Zookeeper分布式服务架构是谷歌的Chubby一个开源的实现，是Hadoop和Hbase的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。

        实际上一般是用于 用于分布式中一致性处理的框架 由于分布式系统中一致性处理较为困难，其他的分布式系统没有必要费劲重复造轮子，故随后的分布式系统中大量应用了zookeeper，以至于zookeeper成为了各种分布式系统的基础组件(比如著名的分布式架构: hadoop、kafka、dubbo 都是基于zookeeper而构建)

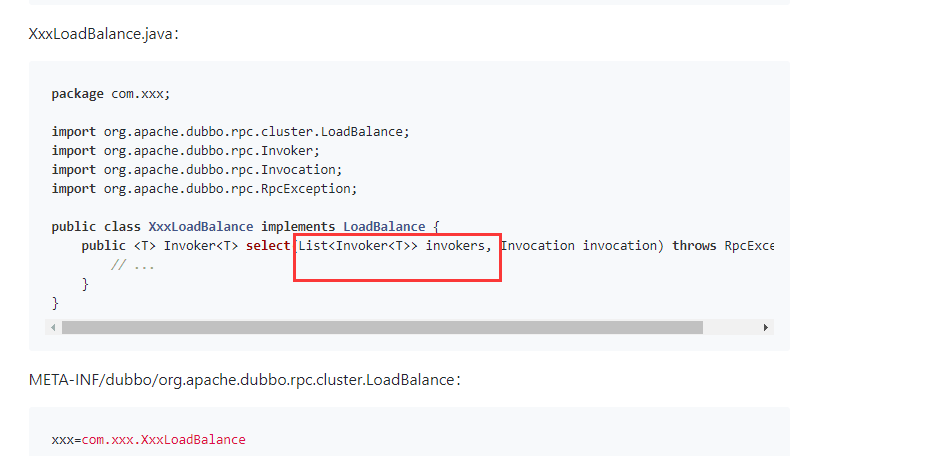
        所以回答第一个问题:用户使用的多,相应的资料与生态也想对的成熟,稳定性也是相对于其他分布式比较稳定,发展与版本的迭代也很迅速。

问题二：加减服务实例是需要配置的serverID，去看paxos算法或者zab就知道需要配置！当前zk集群是3台，那么选举的大多数是2；如果加一台，选举的大多数就是3，如果是挂了一台，当前zk集群并不知道是挂了一台 还是 集群下掉一个节点，所以需要进行配置变更；

问题三：对于多个服务实例使用同一ID的时候，用于服务更新场景，能够无缝切换这个是不可以的，ID是不可以重复的！

问题四：扩容问题：一般就3-7台，就行了，太多的话强一致性，可用性会降低而且zk启动后集群就固定了，除非你、改配置文件后重启才行；

问题五:也是最关键zk能不能,将对同一设备的请求，转发到同一服务实例，转发是自己做的，比如dubbo的话应该是对应负载均衡这块吧，invokers是当前注册中心服务的服务实例，你就看怎么挑或者干脆用默认策略也行。



应用的场景：

1. 配置管理：在分布式中很常见，同一个应用系统的多个服务由zk来修改配置等等；
2. 集群管理：作为“主管”知道当前集群中每台机器的服务状态，一旦有机器不能提供服务，就做从新调整分配服务
3. 分布式锁，主要是因为zk保证数据的强一致性，zookeeper的znode节点创建的唯一性和递增性能保证所有来抢锁的worker的原子性

**结论：最简单的方法就是第一次获取到服务实例ID用个map存起来!**