

架构篇

架构设计原则

- 单一职责原则
- 开放封闭原则
- 里氏替代原则
- 依赖倒置原则
- 接口分离原则

分布式

- 分布式理论
  - 2PC
  - 3PC
  - CAP
    - CAP 理论的证明
    - 如何在 CA 之间做选择
  - BASE
- 可用性
  - SLA指标
  - 心跳检测
- 分布式协调 Zookeeper
  - 基本概念
  - 常见用法
  - ZAB算法
  - 脑裂
- 分布式事务
  - 2PC 和 3PC
  - 本地事务&分布式事务
  - 可靠消息最终一致性
  - 最大努力通知
  - TCC
- Dubbo
  - 服务注册
  - 服务发现
  - 服务治理
  - 优雅上下线
- 分布式数据库
  - 怎样打造一个分布式数据库
  - 什么时候需要分布式数据库
  - mycat
  - otter
  - HBase
- 分布式文件系统
  - mfs
  - fastdfs
- 分布式缓存
  - 本地缓存与分布式缓存
    - Guava
    - Caffeine
    - 多级缓存
  - 为什么使用分布式缓存
  - 缓存一致性
  - 缓存命中率
  - 缓存冗余
  - 缓存穿透
  - 缓存击穿
  - 缓存雪崩
  - 缓存预热
  - Redis
    - 为什么单线程也能这么快?
    - Redis 6.0的多线程
    - 支持的数据结构
    - 数据存储的结构
    - 过期策略和内存淘汰机制
  - 如何保证缓存与数据库的一致性
- 限流降级
  - 熔断器模式
  - Hystrix
  - Sentinal
  - resilience4j
- 分布式算法
  - 拜占庭问题与算法
  - 2PC
  - 3PC
  - 共识算法
  - Paxos 算法与 Raft 算法
  - ZAB算法

- 实体、值对象
- 聚合、聚合根
- 限界上下文
- DDD如何分层
- 充血模型和贫血模型
- DDD和微服务有什么关系

微服务

- SOA
- 康威定律
- ServiceMesh — sidecar
- Docker & Kubernetes
- Spring Boot
- Spring Cloud

- 分库分表
  - 横向拆分与水平拆分
  - 分库分表后的分布式事务问题
- CDN技术
- 消息队列
  - RabbitMQ、RocketMQ、ActiveMQ、Kafka
  - 各个消息队列的对比
  - 消息队列应用场景
    - 异步处理
    - 应用解耦
    - 流量削峰
    - 依赖可靠消息实现最终一致性
    - 怎么保证消息不丢失
    - 消息的幂等处理

高并发

高可用

- 双机架构
  - 主备复制
  - 主从复制
  - 主主复制
- 异地多活
- 预案
- 预热
- 限流

高性能

- 高性能数据库
  - 读写分离
  - 分库分表
- 高性能缓存
  - 缓存穿透
  - 缓存雪崩
  - 缓存热点
- 负载均衡
- PPC、TPC

监控

- 监控什么
  - CPU
  - 内存
  - 磁盘I/O
  - 网络I/O
- 监控手段
  - 进程监控
  - 语义监控
  - 机器资源监控
  - 数据波动
- 监控数据采集
  - 日志
  - 埋点
- Dapper

负载均衡

- 负载均衡分类
  - 二层负载均衡
  - 三层负载均衡
  - 四层负载均衡
  - 七层负载均衡
- 负载均衡工具
  - LVS
  - Nginx
  - HAProxy
- 负载均衡算法
  - 静态负载均衡算法
    - 轮询
    - 比率
    - 优先权
  - 动态负载均衡算法
    - 最少连接数
    - 最快响应速度
    - 观察方法
    - 预测法
    - 动态性能分配
    - 动态服务器补充
    - 服务质量
    - 服务类型
    - 规则模式

CDN

- 数据一致性

搜索引擎

- 倒排索引
  - 什么是倒排索引
  - 为什么搜索引擎使用倒排索引
- Solr
- Lucene
- Nutch
- Elasticsearch