ZooKeeper介绍ZooKeeper是一个高可用的分布式数据管理和协调框架，可以很好地保证分布式环境中数据的一致性。ZooKeeper 的设计目标是将那些复杂且容易出错的分布式一致性服务封装起来，构成一个高效可靠的原语集，并以一系列简单易用的接口提供给用户使用。ZooKeeper主要是用来解决分布式应用中经常遇到的一些数据管理问题，如：统一命名服务、状态同步服务、集群管理、分布式应用配置项的管理等。在Hadoop,Hbase，Kafka中，ZooKeep都作为核心组件使用。ZooKeeper中有一个很重要的算法是Paxos。Zookeeper 采用paxos一致性算法保证了数据的一致性，Paxos算法是一种基于消息传递且具有高度容错特性的一致性算法。其中一致性是指：在一个分布式系统中，有一组进程，它们需要确定一个value，于是每个进程都提出了一个value，一致性就是指只有其中一个value可以被选中作为最后确定的值，并且当这个值被选出来后，所有的进程都需要被通知到。一致性是一篇bigtable的的论文中提到的Chubby所具有的功能，ZooKeeper借鉴这种思想被创造了出来，ZooKeeper采用Paxos的变种Zab（ZooKeeper原子消息广播协议）来实现消息传输的一致性。在Paxos小岛的基础上采用Zab而不是完全使用原来的Paxos算法是因为paxos算法是时间轴在后面的优先被选择，可能存在本场议员投票数过半就已经选择这个提案，但后面的议员提出新的议案时，提案号码会增加，原来的提案已经被采用，缺少一个“总统”来同步（Sync）统一议案再同时将所有收到的议案发送给所有议员再进行投票。Zab算法可以很好地避免这种问题发生。但是也存在总统缺失的情况，议员各自发表声明，再进行新的总统推选。总统大选期间政府停业，拒绝普通民众的请求。ZAB协议有两种基本的模式，分别为崩溃恢复和消息广播（即选举中总统缺失和正常情况）。当整个框架在运行的过程中，或者是当Leader出现网络中断、崩溃重启等异常的情况的时候，ZAB协议就会进入恢复模式，来重新选举一个新的Leader服务器来处理事物请求，当新的Leader服务器选择出来，并且和过半以上的机器实现了状态同步后，ZAB的恢复模式就结束了。这个时候就可以进行ZAB的消息广播模式了，如果在这个过程中有一个新的Zookeeper加入了当前的集群中，就会启动恢复模式，直到实现了和Leader的状态同步为止。Zookeeper工作原理为了保证事务的顺序一致性，zookeeper采用了递增的事务id号（zxid）来标识事务。所有的提议（proposal）都在被提出的时候加上了zxid。实现中zxid是一个64位的数字，它高32位是epoch用来标识leader关系是否改变，每次一个leader被选出来，它都会有一个新的epoch，标识当前属于那个leader的统治时期。低32位用于递增计数。每个Server在工作过程中有三种状态：LOOKING：当前Server不知道leader是谁，正在搜寻LEADING：当前Server即为选举出来的leaderFOLLOWING：leader已经选举出来，当前Server与之同步Zookeeper系统中的角色主要有三类，Leader，Learner和Observer选择leader leader失去多数follower或当leader崩溃时，这时候zookeeper进入恢复模式，恢复模式就要重新选举，得出一个新的leader，让所有的Server都恢复到一个正确的状态。选举中Paxos算法与Zab协议：1.选举线程由当前Server发起选举的线程担任，其主要功能是对投票结果进行统计，并选出推荐的Server；2.选举线程首先向所有Server发起一次询问(包括自己)；3.选举线程收到回复后，验证是否是自己发起的询问(验证zxid是否一致)，然后获取对方的id(myid)，并存储到当前询问对象列表中，最后获取对方提议的leader相关信息(id,zxid)，并将这些信息存储到当次选举的投票记录表中；4.收到所有Server回复以后，就计算出zxid最大的那个Server，并将这个Server相关信息设置成下一次要投票的Server；5.线程将当前zxid最大的Server设置为当前Server要推荐的Leader，如果此时获胜的Server获得n/2 + 1的Server票数， 设置当前推荐的leader为获胜的Server，将根据获胜的Server相关信息设置自己的状态，否则，继续这个过程，直到leader被选举出来。选完leader以后，ZooKeeper就进入状态同步过程。同步过程：1.leader等待server连接；2.Follower连接leader，将最大的zxid发送给leader；3.Leader根据follower的zxid确定同步点；4.完成同步后通知follower 已经成为uptodate状态；5.Follower收到uptodate消息后，又可以重新接受client的请求进行服务了。