

Архитектура программных  
систем. Лекция 2  
Распределённые и одноранговые системы

Михаил Пожидаев

27 февраля 2024 г.

# Распределённые системы

## *Ключевые особенности обобщённого понятия*

Под распределёнными системами понимается класс вычислительных систем, которые используют для хранения и обработки информации динамически изменяемое множество узлов.

### **Не одноранговая архитектура!**

Распределённые системы имеют узел-контроллер, которого нет в одноранговых системах.

### **Не микросервисная архитектура!**

Вычислительная модель в распределённых системах обычно однородная и больше ориентирована на однотипные операции сортировки и поиска.

# Составные части

## *Компоненты и типы распределённых вычислений*

Распределённая система состоит из:

- ▶ управляющего узла-контроллера, отказ которого приводит к неработоспособности системы;
- ▶ множества узлов-вычислителей, отдельные отказы которых не приводят к неработоспособности системы.

Типы распределённых вычислений:

- ▶ вычисления с сохранением промежуточных данных на диск;
- ▶ вычисления в памяти (in memory).

# Плюсы

## *Достоинства распределённых систем*

1. Способность обрабатывать огромные массивы слабоструктурированных данных.
2. Низкая чувствительность к отказу отдельного вычислительного узла, кроме контролирующих.
3. Простота решения задач сортировки и поиска информации.
4. Возможность географического распределения данных.

# Минусы

## *Недостатки распределённых систем*

1. Наличие критического узла-контроллера., нарушение работы которого приводит к утрате всей системы.
2. Высокая потребность в вычислительных мощностях из-за многократного дублирования хранимой информации.
3. Ограниченность вычислительных задач, выполняемых с хранимыми данными.

# Apache Hadoop

## Платформа распределённых вычислений

- ▶ *HDFS*: распределённая файловая система для хранения сверхбольших массивов данных. Не поддерживает изменения файлов.
- ▶ *MapReduce*: фреймворк для распределённых вычислений.

# MapReduce

## *Порядок вычислений*

Рассмотрим алгоритм подсчёта всех слов в тексте с использованием MapReduce:

1. Map: для каждого слова в тексте формируется пара ключ-значение, в которой в ключ записывается слово, а в значение единица.
2. Получившиеся пары группируются в блоки с одинаковым ключом.
3. Reduce: каждый блок редуцируется до новой пары, состоящей из слова и количества исходных пар.

# Apache Spark

## *Распределённые резидентные вычисления*

1. Способна обрабатывать слабоструктурированные данные.
2. Способна работать в потоковом режиме.
3. Предлагает отклик заметно лучше, чем в обычном MapReduce.
4. Поддерживает некоторые алгоритмы машинного обучения.



# Одноранговые системы

## *Основные характеристики и особенности*

### Общая черта

Все узлы системы играют в ней одинаковую роль, и отсутствует управляющий или координирующий узел. Обычно разрабатываются не от хорошей жизни.

1. Обладают повышенной отказоустойчивостью и защищённостью от недобросовестного администрирования.
2. Нередко подразумевают многократное дублирование одного и того же функционала (особенно систем хранения).
3. Требуют проработки глубоких механизмов согласованной работы.

# Блокчейн

## *Криптовалюты, смарт-контракты и NFT*

1. Все узлы хранят полную историю операций (цепочку блоков).
2. Каждый блок содержит хэш-сумму предыдущего блока.
3. Попытка добавления нового блока требует одобрения определённого количества других участников.
4. Требуется наличие механизма защиты от добавления несогласованных блоков.

# Биткойн

## Блоки с «денежными» транзакциями

Блок включает:

- ▶ информацию о переводе средств;
- ▶ хэш-сумму предыдущего блока;
- ▶ специальную «добавку» для корректировки хэш-суммы нового блока.

### Майнинг

Вычисление «добавки» требует ресурсов и времени.

Введено для ограничения скорости порождения блоков и предполагает вознаграждение за проделанную работу.

# Ethereum

## Смарт-контракты и NFT

### Почему только деньги?

Вместо криптовалюты могут использоваться любые другие цифровые объекты, которые принято называть токенами. Пользователи могут их создавать и передавать. Что и для чего — решает сам пользователь.

1. Правила передачи могут определяться самими пользователями (смарт-контракты).
2. Оказалась возможной доселе невиданная вещь — подтверждение права владения цифровым объектом.
3. Безграничный простор для перевода объектов реального мира в цифровой.

# Теорема CAP

## *Эмпирическое наблюдение*

Любая распределённая система будет удовлетворять только двум любым требованиям из списка:

- ▶ согласованность (consistency);
- ▶ доступность (availability);
- ▶ устойчивость к разделению (partitions tolerance).

# DHT

## Распределённая хеш-таблица

DHT является примером одноранговой системы со следующими ключевыми характеристиками:

1. Практическое отсутствие ограничений масштабирования.
2. Выдающаяся отказоустойчивость, практически недостижимая в клиент-серверной и микросервисной архитектурах.

*Обычно применяется для отказоустойчивого хранения информации (распределённые СУБД, пиринговые системы и пр.).*

# Ключи записей

## *Формирование адресного пространства*

### Область близости

Пространство ключей позволяет формировать множество близлежащих узлов, логическое расстояние до которых невелико.

### Логическое и физическое пространство

Множество логических адресов отображается во множество физических, причём близлежащие логические узлы могут находиться далеко друг от друга в физическом пространстве.

# План Б

## *Дублирование и восстановление записей*

### Запись

В процессе записи данных в DHT данные записываются не только на узел с нужным ключом, но сразу же и на близлежащие узлы (BitTorrent делает это в процессе скачивания).

### Чтение

В процессе чтения при недоступности необходимого узла требуется проверить близлежащие узлы, которые могут подсказать, где искать данные.



# CAD

## *Content addressable storage*

### Контент вместо адреса

Для построения адреса хранимого блока можно использовать содержимое блока после обработки хеш-функцией.

### Свойства:

- ▶ неизменяемость данных;
- ▶ изменение блока подразумевает получение нового адреса (хотя старый блок можно не удалять);
- ▶ гармоничное сочетание с DHT.

# Архитектура

## Когда применять DHT

1. Требуется особая отказоустойчивость без точек отказа.
2. Существует возможность формирования пространства ключей.
3. Система ориентирована на хранение информации.
4. Доступно большое количество вычислительных ресурсов. Их количество важнее, чем уровень ожидаемой доступности.

Спасибо за внимание!

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

E-mail: [msp@luwrain.org](mailto:msp@luwrain.org)

Канал в Телеграм: @MarigostraRu