Распределённые Информационные Системы

- В некомпьютерных ИТ информационные ресурсы предприятия разделены и логически распределены по различным подразделениям и службам. С другой стороны, информационные ресурсы используются в целом коллективно, т.е. с одними и теми же документами, картотеками и др. могут работать одновременно несколько **сотрудников или подразделений.**
- Первоначальные подходы к созданию баз данных АИС заключались в сосредоточении данных на одной вычислительной установке, т.е. все информационные ресурсы предприятия сосредоточиваются централизованно, что порождает ряд проблем (кадровых, технологических и материальных). Опыт внедрения АСУ в 70-80 годы показал невысокую эффективность такого подхода. Осознание этих проблем стало приводить к мысли о распределенных информационных системах.

Идеи распределенных систем

В основе распределенных систем лежат две основные идеи:

- много организационно и физически распределенных пользователей, одновременно работающих с общими данными
- логически и физически распределенные данные, составляющие единое взаимосогласованное целое – базу данных (отдельные таблицы, записи и даже поля могут располагаться на различных вычислительных установках или входить в различные локальные базы данных).

Основные принципы функционирования распределенных БД.

Впервые задача создания и функционирования распределенных ИС была поставлена специалистом в области баз данных Крисом Дейтом.

Крис Дейт сформулировал **основные принципы** функционирования распределенных БД:

- прозрачность расположения данных для пользователя означает , что для пользователя распределенная БД должна представляться точно также, как и нераспределенная
- изолированность пользователей друг от друга означает, что пользователь не должен чувствовать работу других пользователей в тот момент, когда он обновляет, изменяет или удаляет данные
- синхронизация и согласованность (непротиворечивость) состояния данных в любой момент времени.

Дополнительные принципы

Из основных вытекает ряд дополнительных принципов:

- локальная автономия, т.е. ни одна вычислительная установка не должна зависеть от любой другой установки
- отсутствие центральной установки
- независимость от местоположения (пользователю все равно, где находятся данные, он работает так, как будто они находятся на его локальной установке)
- непрерывность функционирования (отсутствие плановых отключений системы в целом)
- независимость от фрагментации данных, как от горизонтальной фрагментации, когда записи одной таблицы размещены на различных установках, так и от вертикальной фрагментации, когда различные поля одной таблицы размещены на разных установках
- независимость от реплицирования (дублирования) данных, когда какая-либо таблица БД физически может быть представлена несколькими копиями, расположенными на различных установках
- распределенная обработка запросов (оптимизация запросов должна носить распределенный характер – сначала глобальная оптимизация, а затем локальная оптимизация на каждой из задействованных установок)
- распределенное управление транзакциями (транзакция считается завершенной, если она успешно завершена на всех задействованных установках
- независимость от аппаратуры (система должна успешно функционировать на компьютерах различного типа)
- независимость от типа операционной системы
- независимость от коммуникационной сети
- независимость от СУБД (на практике СУБД, поддерживающие язык SQL).

Техника “представлений”

Представлением называется сохраняемый в базе данных авторизованный глобальный запрос на выборку данных. Авторизованность означает конкретно поименованного в систем пользователя, а глобальность означает, что выборка данных может осуществляться со всей базой данных. В результате таких глобальных авторизованных запросов для конкретного пользователя создается некая виртуальная БД со своей схемой данных и своими данными. В принципе, с точки зрения информационных задач пользователю безразлично, где и в каком виде находятся сам данные.

Схема техники «представлений»

При входе пользователя в распределенную систему ядро СУБД, идентифицируя пользователя, запускает запросы его ранее определенного и хранимого в БД представления и формирует ему его видение базы данных, которое воспринимается пользователем как обычная локальная база данных. Так как представление БД виртуально, то данные физически находятся на своем обычном месте. При осуществлении пользователем манипуляций с данными ядро распределенной СУБД по системному каталогу БД находит данные, определяет стратегию действий с ними. Большая часть таких операций невидима для пользователя, и он воспринимает работу в распределенной БД, как в обычной локальной БД.

Несмотря на простоту и изящность идеи представлений, практическая реализация подобной технологии встречает ряд серьезных проблем.

Первая из них связана с размещением системного каталога базы данных, поскольку при запросе пользователя ядро распределенной СУБД в первую очередь должно узнать где и в каком виде находятся данные. Требование отсутствия центральной установки означает, что системный каталог должен быть на любой локальной установке. Тогда возникает проблема обновлений. Если пользователь изменил данные или структуру в системе, то эти изменения должны отразиться во всех системных каталогах, что может встретить трудности в виде недоступности системных каталогов в этот момент.

Решение подобных проблем и практическая реализация ИС осуществляется путем отступления от некоторых рассмотренных выше принципов создания и функционирования распределенных систем.

В зависимости от того, какой принцип приносится в жертву, выделились несколько направлений в технологиях распределенных систем –

- технология клиент-сервер
- технология реплицирования
- технология объектного связывания, причем реальные распределенные ИС построены, как правило, на сочетании всех трех технологий, но рассматривать их целесообразно отдельно.

Технологии и модели Клиент-сервер

Системы на основе технологии клиент-сервер исторически выросли из первых централизованных многопользовательских АИС. В технологиях клиент-сервер отступают от одного из главных принципов функционирования распределенных систем - отсутствия центральной установки.

Две основные идеи, лежащие в основе клиент-серверных технологий

- общие для всех пользователей данные на одном или нескольких серверах;
- много пользователей на различных вычислительных установках, совместно обрабатывающих данные.

Иначе говоря, системы, основанные на клиент-серверных технологиях, распределены только в отношении пользователей, поэтому их часто не относят к распределенных системах, а считают отдельным классом многопользовательских систем.

Под сервером в широком смысле понимается любая система, процесс или компьютер, владеющие каким-либо вычислительным ресурсом (памятью, временем, процессором и т.д.).

Клиентом называется также любая система, процесс, компьютер, запрашивающие у сервера какой-либо ресурс, пользующиеся каким-либо ресурсом или обслуживаемые сервером иным способом.

В своем развитии системы клиент-сервер прошли несколько этапов, в ходе которых сформировались различные модели систем клиент-сервер. Их реализации основаны на разделении структуры СУБД на три компонента:

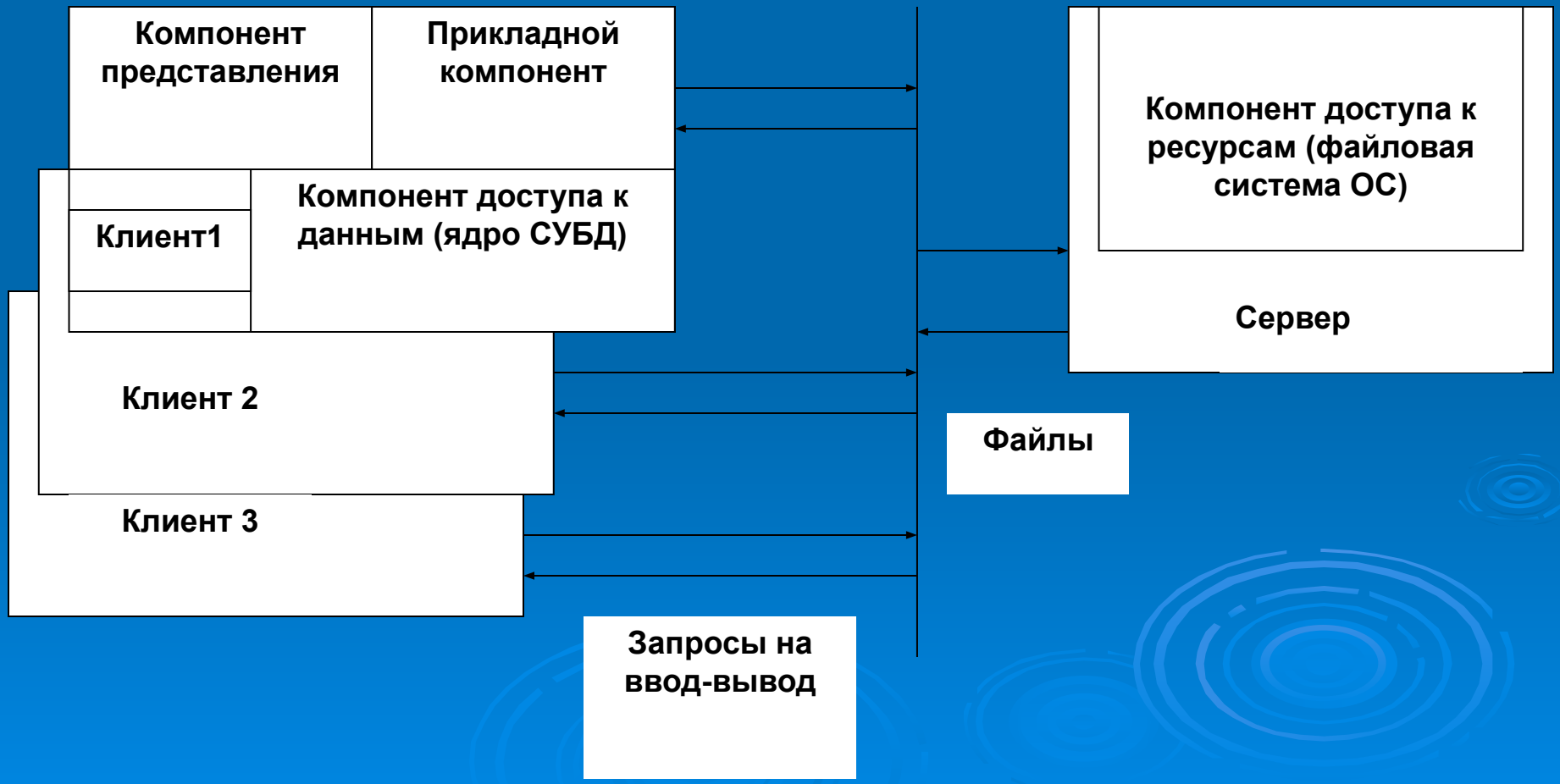
- компонент представления (функции ввода и отображения данных - интерфейс пользователя);
- прикладной компонент (набор запросов, событий, процедур и др. вычислительных функций);
- компонент доступа к данным (функции хранения, извлечения, обновления данных – машина данных).

Исходя из особенностей реализации и распределения в системе этих трех компонентов различают четыре модели технологий клиент-сервер:

- модель файлового сервера (File Server – FS);
- модель удаленного доступа к данным (Remote Data Access – RDA);
- модель сервера базы данных (Data Base Server – DBS);
- модель сервера приложений (Application Server – AS).

Модель файлового сервера

Один из компьютеров сети выделяется и определяется файловым сервером, т.е. общим хранилищем любых данных. Суть FS-модели иллюстрируется схемой, приведенной на рисунке:



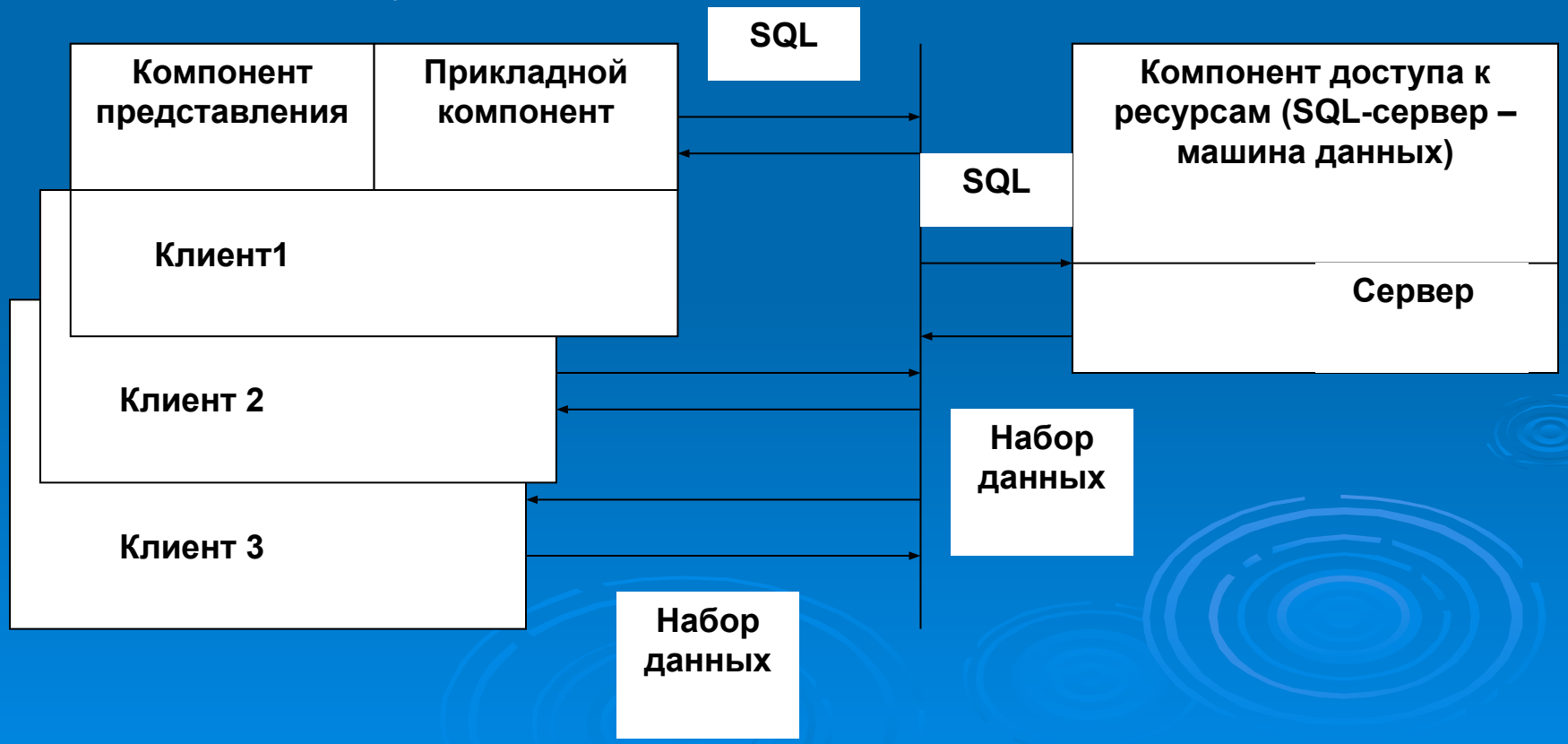
В FS-модели все основные компоненты размещаются на клиентской установке. При обращении к данным ядро СУБД обращается с запросом на ввод-вывод данных к файловой системе. ОС копирует файл БД полностью или частично в оперативную память клиентской установки, т.е. сервер выполняет пассивную функцию.

Достоинства и недостатки.

- Достоинством данной модели являются ее простота, отсутствие высоких требований к производительности сервера (главное – объем дискового пространства). Программные компоненты СУБД не распределены.
- Недостатки такой модели очевидны: это высокий сетевой трафик, достигающий пиковых значений в начале рабочего дня. Более существенным является отсутствие специальных механизмов безопасности файлов базы данных со стороны СУБД.

Модель удаленного доступа к данным

В этой модели компонент доступа к данным в СУБД полностью отделен от двух других компонентов и размещается на сервере системы. Компонент доступа к данным реализуется в виде самостоятельной программной части, называется SQL-сервером и инсталлируется на сервере системы. Функции SQL-сервера ограничиваются операциями по организации, размещению, хранению и манипулированию данными в дисковой памяти сервера. Схема RDA-модели приведена на рисунке:



В файле БД на сервере системы находится системный каталог БД, в котором содержатся в том числе и сведения о зарегистрированных клиентах, их полномочиях и т.д.

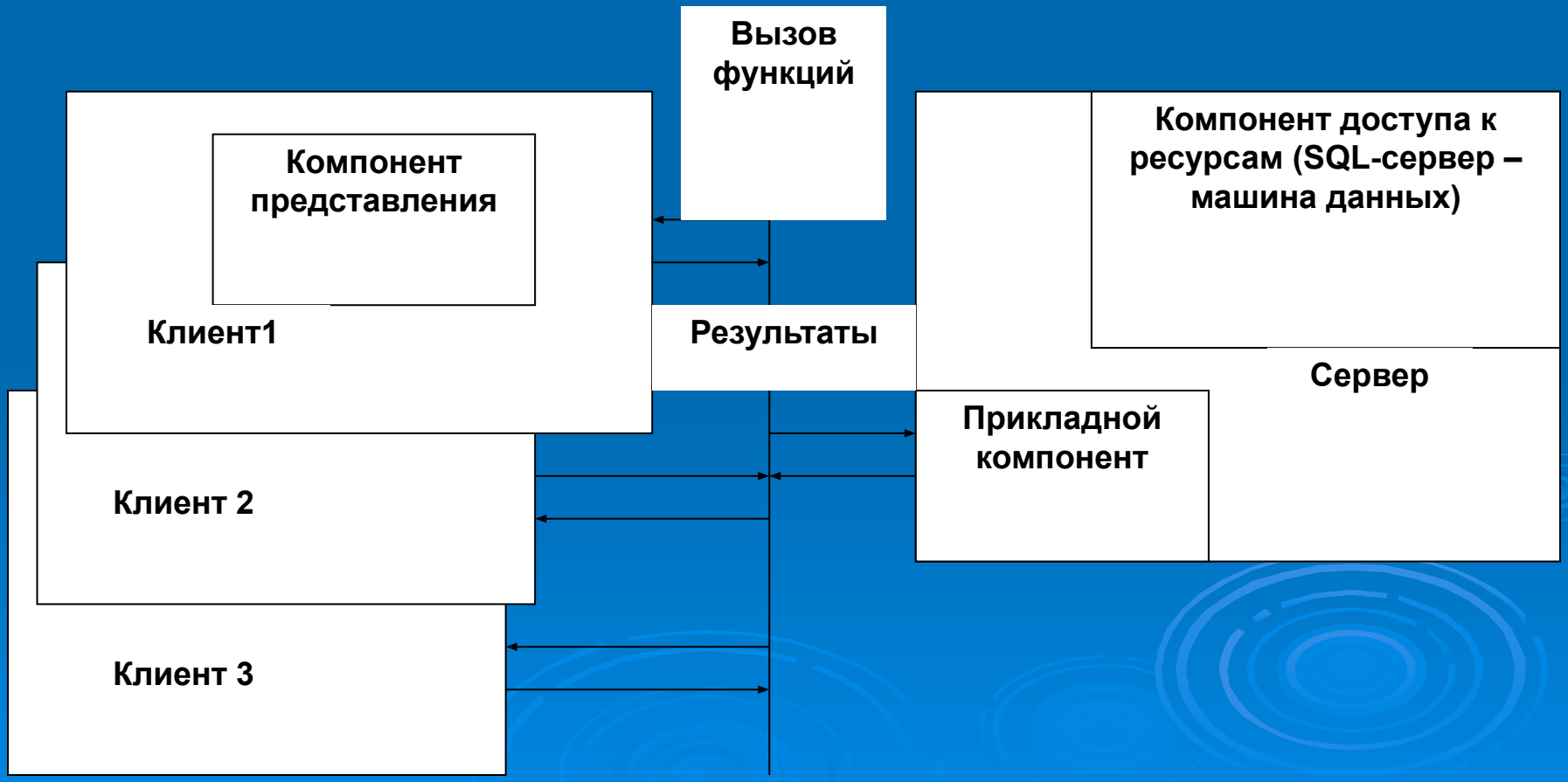
На клиентских установках инсталлируются отдельные программные части СУБД, реализующие интерфейсные и прикладные функции. Прикладной компонент формирует все необходимые SQL-инструкции и направляет их SQL-серверу. SQL-сервер принимает инструкции, выполняет их проверяет и обеспечивает ограничения целостности данных и направляет клиенту результаты обработки, т.е. наборы данных, которые могут быть существенно меньше по объему всей базы данных. В результате резко уменьшается загрузка сети, а сервер приобретает активную функцию. Ядро СУБД обеспечивает важные функции по обеспечению целостности и безопасности данных при совместной работе нескольких пользователей.

Достоинства и недостатки.

- Достоинством RDA-модели является унификация интерфейса взаимодействия прикладных компонентов ИС с общими данными. Такое взаимодействие стандартизовано в рамках языка SQL специальным протоколом ODBC(Open DataBase Connectivity), играющим важную роль в обеспечении интероперабельности, т.е. независимости от типа СУБД. Специальный компонент ядра СУБД на сервере (драйвер ODBC) способен принимать, обрабатывать и направлять результаты обработки на клиентские установки, функционирующие под управлением реляционных СУБД других, не родных типов.
- К недостаткам RDA-модели можно отнести высокие требования к клиентским установкам, ток как прикладные программы выполняются на них. Другим недостатком является все же существенный трафик сети, обусловленный тем, что с сервера БД клиентам направляются таблицы, которые могут занимать существенный объем.

Модель сервера базы данных

Эта модель стала развитием модели RDA. В отличие от нее прикладной компонент полностью размещается и выполняется на сервере системы. Схема DBS-модели приведена на рисунке:



На клиентских установках размещается только интерфейсный компонент, что существенно снижает требования к вычислительной установке клиента. Пользователь направляет на сервер БД только лишь необходимые вызовы процедур, запросов и других функций по обработке данных. Все операции по доступу и обработке данных выполняются на сервере. Клиенту направляются лишь результаты обработки, этим обеспечивается существенное снижение трафика сети по сравнению с RDA-моделью.

Достоинства и недостатки.

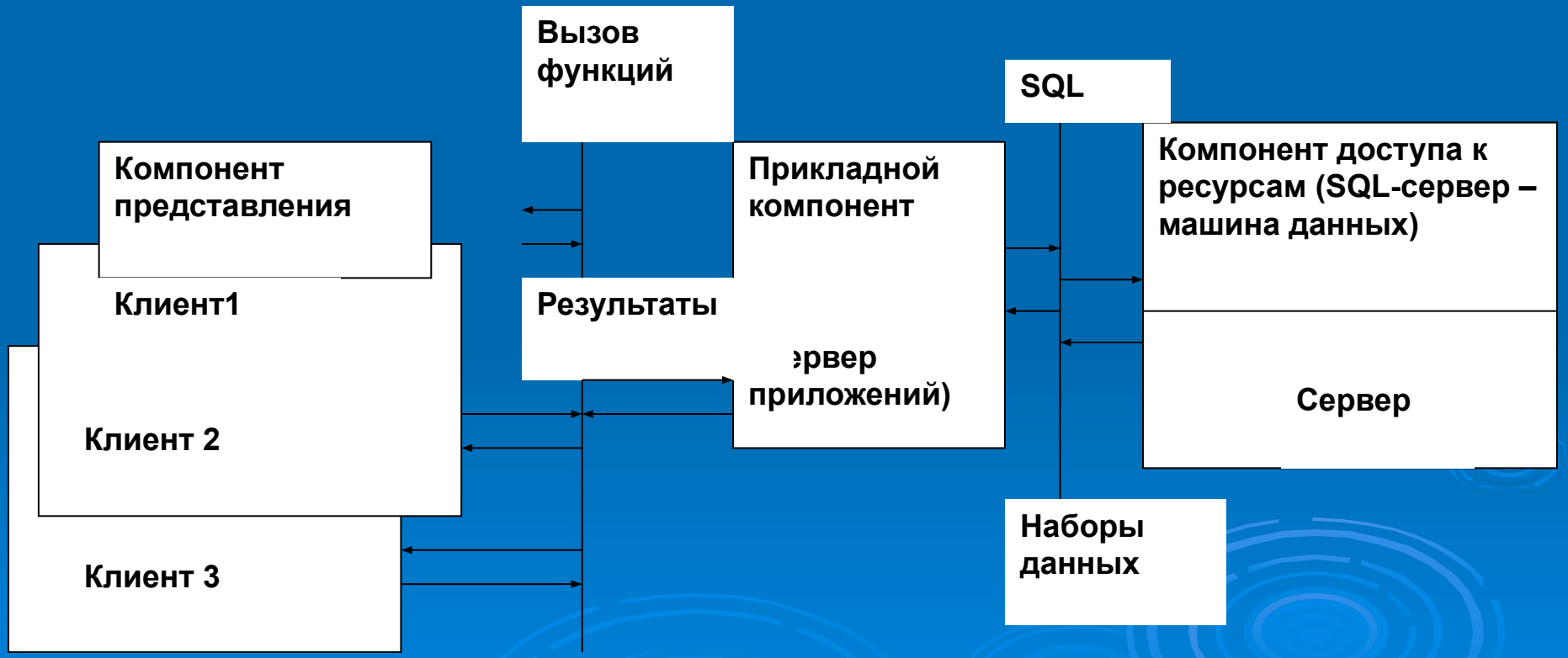
На сервере системы выполняются процедуры прикладных задач всех пользователей системы, в результате резко возрастают требования к вычислительной установке сервера (к объему дискового пространства и оперативной памяти, а также к быстродействию). Это основной недостаток DBS-модели.

Достоинства:

- разгрузка сети;
- более активная роль сервера сети (размещение, хранение и выполнение на нем механизма событий, правил и процедур дает возможность более эффективно настраивать АИС на все нюансы предметной области);
- повышается надежность хранения и обработки данных.

Модель сервера приложений

Чтобы разнести требования к вычислительным ресурсам сервера в отношении быстродействия и памяти по разным вычислительным установкам, используется модель сервера приложений. Суть модели сервера приложений заключается в переносе прикладного компонента АИС на специализированный дополнительный с**ервер системы**. Схема AS-модели приведена на рисунке:



Как и на DBS-модели, на клиентских установках располагается только клиентская часть. Вызовы функций обработки данных направляются на сервер приложений, где они выполняются для всех пользователей системы. За выполнением низкоуровневых операций сервер приложений обращается к SQL-серверу и получает от него наборы данных.