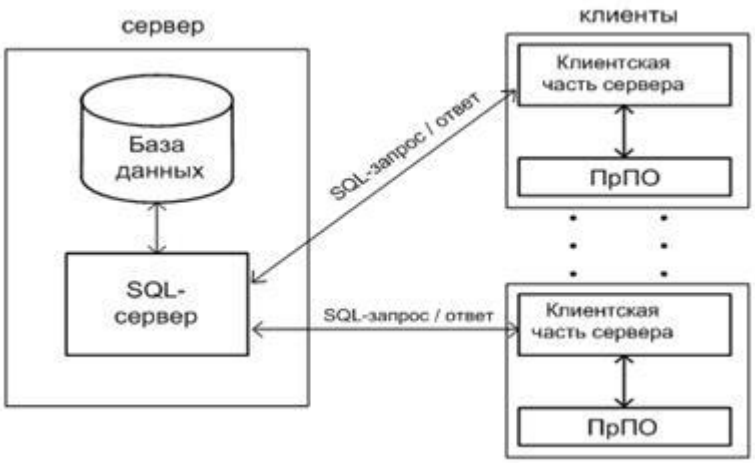
Система работает по следующему принципу:

1. Клиент отправляет запрос серверной машине.
2. Сервер принимает обращение с требованием выполнить определенное действие и выполняет поставленную задачу.
3. Программно-аппаратный комплекс отправляет клиенту результат выполненной работы, обработанного запроса. Модель клиент-сервер предоставляет возможность разграничить поставленные задачи и работу над вычислениями между теми, кто заказывает услуги и теми, кто их поставляет.

Основные компоненты системы:

· **клиент.** Рабочая станция считается входной точкой конечного пользователя в данной системе. Отправляет запросы, получает ответы;
· **сервер.** Взаимодействует с многочисленными клиентами и решает поставленные ими задачи;
· **сеть.** Здесь происходит передача данных. Посредством сети можно соединить рабочие машины общими ресурсами;
· **приложения.** Могут обрабатывать информацию, организовывать физическое распределение данных между сервером и клиентом. Программным обеспечением оснащают серверные устройства для сбора данных, работы с ними и хранения. А также ПО устанавливают на компьютерной станции-клиенте.



О технологии клиент-сервер

Серверное устройство поддерживает многопользовательский режим и обеспечивает одновременно работу с несколькими клиентами. Конечно, машина не может решать в прямом смысле слова одновременно несколько поставленных задач, она выстраивает запросы в очередь по мере поступления, обрабатывает обращения и отправляет результаты работы. Запросы можно выстраивать в списке по приоритетности. Чем важнее запрос, тем быстрей его обрабатывают, даже, если он поступил позже.
Рядовые пользователи сети интернет даже не догадываются о том, как их запросы моментально обслуживаются, чтобы они читали новости, книги, тематические статьи, смотрели интересные видео и фильмы, ходили по форумам, «зависали» в социальных сетях, оплачивали счета, общались с друзьями, оформляли заказы на покупку товаров и т.д. Главное, что ответная реакция быстрая.
Именно технология клиент сервер предоставляет возможность реализовать вышеуказанные многочисленные поставленные задачи. Обычно клиент – это браузер конкретного пользователя. А серверами зачастую выступают:

·любые серверы http;
·наборы серверных машин (например, Denwer);
·локальный веб-сервер.

Обмен информацией между клиентом и сервером происходит благодаря сетевым протоколам в интернете. Каждой услуге соответствует определенный протокол, их предостаточно. Запросы, отсылаемые клиентом, классифицируют как http сообщения. Здесь четко указано, какие сведения нужно предоставить, в каком оформлении. Серверное устройство после анализа и обработки запроса, обычно отвечает html документом – дает свой http ответ. Сообщение от клиента поступает с дополнительными данными, чтобы серверу было понятно, как с ним работать. Ответ машины также отправляется с кодами помимо полезных запрашиваемых данных, чтобы браузер оценил понятливость аппаратно-программного комплекса при обработке его запроса. Смотря на каком уровне осуществляется взаимосвязь клиента с сервером, отсылаемые сообщения браузером упаковываются по-разному. Как будто они оборачиваются клиентом в несколько

слоем обертки. После того, как послание поступило серверной станции, она приступает к разворачиванию всех этих слоев, проводит анализ информации и сбор данных. Говоря больше о технологии клиент-сервер, следует уточнить, что браузер первый выходит на контакт и делает запрос серверной машине, которая лишь предоставляет услуги в ответ на сообщения и указывает, какие условия нужно при этом соблюдать. Разные компьютерные устройства используют, чтобы установить программное обеспечение клиента и серверного оборудования. Но есть случаи, когда они работали на одном ПК.

Когда на одном сайте одновременно находятся несколько посетителей, к серверу в один момент обращается много клиентов. Однако одномоментное поступление запросов ограничено мощностью и возможностями серверных устройств, а также характером отправляемых сообщений.

Архитектура клиент-сервер

Благодаря архитектуре клиент и сервер определены позиции взаимной связи между компьютерными машинами лишь в целом. Что же касается нюансов взаимодействия, они определены протоколами. Технология вполне прозрачно намекает на разделение в сети рабочих машин: серверы и клиенты. Рабочий контакт всегда инициирован клиентской машиной. Протокол же описывает, по каким правилам этот контакт установлен и действует. Архитектура взаимодействия между клиентом и сервером подразделяется на два вида:

· **двухзвенная**. Сторонние ресурсы не задействованы. Одна машина обрабатывает поступившие сообщения. В этом случае сервер должен быть высокопроизводительным. Несмотря на эти жесткие требования, архитектура очень надежная. Первый уровень – клиент отправляет запрос. Второй уровень сервером принимается сообщение, обрабатывается и отправляется ответ.

· **многоуровневая.** Речь идет о любой современной архитектуре СУБД. Принципиальное отличие и особенность: запросом клиента занимаются одновременно несколько серверных устройств. Операции перераспределяются, нагрузка на серверную машину снижена и оптимальная. Единственный минус: низкая надежность по сравнению с предыдущим вариантом.

Многоуровневая клиент-серверная архитектура

Обработкой данных занимаются несколько разных серверов. Благодаря такому подходу возможности серверов и клиентов используются более эффективно за счет разделения функций:

· хранения;

· обработки;

· предложения информации.

К тому же, систему можно точнее разделить на функциональные блоки для выполнения конкретной роли. Для этого между собой взаимодействуют разнообразные серверы приложений. К примеру, реально выделить сервер, необходимый для выполнения всего функционала по управлению персоналом. При этом реально сделать такую настройку, что пользователи смогут пользоваться только его общедоступным функционалом, а детали реализации серверной машины будут недоступны, так как с ней свяжут отдельную базу данных. Подобные системы легко адаптируются под веб, ведь легче организовать доступ пользователей к конкретному функционалу БД посредством html форм, чем ко всей БД. На веб-технологию очень просто перевести многоуровневую систему. Заменяют клиентскую часть браузером спецтипа или универсального назначения. При этом дополняют веб-сервером и компактными программными модулями сервер приложений. Многоуровневая архитектура также использует менеджеры транзакций. Обмен информацией одновременно происходит между одной серверной машиной приложений и несколькими серверами БД.

Преимущества системы:

· информация защищена и безопасно хранится. Так как серверная машина БД ведет базы данных, можно независимо от программ пользователя обрабатывать информацию в базе;
· повышенная стойкость к сбоям. Сохранена целостность информационных запросов, они доступны другим пользователям, если во время работы клиента случился сбой;
· масштабируемость. Архитектура адаптируется к увеличению количества пользователей. База данных также расширяется в объеме. Однако при этом не поставлена задача менять ПО. Система наращивает аппаратные средства, так происходит подстройка под меняющиеся факторы;
· повышенная защита данных от взлома и опасных атак;
· один пользователь меньше нагружает сеть, поэтому увеличивается ее пропускная способность. Можно удовлетворить запросы большего количества пользователей;
· гибкость системы.
Преимущества и недостатки архитектуры клиент-сервер
Разделен код программы клиентского и серверного приложения. Это главное преимущество архитектуры. Выбрана локальная сеть. Поэтому плюсы следующие:

· к клиентским рабочим станциям выдвигают низкие запросы;
· преимущественно все вычислительные операции выполняются на серверах;
· гибкая система;
· реально повысить защиту локальной сети.

Но не все так гладко с клиент-серверной архитектурой, есть и недостатки:
· серверные машины стоят в разы дороже, чем клиентские рабочие станции;
· обслуживание серверов доверяют только квалифицированным и профессионально подготовленным специалистам;
· работа клиентских компьютерных устройств остановлена, если в локальной сети «полетело» серверное оборудование.

Важно понимать, что нет четкого разделения оборудования на клиентское и серверное. Просто архитектура к/с дает возможность перераспределить и оптимизировать загруженность и распределить функциональность между этими рабочими станциями.