке файлов другим компьютерам.  Файловый  сервер  работает  под  управлением  сетевой  операционной системы  и  играет  роль  компонента  доступа к информационным ресурсам (т. е. к файлам). На всех остальных  компьютерах функционирует приложение, в кодах которого совмещены компонент представления и прикладной компонент. Протокол обмена представляет собой набор низкоуровневых вызовов, обеспечивающих приложению доступ к файловой системе на файл-сервере.

FS-модель послужила фундаментом для расширения возможностей персональных СУБД в направлении поддержки многопользовательского режима. В таких системах на каждом персональном компьютере выполняются копии приложения и ядра СУБД, а база данных содержится в разделяемых файлах, которые находятся на файловом сервере. Когда прикладная программа обращается к базе данных, СУБД направляет запрос на файловый сервер. В этом запросе указаны файлы, содержащие запрашиваемые данные. В ответ на запрос файловый сервер направляет по сети требуемый блок данных. Система управления, получив его, выполняет над данными действия, которые были декларированы в прикладной программе.

Размещение компонента представления и прикладного компонента на компьютерах-клиентах существенно уменьшает общее число процессов операционной системы. К технологическим недостаткам модели относят высокий сетевой трафик (передача множества файлов, необходимых приложению), отсутствие адекватных средств безопасности доступа к данным (защита только на уровне файловой системы).

Более технологичная **RDA-модель** отличается от FS-модели  компонентом доступа к информационным ресурсам. В RDA-модели коды компонента представления и прикладного компонента также совмещены и также выполняются на компьютере-клиенте, однако для доступа к информационным ресурсам используются либо операторы специального языка (например, языка запросов SQL, описываемого в разд. 4), либо вызовы функций специальной библиотеки. Клиент направляет по сети запрос на доступ к базе данных удаленному компьютеру. На этом компьютере функционирует ядро СУБД, которое обрабатывает запросы, выполняя предписанные в них действия, и возвращает клиенту результат, оформленный как блок данных. Сервер базы данных выполняет операции обработки данных,  отрабатывает запросы и транзакции, что резко уменьшает загрузку сети, так как по ней передаются от клиента к серверу не файлы, а запросы на языке высокого уровня (обычно, языке SQL) с существенно меньшим объемом. Важное достоинство RDA-модели – унификация интерфейса "клиент-сервер" в виде языка SQL, уже используемого как язык запросов в СУБД.

Наряду с RDA-моделью все большую популярность приобретает считающаяся перспективной **DBS-модель**. Она реализована в современных реляционных СУБД Informix, Ingres, Sybase, Oracle. Ее основу составляет механизм хранимых процедур как средство программирования SQL-сервера. Процедуры хранятся в словаре базы данных на SQL-сервере и разделяются между несколькими клиентами. Язык, на котором разрабатываются хранимые процедуры, представляет собой процедурное расширение языка запросов SQL и уникален для каждой конкретной СУБД.

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. Достоинства DBS-модели заключены в возможности централизованного администрирования прикладных функций, в некотором снижении трафика (вместо SQL-запросов по сети направляются вызовы хранимых процедур), в возможности разделения процедуры между несколькими приложениями и в экономии ресурсов компьютера за счет использования единожды созданного плана выполнения процедуры. К недостаткам модели можно отнести ограниченность средств, используемых для написания хранимых процедур, которые представляют собой разнообразные процедурные расширения SQL, заметно уступающие по изобразительным средствам и функциональным возможностям таким языкам программирования, как Си или Паскаль. Сфера их использования ограничена конкретной СУБД; в большинстве СУБД отсутствуют возможности отладки и тестирования разработанных хранимых процедур.

В современных многопользовательских СУБД используются также смешанные модели, в которых поддержка целостности базы данных и некоторые простейшие прикладные функции осуществляются хранимыми процедурами (DBS-модель), а более сложные функции реализуются непосредственно в прикладной программе, которая выполняется на компьютере-клиенте (RDA-модель).

В **AS-модели** процесс, выполняющийся на компьютере-клиенте, отвечает за интерфейс с пользователем, т. е. является компонентом представления. Обращаясь за выполнением услуг к прикладному компоненту, этот процесс играет роль клиента приложения (Application Client – AC). Прикладной компонент реализован как группа процессов, выполняющих прикладные функции, и называется сервером приложения (Application Server – AS). Все операции над информационными ресурсами выполняются соответствующим компонентом, по отношению к которому AS играет роль клиента.

Из прикладных компонентов доступны ресурсы различных типов – базы данных, очереди, почтовые службы и др.

RDA- и DBS-модели опираются на двухзвенную схему разделения функций. В RDA-модели прикладные функции приданы программе-клиенту, в DBS-модели ответственность за их выполнение берет на себя ядро СУБД на сервере. В первом случае прикладной компонент сливается с компонентом представления, во втором – интегрируется в компонент доступа к информационным ресурсам.

В AS-модели реализована трехзвенная схема разделения функций, причем прикладной компонент выделен как важнейший изолированный элемент приложения  и стандартизованы его интерфейсы с двумя другими компонентами. AS-модель является фундаментом для мониторов обработки транзакций.