Zookeeper分布式系统开发实战 第8课

DATAGURU专业数据分析社区

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

http://edu.dataguru.cn

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

第八讲 分布式锁案例—场景

- 场景一
  - 同一个应用部署在多台机器上，用于支撑高并发访问
  - 以下这段代码有什么问题？

```
@Transactional
public boolean doOrder(Order o) {
    //获取当前的产品库存数量
    Product nowp = productMapper.selectProductById(o.getProductId());
    if(nowp.getSize()>=o.getPnum()){
        orderMapper.saveOrder(o);
        HashMap<String,Integer> hm = new HashMap<String,Integer>();
        hm.put("nums", o.getPnum());
        hm.put("id",nowp.getId());
        productMapper.reduceNum(hm);
        System.out.println("库存充足，购买成功");
    }else{
        System.out.println("库存不足，购买失败");
        return false;
    }
    return true;
}
```

第八讲 分布式锁案例—场景

- 场景二
  - 有些数据，当有线程在读时，只允许别的线程再读，而不允许进行写操作
  - 有些数据，当有线程在写时，其它线程都不能再进行操作
  - 比如：数据库需要某种机制来保障数据的一致性并且尽量兼顾性能

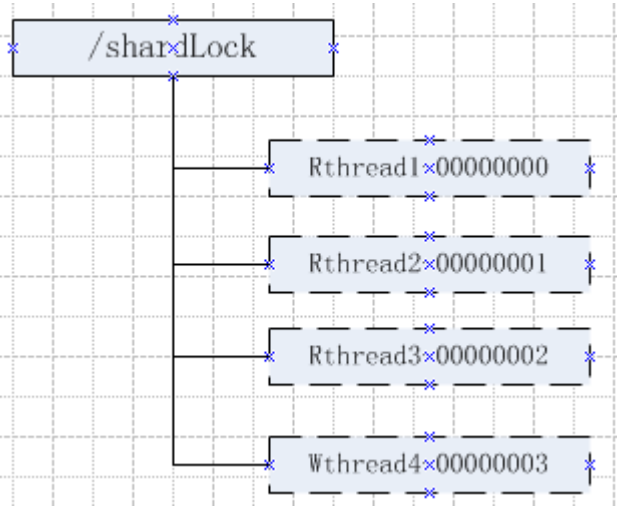
第八讲 分布式锁的类型

- 排他锁
  - 定义：只能允许一个线程获得，其它线程都需要等待已经获取的线程完成才能再次争抢锁资源
  - Zk实现：
    - 获得锁：通过构建一个目录，当叶子节点能创建成功，则认为获取到锁，因为一旦一个节点被某个会话创建，其它会话再次创建这个节点时，将会抛出异常，比如目录为：

    - 释放锁：删除节点或者会话失效

第八讲 分布式锁的类型

- 共享锁
  - 定义：
    - 读锁，如果前面线程使用的是读锁，则后面的线程还可以获取读锁，从而可以继续进行读操作
    - 写锁，如果在线程打算获取锁从而进行操作时，无论前面已经有读锁或者写锁都必须进入等待
  - Zk实现：
    - 获得读锁：利用zk节点的顺序性，对于读操作，节点名称带一个R标识，如果前面存在序列数比自己小，并且都是带R标识，则说明前面加的都是读锁，还可以继续获取读锁；否则，等待锁释放后有机会再抢
    - 获得写锁：只有自己创建的节点序列最小，才能获得读锁，否则，进入等待，直到有锁资源被释放，然后再判断是否有机会得到锁
    - 释放锁：删除节点或者会话失效



Thanks

FAQ时间