

suolu

公告

昵称: suolu
园龄: 1年
粉丝: 3
关注: 0
+加关注

<	2018年4月					>
日	一	二	三	四	五	六
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Go(9)
GC(8)
Kubernetes(5)
mysql(5)
Docker(4)
nginx(2)
zookeeper(2)
分布式锁(1)

博客园 首页 新随笔 联系 管理 订阅 XML

随笔 - 34 文章 - 0 评论 - 3

分布式锁的一点理解

以下内容是目前自己理解的总结，如有错误请大家指正。

什么是锁

- 在单进的系统程中，当存在多个线程可以同时改变某个变量（可变共享变量）时，就需要对变量或代码块做同步，使其在修改这种变量时能够线性执行消除并发修改变量。
- 而同步的本质是通过锁来实现的。为了实现多个线程在一个时刻同一个代码块只能有一个线程可执行，那么需要在某个地方做个标记，这个标记必须每个线程都能看到，当标记不存在时可以设置该标记，其余后续线程发现已经有标记了则等待拥有标记的线程结束同步代码块取消标记后再去尝试设置标记。这个标记可以理解为锁。
- 不同地方实现锁的方式也不一样，只要能满足所有线程都能看得到标记即可。如java中synchronize是在对象头设置标记，Lock接口的实现类基本上都只是某一个volatile修饰的int型变量其保证灭个线程都能拥有对该int的可见性和原子修改，linux内核中也是利用互斥量或信号量等内存数据做标记。
- 除了利用内存数据做锁其实任何互斥的都能做锁（只考虑互斥情况），如流水表中流水号与时间结合做幂等校验可以看作是一个不会释放的锁，或者使用某个文件是否存在作为锁等。只需要满足在对标记进行修改能保证原子性和内存可见性即可。

分布式

分布式情况

事物(1)
索引(1)
更多

随笔分类

Go(9)
Java(1)
JVM(8)
Linux(6)
nginx(2)
并发(1)
分布式(2)
数据库(5)

随笔档案

2017年9月 (1)
2017年5月 (3)
2017年4月 (21)
2017年3月 (9)

最新评论

1. Re:分布式锁的一点理解
好文章，目前也是打算把redis分布式锁替换成zk分布式锁。

--youyajike

2. Re:分布式锁的一点理解
有的不是太明白，但也极大的拓展了我对分布式锁的认知，多谢博主。

--九陌

3. Re:分布式锁的一点理解
对我等菜鸟来说，受益匪浅
--新生代程序员

阅读排行榜

1. 分布式锁的一点理解(6573)
2. 《Kubernetes权威指南》——入门(4229)
3. 《Kubernetes权威指南》——Kubelet运行机制与安全机制(2073)
4. 《Java多线程编程核心技术》——多线程与同步(951)
5. 《Mysql技术内幕, Innodb存储引擎》——Innodb体系结构(783)

此处主要指集群模式下，多个相同服务同时开启。

- 分布式与单机情况下最大的不同在于其不是多线程而是多进程。
- 多线程由于可以共享堆内存，因此可以简单的采取内存作为标记存储位置。而进程之间甚至可能都不在同一台物理机上，因此需要将标记存储在一个所有进程都能看到的地方。

分布式锁

- 当在分布式模型下，数据只有一份（或有限制），此时需要利用锁的技术控制某一时刻修改数据的进程数。
- 与单机模式下的锁不仅需要保证进程可见，还需要考虑进程与锁之间的网络问题。（我觉得分布式情况下之所以问题变得复杂，主要就是需要考虑到网络的延时和不可靠。。。一个大坑）
- 分布式锁还是可以将标记存在内存，只是该内存不是某个进程分配的内存而是公共内存如Redis、Memcache。至于利用数据库、文件等做锁与单机的实现是一样的，只要保证标记能互斥就行。

单机Redis锁

基本锁

- 原理：利用Redis的setnx如果不存在某个key则设置值，设置成功则表示取得锁成功。
- 缺点：如果获取锁后的进程，在还没执行完的时候挂调了，则锁永远不会释放。

改进型

- 改进：在基本型是锁上的setnx后设置expire，保证即使获取锁的进程不主动释放锁，过一段时间后也能自动释放。
- 缺点：
 - setnx与expire不是一个原子操作，可能执行完setnx该进程就挂了。
 - 当锁过期后，该进程还没执行完，可能造成同时多个进程取得锁。（貌似这个问题目前还没有很优雅的方案）

再改进

- 改进：利用Lua脚本，将setnx与expire变成一个原子操作，可解决一部分问题。
- 缺点：还是锁过期的问题。

步骤

评论排行榜

1. 分布式锁的一点理解(3)

推荐排行榜

1. 分布式锁的一点理解(2)

Copyright ©2018 suolu

1. 直接调用Lua脚本原子setnx同时expire，设置一个随机值。
2. 获取到锁则执行同步代码块，没获取则根据业务场景可以选择自旋、休眠、或做一个等待队列等拥有锁进程来唤醒（类似Synchronize的同步队列）。
3. 当同步代码块执行完成，先判断锁的key是否是自己设置的，如果是则删除key（可利用Lua做成原子操作），不是则表明自己的锁已经过期，不需要删除。（这时候就出现了多进程同时有锁的问题了）

总结

一般情况下直接用setnx加expire就够了，但从安全性的角度看还是存在一下几个问题：

1. 单点问题。单机Redis只在单机上，如果单机down了，那么所有需要用分布式锁的地方均获取不到锁，全部阻塞。需要做好降级的处理。
2. 可能出现多进程同时拥有锁。

Redlock

Redlock是Redis的作者antirez给出的集群模式的Redis分布式锁，它基于N个完全独立的Redis节点（通常情况下N可以设置成5）。

步骤

1. 获取当前时间（毫秒数）。
2. 按顺序依次向N个Redis节点执行获取锁的操作。获取锁的操作与单机锁一样。
3. 如果获取锁成功的节点数 $\geq N/2+1$ ，则再计算获取锁的时间有没有超过锁过期时间（可考虑设置一个必须留多长的时间给代码块执行），如果超过了则认为取锁失败。
4. 如果取锁失败则应该对所有节点进行释放锁的操作。

优化

- 当有5个节点，某次上锁对a,b,c三个节点上锁成功，而后c马上down了，此时还没通过AOF或RDB写入磁盘。而后c又马上恢复，此时c没有上锁数据，因此此时可能出现c,d,e三个节点被别的进程上锁。所以在节点恢复时应该延时起码一个锁的过期时间。

Zookeeper锁

zookeeper锁相关基础知识

- zk一般由多个节点构成（单数），采用zab一致性协议。因此可以将zk看成一个单点结构，对其修改数据其内部自动将所有节点数据进行修改而后才提供查询服务。
- zk的数据以目录树的形式，每个目录称为znode，znode中可存储数据（一般不超过1M），还可以在其中增加子节点。
- 子节点有三种类型。序列化节点，每在该节点下增加一个节点自动给该节点的名称上自增。临时节点，一旦创

建这个 znode 的客户端与服务器失去联系，这个 znode 也将自动删除。最后就是普通节点。

- Watch机制，client可以监控每个节点的变化，当产生变化会给client产生一个事件。

zk基本锁

- 原理：利用临时节点与watch机制。每个锁占用一个普通节点/lock，当需要获取锁时在/lock下创建一个临时节点，创建成功则表示获取锁成功，失败则watch/lock节点，有删除操作后再去争锁。临时节点好处在于当进程挂掉后能自动上锁的节点自动删除即取消锁。
- 缺点：所有取锁失败的进程都监听父节点，很容易发生羊群效应，即当释放锁后所有等待进程一起来创建节点，并发量很大。

zk锁 优化

- 原理：上锁改为创建临时有序节点，每个上锁的节点均能创建节点成功，只是其序号不同。只有序号最小的可以拥有锁，当需要不是最小的则watch序号排在前面的一个节点(公平锁)。
- 步骤：

1. 在/lock节点下创建一个有序临时节点 (EPHEMERAL_SEQUENTIAL)。
2. 判断创建的节点序号是否最小，如果是最小则获取锁成功。不是则取锁失败，然后watch序号比本身小的前一个节点。
3. 当取锁失败，设置watch后则等待watch事件到来后，再次判断是否序号最小。
4. 取锁成功则执行代码，最后删除本身节点，释放了锁。

分布式锁总结

分布式锁存在的问题

1. 均可能存在多进程拥有锁的情况。redis锁主要是expire时间与代码执行时间的问题，zk锁的问题在于zk是通过心跳监控进程存活状态，如果进程进行GC pause或者因为网络原因导致很长时间没与zk联系，则将导致zk认为进程已挂，而后锁自动释放，而此时进程并未挂任然在执行。
2. Redlock锁的时间问题。由于redis的expire的实现是通过pexpireat，如果某个节点发生时钟跳跃，则该节点可能过早释放锁导致一系列问题。

解决方案

1. 获取锁时提供一个fencing token(两种说法，一种说需要有序，一种说随机值就可以，我觉得随机值就可以)，在进程获取锁后对数据进行操作时，数据所在的

资源服务器需要去锁中查看当前token，如果token对的才执行，不对则放弃执行。

2. 我觉得对于放弃执行的应该在我们的代码块中增加类似事物的rollback的操作。因此如果资源服务器拒绝了我们的操作则表明此时起码已经存在了另外一个进程拥有锁了，为了保证数据安全性不能继续执行，因此需要回滚到执行代码块之前而继续去竞争锁。
3. 至于Redis锁的时间问题，Antirez说在运维层面是可以控制时钟跳跃的区间的，只要能控制跳跃区间与expire的比例就没问题，详细可看《基于Redis的分布式锁真的安全吗？》

总结

1. 大多数时候采用zk锁就好了，没必要再考虑安全性的问题。其实也可以通过zk锁+幂等校验来达到双层保障。
2. fencing 机制需要对数据服务进行修改适配，个人觉得没这个必要吧。。。

目前就这些了。。。。后面想到再补充吧。

引用：基于Redis的分布式锁真的安全吗？

[基于Redis的分布式锁真的安全吗？上](#)

[基于Redis的分布式锁真的安全吗？下](#)

分类: [分布式](#)

标签: [分布式锁](#), [redis](#), [zookeeper](#)

好文要顶

关注我

收藏该文



suolu

关注 - 0

粉丝 - 3

[+加关注](#)

2

0

» 下一篇: [zookeeper基础知识](#)

posted on 2017-03-20 14:45 [suolu](#) 阅读(6573) 评论(3) [编辑](#) [收藏](#)

发表评论

#1楼 2017-03-20 15:44 | 新生代程序员

对我等菜鸟来说，受益匪浅

支持(0) 反对(0)

#2楼 2017-11-08 14:06 | 九陌

有的不是太明白，但也极大的拓展了我对分布式锁的认知，多谢博主。

支持(0) 反对(0)

#3楼 2018-01-29 14:33 | youyajike

好文章，目前也是打算把redis分布式锁替换成zk分布式锁。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻:

- 科技公益有了新玩法：腾讯公益启动“行为公益季”
 - 对话苏宁新零售：O2O从未消失 无界零售不着调
 - 商汤科技宣布获得C轮战略融资6亿美元 阿里巴巴领投
 - IPO热潮带动更多公司进入 今年将是科技行业并购年
 - 乐视危机的又一个受伤者 毅昌股份去年净利同比降2575%
- » 更多新闻...

最新知识库文章:

- 写给自学者的入门指南
 - 和程序员谈恋爱
 - 学会学习
 - 优秀技术人的管理陷阱
 - 作为一个程序员，数学对你到底有多重要
- » 更多知识库文章...