|  |  |
| --- | --- |
| **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 |

# Zk Shell使用

$ zkCli.sh [-server ip:port]  
常用命令

ls、ls2 列出zk中的节点

create [-s] [-e] path data acl 创建节点

zk的节点分为两种：临时节点（随着zk session消亡而自动删除）、持久节点（一直存在）

delete 删除命令

rmr path 递归删除一个znode

get 获取节点信息

set 更新节点数据

connect host:port 连接到指定节点

setquota -n|-b val path 某个Znode指定多少存储空间或者允许创建多少个节点

n 指定可以设置多少个子节点

b 指定可以设置多大空间（byte）

listquota path

对于配额不是硬性的提示，超过配额还是可以继续创建，只不过在日志里面有提示

stat path查看节点的状态

# Zk Client调用示例

Zkdemo 解释怎么去连接zk并监听事件

ZkOperateDemo 解释如何去创建znode并监听事件

ZkDataDemo 解释如何改变znode数据并监听事件

ZKChildrenDemo 解释如何改变子节点并监听事件

ZKChildrenAsyncDemo 解释如何异步调用并完成回调

# 分布式发展历程

# 分布式技术详解

并发性

分布性

缺乏全局时钟

对等性

故障肯定会发生

# 分布式事务

# ACID

# CAP理论



一致性（Consistency）

可用性（Availability）

分区容错性（网络分区）Partition tolerance

CAP理论的特点，就是CAP只能满足其中2条。