**Содержание**

Введение...................................................................................................................3

Конфигурация и архитектура информационной системы...............................4-9

Заключение.............................................................................................................10

Список литературы................................................................................................11

**Введение**

В основании решения большинства задач лежит обработка данных. Для облегчения обработки сведений формируются информационные системы.

Сама идея систем информации и определенные принципы их организации появилась задолго до возникновения ЭВМ. Библиотеки, архивы, адресные бюро, справочники, словари – это информационные системы. Для того, чтобы реализовать традиционные информационные системы, применялись разные картотеки, папки, журналы, каталоги библиотеки.

Обслуживание систем информации, которые реализуются на бумажном носителе, сопрягаются с большинством трудностей: чем больше информационная система, тем больше бумаги и места нужно для ее хранения, много времени затрачивается на поиск необходимой информации. Трудности появляются и во время обновления, анализа и обработке сведений.

Информационная система представляет собой совокупность средств, способов и персонала, который применяется для хранения, обработки и выдачи сведений в интересах достижения поставленной цели.

Цель работы – анализ конфигурации и архитектуры информационной системы.

Объект исследования – информационная система.

Предмет исследования – конфигурация и архитектура.

**Конфигурация и архитектура информационной системы.**

Эффективность работы информационной системы в большинстве своем зависит от ее архитектуры. На данный момент перспективной считается архитектура клиент-сервер. В довольно распространенном варианте она предполагает наличие сети компьютера и распределенной базы сведений, которая включает корпоративную базу данных и персональные базы данных. КБД будет размещено на компьютере-сервере, ПБД помещаются на компьютерах работников подразделений, которые являются клиентами корпоративной базы данных.

Если по-простому говорить – архитектура ИС является абстрактным определением, которое выявляется из определенных составных частей и приложения и каким образом данные части связываются между собой. По составным частям приложения обычно подразумевают программы или программные модули, которые выполняют отдельные, относительно изолированные задачи.

Также необходимо сказать о такой технологии, как «клиент-сервер» - это модель вычислений, которая предполагает распределение функционала обработки в многопользовательской базе данных по определенным компьютерам. Распределение исполнения функционала обработки между компьютерами происходит с применением протокола сервисных запросов, то есть один компьютер «клиент» будет запрашивать обслуживание у иного компьютера «сервера», реализующего обслуживание и отсылку его результатов «клиенту».

При разработке и внедрении на уровне серверов баз данных механизма хранимых процессов возникла концепция активного сервера БД. В этой ситуации часть функционала прикладного компонента реализуются по типу хранимых процессов, исполняемых на стороне сервера. Друга прикладная логика исполняется на клиентской стороне.

Реализация прикладного компонента на стороне сервера представляет собой – сервер приложений. Перенос функционала прикладного компонента на сервер сокращает требования к конфигурации клиентов и делает проще администрирование, однако представляет высокие требования к производительности, безопасности и надежности сервера.

Еще одной тенденцией считается все больше применение распределенных вычислений. Они будут реализоваться на основании модели сервера приложений, где сетевое приложение делится на несколько частей, каждая из которых способна исполняться на отдельном компьютере. Выделенные доли приложения будут взаимодействовать друг с другом, обмениваясь сообщениями в заранее согласованном формате.

Нынешние программы и информационные системы достигают такого уровня развития, что определение «архитектура» в применении к ним уже давно не будет удивлять. Правильно выстроить информационную систему, результативно и надежно работающую не проще, чем сконструировать и возводить нынешнее многофункциональное здание.

Архитектура ПО системы или набора систем состоит из всех важнейших проектных решений по поводу структур программы и взаимодействии среди данных структур, которые составляют системы. Проектные решения помогают обеспечить желаемый набор свойств, которые обязана поддерживать система, чтобы становиться успешной. Проектные решения предоставляют концептуальное основание для разработки системы ее поддержания и обслуживания.

Хоть термины несколько различаются, можно сказать о немалой степени схожести. К примеру, многие определения указывают на то, что архитектура связывается со структурой и поведением, а также лишь со значимыми решениями, способна соответствовать определенному архитектурному стилю, на нее воздействуют заинтересованные в ней лица и ее окружение, она будет воплощать решения на основании логики.

Архитектура ПО будет охватывать не только ее структурные и поведенческие аспекты, но и правила ее применения и интеграции с иными системами, функциональность, производительность, гибкость, надежность, способность повторного использования, полноту, экономические и технологические ограничения, а также вопрос пользовательского интерфейса.

По мере развития программных систем огромное значение приобретает их интегрированность друг с другом для выстраивания единого информационного пространства организации. Интеграция считается важным компонентом архитектуры.

Для выстраивания правильной и надежной архитектуры и грамотно спроектированной интеграции программных систем нужно четко следовать нынешним стандартам в данных сферах. Без этого огромная вероятность сделать архитектуру, которая не может развиваться и удовлетворять растущие потребности пользователей ИТ.

Как и любая классификация, архитектуры информационных систем не считаются вовсе жесткими. В архитектуре каждой определенной информационной системы зачастую можно отыскать влияния сразу нескольких архитектурных решений.

Расположение Базы Данных выявляет определенную архитектуру базы данных, и они подразделяются на:

1. Локальные.
2. Удаленные.

Для исполнения операций с локальными БД будут разрабатываться и применяться определенные локальные приложения, а для операции с удаленными БД – клиент-серверные приложения.

Расположение БД в значимой мере воздействует на разработку приложения, который обрабатывает содержащиеся в данной базе сведения.

Локальные БД находятся на том же компьютере, что и работающие с ними приложения. В данной ситуации информационная система располагает локальной архитектурой. Работа с БД осуществляется, в основном, в однопользовательском режиме. При необходимости реально запускать на компьютере иное приложение, совместно производящее доступ к данным сведениям.

Для управления совместным доступом к БД нужны специализированные средства контроля и защиты. Представленные средства способны понадобиться, к примеру, в ситуации, когда приложение старается поменять запись, которую редактирует иное приложение. Любая разновидность БД производит подобный контроль собственными методами, и обычно обладает встроенными средствами разграничения доступности.

Во время применения локальной БД в сети вероятна организация многопользовательского доступа к ней. В этой ситуации файлы БД и предназначенное для работы с ней приложения находятся на сервере сети. Любой пользователь запускает со своего компьютера данное расположенное на сервере приложение, при этом у него будет запускаться ксерокопия приложения. Подобный сетевой вариант применения локальной БД соответствует архитектуре «файл-сервер». Приложение при архитектуре «файл-сервер» также способен записываться и на каждый компьютер сети, в такой ситуации приложению отдельного компьютера должно быть известно расположение общей БД.

Архитектура «файл-сервер» в основном используется в сетях с маленьким количеством пользователей, для ее реализации подойдут персональные СУБД, к примеру, Paradox или dBase. Достоинствами данной архитектуры считаются простота реализации, и то, что приложение в основном разрабатывается в расчете на одного пользователя и не находится в зависимости от того, на каком компьютере оно будет установлено.

Необходимо также отметить достоинства архитектуры «клиент-сервер»:

1. Сокращение нагрузки на сеть, так как теперь в ней циркулирует лишь необходимая информация.
2. Рост безопасности данных, который связывается с тем, что обработка запросов всех клиентов исполняется единой программой, которая располагается на сервере. Сервер будет устанавливать общие для каждого пользователя правила применения БД, управляет режимами доступа клиентов к сведениям, запрещая совместное изменение одной записи разными пользователями.
3. Сокращение сложности клиентских приложений благодаря отсутствию в них кода, который связывается с контролем БД и разграничением доступа к ней.

К отрицательным сторонам требуется причислить:

1. Неработоспособность сервера способна делать неработоспособной целую вычислительную сеть.
2. Администрирование этой системы обязывает иметь квалифицированный персонал.
3. Высокая цена оборудования.
4. Бизнес логика приложений остается в клиентском ПО.

Для того, чтобы реализовать архитектуру «клиент-сервер» обычно применяются многопользовательские СУБД, к примеру, Oracle. Такие СУБД также будут называть промышленными, так как они помогают формировать информационную систему организации или предприятия с огромным числом пользователей.

Трехуровневая архитектура является моделью программного комплекса, которая предполагает наличие в нем нескольких компонентов: клиента, сервера приложений и сервера баз данных.

В простых конфигурациях каждый компонент или часть из него может совмещаться на одном вычислительном узле. В продуктивных конфигурациях в основном применяется выделенный вычислительный узел для сервера баз данных или кластер серверов баз данных, для серверов приложений – выделенная группа узлов вычисления, к которым напрямую будут подключаться клиенты.

В сопоставлении с клиент-серверной или файл-серверной архитектурой, трехуровневая может обеспечить в основном большую масштабируемость, более обширные возможности по обеспеченности безопасности и отказоустойчивости. Помимо этого, в сопоставлении с клиент-серверными приложениями, которые применяют прямые подключения к серверам баз данных, сокращаются требования к скорости и стабильности каналов связи между клиентом и частью сервера.

Реализация приложений, которые доступны из веб-браузера или тонкого клиента, в основном, предполагает развертывание программного комплекса в трехуровневой архитектуре. При этом в основном разработка трехуровневых программных комплексов труднее, чем для клиент-серверных приложений, также наличие дополнительного связующего ПО способно налагать новые издержки в администрировании подобных комплексов.

Для доступа к определенным сетевым сервисам применяются клиенты, возможности которых характеры определением «толщина». Оно выявляет конфигурацию оборудования и ПО, которое имеется у клиента.

**Заключение**

Таким образом, на основании проведенного анализа можно сказать, что клиент-серверная архитектура считается самой приемлемой для информационных систем и ее выбор можно считать обоснованным.

Сервис-ориентированная архитектура является концепцией построения архитектуры информационной системы из слабосвязанных между собой частей на основании сервисов – конкретных элементов с обозначенными интерфейсами, которые исполняют конкретные функции. Основное отличие данной архитектуры в том, что применение независимых сервисов с четко конкретными интерфейсами, которые для исполнения личных задач способны вызываться определенным стандартным методом, при условии, что сервисы заранее ничего не знают о приложение, вызывающее их, а приложение не знает, как сервисы исполняют собственную задачу.

**Список литературы**

1. Сайт лаборатории ММиИС [Электронный ресурс]. - Режим доступа: http://www.mmis.ru/.

2. PLANY: развернутое руководство [Электронный ресурс]. - Режим доступа:

http ://www. mmis.ru/Portals/0/Plany. pdf.

3. 1С: Университет [Электронный ресурс]. - Режим доступа: http://v8.1c.ru/solutions/ prod-uct.jsp?prod\_id= 170.

4. Резник С.Д. Управление кафедрой: Учебник. - 3-е изд., перераб. и доп. - М.: ИНФРА-М, 2015. - 606 с.

5. 1С: Предприятие 8.2. Руководство разработчика. Ч. 1, 2. - М.: Фирма «1С», 2017. - 1109 с.

6. Радченко М.Г., Хрусталева Е.Ю. 1С: Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы. - М.: ООО «1С-Паблишинг», 2016. - 872 с.

7. Габец А.П., Козырев Д.В., Кухлевский Д.С., Хрусталева Е.Ю. Реализация прикладных задач в системе «1С:Предприятие 8.2». - М.: ООО «1С-Паблишинг», 2015. - 714 с.