

Термин «клиент-серверная архитектура» – сборное понятие, состоящее из двух взаимодополняющих компонентов: сервера и, собственно, клиента.

Клиент – локальный компьютер на стороне виртуального пользователя, который выполняет отправку запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

Сервер – очень мощный компьютер или специальное системное оборудование, которое предназначается для разрешения определенного круга задач по процессу выполнения программных кодов. Он выполняет работы сервисного обслуживания по клиентским запросам, предоставляет пользователям доступ к определенным системным ресурсам, сохраняет данные или БД.

Особенности такой модели заключаются в том, что пользователь отправляет определенный запрос на сервер, где тот системно обрабатывается и конечный результат отсылается клиенту. В возможности сервера входит одновременное обслуживание сразу нескольких клиентов.

Если одновременно поступает более одного запроса, то такие запросы устанавливаются в определенную очередь и сервером выполняются по очереди. Порой запросы могут иметь свои собственные приоритеты. Часть запросов с более высокими приоритетами будут постоянно выполняться в первоочередном порядке!

Параметры, которые могут реализоваться **на стороне сервера**:

1. Хранение, защита и доступ к данным;
2. Работа с поступающими клиентскими запросами;
3. Процесс отправки ответа клиенту.

Параметры, которые могут реализоваться **на стороне клиента**:

1. Площадка по предоставлению пользовательского графического интерфейса;
2. Формулировка запроса к серверу и его последующая отправка;
3. Получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

Архитектура системы клиент-сервер формулирует принципы виртуального общения между локальными компьютерами, а все правила и принципы взаимодействия находятся внутри протокола.

Сетевой протокол – это особый набор правил, на основании которого выполняется точное взаимодействие между компьютерами внутри виртуальной сети.

ЕСТЬ 2 ВИДА КЛИЕНТ-СЕРВЕРНЫХ АРХИТЕКТУР:

1. Двухуровневая, состоящая из 2 узлов:

- сервер, который ответственен за получение входящих запросов и отправку ответа пользователю, применяя при этом собственные ресурсы системы;
- клиент, который может предоставлять пользовательский графический интерфейс.

Особенности работы заключаются в том, что на сервер приходит определенный запрос, потом его обрабатывают и дают напрямую, без дополнительного применения группы внешних ресурсов.



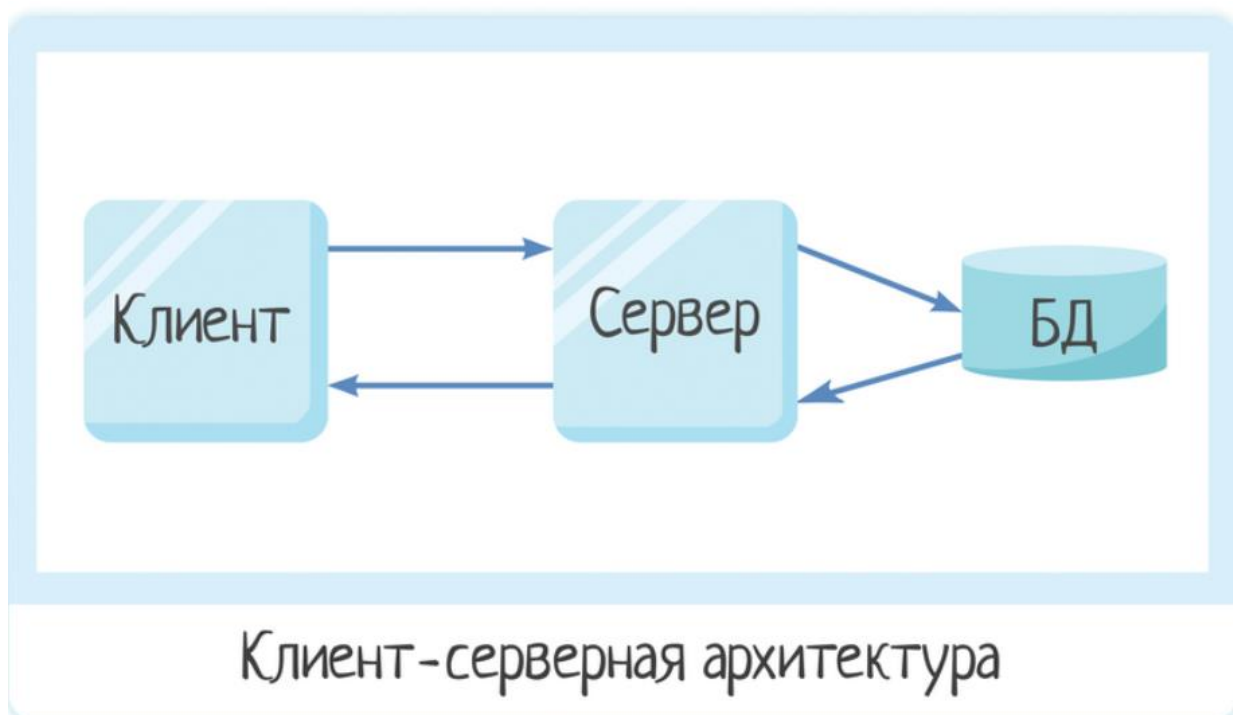
2. Трехуровневая система состоит из использования таких компонентов:

Клиент

Сервер

БД (база данных) — отдельный программный продукт, который позволяет:

- быстро делать выборки информации;
- сохранять информацию даже при рестарте системы.



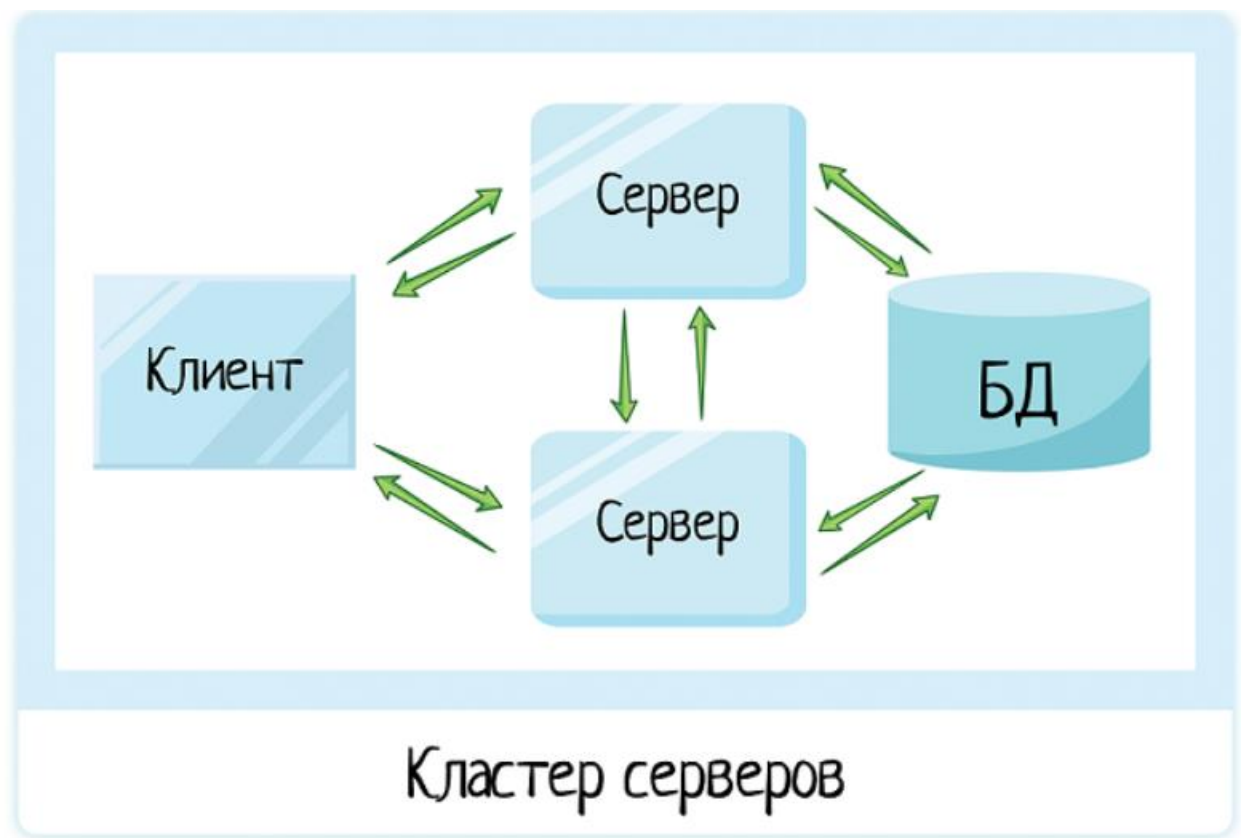
Плюсы архитектуры

Резюмируем плюсы архитектуры:

1. **Мощный сервер дешевле 100+ мощных клиентских машин** — если мы хотим, чтобы приложение не тормозило, нужна хорошая машина. Она у вас будет одна. Или несколько, если нагрузка большая, но явно меньше, чем количество клиентов.
2. **Нет дублирования кода** — основной код хранится на сервере, клиент отвечает только за «нарисовать красивенько» и простенькие проверки на полях «тут число, тут строка не длиннее 100 символов».
3. **Персональные данные в безопасности** — простой пользователь не видит лишнего. Он не знает ваше ключевое слово, паспортные данные и количество денег на счете.

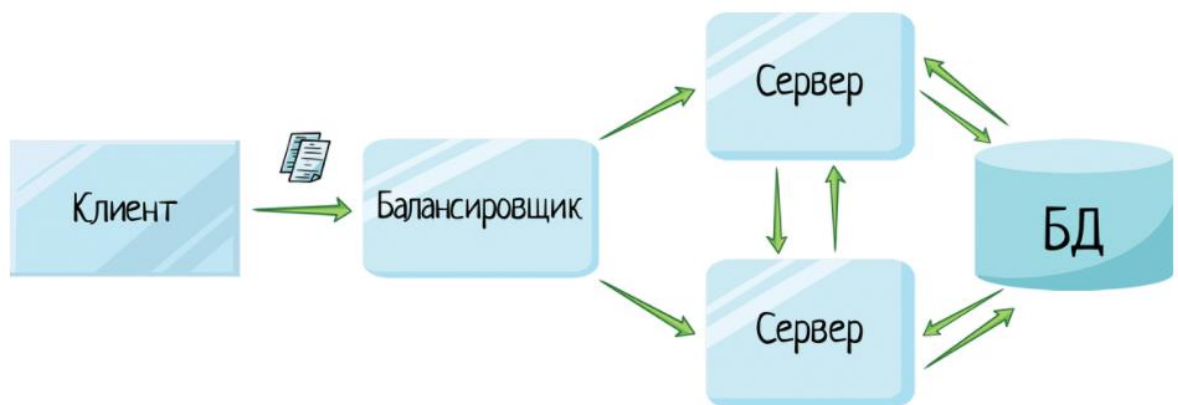
Минусы архитектуры: При падении сервера или БД, все клиенты не могут работать.

Для решения этой проблемы применяют кластер серверов:

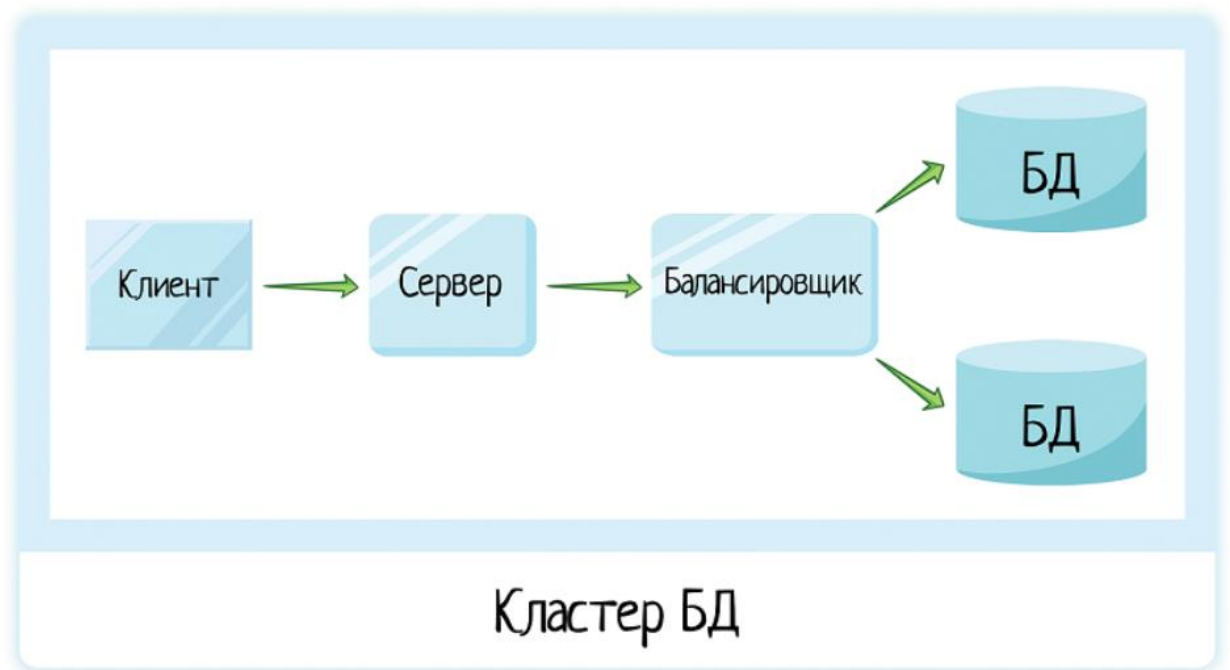


Как в таком случае клиент понимает, куда ему отправлять запрос?

Перед серверами ставят балансировщик, и клиент шлет запрос туда. Сколько бы серверов не поставили в кластер, клиенту это не интересно. У него есть один URL — адрес балансировщика.



Балансировщик распределяет нагрузку между серверами сколько бы их не было, так же может регулировать запросы сервера к базам данных



Есть две принципиальные схемы работы кластеров:

горячий резерв — когда у нас есть несколько серверов, работающих в параллель, и балансировщик распределяет нагрузку между ними.

холодный резерв — когда у нас второй сервер является резервной копией «на всякий случай». Все запросы идут на первый сервер, второй отдыхает, но если связь с первым пропадет, балансировщик отправит нагрузку на второй.