参考：<http://www.cnblogs.com/raphael5200/p/5285267.html>

# [Zookeeper 1、Zookeeper 定义与工作原理](http://www.cnblogs.com/raphael5200/p/5285267.html)

****

**1、什么是Zookeeper**

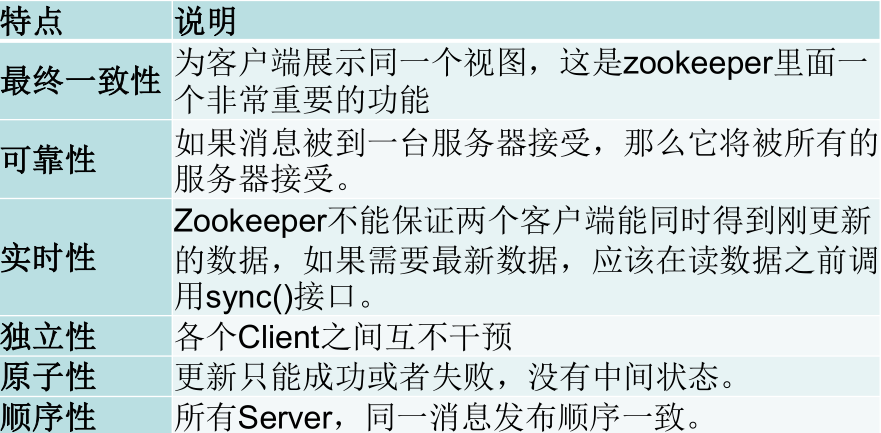
　　» Zookeeper 是 Google 的 Chubby一个开源的实现，是 Hadoop 的分布式协调服务

　　» 它包含一个简单的原语集，分布式应用程序可以基于它实现同步服务，配置维护和命名服务等

**2、为什么要用Zookeeper**

　　» 大部分分布式应用需要一个主控、协调器或控制器来管理物理分布的子进程（如资源、任务分配（me：如yarn等）等）  
　　» 目前，大部分应用需要开发私有的协调程序，缺乏一个通用的机制（zookeeper产自于hadoop的生态系统，它后期从管理配置组件分离出来的）  
　　» 协调程序的反复编写浪费，且难以形成通用、伸缩性好的协调器  
　　» Keepalived：提供通用的分布式锁服务，用以协调分布式应用但是：  
　　　　• Keepalived监控节点不好管理  
　　　　• Keepalive 采用优先级监控  
　　　　• 没有协同工作  
　　　　• 功能单一  
　　　　• Keepalive可扩展性差

**3、Zookeeper的优点**



这些优点都是由zookeeper内部设计的相关机制来保证的：

最终一致性：zookeeper集群为客户端提供了一个同一视图，即服务器集群对客户端是透明的；

可靠性：如果消息被一台服务器接收，那么他将被所有的服务器接收。这主要是因为当一台服务器接受到写请求后，他会自动将请求同步到其他的服务器，只有大多数服务器都返回成功后才代表本次操作成功；

实时性：zookeeper不能保证两个客户端能同时得到刚刚更新的数据，如果需要获取最新的数据应该调用sync接口；

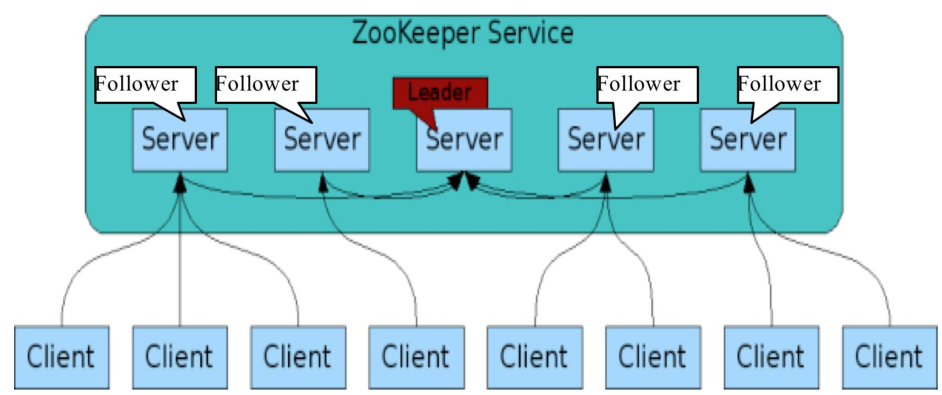
独立性：各个client之间互不干预；

原子性：写/更新操作只能成功或者失败，没有中间状态；

顺序性：所有server，同一消息发布顺序一致。

**4、Zookeeper的工作原理**

　　1.每个Server在内存中存储了一份数据（me：在内存中存储数据保证了快速响应客户端的请求）；  
　　2.Zookeeper启动时，将从实例中选举一个leader（Paxos协议）（me：zookeeper只有两种状态-leader和follower）  
　　3.Leader负责处理数据更新/写等操作  
　　4.一个更新操作成功，当且仅当大多数Server在内存中成功修改数据。



**5、Zookeeper能做什么**

**管理分布式集群，如**

　　» 管理Hadoop,使用Zookeeper的事件处理确保整个集群只有一个NameNode,存储配置信息等.  
　　» 管理HBase,使用Zookeeper的事件处理确保整个集群只有一个HMaster,察觉HRegionServer联机和宕机,存储访问控制列表等.

　　...

**6、Zookeeper的特性**

　　» Zookeeper是简单的  
　　» Zookeeper是富有表现力的  
　　» Zookeeper具有高可用性  
　　» Zookeeper采用松耦合交互方式  
　　» Zookeeper是一个资源库