|  |  |
| --- | --- |
| **第一节** | **第二节** |
| Zokeeper安装配置及简单使用 | Zookeeper API如何进行调用 |
| 分布式架构详解 | 一致性协议详解 |
| Paxos算法及ZAB协议 | Chubby技术架构解析 |
| ZkClient、Curator | ZK大型分布式系统中的应用 |
| Zk分布式锁、发布订阅 | Zk命名服务、Master选举 |
| Zk高级特性：服务器角色 | Zk高级特性：系统模型 |
| Zk高级特性：会话与客户端 | Zk高级特性：数据与存储 |
| Zk高级特性：序列化与通信协议 | Zk高级特性：Leader选举解析 |

# Zookeeper客户端

## StaticHostProvider

Host\_6

Host\_3

Host\_1

Host\_5

Host\_44

Host\_2

lastIndex=-1

currentIndex=-1

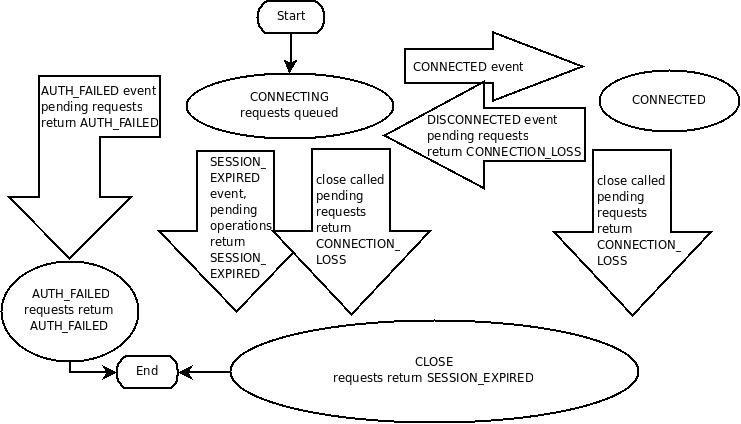
## Chroot命名空间

主要为了对业务进行隔离性

Zookeeper client=new Zookeeper(“192.168.56.101:2181/zk-client”, ........)

/zk-client就是Chroot命名空间。Chroot命名空间可以多级

# Zookeeper会话



CONNECTING 正在连接

CONNECTED 已经连接

RECONNECTING 重新连接

RECONNECTED 重新连接上

CLOSE 会话关闭

## Session

SessionTracker, LearnerSessionTracker, SessionTrackerImpl

org.apache.zookeeper.server.SessionTrackerImpl.initializeNextSession(long)

SessionID的分配（初始化）函数，策略如下：

1. 取时间，并且左移24位得到的结果再右移8位（高8位，低16位都是0）
2. sid拿出来进行左移56位
3. 和第一步的结果做或运算

Session分桶

按照Session会话过期时间进行分区块保存。

session激活过程：

1. 检测会话是否过期
2. 计算会话下一次超时时间
3. 定位会话的所在区块
4. 迁移会话

# Zookeeper面试问题

1. Zookeeper是如何保证事务的顺序一致性的
2. Zookeeper是如何选取主leader的？
3. Zookeeper中的节点
4. Zookeeper的通知机制
5. Zookeeper分布式锁/分布式队列/配置管理/Master选举
6. Paxos、ZAB协议（2P/3P/CAP/BASE）

# Zookeeper实现配置中心