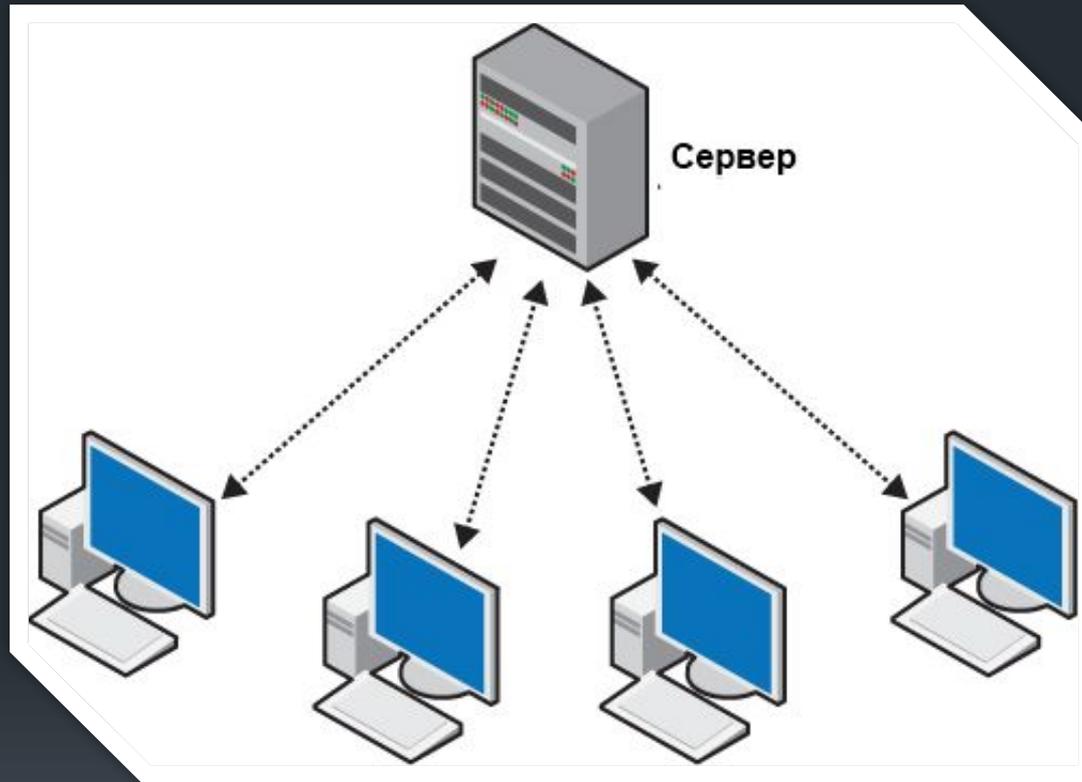


Клиент Сервер



Вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.



Программы — сервера, ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством HTTP, FTP, BitTorrent, потоковое мультимедиа или работа с базами данных) или сервисных функций (например, работа с электронной почтой, общение посредством систем мгновенного обмена сообщениями, просмотр web-страниц во всемирной паутине).

Основной принцип технологии «Клиент - Сервер» заключается в разделении функций приложения на три группы:

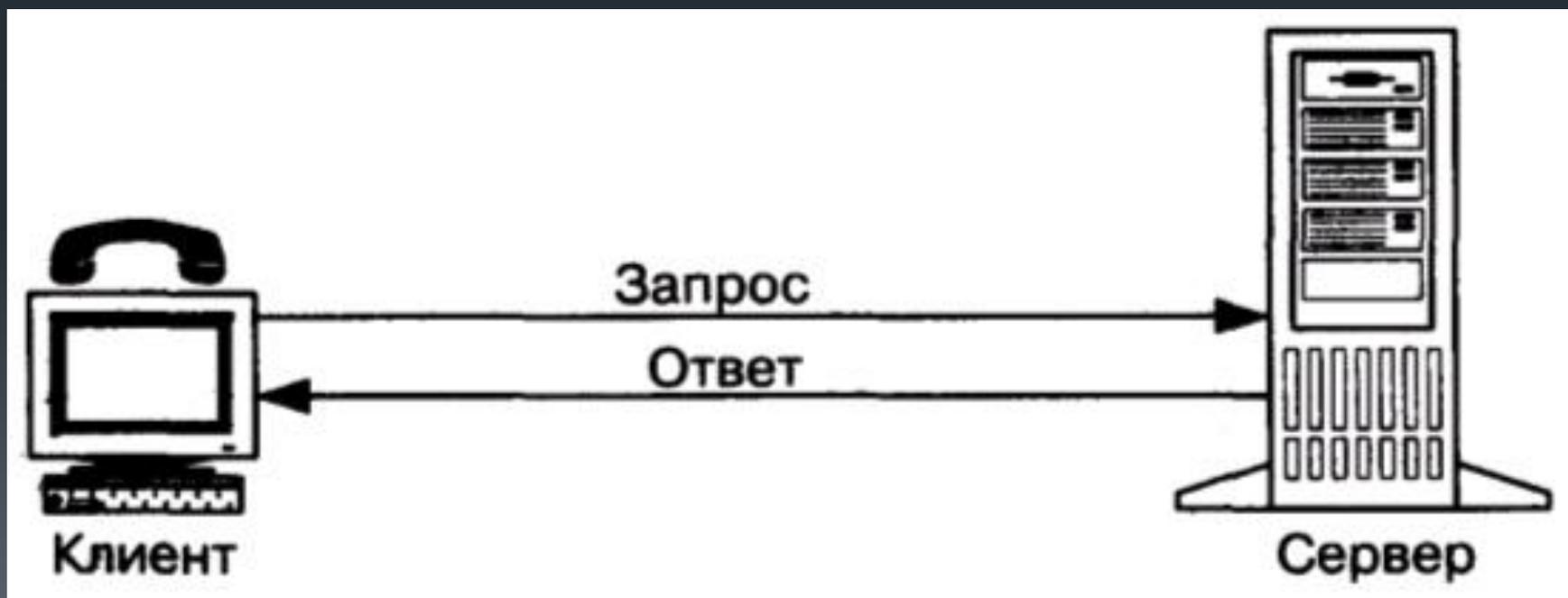
- ввод и отображение данных (взаимодействие с пользователем);
- прикладные функции, характерные для данной предметной области;
- функции управления ресурсами (файловой системой, базой данных и т.д.)

Серверная часть приложения

Обеспечение хранения и их обработка

Клиентская часть приложения

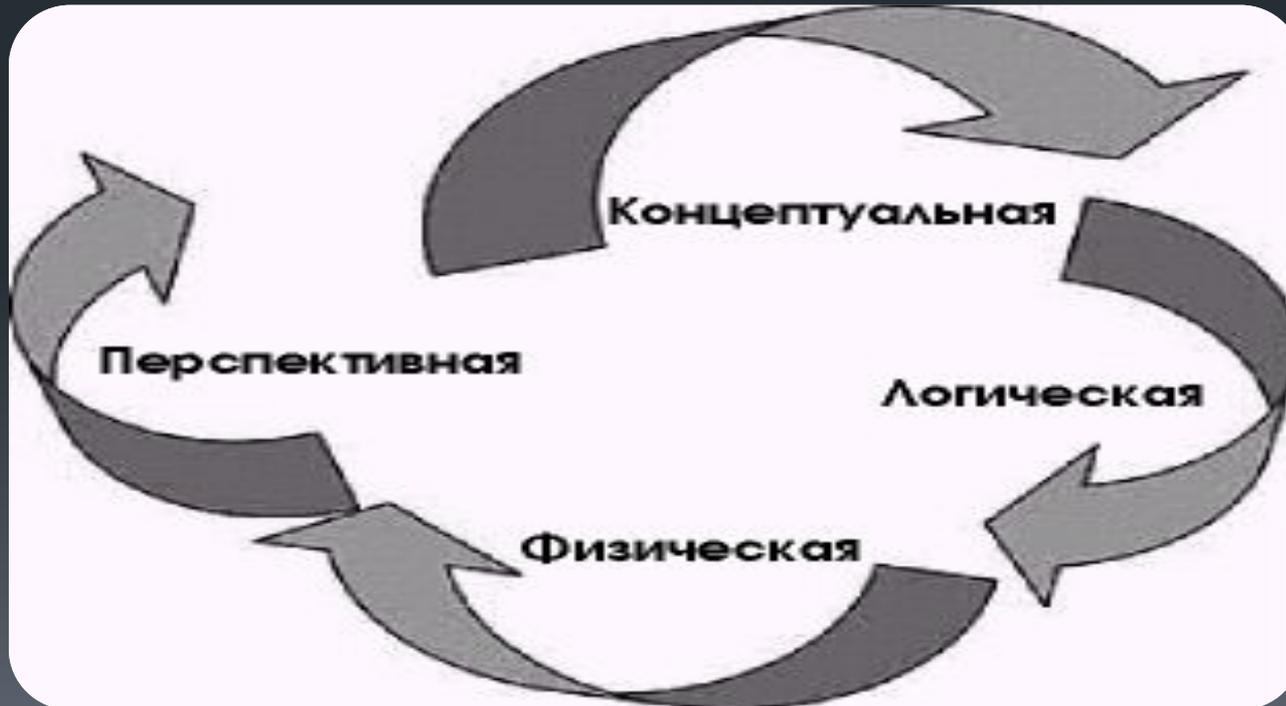
Передача серверу соответствующих запросов



Проектирование клиент-серверной системы

Стадия разработки

- Клиент-серверное проектирование оптимизированной системы управления базой данных состоит из четырех стадий: концептуальной, логической, физической и перспективной.



Стадии проектирования клиент-серверной базы данных

- КОНЦЕПЦИЯ
- ЛОГИКА
- ФИЗИЧЕСКОЕ РЕШЕНИЕ
- ПЕРСПЕКТИВА

Системы клиент-сервер



- «интеллектуальные» клиенты;
- «интеллектуальный» сервер;
- смешанные системы;
- многоуровневые системы. Схему реализации выбирают на основе анализа требований к:
 - *сетевому графику;*
 - *ресурсам клиента и сервера;*
 - *производительности базы данных.*

«Интеллектуальные» клиенты



Достоинства «интеллектуальных» клиентов

- Простота архитектуры, что облегчает разработку и сопровождение системы.
- Наличие хорошо известных и достаточно мощных средств разработки (например, Visual Basic 5.0).
- Клиент хорошо подходит для хранения текущей информации о состоянии, например первичного ключа записи, которую сейчас просматривает пользователь.

Недостатки «интеллектуальных» клиентов

- Выполнение бизнес-правил на клиенте иногда увеличивает сетевой трафик из-за необходимости передавать клиенту все данные для принятия решения на основе правил.
- Для модификации бизнес-логики необходимо повторное развертывание всех клиентов.

«Интеллектуальный сервер»



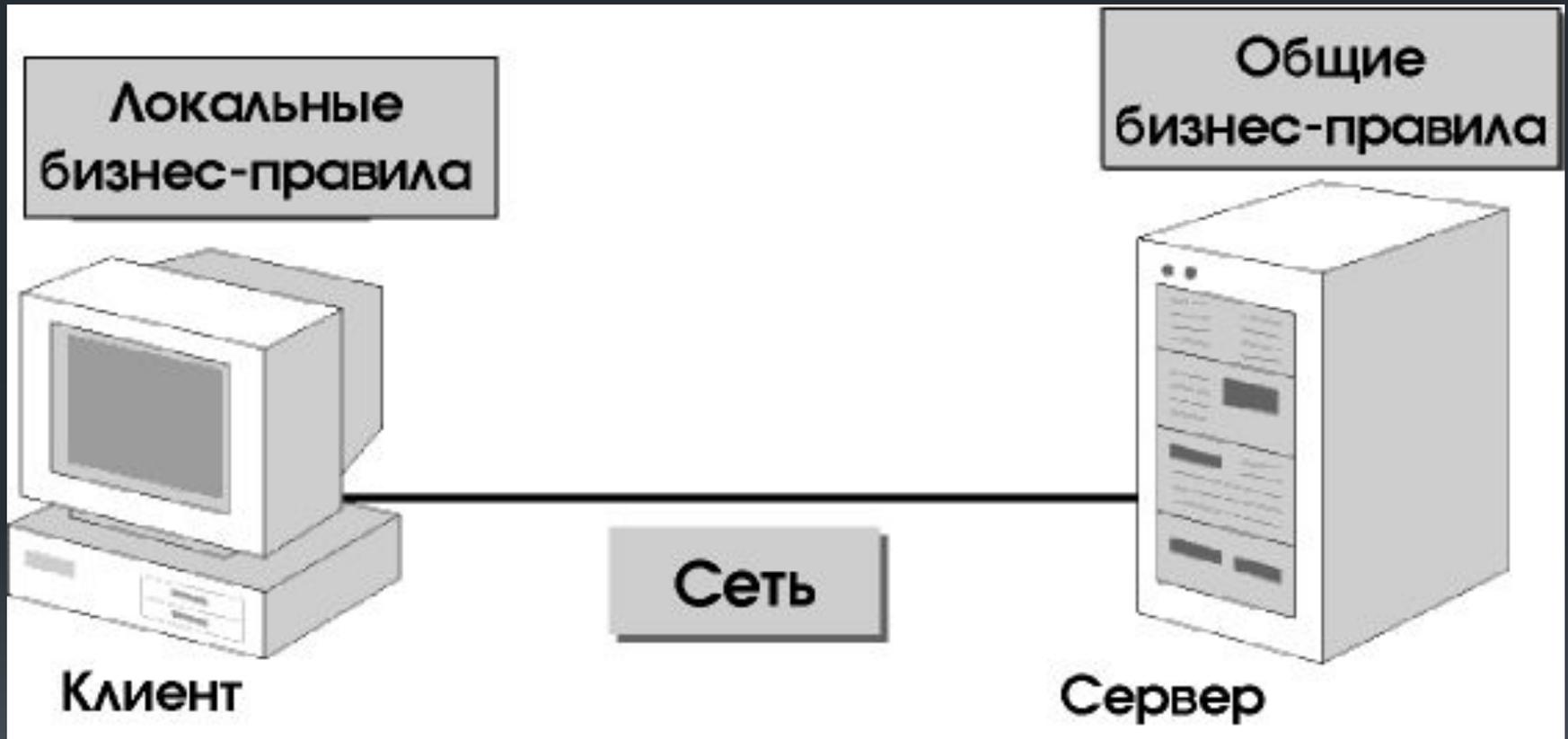
Достоинства «интеллектуальных» серверов

- Увеличение производительности: бизнес-логика выполняется в том же адресном пространстве, что и код доступа к базе данных, и, кроме того, тесно интегрирована с механизмом поиска данных SQL Server. Это означает, что данные не нужно перемещать или копировать перед обработкой, а значит, сетевой трафик минимизируется.
- На сервере легче обеспечивать целостность данных.
- При необходимости бизнес-логика модифицируется централизованно, без изменения клиентов.

Недостатки «интеллектуальных» клиентов

- Повышение требований к ресурсам сервера, где выполняются все запросы и манипуляции с данными.

Смешанные системы



Достоинства смешанных систем

- Часть бизнес-логики может быть реализована в клиентской части.
- Серверный код (например, хранимые процедуры SQL Server) одновременно доступен многим клиентам, что снижает накладные затраты при выполнении однотипных запросов.
- Эффективность работы клиентов меньше зависит от сетевого трафика.

Недостатки смешанных систем

- Бизнес-логика распределена между клиентом и сервером.
- Модернизация приложения требует распространения новых версий клиентской части среди широкой аудитории

Многоуровневые системы



Сервисы представления данных

Бизнес-логика

Хранилище данных

Сервис — это набор связанных функций, выполняющих определенные действия и/или предоставляющих информацию на основе взаимодействия с пользователем. Доступ к сервису обеспечивает интерфейс, инкапсулирующий его реализацию.

Сервисная модель — это метод рассмотрения приложения как набора средств или сервисов, которые удовлетворяют запросы клиентов.

Моделирование программы в виде набора отдельных сервисов позволяет повторно использовать компоненты, предоставляет доступ к ним другим приложениям и помогает распределять их выполнение между несколькими компьютерами сети.

Достоинства многоуровневых систем

- Разделение компонентов интерфейса, бизнес-правил и хранения данных. Возможность применения интеллектуальных клиентов. Возможность применения сервисов.

Недостатки многоуровневых систем

- Необходимы сервер и сеть. Увеличивается сетевой трафик.

Преимущества

- Отсутствие дублирования кода программы-сервера программами-клиентами.
- Так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются.
- Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще организовать контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Недостатки

- Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т. п.
- Поддержка работы данной системы требует отдельного специалиста — системного администратора.
- Высокая стоимость оборудования.

Спасибо за
внимание!!!

