

分布式锁选型

分布式锁出现背景

现在越来越多的项目使用分布式架构，JVM层面的Synchronized和ReentrantLock只能保证单个JVM的线程安全，当大量的并发访问出现，请求会被分配到不同的节点上，同一时刻还是会有多个线程操作共享资源，造成数据不一致，这时就需要分布式锁来保证数据一致性。本篇文章我们就来介绍一下分布式锁的选型，以及主流分布式锁的应用。

分布式锁的核心原理是在一块共享的存储区域，存一个锁的标志，然后对Lock接口功能进行实现，提供tryLock, lock, unLock等功能，主流的几种分别是基于MySQL实现，基于Redis实现，基于Zookeeper实现的，下面我们一起探究下每种实现的优缺点，以及应用。

基于MySQL（数据库）实现

基于MySQL（数据库）实现分布式锁

优势

1. 无需引入新的中间件，基本所有的服务都会依赖于MySQL（数据库）
2. 实现简单，加锁方便，只需要使用select ... for update即可加锁
3. 增加版本号字段，还可以基于数据库实现乐观锁形式的分布式锁

劣势

1. MySQL很难抗住高并发
2. 并发冲突容易引起数据库死锁
3. MySQL一般存在单点问题
4. 无法自动释放锁，可能还需借助额外定时任务来释放锁

应用

很多常见的框架都是通过MySQL来实现分布式锁，比如xxl-job, liquibase等，在日常业务中不是很建议使用基于MySQL的分布式锁

基于Redis实现

优势

1. Redis高性能，能抗住高并发
2. Redis单线程，不会出现加锁阻塞，死锁问题
3. Redis一般都是集群化部署，没有单点问题
4. Redis支持过期时间，可以自动释放锁
5. Redis支持发布订阅模式，可用于释放锁的通知
6. Redis丰富的数据结构，能够很方便的对锁进行扩展使其支持可重入锁
7. Redis支持Lua脚本，可以保证原子性
8. 有Redisson这种开箱即用的客户端框架

劣势

1. Redis的锁无法实现自动续期，业务没有执行完，过期时间到，锁会被释放（Redisson Watch dog可解决）
2. Redis主从复制，可能会出现数据不一致（RedLock可以保证）
3. RedLock 可靠性高，成本高
4. Redisson使用通知机制释放锁，等待的客户端多，会出现惊群效应，导致瞬时流量大
5. Redisson支持公平锁和非公平锁
6. Redis集群出现脑裂，也会造成锁的判断失误
7. Redis集群出现时钟漂移

应用

基于Redis实现的分布式锁，是目前最为主流的分布式锁形式，随着硬件价格的下调，以及大家对软件性能的要求越来越高，Redis也和MySQL（数据库）一样成为了每个应用的标配，于此同时Redis丰富的数据结构和超高的性能也让其成为分布式锁领域的有力竞争者，并且Redis支持设置过期时间，支持分布式，可以说是完美的弥补了MySQL实现分布式锁的不足，加上Redisson对分布式锁实现的封装，让我们可以开箱即用，避免很多问题，现在大部分的应用都是使用Redisson封装的基于Redis的分布式锁，在一些可靠性要求高，集群化Redis部署，可以使用Redisson提供的RedLock。

基于Zookeeper实现

优势

1. 具有很好的一致性，因为Zookeeper是CP的设计
2. 临时节点，使用结束，支持自动释放
3. Zookeeper一般是集群部署，没有单点故障
4. Zookeeper临时节点是基于心跳保持的，心跳不停就可以自动续约
5. 有curator这种开箱即用的客户端框架
6. Zookeeper是公平锁

劣势

1. 大部分项目不依赖Zookeeper，只用作分布式锁有浪费
2. Zookeeper在处理大规模并发时，性能不行

应用

基于Zookeeper实现，一般是对可靠性要求高，并发低的项目使用，也要考虑引入新中间件的成本。

> 感觉有帮助的朋友可以点赞、收藏加，感谢！

原文链接: <https://juejin.cn/post/7370901494595485711>