Лекция 2. Архитектура информационных систем

Опыт последних лет разработки программного обеспечения (ПО) показывает, что архитектура информационной системы должна выбираться с учетом нужд бизнеса, а не личных пристрастий разработчиков. В данной лекции рассмотрим существующие клиент- серверные архитектуры построения информационных систем.

Существует несколько определений, что такое **информационная система** и **архитектура информационной системы**. Приведем некоторые из них.

Информационная система – организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы. Информационные системы предназначены для хранения, обработки, поиска, распространения, передачи и предоставления информации.

Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы**.**

Можно сформулировать проще:

Архитектура информационной системы – абстрактное понятие, определяющее, из каких составных частей (элементов, компонент) состоит приложение и как эти части между собой взаимодействуют.

Под составными частями (элементами, компонентами) приложения обычно понимаются программы или программные модули, выполняющие отдельные, изолированные задачи.

Рисунок 1 – Компоненты информационной системы

Компоненты информационной системы (рис. 1) по выполняемым функциям можно разделить на три слоя:

- ***Слой представления (пользовательский интерфейс)*** – все, что связано с взаимодействием с пользователем: нажатие кнопок, движение мыши, отрисовка изображения, вывод результатов поиска и т. д.
- ***Бизнес-логика*** – правила, алгоритмы реакции приложения на действия пользователя или на внутренние события, правила обработки данных.
- ***Слой доступа к данным*** – хранение, выбор, модификация и удаление данных, связанных с решаемой приложением прикладной задачей.

Классификация архитектур ИС

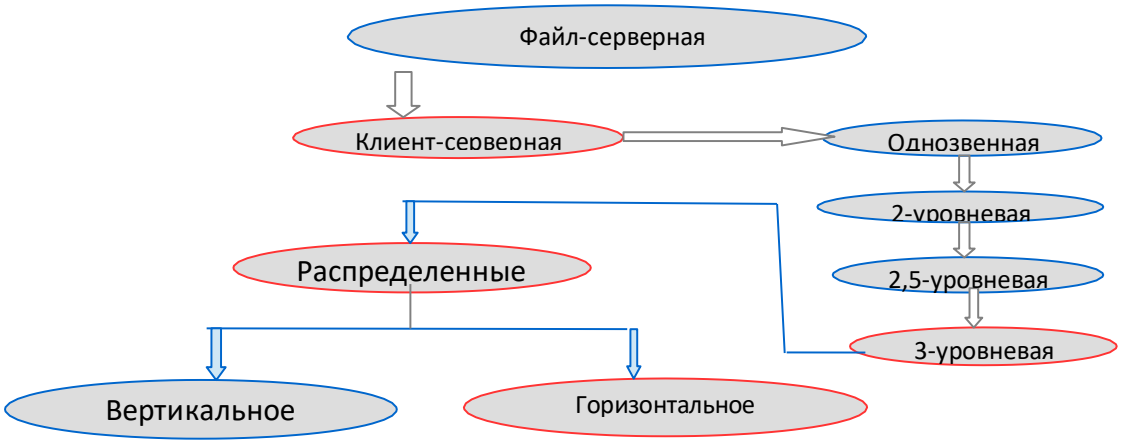

Рисунок 2 – Архитектура информационной системы

Файл-серверная архитектура

Все общедоступные файлы хранятся на выделенном компьютере ***файл-сервере*** (рис. 3).

Файл-серверные приложения – приложения, использующие сетевой ресурс для хранения программы и данных.
Функции сервера: хранение данных и кода программы.
Функции клиента: обработка данных.

Количество клиентов ограничено десятками.

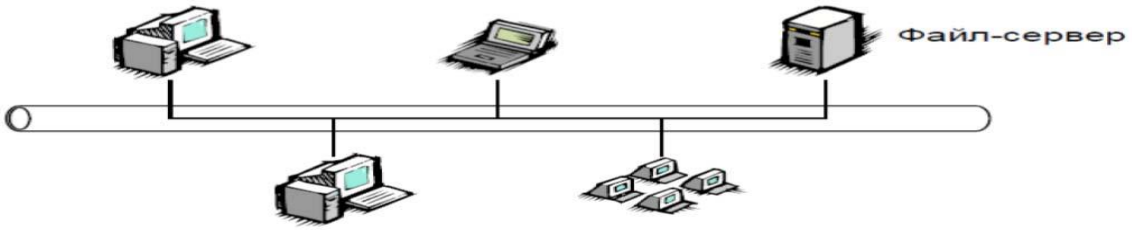

Рисунок 3 – Файл-серверная архитектура

Рисунок 4 – Модель файлового сервера

Положительные стороны:

- многопользовательский режим работы с данными;
- удобство централизованного управления доступом;
- низкая стоимость разработки.

Отрицательные стороны:
низкая производительность;

- низкая надежность;
- слабые возможности расширения.

Отрицательные стороны архитектуры с файловым сервером (рис. 4) вытекают, главным образом, из того, что данные хранятся в одном месте, а обрабатываются в другом. Их нужно передавать по сети, что приводит к очень высоким нагрузкам на сеть и резкому снижению производительности приложения при увеличении числа одновременно работающих клиентов. Вторым важным недостатком такой архитектуры является децентрализованное решение проблем целостности и согласованности данных и одновременного доступа к данным [11].

Клиент-серверная архитектура

Ключевое отличие от архитектуры ***файл-сервер*** **–** абстрагирование от физической схемы данных и манипулирование данными клиентскими программами на уровне *логической схемы* (рис. 5). Это позволило создавать надежные многопользовательские ИС с централизованной базой данных (БД), независимые от аппаратной (а часто и программной) части сервера БД и поддерживающие графический интерфейс пользователя (ГИП) на клиентских станциях, связанных локальной сетью.

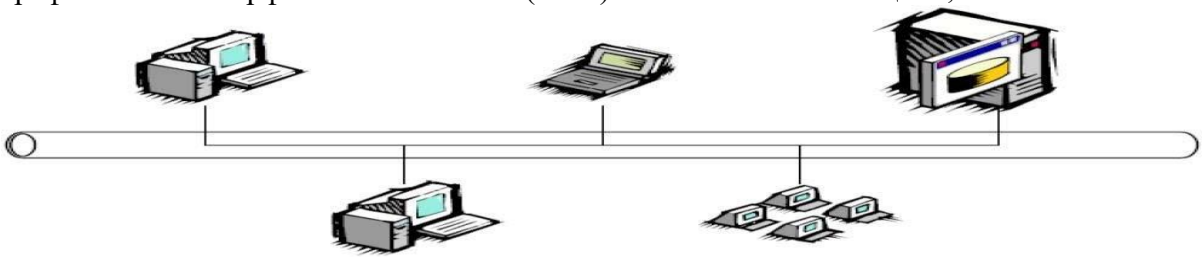

Рисунок 5 – Клиент-серверная архитектура

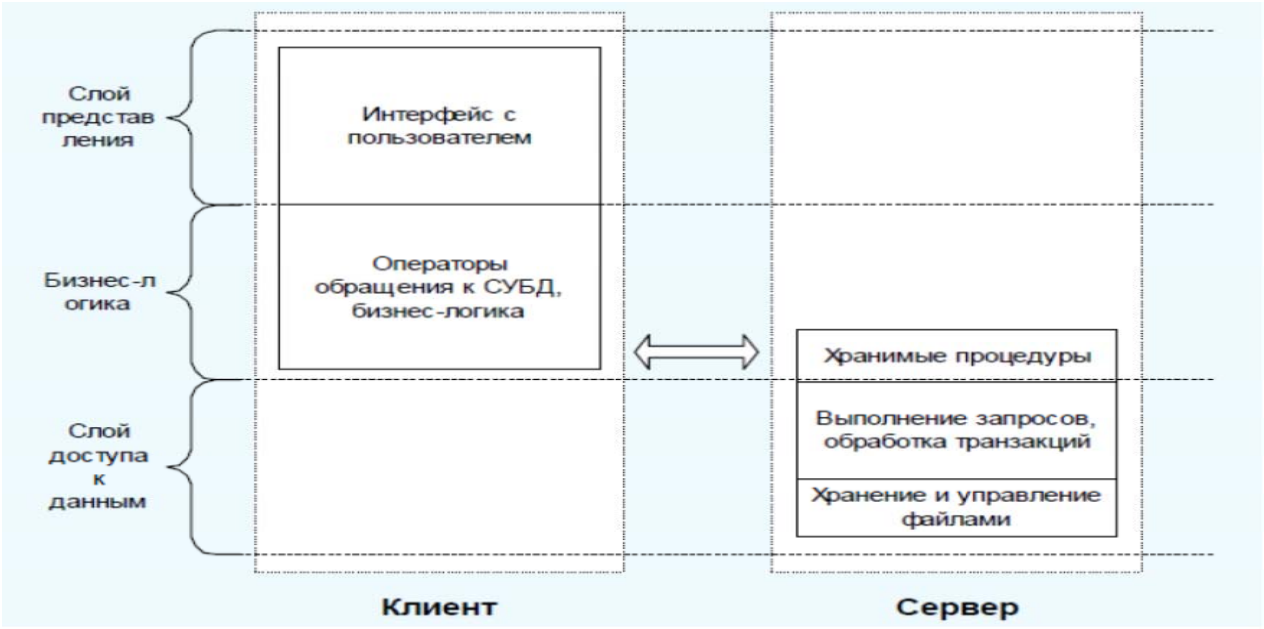

Рисунок 6 – Модель сервера СУБД

Оособенности:

- клиентская программа работает с данными через запросы к серверному ПО;
- базовые функции приложения разделены между клиентом и сервером.

Положительные стороны:

- полная поддержка многопользовательской работы;
- гарантия целостности данных.

Отрицательные стороны:

- Бизнес логика приложений осталась в клиентском ПО. При любом изменении алгоритмов, надо обновлять пользовательское ПО на каждом клиенте.
- Высокие требования к пропускной способности коммуникационных каналов с сервером.
- Слабая защита данных от взлома, в особенности от недобросовестных пользователей системы.
- Высокая сложность администрирования и настройки рабочих мест пользователей системы.
- Необходимость использовать мощные ПК на клиентских местах.
- Высокая сложность разработки системы из-за необходимости выполнять бизнес-логику и обеспечивать пользовательский интерфейс в одной программе.

Нетрудно заметить, что большинство недостатков классической или 2-слойной (2-уровневой) архитектуры клиент-сервер (рис. 6) проистекают от использования клиентской станции в качестве исполнителя бизнес-логики ИС. Поэтому очевидным шагом дальнейшей эволюции архитектур ИС явилась идея «тонкого клиента»: алгоритмы обработки данных разбивались на части, связанные с выполнением бизнес-функций и отображением
информации в удобном для человека представлении, часть, связанная с первичной проверкой и отображением информации, оставалась на клиентской машине, а вся реальная функциональность системы переносилась на серверную часть.

Переходная архитектура (2,5-слойный клиент-сервер)

Особенности:

- Использование хранимых процедур и вычисление данных на стороне сервера;
- использование систем управления базами данных (СУБД) со всеми их преимуществами;
- написание программ для серверной части, в основном, на специализированных встроенных языках СУБД, которые не позволяют написать всю бизнес-логику приложения,вследствие чего часть бизнес-логики все равно реализуется на стороне клиента;
- физически ИС состоит из двух компонентов.

Положительные стороны:

- реализация вычислений на серверной стороне и передача по сети готовых результатов вычислений, что ведет к снижению требований к скорости передачи данных между клиентской и серверной частями;
- существенное улучшение защиты информации, так как пользователям даются права на доступ к функциям системы,а не к ее данным и т. д.

Отрицательные стороны:

- ограниченная масштабируемость;
- зависимость от программной платформы;
- ограниченное использование сетевых вычислительных ресурсов;
- написание программ для серверной части системы на слабо предназначенных для этого встроенных в СУБД языках описания хранимых процедур;
- низкое быстродействие системы;
- высокая трудоемкость создания и модификации ИС;
- высокая стоимость аппаратных средств, необходимых для функционирования ИС.

Трехуровневая клиент-серверная архитектура

Основное отличие от *архитектуры 2.5* – **физическое** разделение программ, отвечающих за хранение данных (СУБД) и их обработку (сервер приложения (СП), applica-tion server (AS)). Такое разделение программных компонент позволяет оптимизировать нагрузки как на сетевое, так и на вычислительное оборудование комплекса (рис.7). **Сервер приложений** – комплекс программ, выполняемых на сервере и реализующих бизнес-логику ИС.

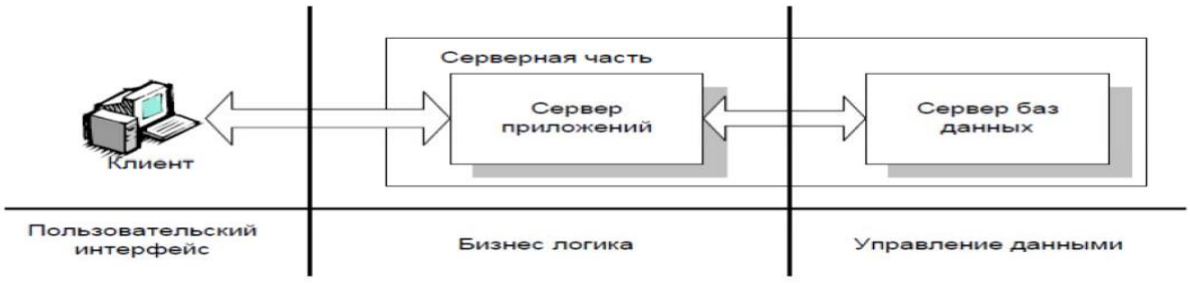

Рисунок 7 – Трехуровневый клиент-сервер

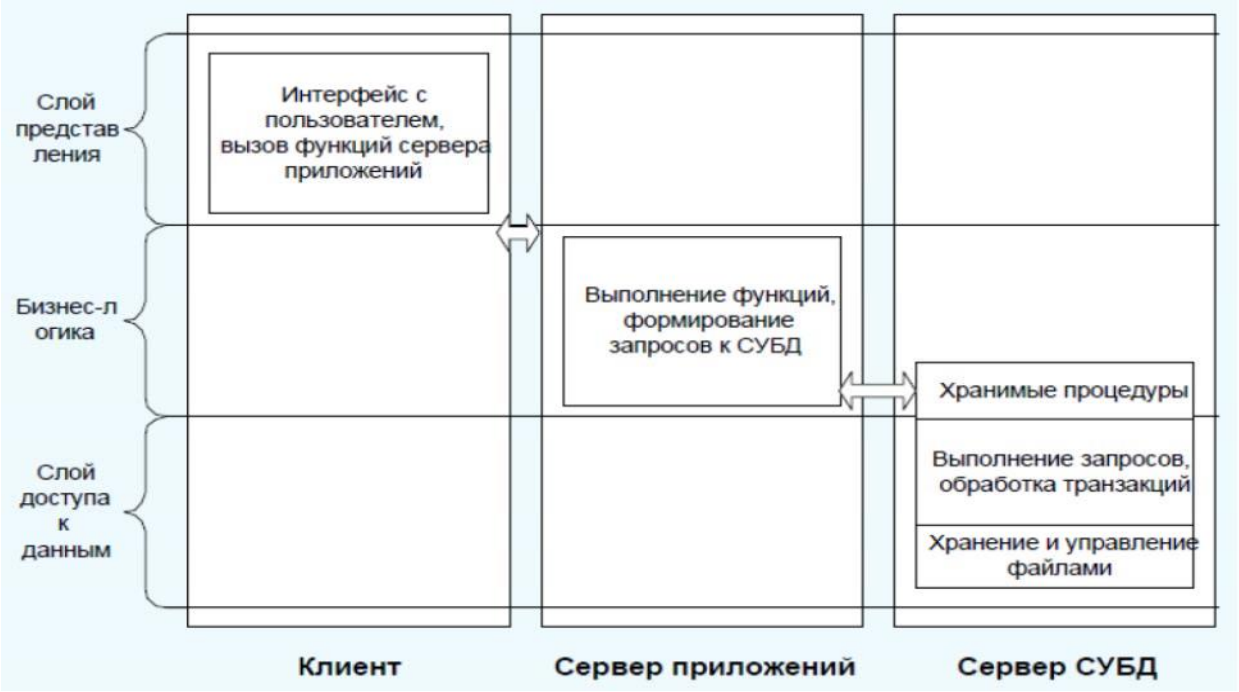

Рисунок 8 – Модель сервера приложений

Положительные стороны:

- «Тонкий клиент».
- Между клиентской программой и сервером приложения передается лишь минимально необходимый поток данных – аргументы вызываемых функций и возвращаемые от них значения. Это теоретический предел эффективности использования линий связи.
- Сервер приложения ИС (рис. 8) может быть запущен в одном или нескольких экземплярах на одном или нескольких компьютерах, что позволяет использовать вычислительные мощности организации столь эффективно и безопасно, как этого пожелает администратор ИС.
- Дешевый трафик между сервером приложений и СУБД. Трафик между сервером приложений и СУБД может быть большим,однако это всегда трафик локальной сети, а их пропускная способность достаточно велика и дешева. В крайнем случае, всегда можно запустить СП и СУБД на одной машине, что автоматически сведет сетевой трафик к нулю.
- Снижение нагрузки на сервер данных по сравнению с 2.5- слойной схемой, а значит, и повышение скорости работы системы в целом.
- Дешевле наращивать функциональность и обновлять ПО.

Отрицательные стороны:

- Выше расходы на администрирование и обслуживание серверной части.

Особенности:

Широкие возможности масштабирования. Одна и та же система может работать как на одном отдельно стоящем компьютере, выполняя на нем программы СУБД, СП и клиентской части, так и в сети, состоящей из сотен и тысяч машин. Единственным фактором, препятствующим бесконечной масштабируемости, является лишь требование ведения единой базы данных.

Упрощение расширения функциональных возможностей.

В отличие от 2,5-слойной схемы нет необходимости менять всю систему – достаточно установить новый СП с требуемой функцией.

- По сравнению с 2-слойной схемой уменьшается число проблем, связанных с переустановкой клиентских частей программы на множестве компьютеров, быть может, весьма удаленных.

Многозвенные архитектуры клиент-сервер

Многозвенные архитектуры клиент-сервер являются прямым продолжением разделения приложений на уровни пользовательского интерфейса, компонентов обработки и данных. Различные звенья взаимодействуют в соответствии с логической организацией приложения. Во множестве бизнес-приложений распределенная обработка эквивалентна организации многозвенной архитектуры приложений клиент-сервер. Такой тип распределения называется ***вертикальным (ВР)***. Характерной особенностью вертикального распределения является то, что оно достигается *размещением логически различных компонентов на разных машинах.*

В современных архитектурах распределение на клиенты и серверы происходит способом, известным как ***горизонтальное распределение (ГР)***. При таком типе распределения *клиент или сервер могут содержать физически разделенные части логически однородного модуля*, причем работа с каждой из частей может происходить независимо. Это делается для выравнивания загрузки.