Java进阶知识

- 分布式消息队列
  - 系统间通信方式
    - 一种方式
      - MQ
    - 文件
    - Socket
    - 数据库
    - RPC
  - 消息队列概念
    - 从队列到服务
      - 消息队列作用
        - 异步通信
        - 流量削峰
        - 解耦事件
        - 可靠通信
      - 消息模式
        - 点对点 Queue
        - 订阅发布 Topic
        - 消息传递模式
          - At Most Once
          - At Least Once
          - Exactly Once
          - 二种消息保障
            - 通过确认机制实现最多投递
            - 可以被事务管理保障，甚至可以实现EA
            - 有序性：同一个Topic或Queue的消息，保障按顺序投递
            - 消息协议
            - JMS
            - 消息队列通用协议
            - 并发消费中间件
              - ActiveMQ/RabbitMQ
              - Kafka/RocketMQ
              - Apache Pulsar
        - 消息模式与消费协议
          - 消息处理事务性
            - 可以被事务管理保障，甚至可以实现EA
            - 消息可靠性
            - 消费者Group/
            - Offset的存储位
            - Offset的手动提交
            - Offset自动提交
            - Offset Seek
          - 消费者
          - 消费
          - 生产者
            - 核心模式
            - 同步发送
            - 异步发送
            - 单向异步
            - 消息可靠性保证
            - 高吞吐量
            - 顺序性
            - 事务性
            - 消息可靠性保证
            - 高吞吐量
            - 顺序性
            - 事务性
          - 高容错性
          - 高可用性
          - 部署结构
            - Topic/Partition
            - Partition的Replica
            - Topic
          - 特性
          - 部署结构
          - Kafka
            - 基本概念
              - Broker
              - Topic
              - Partition
              - Producer
              - Consumer
              - ConsumerGroup
            - 主要功能
            - ActiveMQ
            - 第一版本：内存MQ
            - 第二版本：基于JDBC的store
            - 第三版本：基于zookeeper和文件的store
            - 第四版本：功能完整的MQ
            - 第五版本：体系完整的MQ
            - 功能不如MQ
            - 其他MQ产品
              - RabbitMQ
              - RocketMQ
              - Pulsar
              - EIP
- 分布式缓存
  - 缓存基础
    - 数据类型
    - 缓存加速时机
      - 启动全量加载
      - 懒加载
      - 同步更新加载
      - 延迟异步加载
      - 缓存有效性与数据同步
        - 读写穿
        - 写透
        - 缓存命中率
        - 一致性高的数据，不适用缓存
        - 变动频率大，一致性要求高
        - 写多读少
        - 预热启动缓
        - 缓存不当导致的问题
        - 系统性能资源耗尽
        - 对数据一致性、性能、成本综合衡量引入缓存的指标
    - 缓存实现
      - 本地缓存
        - 简单的本地缓存实现
        - 常见框架的本地缓存
          - Hibernate
          - MyBatis
          - Guava
          - Spring
      - 远程缓存
        - 本地缓存问题
        - 常见远程缓存
          - Redis
          - memcached
          - HazelCast
          - 内存网格
          - Ignite
    - 缓存策略
      - 容量
      - 过期策略
        - FIFO
        - LRU
    - 缓存穿透问题
      - 缓存击穿
        - 大量并发查询不存在的KEY，导致请求直接压力透传到数据库
        - 解决
        - 缓存空值的KEY，这样第一次不存在也会被加载，但记录，下次查询就了KEY
        - Bloom过滤器(bearingBloop 判断KEY是否存在
        - 解决
        - 缓存雪崩
        - 缓存大面积失效，使用 延迟淘汰策略 的策略2，这样用不会触发更新
        - 缓存雪崩
        - 当某一时刻发生大规模的缓存失效的情况，会有大量的请求涌来直接打到数据库，导致数据库压力无力直接扛住。
        - 解决
        - 更新策略是对缓存上数据过期时间
        - 使用的热数据尽量分散到不同的机器上
        - 多台机器做主从复制或者多副本，实现高可用
        - 实现细粒度流机制，对系统进行容错能力控制
  - 分布式锁
    - 基本功能
    - 几大使用场景
      - 业务数据操作
      - 业务数据处理
      - 业务一致计数
      - 高并发计数
      - 分布式锁
      - 发布订阅与Stream
    - Redis
      - Redis使用
        - Jedis
        - Lettuce
        - Redisson
        - 客户端
        - Spring整合
        - MyBatis整合
        - Redis高级功能
        - 事务
        - 管道技术
        - 数据备份与恢复
        - 性能优化
        - 分区
        - Redis高可用
        - 主从复制
        - Sentinel高可用
        - Cluster分片
        - Sublimpic 6
- RPC和微服务
  - RPC基本原理
    - RPC是干什么
      - 核心机制
      - 实现原理
        - 序列化
        - 网络传输
        - 数据反序列化
        - 管理要点
        - 设计
        - 代理
        - 异步化
        - 网络传输
        - 数据反序列化
    - RPC技术框架
      - 常见RPC技术
      - 设计一个RPC框架
      - 分布式业务间调用要考虑的点
      - JdPC到分布式服务化
        - 典型的分布式框架
        - RPC性能测试实例
      - Dubbo技术
        - Dubbo介绍
        - Dubbo功能
        - Dubbo技术原理
        - Dubbo应用场景
        - Dubbo最佳实践
    - 微服务
      - 微服务发展历史
      - 分布式微服务化与SOA的区别
      - 配置/注册/发现中心
        - 配置中心
        - 注册中心
        - 元数据中心
        - 如何实现HA/高容中心
        - 服务注册
        - 服务发现
        - 服务高容
        - 服务发现
        - 服务高容
        - 服务过滤
        - 服务高容
        - 服务过滤与流控
        - 服务高容
        - 服务过滤与流控
        - 多语言
        - 多语言
        - 服务间通信
        - 微服架构
        - 微服架构
        - 微服架构