



Zookeeper 的认识

作者: [JackLei](#)

原文链接: <https://ld246.com/article/1655026380736>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

ZooKeeper是什么---

ZooKeeper 是什么 :tada: ?

ZooKeeper 是一个分布式的，开放源码的 **分布式应用程序协调服务**，为了解决分布式架构下数据一致性问题，典型的应用场景有分布式配置中心、分布式注册中心、分布式锁、分布式队列、集群选举、分布式屏障、发布/订阅等场景。

ZooKeeper有哪些数据结构-

1-持久化目录节点-

客户端与 zookeeper 断开连接后，该节点依旧存在，只要不手动删除该节点，他将永远存在。

2-持久化顺序编号目录节点-

客户端与 zookeeper 断开连接后，该节点依旧存在，只是 zookeeper 给该节点名称进行顺序编号。

3-临时目录节点-

客户端与 zookeeper 断开连接后，该节点被删除。

4-临时顺序编号目录节点-

客户端与 zookeeper 断开连接后，该节点被删除，只是 zookeeper 给该节点名称进行顺序编号

Zookeeper的核心功能

监听机制

监听到七种类型变化：

1、None：连接建立事件

2、NodeCreated：节点创建

3、NodeDeleted：节点删除

4、NodeDataChanged：节点数据变化

5、NodeChildrenChanged：子节点列表变化

6、DataWatchRemoved：节点监听被移除

7、ChildWatchRemoved：子节点监听被移除

我们在开发中使用Synchronized就可以保证线程安全了-为什么还需要使用zookeeper来实现分布式锁呢-

我们在开发中使用 Synchronized 就可以保证线程安全了，为什么还需要使用 zookeeper 来实现分布式锁呢？

如果是单机情况下也就是只有一个进程的情况下使用 Synchronized 是可以保证线程安全的但是分布式情况下是多个不同的进程，而不是一个进程里面不同的线程，所以 Synchronized 是无法证多个进程安全的。

zookeeper节点有哪些重要信息呢-怎么进行查看-

1、cZxid：创建 znode 的事务 id (Zxid 的值)。

2、mZxid：最后修改 znode 的事务 id。

3、pZxid：最后添加或删除子节点的事务 id (子节点列表发生变化才会发生改变)。

4、ctime：znode 创建时间。

5、mtime：znode 最近修改时间。

6、dataVersion：znode 的当前数据版本。

7、cversion：znode 的子节点结果集版本 (一个节点的子节点增加、删除都会影响个版本)。

8、aclVersion：表示对此 znode 的 acl 版本。

9、ephemeralOwner：znode 是临时 znode 时，表示 znode 所有者的 sessionid 如果 znode 不是临时节点，则该字段设置为零。

10、dataLength：znode 数据字段的长度。

zookeeper创建节点-和监听节点有哪些命令-

1、创建持久节点：create /节点名称

2、创建持久顺序节点：create -s /节点名称

3、创建临时节点：create -e /节点名称

4、创建临时顺序节点：create -e -s /节点名称

5、监听节点：get -w /节点名称</p>

<h2 id="Zookeeper有哪些权限控制ACL-Access-Control-List-">Zookeeper 有哪些权限控制 ACL Access Control List) </h2>

<h2 id="权限模式">权限模式</h2>

<p>就是 zookeeper 服务器进行权限验证的方式，比如手机解锁是用指纹还是用密码呢，如果用密码，知道密码的都可以解锁你的手机，如果用指纹，只有你自己的手指头可以解锁。所以权限模式大体为两种类型：</p>

<h3 id="1-范围验证-">1、范围验证：</h3>

<p>范围就是 zookeeper 可以针对一个 ip 或者一段 ip 地址授予权限，相当于指纹解锁，我给我一手指头，或者所有的脚指头都授予权限。</p>

<h3 id="2-口令验证-">2、口令验证：</h3>

<p>可以理解为用户名密码的方式，这种就相当于密码解锁了，知道了用户名密码后所有的人都相当授予了权限。</p>

<h2 id="授权对象-">授权对象：</h2>

<p>授权对象就是把权限授予给谁，如果是范围验证方式，那么授权对象就是 ip 地址，如果是口令证，授权对象就是用户名。</p>

<h2 id="授权信息-">授权信息：</h2>

<p>授权信息就是指我们具体的权力是什么，比如我们解锁手机后可以打游戏还是可以听歌，还是可用手机砸核桃呢。</p>

<p>zookeeper 中定义好的权限有 5 种：</p>

<p>1、数据节点 (c:create) 创建权限，授予权限的对象可以在数据节点下创建子节点。

2、数据节点 (w:wirte) 更新权限，授予权限的对象可以更新该数据节点。

3、数据节点 (r:read) 读取权限，授予权限的对象可以读取该节点的内容以及子节点的列表信息。

4、数据节点 (d:delete) 删除权限，授予权限的对象可以删除该数据节点的子节点。

5、数据节点 (a:admin) 管理者权限，授予权限的对象可以对该数据节点体进行 ACL 权限设置。</>

<h2 id="怎么查看和设置某个节点的ACL权限信息呢">怎么查看和设置某个节点的 ACL 权限信息呢</h2>

<p>可以通过 getAcl 来获取某个节点的权限信息，通过 setAcl 来设置某个节点的权限信息。</p>

<h2 id="Zookeeper数据持久化">Zookeeper 数据持久化</h2>

<p>持久化也是两种方式，一种是记录事务日志，一种是快照方式。</p>

<h2 id="记录事务日志-">记录事务日志：</h2>

<p>磁盘会进行 IO 操作，事务日志的不断增多会触发磁盘为文件开辟新的磁盘块，所以为了提升磁的效率，可以在创建文件的时候就向操作系统申请一块大一点的磁盘块，通过参数 zookeeper.preAll cSize 配置。</p>

<p>事务日志的存放地址通过 zoo.cfg 配置文件中的 dataDir 来指定。</p>