

Архитектура программных
систем, лекция 2.
Клиент-серверная архитектура

Пожидаев М. С.

7 марта 2023 г.

Сервер

Центральный узел архитектуры

Сервер — компьютер или виртуальный хост для обработки запросов клиентских систем. Обладает следующими характеристиками:

- ▶ предполагает повышенные вычислительные ресурсы, т. к. от их количества зависит множество клиентских систем;
- ▶ не предполагает работу пользователей, кроме администраторов;
- ▶ представлен в единственном варианте или в виде кластера из нескольких соединённых систем.

Клиент

Множество узлов для работы пользователей

Клиент — компьютер или приложение для работы пользователей. Обладает следующими характеристиками:

- ▶ функционально содержит реализацию только интерфейса пользователя, перекладывая основную работу на сервер, с которым поддерживает постоянное подключение;
- ▶ множество клиентов определяется вычислительными мощностями сервера и пропускной способностью канала подключения к нему;
- ▶ может иметь ограниченные вычислительные ресурсы, но становится несамостоятельным при утрате подключения к серверу.

Протокол

Соглашение для связи клиента и сервера

Критический компонент

Протокол выполняет критическую функцию поддержания согласованного взаимодействия всех клиентов и сервера. Нарушения его поддержки немедленно приводят к разрушению всей архитектуры.

1. Требуется относиться предельно тщательно к проектированию протоколов, максимально отдаляя момент его обновления.
2. Протокол не должен быть перегруженным, поскольку это приводит к исчерпанию ресурсов каналов связи.
3. Фиксированный протокол позволяет проводить обновления клиентского и серверного ПО независимо друг от друга.

Применение

Примеры использования клиент-серверной архитектуры

1. Сервисы со множественными реализациями:
 - ▶ HTTP: браузеры и веб-серверы;
 - ▶ SMTP и электронная почта;
 - ▶ DNS и серверы имён.
2. Сервисы с фиксированной реализацией:
 - ▶ большинство SQL-серверов: PostgreSQL, MySQL и др.;
 - ▶ Закрытые специализированные системы, протоколы компьютерных игр.

Составные протоколы

Построение сложных систем

1. Архитектура может расширяться до случаев, когда в роли клиентов выступают другие серверы (SMTP, DNS).
2. Архитектура может расширяться до случаев, когда один протокол использует для передачи данных другой протокол (RESTful службы поверх HTTP).

Однопоточный сервер

Все запросы в одном потоке

Плюсы:

- ▶ отсутствие конкурентного доступа к данным;
- ▶ высокая вычислительная устойчивость.

Минусы:

- ▶ задействование только одного ядра процессора;
- ▶ задержка ответа клиентам при «тяжёлых» запросах.

Многопоточный сервер

Каждый запрос имеет свой поток

Плюсы:

- ▶ вычислительная эффективность с задействованием всех ядер процессора;
- ▶ быстрый отклик на запросы клиентов.

Минусы:

- ▶ сложность проектирования с высоким уровнем конкурентного доступа к данным;
- ▶ сложность расчёта оптимальных параметров системы.

Спасибо за внимание!

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

E-mail: msp@luwrain.org

Канал в Телеграм: @MarigostraRu