

Zookeeper集群搭建步骤及相关知识点深入了解

# 1、zookeeper概念

zookeeper是一个**分布式协调服务**：a:zookeeper是为别的分布式程序服务的

b:zookeeper本身就是一个分布式程序（只要半数以上节点存活，zookeeper就能正常服务。）

c:zookeeper的服务范围：主从协调、服务器节点动态上下线、统一配置管理、分布式共享锁、统一名称服务...

d:zookeeper底层其实只提供了两个功能：

（1）管理（存储、读取）用户程序提交的数据

（2）为用户程序提交数据节点监听服务

# 2、zookeeper集群机制

半数机制：集群中半数以上机器存活，集群可用。

zookeeper集群适合搭建在奇数台机器上。

# 3、zookeeper特性

- Zookeeper：一个leader，多个follower组成的集群
- 全局数据一致：每个server保存一份相同的数据副本，client无论连接到哪个server，数据都是一致的
- 分布式读写，更新请求转发，由leader实施
- 更新请求顺序进行，来自同一个client的更新请求按其发送顺序依次执行
- 数据更新原子性，一次数据更新要么成功，要么失败
- 实时性，在一定时间范围内，client能读到最新数据

# 4、zookeeper数据结构

- 层次化的目录结构，命名符合常规文件系统规范(见下图)

- 每个节点在zookeeper中叫做Znode,并且其有一个唯一的路径标识

- 节点Znode可以包含数据和子节点（但是EPHEMERAL类型的节点不能有子节点）

< 2018年7月 >						
日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

公告

昵称：[ahu-lichang](#)

园龄：[1年4个月](#)

粉丝：[47](#)

关注：[2](#)

[+加关注](#)

搜索

找找看

谷歌搜索

常用链接

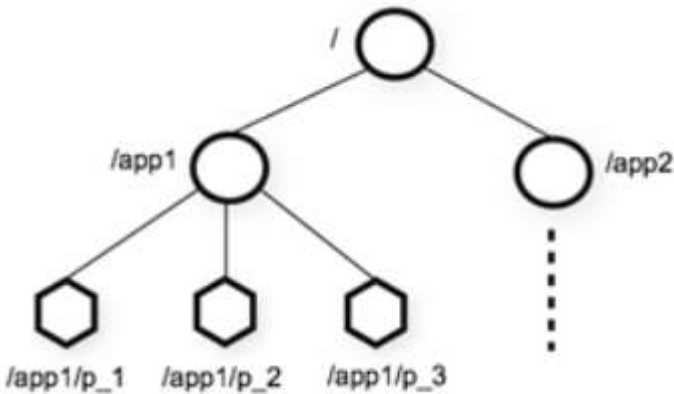
- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

最新随笔

- [1. Spark学习入门](#)
- [2. Pig和Hive的对比](#)
- [3. Redis简单了解](#)
- [4. 随机森林（分类与回归）](#)
- [5. 如何寻找数组中的最小值与最大值](#)
- [6. 如何实现单链表的增删操作](#)
- [7. Scala入门教程---《chang哥教你一天搞定Scala》](#)
- [8. win10下安装scala](#)
- [9. SVM（支持向量机）分类算法](#)
- [10. 协同过滤推荐算法](#)

我的标签

- 客户端应用可以在节点上设置监视器



- 节点类型：

1、Znode有两种类型：

短暂（ephemeral）（断开连接自己删除）

持久（persistent）（断开连接不删除）

2、Znode有四种形式的目录节点（默认是persistent）

PERSISTENT

PERSISTENT\_SEQUENTIAL（持久序列/test0000000019）

EPHEMERAL

EPHEMERAL\_SEQUENTIAL

3、创建znode时设置顺序标识，znode名称后会附加一个值，顺序号是一个单调递增的计数器，由父节点维护

4、在分布式系统中，顺序号可以被用于为所有的事件进行全局排序，这样客户端可以通过顺序号推断事件的顺序

## 5、zookeeper原理及内部选举机制

原理：zookeeper在配置文件中并没有指定master和slave，但是，zookeeper在工作时，只有一个节点为leader，其余节点为follower，leader是通过内部的选举机制临时产生的。

选举机制：（两种情况）

（1）全新集群paxos

假设有五台服务器组成的zookeeper集群,它们的id从1-5,同时它们都是最新启动的,也就是没有历史数据,在存放数据量这一点上,都是一样的.假设这些服务器依序启动,来看看会发生什么.

1) 服务器1启动,此时只有它一台服务器启动了,它发出去的报没有任何响应,所以它的选举状态一直是LOOKING状态

2) 服务器2启动,它与最开始启动的服务器1进行通信,互相交换自己的选举结果,由于两者都没有历史数据,所以id值较大的服务器2胜出,但是由于没有达到超过半数以上的服务器都同意选举它(这个例子中的半数以上是3),所以服务器1,2还是继续保持LOOKING状态.

3) 服务器3启动,根据前面的理论分析,服务器3成为服务器1,2,3中的老大,而与上面不同的是,此时有三台服务器选举了它,所以它成为了这次选举的leader.

4) 服务器4启动,根据前面的分析,理论上服务器4应该是服务器1,2,3,4中最大的,但是由于前面已经有半数以上的服务器选举了服务器3,所以它只能接收当小弟的命了.

5) 服务器5启动,同4一样,当小弟.

（2）非全新集群（数据恢复）

初始化的时候，是按照上述的说明进行选举的，但是当zookeeper运行了一段时间之后，有机器down掉，重新选举时，选举过程就相对复杂了。

需要加入数据id、leader id和逻辑时钟。

数据id：数据新的id就大，数据每次更新都会更新id。

Leader id：就是我们配置的myid中的值，每个机器一个。

- WordCountApp(2)
- XmlPullParser解析xml文件(1)
- XUtils(1)
- yarn(1)
- Zookeeper服务器集群的搭建与操作(1)
- 常见推荐算法(1)
- 带断点续传的多线程下载(1)
- 带缓存功能的网络图片下载查看器(1)
- 发送GET和发送POST请求的几种方式（javaSE、HttpClient、AsyncHttpClient(1)
- 混合推荐机制(1)
- 更多

随笔分类(113)

- Android(11)
- Azkaban(1)
- Flume(3)
- Hadoop1.x(17)
- Hadoop2.x(7)
- Hadoop数据分析(3)
- HBase(7)
- Hive(4)
- Java(6)
- Kafka(4)
- Linux(7)
- Mahout(1)
- ML(11)
- Pig(2)
- Python(2)
- Redis(1)
- Scala(2)
- Shell(2)
- Spark(1)
- Sqoop(2)
- Storm(11)
- ZooKeeper(4)
- 推荐系统(3)
- 用户画像(1)

随笔档案(106)

- 2017年12月（1）
- 2017年8月（5）
- 2017年7月（13）
- 2017年6月（8）
- 2017年5月（18）
- 2017年4月（42）
- 2017年3月（19）

### 积分与排名

- 积分 - 65246
- 排名 - 6041

### 最新评论

- 1. Re:Spark学习入门

逻辑时钟：这个值从0开始递增,每次选举对应一个值,也就是说: 如果在同一次选举中,那么这个值应该是一致的 ；逻辑时钟值越大,说明这一次选举leader的进程更新.

选举的标准就变成：

- 1、逻辑时钟小的选举结果被忽略，重新投票
- 2、统一逻辑时钟后，数据id大的胜出
- 3、数据id相同的情况下， leader id大的胜出

根据这个规则选出**leader**。

## 6、zookeeper集群的搭建(博主亲手成功搭建步骤，经得起考验！)

( 1 ) 准备三台机器：hadoop1(192.168.33.201)、hadoop2(192.168.33.202)、hadoop3(192.168.33.203)。

这三台机器是在之前安装好常用软件（jdk）的机器克隆来的，克隆来了以后，要分别对其中的物理地址和ip进行修改：

ifconfig	---查看物理设备名称（eth1）
setup	---修改里面的设备名称为eth1，并将修改ip地址，保存退出
service network restart	---重启网卡
ifconfig	---查看是否配置成功

( 2 ) 将zookeeper安装包上传到hadoop1的/usr/local中。

上传方式：使用上传工具或者使用 <b>lrzsz</b>	
<b>lrzsz</b> 是一款在linux里可代替ftp上传和下载的程序。	
要想使用lrzsz进行 <b>上传rz(也可直接鼠标拖拽)</b> 和 <b>下载sz</b> ，则必须安装lrzsz。可以使用 <b>yum install lrzsz</b> 自动安装，必须有网络。没有网络，可以制作yum本地源。	

( 3 ) 解压        tar -zxvf /usr/local/zookeeper-3.4.5.tar.gz

重命名        mv zookeeper-3.4.5.tar.gz zookeeper

( 4 ) 修改环境变量

```
vi /etc/profile
```

添加内容：export ZOOKEEPER\_HOME=/usr/local/zookeeper

```
export PATH=.:$ZOOKEEPER_HOME/bin....
```

刷新环境变量

```
source /etc/profile
```

注意：3台机器都要修改

( 5 ) 修改配置文件

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

```
cp zoo_sample.cfg zoo.cfg
```

```
vi zoo.cfg
```

添加内容：

```
dataDir=/usr/local/zookeeper/data
```

（不能放在临时文件夹中，需新建data文件夹）

- 我是电子工业出版社的编辑，望认识一下。  
谢谢！  
我的微信号：wuwei83744810
- 秋客
- 2. Re:shuffle过程分析
- @丰峰锋我也是在网上看到的，然后整理过来的...
- ahu-lichang
- 3. Re:shuffle过程分析
- @ahu-lichangHadoop权威指南里即便是有也不会是彩图，肯定也是引用官方的图，只有官网才可能给出彩图。但是虽然很多人的博客都在使用这张图，大部分都说说是官方文档的，可是我找了官方文档，并没有.....
- 丰峰锋
- 4. Re:shuffle过程分析
- @丰峰锋在网上看到的，在Hadoop权威指南里好像有这个图...
- ahu-lichang
- 5. Re:shuffle过程分析
- 请问这两幅图是哪来的，能不能分享一下这两幅图所在的官方文档的地址
- 丰峰锋
- 6. Re:Combiner
- @hellenlee22去年六月开始从安卓转过来的，马上上班了...
- ahu-lichang
- 7. Re:Combiner
- 你好，楼主，大数据学了好久了？写了这么多文章好厉害
- hellenlee22
- 8. Re:K-means聚类算法及python代码实现
- @ahu-lichang嗯，懂了，谢谢楼主~...
- Asshead
- 9. Re:K-means聚类算法及python代码实现
- @Asshead簇类中心点坐标，中心点坐标没有发生了变化了，就分类结束了...
- ahu-lichang
- 10. Re:K-means聚类算法及python代码实现
- 楼主您好，我想问一下最后跑出来的结果是什么意思呀
- Asshead

### 阅读排行榜

- 1. MapReduce的原理及执行过程(26251)
- 2. K-means聚类算法及python代码实现(17329)
- 3. Mahout介绍和简单应用(7716)
- 4. Zookeeper集群搭建步骤及相关知识点深入了解(7661)
- 5. Scala入门教程---《chang哥教你一天搞定Scala》(7447)
- 6. 朴素贝叶斯分类算法介绍及python代码实现案例(5128)
- 7. KNN分类算法及python代码实现(3586)
- 8. Android studio代码实现打电话+点击事件四种方式(3491)
- 9. Lineage逻辑回归分类算法(3487)
- 10. Android应用程序签名打包（AS）(3255)
- 11. 决策树分类算法及python代码实现案例(3059)

```
dataLogDir=/usr/local/zookeeper/log (新建log文件夹，可不要！！！！)

server.1=hadoop1:2888:3888 (主机名,心跳端口、数据端口)

server.2=hadoop2:2888:3888 (都是默认端口)

server.3=hadoop3:2888:3888 (2888是leader和follow之间通信，3888是投票选举时用的端口)

创建文件夹

mkdir /usr/local/zookeeper/data

mkdir /usr/local/zookeeper/log

在data文件夹中新建myid文件，myid文件的内容为1（一句话创建：echo 1 > myid）

cd data

vi myid

添加内容：1

（6）把zookeeper目录复制到hadoop2和hadoop3中

在这之前要三台机器都要设置映射文件：

vi /etc/hosts

192.168.33.201 hadoop1

192.168.33.202 hadoop2

192.168.33.203 hadoop3

设置三个机器的本机免密登录（三台机器配置一样）：

ssh-keygen -t rsa ---一直回车即可

cd /root/.ssh/ ---生成了公钥和私钥

cat id_rsa.pub >> authorized_keys ---将公钥追加到授权文件中

more authorized_keys ---可以查看到里面追加的公钥

ssh hadoop1

配置两两之间的免密登录：

将hadoop1中的公钥复制到hadoop2中ssh-copy-id -i hadoop2 验证一下：ssh hadoop2

将hadoop3中的公钥复制到hadoop2中ssh-copy-id -i hadoop2 验证一下：ssh hadoop2

这样hadoop2中的授权文件就有三个机器的公钥，再把hadoop2中的授权文件复制给hadoop1和hadoop3

scp /root/.ssh/authorized_keys hadoop1:/root/.ssh/

scp /root/.ssh/authorized_keys hadoop3:/root/.ssh/

这样就ok了！

scp -r /usr/local/zookeeper/ hadoop2:/usr/local/
scp -r /usr/local/zookeeper/ hadoop3:/usr/local/
将hadoop0中的环境变量复制到hadoop2和hadoop3中
scp /etc/profile hadoop2:/etc/
scp /etc/profile hadoop3:/etc/
环境变量复制好了以后，在hadoop2和hadoop3上都要执行source /etc/profile

（7）把hadoop2中相应的myid的值改为2
vi /usr/local/zk/data/myid 将里面的值改为2
```

- 12. 1044-Access denied for user 'root'@'%' to data base 'lc\_db'(2643)
- 13. kafka中生产者和消费者API(2067)
- 14. HBase结合MapReduce批量导入（HDFS中的数据导入到HBase）(1922)
- 15. shuffle过程分析(1897)
- 16. Hive的安装和使用(1827)
- 17. Partitioner(1809)
- 18. hadoop2.6.4的HA集群搭建超详细步骤(1505)
- 19. Storm常用操作命令及WordCount(1490)
- 20. KNN分类算法实现手写数字识别(1341)

### 评论排行榜

- 1. K-means聚类算法及python代码实现(38)
- 2. Scala入门教程---《chang哥教你一天搞定Scala》(9)
- 3. shuffle过程分析(6)
- 4. MapReduce的原理及执行过程(4)
- 5. 朴素贝叶斯分类算法介绍及python代码实现案例(4)
- 6. android网络编程(2)
- 7. Zookeeper集群搭建步骤及相关知识点深入了解(2)
- 8. Combiner(2)
- 9. Spark学习入门(1)

### 推荐排行榜

- 1. K-means聚类算法及python代码实现(3)
- 2. MapReduce的原理及执行过程(2)
- 3. HDFS的Java操作方式(1)
- 4. HDFS详解(1)
- 5. Zookeeper集群搭建步骤及相关知识点深入了解(1)
- 6. win10下安装scala(1)
- 7. Combiner(1)

Powered by: 博客园  
模板提供：沪江博客  
Copyright ©2018 ahu-lichang

把hadoop3中相应的myid的值改为3  
vi /usr/local/zk/data/myid 将里面的值改为3

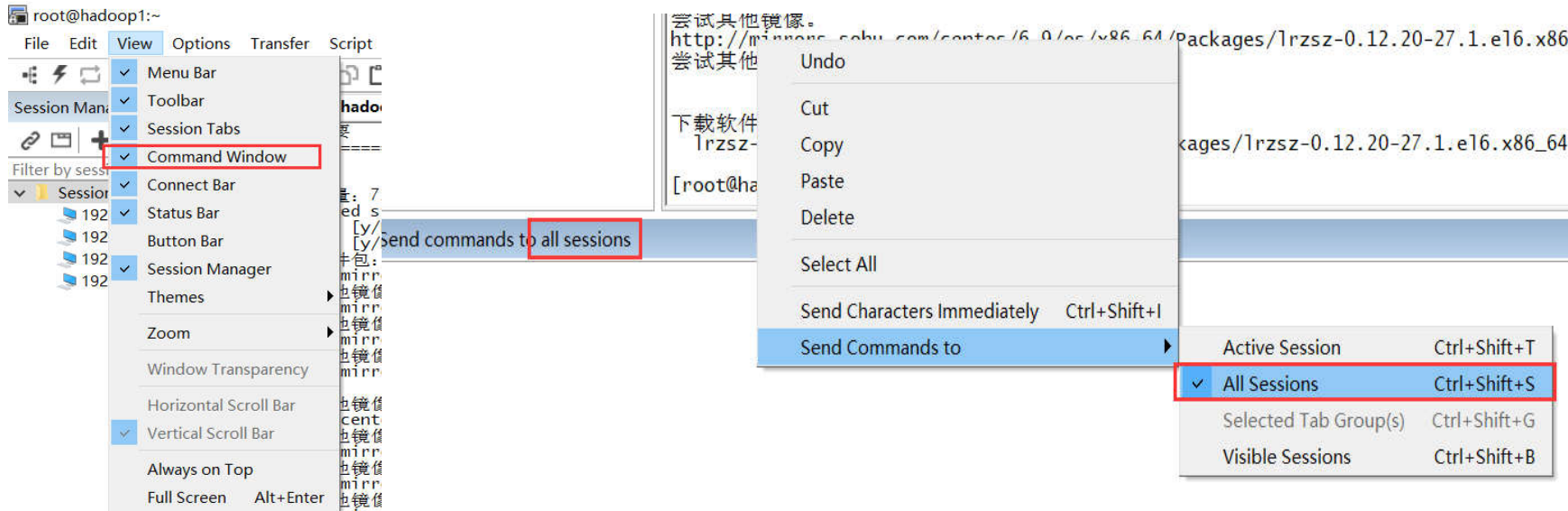
( 8 ) 将三台机器的防火墙关闭掉 , service iptables stop , 查看service iptables status

启动 , 在三个节点上**分别都要**执行命令zkServer.sh start  
cd /usr/local/zk/bin  
ls  
zkServer.sh start

启动完了之后 , 在bin目录下多了一个zookeeper.out

**小技巧 : ( 怎么对三台机器同时执行一个命令 ? )**

在CRT中 , 菜单栏View-->Command Window勾选上 , 工具下面会出现一个窗口 , 在窗口中点击鼠标右键 , 选择all session。这样在窗口中执行一个命令 , 就会对所有session起作用 !



( 9 ) 检验 , jps查看进程 , 会出现进程QuorumPeerMain

在三个节点上**依次执行**命令zkServer.sh status(可以看到MODE , 谁是leader , 谁是follower)

hadoop1 : follower

hadoop2:leader

hadoop3:follower

望志同道合者 , 一起交流探讨 , 有问题可评论 , 会及时回复 !

分类: ZooKeeper

好文要顶

关注我

收藏该文



ahu-lichang

关注 - 2

粉丝 - 47

+加关注

« 上一篇 : MySQL server PID file could not be found!

» 下一篇 : zookeeper命令行客户端



评论

- # 1楼

zookeeper中的znode节点是分布式存储在集群中的其他服务器上是吧

支持(0) 反对(0)

辰~风 评论于 2018-01-03 17:15
- # 2楼[楼主]

@ 辰~风  
znode是zookeeper集群中的一个节点，集群是分布式的，那么znode节点肯定是分布式的啊

支持(0) 反对(0)

ahu-lichang 评论于 2018-01-04 11:38

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。



最新IT新闻：

- 华为手机发货量破亿 国产四强瓜分七成市场
  - 印度二季度手机市场：三星反击 市场份额追平小米
  - 传谷歌拟在5年内用Fuchsia取代Android和Chrome OS
  - 对谷歌处以巨额罚款后，欧盟反垄断机构又瞄准了高通
  - 网贷爆雷潮反思：长期“无证驾驶” 违规经营惹的祸
- » [更多新闻...](#)



最新知识库文章：

- 观察之道：带你走进可观察性
  - 危害程序员职业生涯的三大观念
  - 断点单步跟踪是一种低效的调试方法
  - 测试 | 让每一粒尘埃有的放矢
  - 从Excel到微服务
- » [更多知识库文章...](#)

