УФИМСКИЙ КОЛЛЕДЖ РАДИОЭЛЕКТРОНИКИ ТЕЛЕКОММУНИКАЦИЙ И БЕЗОПАСНОСТИ.

ТЕМА ПРЕЗЕНТАЦИИ:
АРХИТЕКТУРЫ ИНФОРМАЦИОННЫХ СИСТЕМ.

Работу выполнили:
Сафонов Валерий.
Ковырзин Данил.
Студенты группы: 9КСК-101К-19

2020

АРХИТЕКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ

- ПЛАН:
- 1.Виды архитектуры информационной системы
- 2.Локальные информационные системы.
- 3.Файл-серверная архитектура.
- 4.Клиент-серверная архитектура
- 5.Трехуровневая архитектура.

.

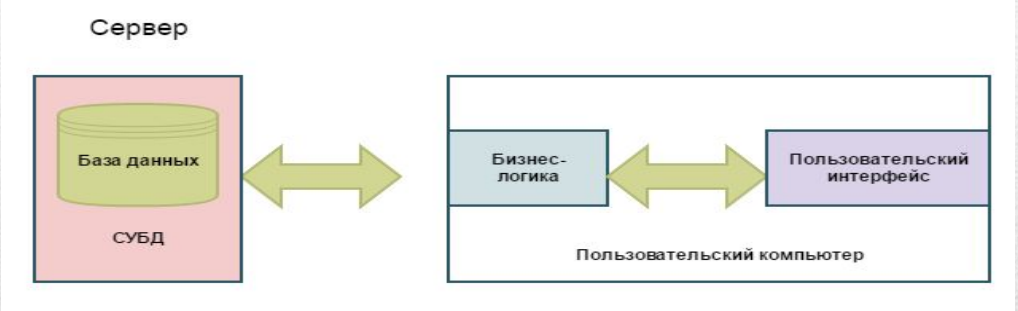

Виды архитектуры информационной системы
ЛОКАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ
Клиент-серверная архитектура
Трехуровневая архитектура
ФОТКА
ИТОГИ:

Виды архитектуры информационной системы

- Виды архитектуры информационной системы Любая информационная система (ИС) включает в себя три компонента: Управление данными; Бизнес-логику; Пользовательский интерфейс. Данные хранятся в базах данных, а управление ими осуществляется с помощью системы управления базами данных (СУБД). Бизнес-логика определяет правила, по которым обрабатываются данные. Она реализуется набором процедур, написанных на различных языках программирования. Пользователь работает с интерфейсом, где логика работы ИС представлена в виде элементов управления – полей, кнопок, списков, таблиц и т.д. Однако, эти три компонента в разных ИС взаимодействуют друг с другом различными способами. Определение 1 Архитектурой информационной системы называется концепция, согласно которой взаимодействуют компоненты информационной системы. Существуют следующие виды архитектур ИС: Локальная; Файл-серверная; Клиент-серверная; Трехслойная.

ЛОКАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ.

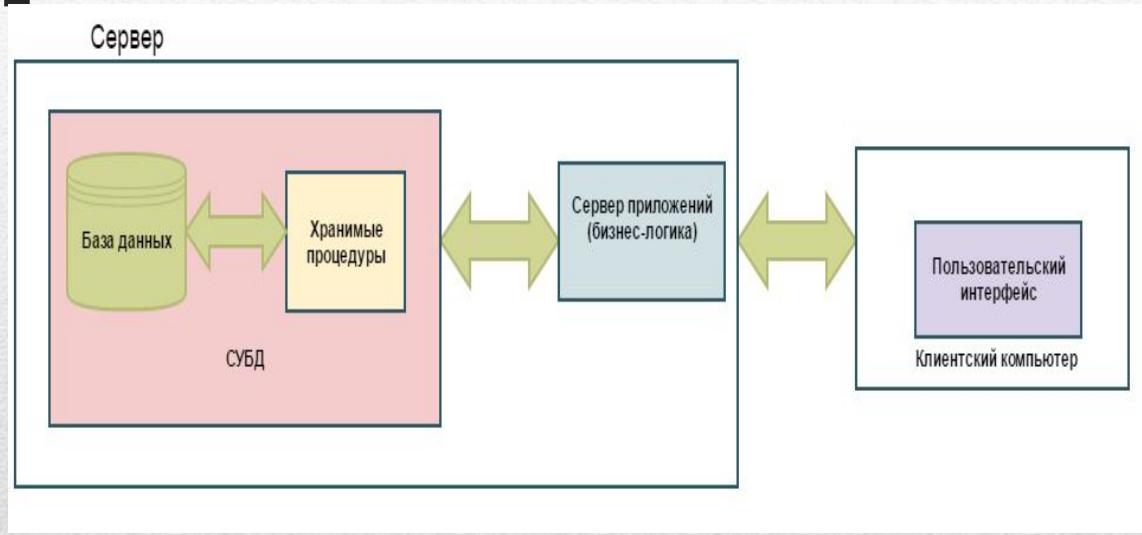
С появлением компьютерных сетей возникла возможность хранить данные в файлах на выделенном специально для этой цели компьютере. Такой компьютер называется файловым сервером или просто сервером. Компьютеры пользователей соединены с сервером сетью, поэтому доступ к данным, могут получить несколько пользователей одновременно. Однако, кроме функции хранения данных и обеспечения доступа к ним, сервер никаких функций не выполняет. Приложения, обрабатывающие данные, находятся на пользовательских компьютерах.



Предположим, что в базе данных на сервере хранится список сотрудников крупного предприятия. На предприятии 1500 сотрудников и 10 подразделений. Пользователю нужно получить число сотрудников, работающих в каждом подразделении. Для решения этой задачи пользователь должен запросить данные всех 1500 сотрудников с сервера по сети, после чего на пользовательском компьютере выполнится процедура, которая осуществит подсчет сотрудников в каждом подразделении. Результатом процедуры будет 10 строк. Таким образом, чтобы получить 10 строк придется передать по сети 1500 строк.

Клиент-серверная архитектура

- До определенного момента на СУБД возлагались лишь задачи хранения данных и организации доступа к ним. С развитием технологий в состав СУБД разработчики стали включать новый компонент – процедурный язык программирования. С его помощью в СУБД стало возможным создавать процедуры для обработки данных, которые можно вызывать повторно. Такие процедуры называются хранимыми процедурами. Наличие хранимых процедур дало возможность осуществлять некоторую часть обработки данных на сервере.



Клиент-серверная архитектура позволяет разгрузить сеть и поддерживать непротиворечивость данных за счет их централизованной обработки. Однако, языки хранимых процедур не приспособлены для полноценной реализации бизнес-логики. Поэтому бизнес-логика в клиент-серверных ИС по-прежнему реализуется на клиентских компьютерах. Такой подход имеет следующие недостатки: любые изменения в бизнес-логике требуют обновления на клиентском компьютере; клиентские компьютеры должны быть достаточно производительными; слабая защита данных от взломов.

Трехуровневая архитектура

- Все недостатки клиент-серверной архитектуры связаны с тем, что на клиентском компьютере лежит слишком большая нагрузка, которую можно было бы перенести на сервер. Поэтому дальнейшее развитие технологий двигалось в направлении переноса нагрузки с клиентских компьютеров на сервер. В дополнение к хранимым процедурам разработчики стали использовать серверные языки программирования. Это дало возможность создавать в ИС промежуточный уровень - сервер приложений.

Сервер приложений – это комплекс программ, выполняемых на сервере и реализующих бизнес-логику ИС .

Использование сервера приложений позволяет максимально разгрузить клиентские компьютеры и сделать обработку данных еще более централизованной, что повышает скорость и надежность ИС.

АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ

ИТОГИ:

- Разработчики и пользователи информационных систем, основанных на архитектуре "клиент-сервер", часто бывают неудовлетворены постоянно существующими сетевыми накладными расходами, которые следуют из потребности обращаться от клиента к серверу с каждым очередным запросом. На практике распространена ситуация, когда для эффективной работы отдельной клиентской составляющей информационной системы в действительности требуется только небольшая часть общей базы данных. Это приводит к идее поддержки *локального кэша* общей базы данных на стороне каждого клиента.
- Фактически, концепция локального кэширования базы данных является частным случаем концепции реплицированных баз данных. Как и в общем случае, для поддержки *локального кэша* базы данных программное обеспечение рабочих станций должно содержать компонент управления базами данных – упрощенный вариант *сервера баз данных*, который, например, может не обеспечивать *многопользовательский режим* доступа. Отдельной проблемой является обеспечение согласованности (когерентности) кэшей и общей базы данных. Здесь возможны различные решения – от автоматической поддержки согласованности за счет средств базового программного обеспечения управления базами данных до полного перекладывания этой задачи на прикладной уровень.
- Преимуществами данной *архитектуры* являются [12, 15]:
- возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети;
- все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов, а также на сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа;
- поддержка многопользовательской работы;
- гарантия целостности данных.
- Недостатки [12, 15]:
- неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;
- администрирование данной системы требует квалифицированного профессионала;
- высокая стоимость оборудования;
- бизнес логика приложений осталась в клиентском ПО.

Спасибо за внимание!