ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ МНОГОУРОВНЕВОЙ КЛИЕНТ-СЕРВЕРНОЙ АРХИТЕКТУРЫ ПРИ РАЗРАБОТКЕКОРПОРАТИВНЫХ ПРИЛОЖЕНИЙ

Асмоловский В.В., Мартынов Я.А.
Научный руководитель: Мартынов Я.А.
Томский политехнический университет
Email:vva1@tpu.ru

Введение

В настоящее время разработка корпоративных приложений и информационных систем для автоматизации различных бизнес-процессов является приоритетным направлением для многих компаний. Эти приложения могут охватывать всё предприятие, либо предназначаться для отдельной группы сотрудников. Несмотря на различия в функционале корпоративных приложений, процесс их разработки и сопровождения *содержит схожий набор этапов*, одним из которых является этап архитектурного проектирования всей информационной системы.

Двухуровневая архитектура

На текущий момент развития сферы информационных технологий самой популярной архитектурой для корпоративных приложений является клиент-серверная архитектура. Данная архитектура подразумевает наличие поставщика услуг – сервера, и потребителя услуг – клиента.[1] Реализация такой архитектуры в чистом виде подразумевает наличие двух уровней (Рис. 1).

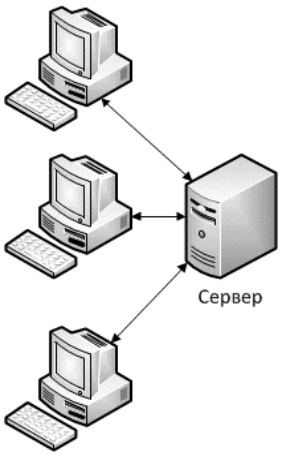

Рис. 18 – Двухуровневая клиент-серверная архитектура

В зависимости от расположения основной бизнес-логики выделяют два крайних случая данной архитектуры – «тонкий» и «толстый» клиент. Если основная часть бизнес-логики находится на клиенте, то такой клиент называют «толстым», если на сервере, то «тонким».

Трёхуровневая архитектура

В разрезе развития клиент-серверной архитектуры наличие двух уровней, практиковалось до вольно долго. Следующим этапом развития стало появление многоуровневой архитектура клиент-сервера, в рамках которой было принято решение разбить серверную часть на несколько частей, каждая из которых выполняет определённую задачу. Как частный пример такого решения можно привести самую часто используемую конфигурацию – трёхуровневую архитектуру (Рис. 2).

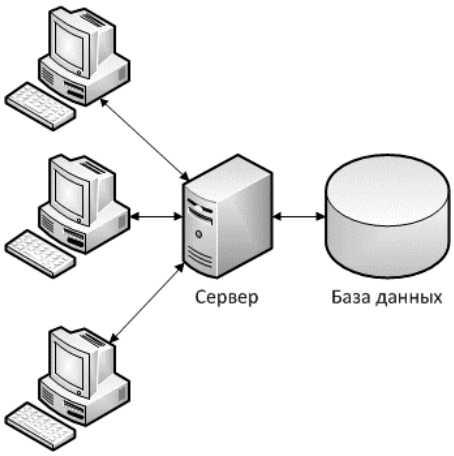

Рис. 19 – Трёхуровневая клиент-серверная архитектура

Отделение базы данных от сервера позволяет повысить производительности всей системы за счёт использования для каждого уровня своего физического узла. Также это позволяет разделить функционал, предоставляемый сервером, для более эффективного использования всех возможностей клиента и сервера.

В трёхуровневой архитектуре принято выделять следующие компоненты:

– Уровень представлений – верхний слой системы, который содержит пользовательский интерфейс и производит все операции с пользователем.

– Уровень логики – слой, который обрабатывает все команды, при необходимости производит вычисления и расчёты, также обрабатывает бизнес-логику. На этом уровне обрабатываются, и интерпретируется нужным образом данные из других слоёв.

– Уровень данных – на данном слое хранятся данные, предоставляемые по запросу из уровня логики.

Возможны разные варианты деления этих компонентов на физическом уровне, и, в простейшем случае, возможно совмещение всех уровней на одном физическом сервере. В лучшем случае должна быть возможность поместить каждый уровень на разных узлах.

Взаимодействие осуществляется только между ближайшими слоями, то есть уровень представлений не может напрямую взаимодействовать с уровнем данных и наоборот. Это позволяет лучше разграничивать компоненты и уменьшать их связанность друг с другом.

Многоуровневая архитектура

Многоуровневая архитектура выполняет дополнительное разделение.[2] Типовое решение при создании многоуровневой архитектуры представлено на рисунке 3.

Рис. 20 – Типовая многоуровневая клиент-серверная архитектура

Уровень клиента напрямую взаимодействует с пользователем и представляет собой интерфейс, реализованный для различных конечных платформ.

Уровень клиентского представления обрабатывает логику, необходимую для построения интерфейса, а также содержит общие элементы для всех платформ.

Уровень бизнес-логики выполняет основные операции, производит вычисления и расчёты.

Уровень доступа к данным обрабатывает все запросы на чтение и запись, поступающие из уровня бизнес-логики.

Уровень данных это некий внешний источник данных, например, база данных.

Представленный типовой шаблон в дальнейшем подстраивается проектировщиком системы под конкретные требования, при этом может быть изменен принцип разделения уровней, изменено их количество, а также определены физические узлы, на которых располагается каждый уровень.

Основными преимуществами такого подхода к архитектуре являются:

– *Масштабируемость* – это свойство связано с возможностью расширения каждого уровня, при этом не затрагивая другие уровни.

– *Безопасность* – позволяет обеспечить более тонкий контроль безопасности всей системы, применяя различные подходы на каждом уровне.

– *Отказоустойчивость* – использование физического разделения уровней снизить вероятность нарушения функционирования всей системы.

– *Независимое изменения уровней* – позволяет выполнять обновление или замену уровней без влияния на остальные уровни. Единственным связующим звеном между слоями являются интерфейсы.

– *Эффективность разработки и внедрения* – связано с возможностью параллельной независимой разработки нескольких уровней.

– *Повторное использование* – означает возможность использования программного кода ранее разработанной системы в новых проектах.

Несмотря на все достоинства многоуровневой архитектуры, она обладает следующими недостатками:

– *Увеличение сложности разработки* приложения по сравнению с приложением, работающим на меньшем количестве уровней.

– Повышенные требования к *аппаратному обеспечению*, *сетевому взаимодействию* и, как следствие, повышенная стоимость этих физических компонентов.

При отсутствии требуемых условий *производительность приложения* может существенно снизиться (по сравнению с отдельным приложением).

Заключение

На текущий момент клиент-серверная архитектура, а конкретно её разновидность – многоуровневая архитектура является самой популярной и востребованной при разработке корпоративных приложений, что в первую очередь связано с большим набором достоинств и преимуществ, получаемых при её использовании. Преимущества от использования многоуровнего подхода, особенно при наличие большого количества клиентов, перекрывают все недостатки.

Литература

1. Многоуровневые системы клиент-сервер [Электронный ресурс]. - Режим доступа: www.webcitation.org/61DUMtwXO, свободный.

2. Многоуровневая архитектура [Электронный ресурс]. - Режим доступа: www.sbras.ru/Report2006/Report321/node30.html, свободный.