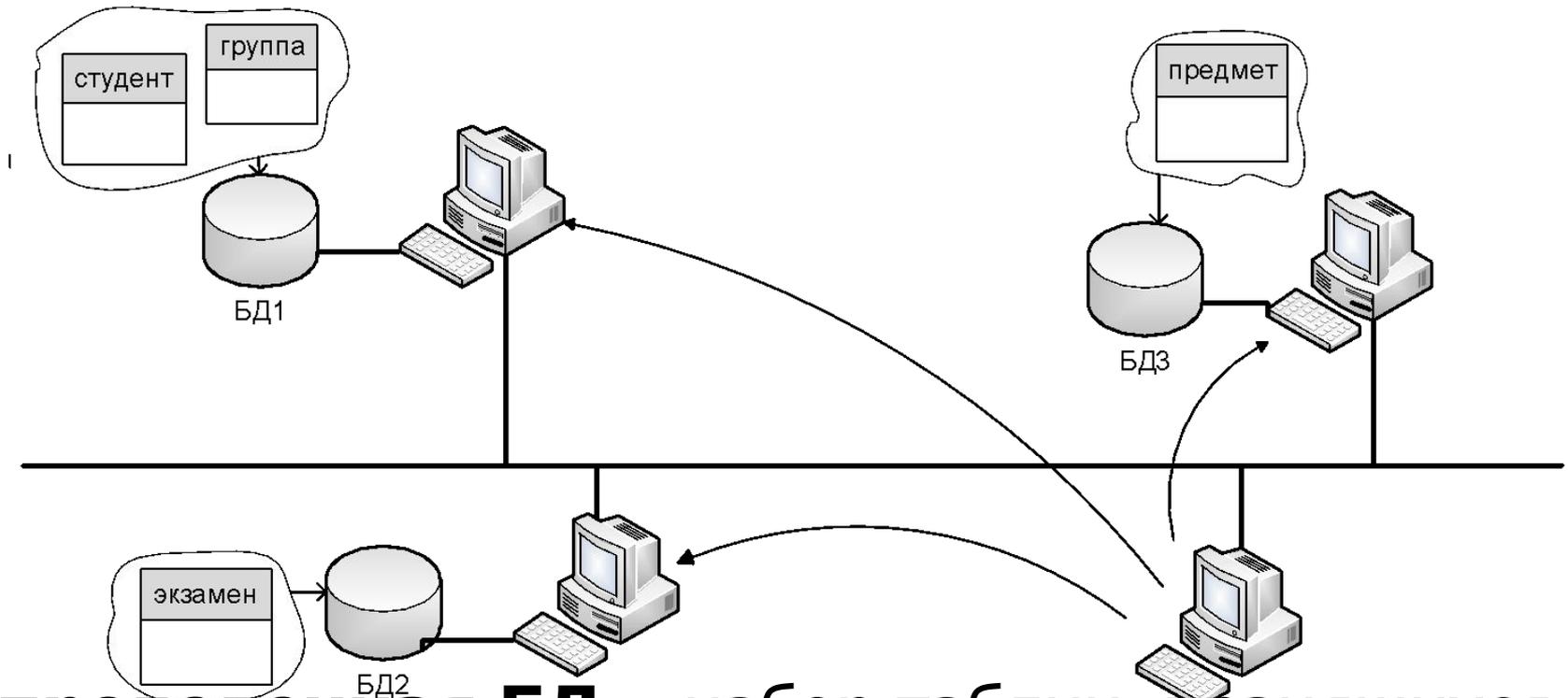


Распределенные и многопользовательские БД



Базы данных. Часть 2

Распределенные и многопользовательские БД



Распределенная БД – набор таблиц, хранящихся в разных узлах компьютерной сети и логически связанных между собой.

Распределенные и многopользовательские БД

Методы выполнения запросов в распределенной БД

Например, нужно реализовать запрос: «Выбрать всех студентов, получивших оценку 5 на экзамене по ОБД».

Метод 1

- Передать таблицу «Экзамен» из БД2 и таблицу «Предмет» из БД3 на компьютер с БД1.
- Выполнить запрос `SELECT stNum, stName FROM Students s, Subject sub, Marks m where s.stNum = m.stNum AND m.subjName = sub.subjName AND sub.subjName = 'ОБД' AND m.mark=5.`

Поскольку таблицы «Предмет» и «Экзамен» большие по объему, то и значительны затраты времени для их пересылки по сети. Кроме того, требуется достаточно места на жестком диске компьютера с БД1.

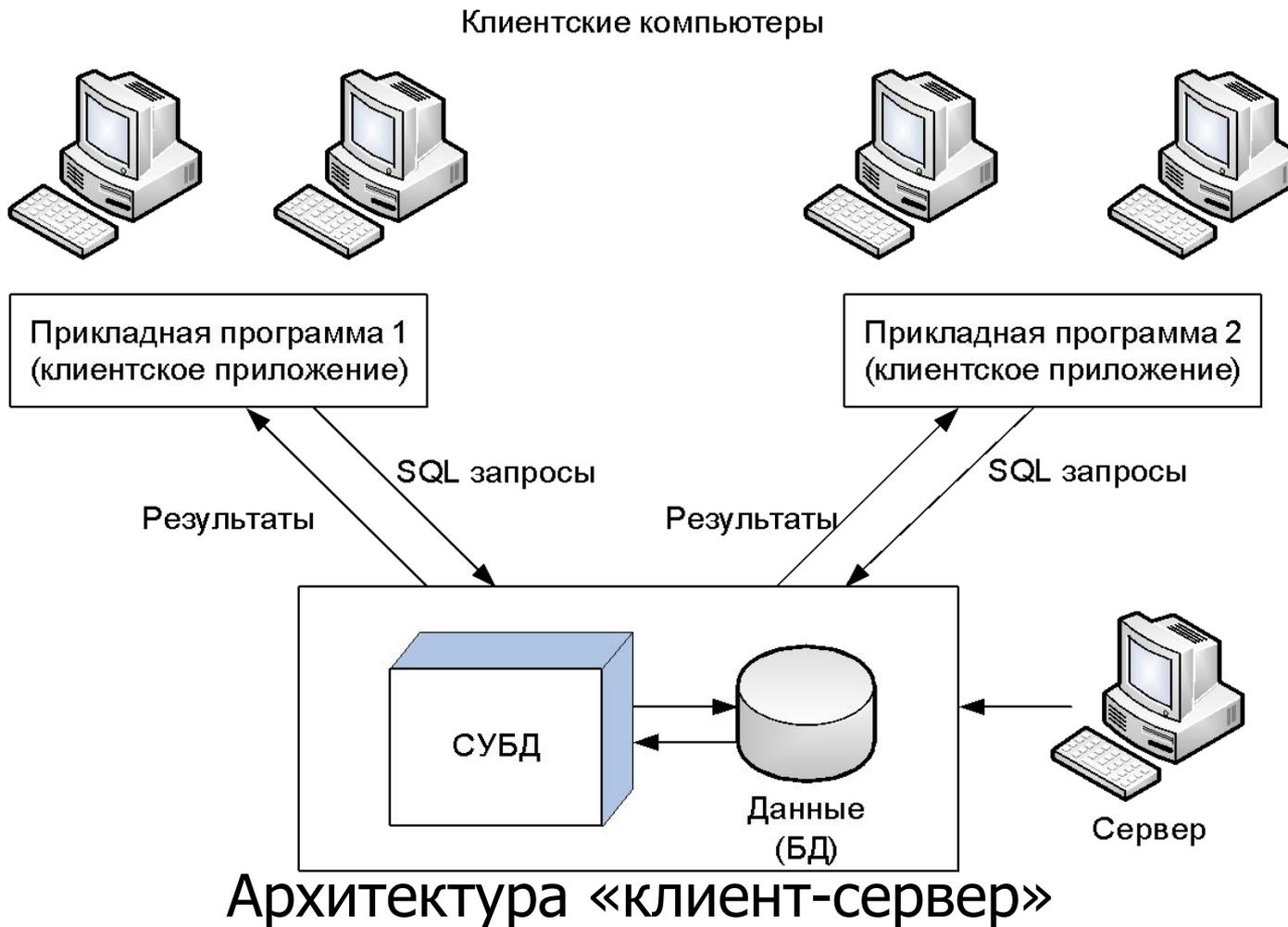
Распределенные и многopользовательские БД

Метод 2

- Послать запрос в БД3: `SELECT subjNum INTO t1 FROM Subject WHERE subjName = 'ОБД'`. Выполнить этот запрос в БД3.
- Переслать `t1` в БД2.
- Выбрать из БД2: `SELECT stNum INTO t2 FROM Marks WHERE subjNum IN (SELECT subjNum FROM t1)`.
- Переслать `t2` в БД1.
- Выполнить в БД1: `SELECT stNum, stName FROM Students WHERE stNum IN (SELECT stNum FROM t2)`.

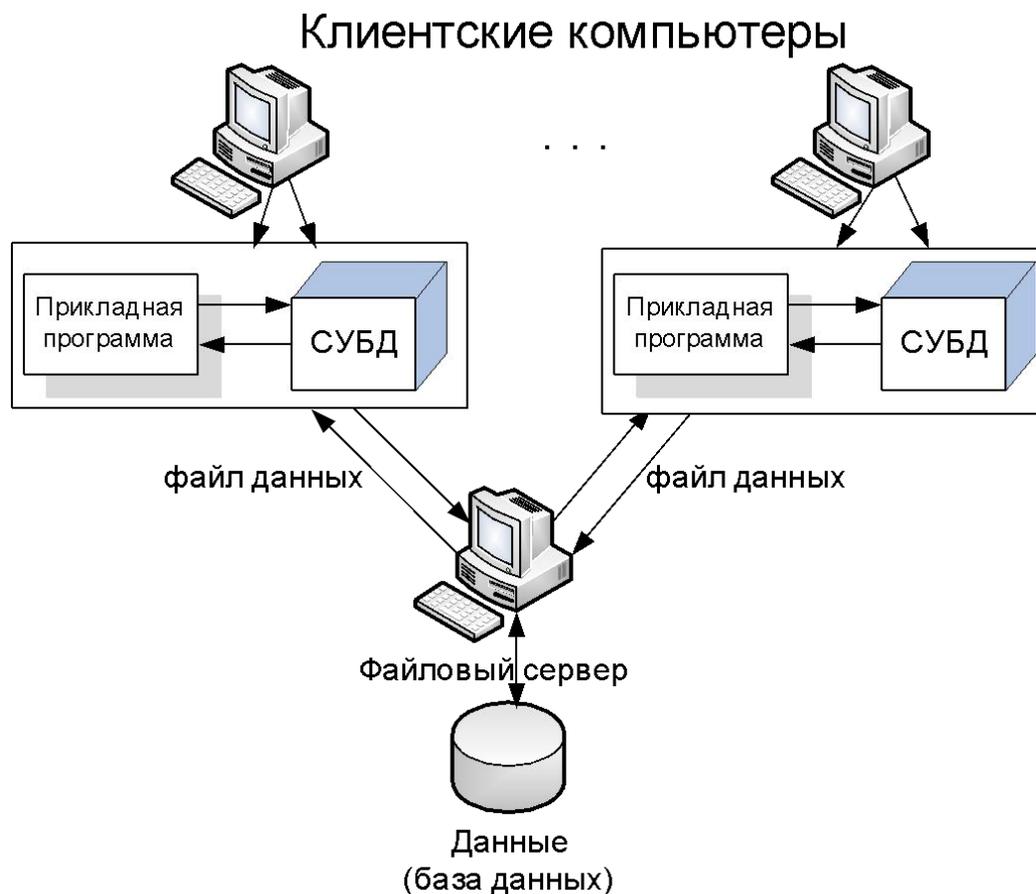
Нагрузка на сеть в этом случае небольшая, т.к. по сети передаются только результаты запросов, что время отклика на общий запрос.

Технология «клиент-сервер»



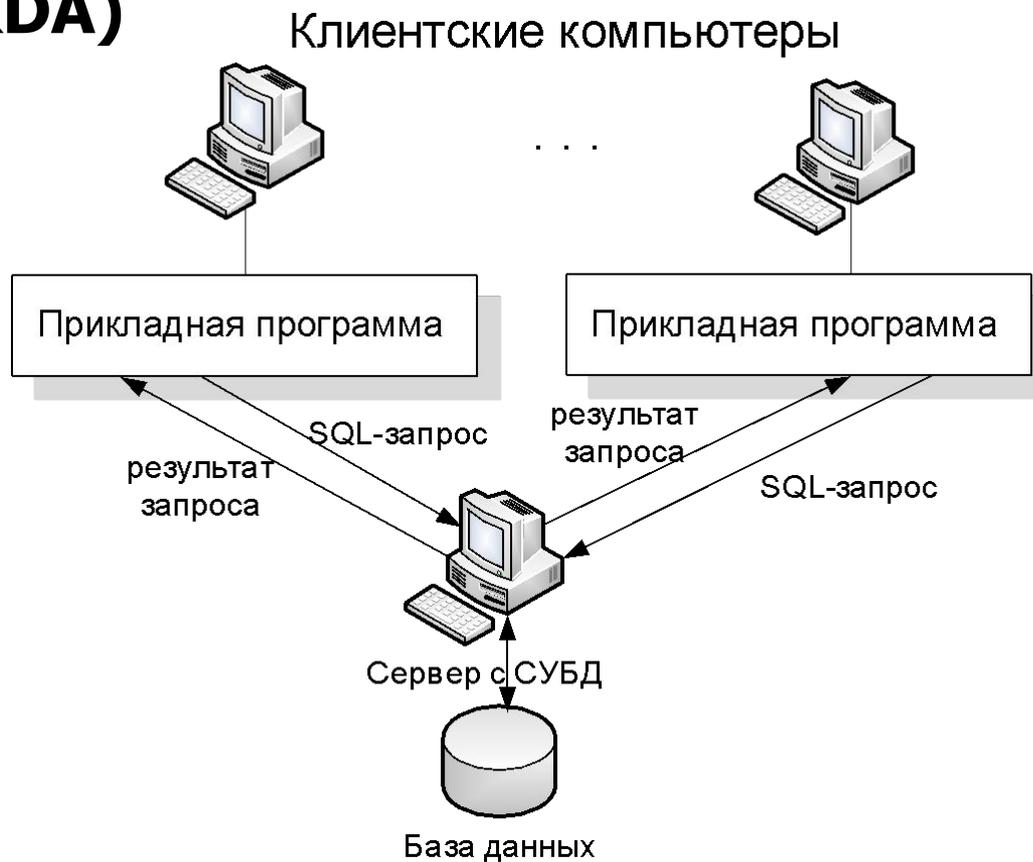
Модели двухзвенной архитектуры «клиент-сервер»

Модель файлового сервера



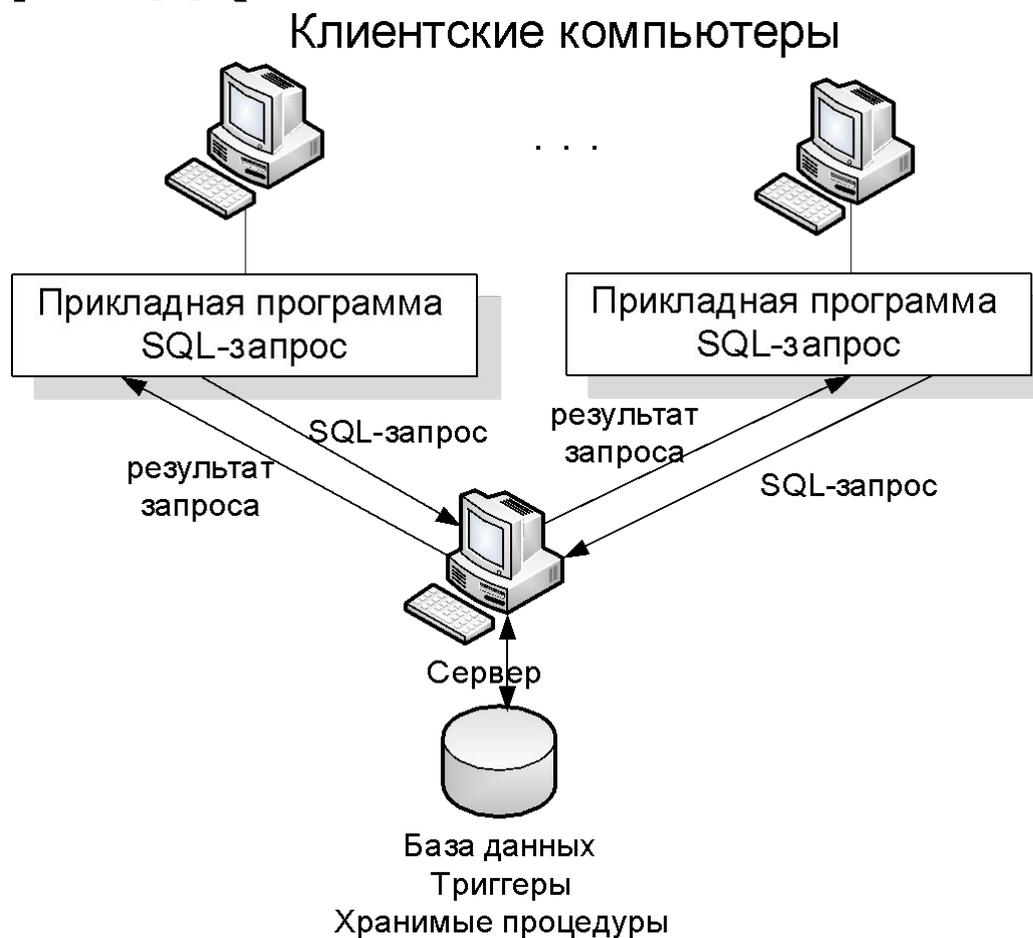
Модели двухзвенной архитектуры «клиент-сервер»

Модель удаленного доступа к данным (Remote Data Access - RDA)



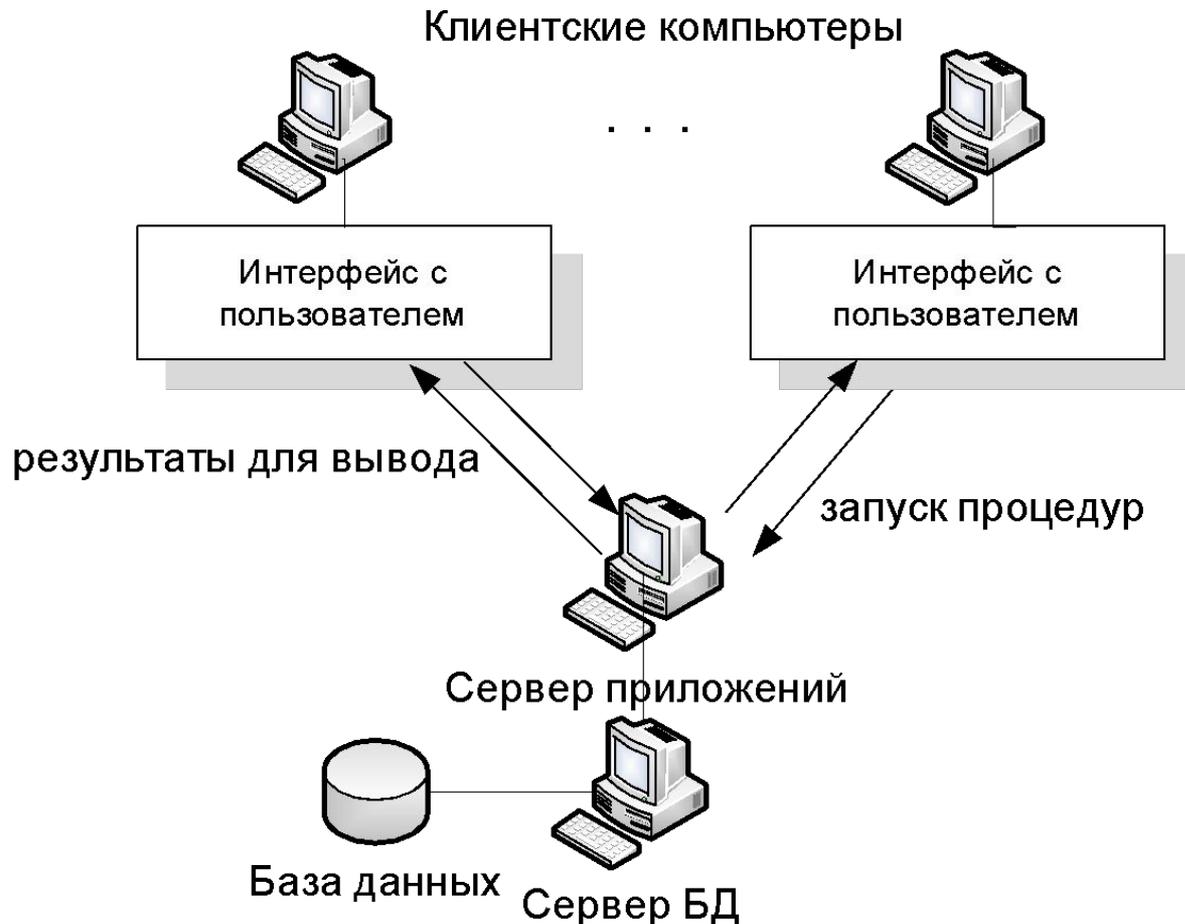
Модели двухзвенной архитектуры «клиент-сервер»

Модель сервера БД (Database Server - DBS)

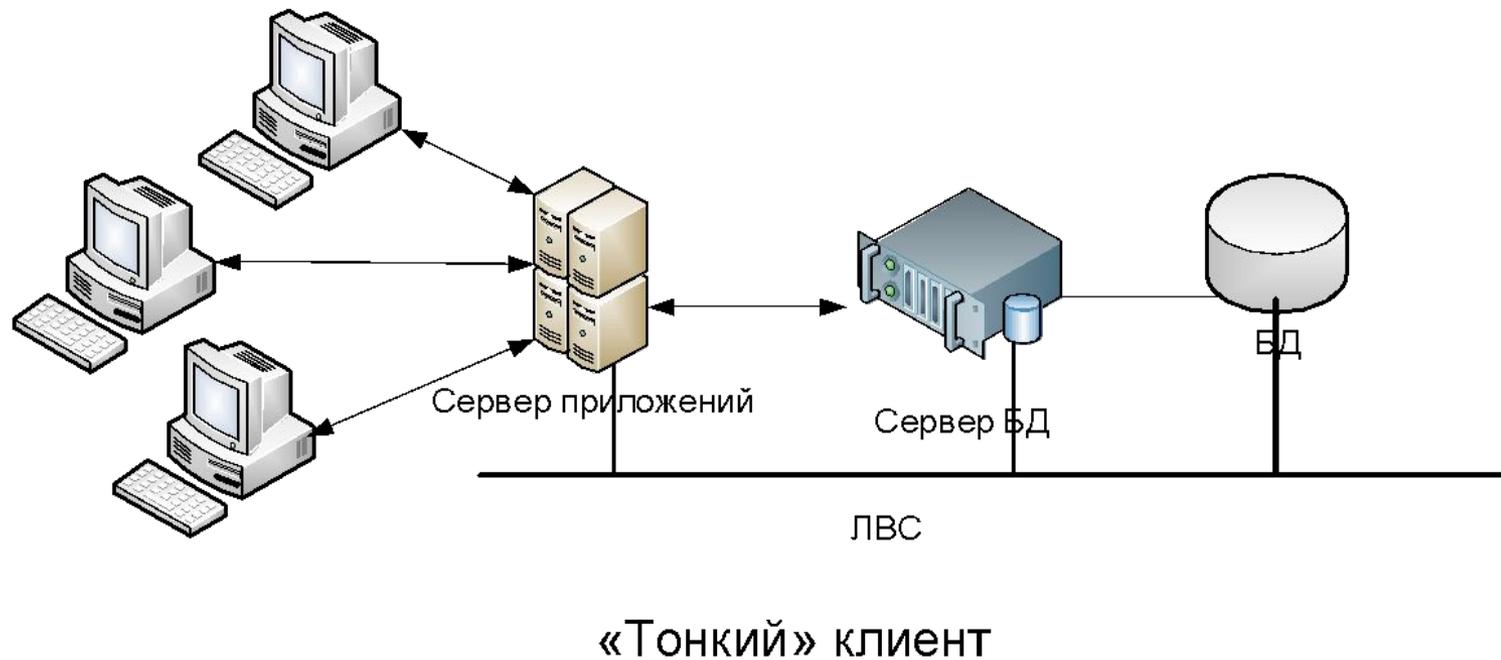


Модель трехзвенной (многозвенной) архитектуры «клиент-сервер»

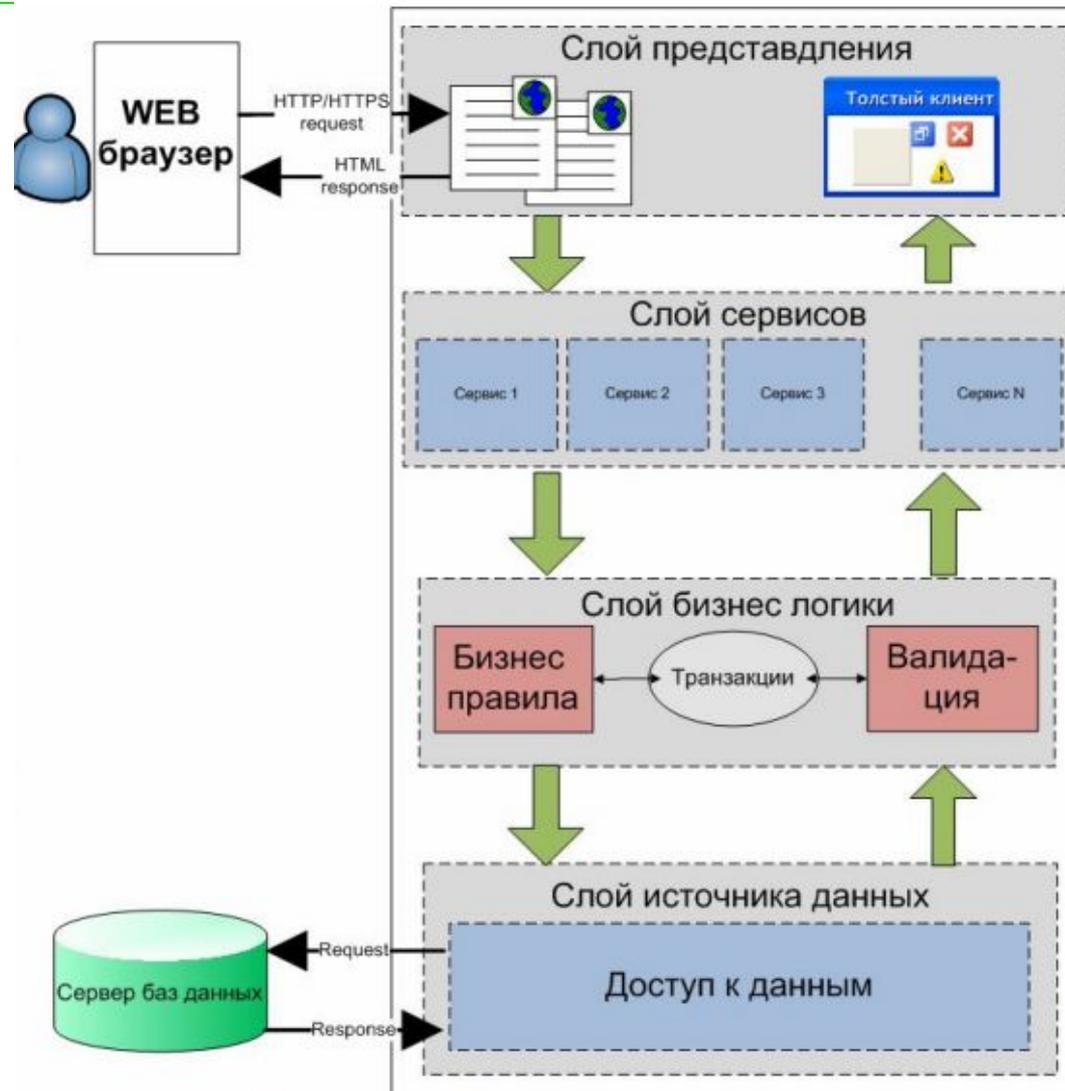
Модель сервера приложений (Application Server - AS)



Модель трехзвенной (многозвенной) архитектуры «клиент-сервер»



Многослойная архитектура корпоративного приложения



Слой доступа к данным

- Доступ к данным по средствам настройки источника данных = url, JDBC, ODBC
- Доступ к данным по средствам объектно-реляционного отображения ORM (Object Relation Mapping) = Файлы маппинга (отображения объектов)

Слой бизнес-логики

- Логика домена (бизнес-логика, логика предметной области)
- Вычисления на основе вводимых и хранимых данных, проверка всех элементов данных и обработка команд, поступающих от слоя представления, а также передача информации слою источника данных
- Вызов выполняемых на сервере БД хранимых процедур и триггеров

Слой сервисов

-
-
-
-
- Определяет границы приложения и множество операций, предоставляемых им для интерфейсных клиентских слоев кода.
- Инкапсулирует бизнес-логику приложения, управляет транзакциями и координирует реакции на действия.

Слой представления

-
- -
 -
 -
 - Тонкий клиент = JSP, PHP
 - Толстый клиент = Java (Eclipse, NetBeans), C++ (Qt, C++Builder)