# 从Paxos到Zookeeper分布式一致性原理与实践

# Zookeeper与Paxos

## 4.1 初始Zookeeper

### 4.1.1 Zookeeper介绍

集群角色：Leader、Follower、Observer

Leader：集群中的所有机器通过一个Leader选举过程来选定一台被称为“Leader”的机器。Leader服务器为客户端提供读和写服务。

Follower和Observer都能够提供读服务，唯一的区别在于，Observer不参与Leader选举过程，也不参与写操作的“过半写成功”策略。

会话（Session）：在Zookeeper中，一个客户端连接是指客户端和服务器之间的一个TCP长连接。当由于服务器眼里太大、网络故障或是客户端主动断开连接等各种原因导致客户端连接断开时，只要在sessionTimeout规定的时间内能够重新连接上集群中任意一台服务器，那么之前创建的会话仍然有效。

节点：第一类构成集群的机器（机器节点），第二类是指数据模型中的数据单元（数据节点）。

ZNode可以分为持久节点和临时节点。

持久节点：一旦ZNode被创建了，除非主动进行ZNode的移除操作，否则这个ZNode将一直保存在Zookeeper上。

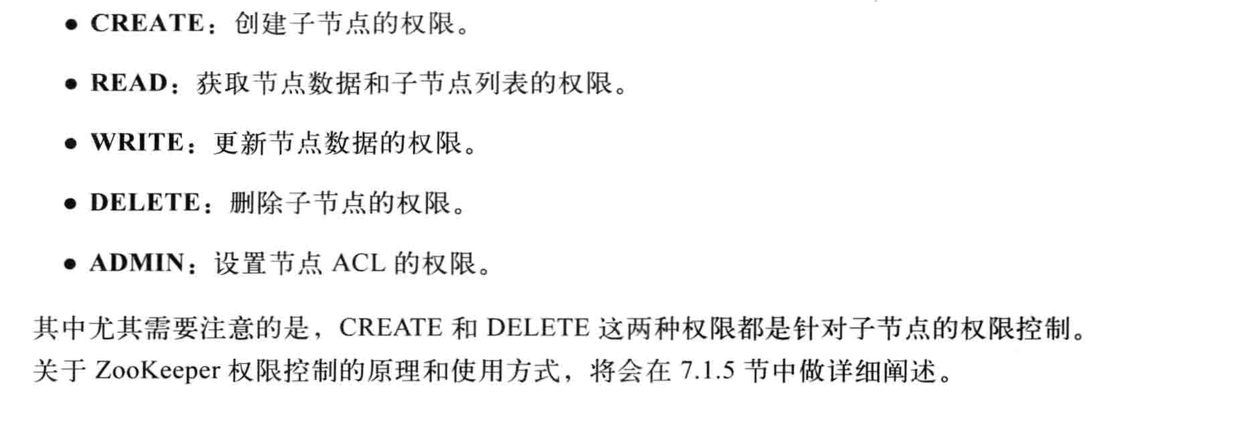
临时节点：生命周期和客户端会话绑定。会话失效，客户端创建的所有临时节点都会被移除。

版本：

Zookeeper为每个ZNode维护一个叫做Stat的数据结构，Stat中记录了这个ZNode的是三个数据版本，分别是version(当前ZNode版本)、cversion（当前ZNode子结点的版本）和aversion(当前ZNode的ACL版本)。

Watcher（事件监听器）：

ACL（Access Control Lists）策略来进行权限控制。



### 4.1.2 Zookeeper介绍

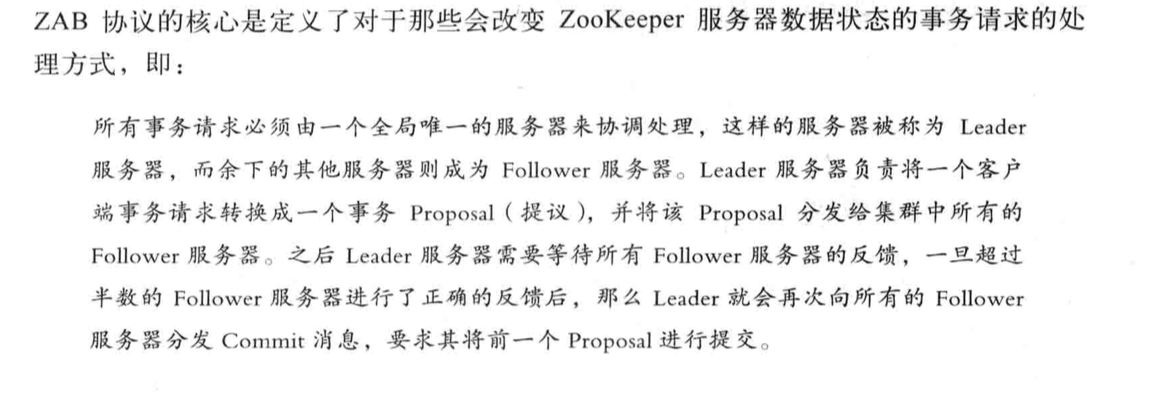
### 4.1.3 Zookeeper介绍

### 4.1.4 Zookeeper介绍

## 4.2 Zookeeper的ZAB协议

Zookeeper Atomic Broadcast(ZAB，Zookeeper原子消息广播协议)的协议作为其数据的一致性核心算法。支持崩溃恢复的原子广播协议。

Zookeeper实现了会中准备模式的系统架构来保持集群中各副本之间数据的一致性。



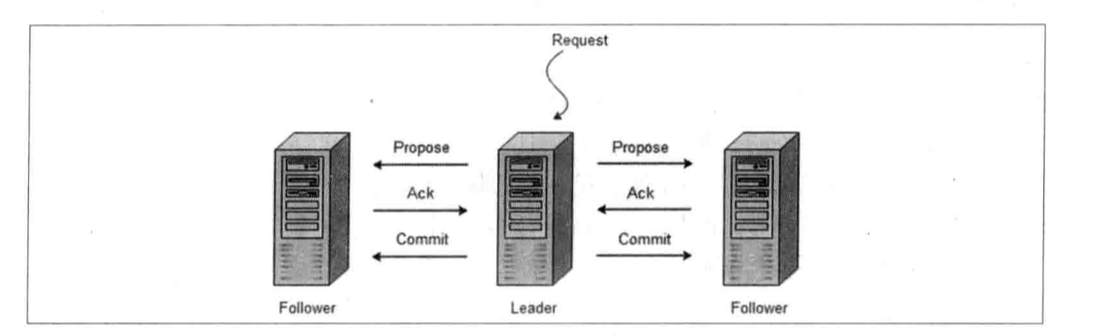
ZAB协议包括两种模式：崩溃恢复和消息广播。

当整个服务器在启动过程中，或是当Leader服务器出现网络中断、崩溃退出与重启等异常情况时，ZAB协议就会进入恢复模式并选举产生新的Leader服务器。当选举产生了新的Leader服务器，同时集群中有过半的服务器与该Leader服务器完成了状态同步之后，ZAB协议就会退出恢复模式。状态同步是指数据同步，保证集群中过半的机器能够和Leader服务器的数据状态保持一致。

集群中过半Follower服务器完成了和Leader服务器的状态同步，那么整个服务框架就可以进入消息广播模式了。当新加入一个台ZAB协议的服务器，且已经存在Leader服务器，那么新加入的服务器自动进入数据恢复模式：找到Leader所在服务器，并与其进行数据同步，然后一起参与到消息广播流程中取。

消息广播：

ZAB协议的消息广播过程使用了一个院子广播协议，



在整个消息广播过程中，Leader服务器会为每个拾取请求生成对应的Proposal来进行广播，

并且在广播拾取Proposal之前，Leader服务器会首先为这个事务Proposal分配一个全局单调递增的唯一ID，我们称之为事务ID(即ZXID)。由于ZAB协议需要保证每一个消息的因果关系，因此必须将每一个事务Proposal按照其ZXID的先后顺序来进行排序与处理。

崩溃恢复：

数据同步：

分别是发现(Discovery)、同步(Synchronization)和广播(Boradcast).

# 使用Zookeeper