

Архитектура программных
систем. Лекция 1
История развития архитектур

Михаил Пожидаев

28 февраля 2023 г.

Архитектура

Понятие и формирование

Понятие архитектуры программы включает в себя наиболее важные статические и динамические аспекты системы. Она зависят от:

- ▶ платформы для работы (компьютерной архитектуры);
- ▶ операционной системы, СУБД, сетевых протоколов;
- ▶ доступности готовых блоков, особенностей развертывания;
- ▶ нефункциональных требований (например, производительности и надежности).

Составные части

Из чего состоит архитектура?

Архитектура программы включает в себя данные:

- ▶ об организации программной системы;
- ▶ о структурных элементах, входящих в систему, и их интерфейсах, а также их поведении, которое определяется кооперациями, в которых участвуют элементы;
- ▶ о составе структурных элементов и элементов поведения наиболее крупных подсистем;
- ▶ о стиле архитектуры, принятом в данной организации, — элементах и их интерфейсах, их кооперации и композиции.

Типы архитектур

Архитектура — многозначный термин

Архитектурой могут называть:

- ▶ архитектуру процессора;
- ▶ архитектуру информационной системы;
- ▶ архитектуру операционной системы;
- ▶ архитектуру сетевых коммуникационных сервисов и пр.

IBM System/360

Основоположник понятия архитектуры

Компьютер IBM System/360, представленный в 1964 г., заложил ясное разделение на архитектуру и её реализацию. Клиент, создав программное обеспечение для архитектуры, мог сравнительно легко запускать его на различных машинах соответствующей серии. Понятие архитектуры стало одновременно обозначать принципы и совместимость. Часть технических особенностей IBM System/360 оказало влияние на всю современную вычислительную технику.

История частично повторилась в 1981 г., когда был представлен персональный компьютер IBM PC, определивший понятие архитектуры и ставший стандартом настольных ПК.

Мейнфреймы

Как пример терминальной системы

Мейнфрейм — высокопроизводительный сервер с повышенными возможностями отказоустойчивости. Являются исторически излюбленным инструментом для построения информационных систем в финансовом и банковском секторах. Отличительная особенность — минимальный вариант понятия клиента, роль которого сводится нередко к терминальному решению или к доступу через веб.

История мейнфреймов началась в 60-ые годы XX в. Популярность стала падать в 90-ые годы, но полностью они не исчезли до сих пор.

Достоинства и недостатки

Плюсы/минусы мейнфреймов

Достоинства мейнфреймов:

- ▶ выдающаяся отказоустойчивость;
- ▶ простота реализации транзакций из-за отсутствия распределённости вычислений;
- ▶ огромный накопленный опыт.

Недостатки мейнфреймов:

- ▶ слабый интерфейс пользователя;
- ▶ невозможность создания географически глобальных сервисов;
- ▶ использование устаревших технологий и языков программирования.

Клиент-серверная архитектура

Расширение роли клиента

Мейнфреймы отвечали требованиям больших корпораций, но потребности малых и средних компаний привели к повышению доступности программных систем. На месте терминальных клиентов появились клиентские настольные системы, которые отличались от сервера только доступными вычислительными ресурсами, но не отличались качественно.

В архитектуре появилось понятие протокола, которое позволяло определить правила общения между сервером и настольными системами пользователей. Коллектив для обслуживания мейнфрейма заменился одним или несколькими системными администраторами.

Надёжность стала ниже, но удобство выше.

Информационную систему могут теперь позволить сравнительно небольшие организации.

Достоинства и недостатки

Плюсы/минусы клиент-серверной архитектуры

Достоинства:

- ▶ высокая доступность для небольших организаций;
- ▶ высокое удобство пользователя;
- ▶ лёгкая тиражируемость решения.

Недостатки:

- ▶ критическая роль сервера;
- ▶ принципиальная ограниченность масштабирования;
- ▶ заметная привязка к аппаратному обеспечению.

Все равны

Одноранговая архитектура

Критическая роль сервера в клиент-серверной архитектуре привела к поиску решений, не имеющих точек отказа или имеющих, но легко восстанавливаемых при утрате. Это породило класс одноранговых систем, которые стали применяться в самых разных областях:

- ▶ распределённые хеш-таблицы и BitTorrent;
- ▶ криптовалюты и блокчейн;
- ▶ различные пиринговые сети.

Достоинства и недостатки

Плюсы/минусы одноранговой архитектуры

Достоинства:

- ▶ автономность функционирования по условиям самоуправления;
- ▶ многократное дублирование данных;
- ▶ сложность злонамеренного использования.

Недостатки:

- ▶ необходимость решения проблем рассинхронизации данных;
- ▶ сложность внесения оправданных изменений;
- ▶ большой потенциал использования в недобросовестных целях.

Выделение сервисов

Сервис-ориентированная архитектура

Привязка клиент-серверной архитектуры к оборудованию и ограниченность производительности мощностью сервера привели к поиску возможности ввести распределение ответственности. Началось выделение сервисов, ответственность которых ограничивалась одной конкретной задачей, решаемой на удалённой системе с предоставлением свободы выбора аппаратной реализации. Резко возрасла сложность реализации бизнес-логики и появилась необходимость разработки протоколов удалённого вызова процедур, но корпоративные системы стали намного производительнее.

Достоинства и недостатки

Плюсы/минусы сервис-ориентированной архитектуры

Достоинства:

- ▶ снижение критичности сервера;
- ▶ снижение привязки к аппаратному обеспечению;
- ▶ рост потенциальной производительности.

Недостатки:

- ▶ резкое возрастание сложности информационных систем;
- ▶ возрастание сложности администрирования информационной системы;
- ▶ появление тяжеловесных компонентов: брокера, протокола удалённого вызова процедур и пр.

Уменьшение сервисов

Микросервисная архитектура

Ясное осознание сильных сторон сервис-ориентированной архитектуры и желание её развития потребовали работы над уменьшением недостатков. Это движение наложилось на развитие технологий контейнерной виртуализации. В результате была сформулирована микросервисная архитектура, которая во многом сохраняет основные принципы сервис-ориентированной архитектуры, но вводит дополнительные правила:

- ▶ сервисы должны быть предельно минимальными;
- ▶ не допускается дублирование функциональности;
- ▶ строго одноранговая компоновка сервисов.

Достоинства и недостатки

Плюсы/минусы микросервисной архитектуры

Достоинства:

- ▶ хороший потенциал горизонтального масштабирования;
- ▶ разработка сервисов отдельными командами;
- ▶ возможность разработки географически распределённых сервисов.

Недостатки:

- ▶ рост сложности брокера и утилит оркестрации контейнеров;
- ▶ сложность поиска причин сбоев из-за крупного масштаба информационной системы;
- ▶ ввод дополнительных специалистов для развёртывания системы (devops).

Спасибо за внимание!

Всё о курсе: <https://marigostra.ru/materials/Architecture.html>

E-mail: msp@luwrain.org

Канал в Телеграм: @MarigostraRu