

## （一）Zookeeper基础知识、体系结构、数据模型

1

zookeeper是一个类似hdfs的树形文件结构，zookeeper可以用来保证数据在(zk)集群之间的数据的事务性一致、

2

zookeeper有watch事件，是一次性触发的，当watch监视的数据发生变化时，通知设置了该watch的client，即watcher

3 zookeeper有三个角色：Learner，Follower，Observer

4 zookeeper应用场景：

统一命名服务（Name Service）

配置管理（Configuration Management）

集群管理（Group Membership）

共享锁（Locks）

队列管理

## （二）Zookeeper配置（搭建zookeeper服务器集群）

### 1.1 结构：一共三个节点

(zk服务器集群规模不小于3个节点),要求服务器之间系统时间保持一致。

### 1.2 上传zk

```
进行解压：      tar zookeeper-3.4.5.tar.gz
重命名：        mv zookeeper-3.4.5 zookeeper
修改环境变量：  vi /etc/profile
                export
```

```
ZOOKEEPER_HOME=/usr/local/zookeeper
```

```
export
```

```
PATH=.:$HADOOP_HOME/bin:$ZOOKEEPER_HOME/bin:$JAVA_HOME/...
```

```
刷新：          source /etc/profile
```

到zookeeper下修改配置文件

```
cd /usr/local/zookeeper/conf
```

```
mv zoo_sample.cfg zoo.cfg
```

```
修改conf:      vi zoo.cfg 修改两处
```

（1）dataDir=/usr/local/zookeeper/data

（2）最后面添加

```
server.0=bhz:2888:3888
```

```
server.1=hadoop1:2888:3888
```

```
server.2=hadoop2:2888:3888
```

服务器标识配置：

创建文件夹： mkdir data

创建文件myid并填写内容为0： vi

myid (内容为服务器标识： 0)

进行复制zookeeper目录到hadoop01和hadoop02

还有/etc/profile文件

把hadoop01、hadoop02中的myid文件里的值修改为1和2  
路径(vi /usr/local/zookeeper/data/myid)

启动zookeeper:

路径: /usr/local/zookeeper/bin

执行: zkServer.sh start

(注意这里3台机器都要进行启动)

状态: zkServer.sh

status(在三个节点上检验zk的mode,一个leader和两个follower)

### 1.3 操作zookeeper (shell)

zkCli.sh 进入zookeeper客户端

根据提示命令进行操作:

查找: ls / ls /zookeeper

创建并赋值: create /bhz hadoop

获取: get /bhz

设值: set /bhz baihezhao

可以看到zookeeper集群的数据一致性

创建节点有俩种类型: 短暂 (ephemeral)

持久 (persistent)

### (三) zoo.cfg详解:

**tickTime:** 基本事件单元, 以毫秒为单位。这个时间是作为 Zookeeper 服务器之间或客户端与服务器之间维持心跳的时间间隔, 也就是每隔 tickTime时间就会发送一个心跳。

**dataDir:** 存储内存中数据库快照的位置, 顾名思义就是 Zookeeper 保存数据的目录, 默认情况下, Zookeeper 将写数据的日志文件也保存在这个目录里。

**clientPort:** 这个端口就是客户端连接 Zookeeper 服务器的端口, Zookeeper 会监听这个端口, 接受客户端的访问请求。

**initLimit:** 这个配置项是用来配置 Zookeeper 接受客户端初始化连接时最长能忍受多少个心跳时间间隔数, 当已经超过 10 个心跳的时间 (也就是 tickTime) 长度后 Zookeeper 服务器还没有收到客户端的返回信息, 那么表明这个客户端连接失败。总的时间长度就是  $10 \times 2000 = 20$  秒。

**syncLimit:** 这个配置项标识 Leader 与 Follower 之间发送消息，请求和应答时间长度，最长不能超过多少个 **tickTime** 的时间长度，总的时间长度就是  $5*2000=10$  秒

**server.A = B:C:D :**

A表示这个是第几号服务器，

B 是这个服务器的 ip 地址；

C 表示的是这个服务器与集群中的 Leader 服务器交换信息的端口；

D 表示的是万一集群中的 Leader

服务器挂了，需要一个端口来重新进行选举，选出一个新的 Leader