

Linux下Zookeeper集群的安装

顾名思义 **zookeeper** 就是动物园管理员，他是用来管 **hadoop**（大象）、**Hive**(蜜蜂)、**pig**(小猪)的管理员，**Apache Hbase** 和 **Apache Solr** 的分布式集群都用到了 **zookeeper**；**Zookeeper**：是一个分布式的、开源的程序协调服务，是 **hadoop** 项目下的一个子项目。

一、**zookeeper** 主要功能如下：

1、配置管理

在我们的应用中除了代码外，还有一些就是各种配置。比如数据库连接等。一般我们都是使用配置文件的方式，在代码中引入这些配置文件。但是当我们只有一种配置，只有一台服务器，并且不经常修改的时候，使用配置文件是一个很好的做法，但是如果我们的配置非常多，有很多服务器都需要这个配置，而且还可能是动态的话使用配置文件就不是个好主意了。这个时候往往需要寻找一种集中管理配置的方法，我们在这个集中的地方修改了配置，所有对这个配置感兴趣的都可以获得变更。比如我们可以把配置放在数据库里，然后所有需要配置的服务都去这个数据库读取配置。但是，因为很多服务的正常运行都非常依赖这个配置，所以需要这个集中提供配置服务的服务具备很高的可靠性。一般我们可以用一个集群来提供这个配置服务，但是用集群提升可靠性，那如何保证配置在集群中的一致性呢？这个时候就需要使用一种实现了一致性协议的服务了。**Zookeeper** 就是这种服务，它使用 **Zab** 这种一致性协议来提供一致性。现在有很多开源项目使用 **Zookeeper** 来维护配置，比如在 **HBase** 中，客户端就是连接一个 **Zookeeper**，获得必要的 **HBase** 集群的配置信息，然后才可以进一步操作。还有在开源的消息队列 **Kafka** 中，也使用 **Zookeeper** 来维护 **broker** 的信息。在 **Alibaba** 开源的 **SOA** 框架 **Dubbo** 中也广泛的使用 **Zookeeper** 管理一些配置来实现服务治理。

2、名字服务

名字服务这个就很好理解了。比如为了通过网络访问一个系统，我们得知道对方的 **IP** 地址，但是 **IP** 地址对人非常不友好，这个时候我们就需要使用域名来访问。但是计算机是不能是别域名的。怎么办呢？如果我们每台机器里都备有一份域名到 **IP** 地址的映射，这个倒是能解决一部分问题，但是如果域名对应的 **IP** 发生了变化了又该怎么办呢？于是我们有了 **DNS** 这个东西。我们只需要访问一个大家熟知的(**known**)的点，它就会告诉你这个域名对应的 **IP** 是什么。在我们的应用中也会存在很多这类问题，特别是在我们的服务特别多的时候，如果我们在本地保存服务的地址的时候将非常不方便，但是如果只需要访问一个大家都熟知的访问点，这里提供统一的入口，那么维护起来将方便得多了。

3、分布式锁

其实在第一篇文章中已经介绍了 **Zookeeper** 是一个分布式协调服务。这样我们就可以利用 **Zookeeper** 来协调多个分布式进程之间的活动。比如在一个分布式环境中，为了提高可靠性，我们的集群的每台服务器上部署着同样的服务。但是，一件事情如果集群中的每个服务器都进行的话，那相互之间就要协调，编程起来将非常复杂。而如果我们只让一个服务进行操作，那又存在单点。通常还有一种做法就是使用分布式锁，在某个时刻只让一个服务去干活，当这台服务出问题的时候锁释放，立即 **fail over** 到另外的服务。这在很多分布式系统中都是这么做，这种设计有一个更好听的名字叫 **Leader Election**(**leader** 选举)。比如 **HBase** 的 **Master** 就是采用这种机制。但要注意的是分布式锁跟同一个进程的锁还是有区别的，所以使用的时候要比同一个进程里的锁更谨慎的使用。

4、集群管理

在分布式的集群中，经常会由于各种原因，比如硬件故障，软件故障，网络问题，有些节点会进进出出。有新的节点加入进来，也有老的节点退出集群。这个时候，集群中其他机器需要感知到这种变化，然后根据这种变化做出对应的决策。比如我们是一个分布式存储系统，有一个中央控制节点负责存储的分配，当有新的存储进来的时候我们要根据现在集群目前的状态来分配存储节点。这个时候我们就需要动态感知到集群目前的状态。还有，比如一个分布式的 **SOA** 架构中，服务是一个集群提供的，当消费者访问某个服务时，就需要采用某种机制发现现在有哪些节点可以提供该服务(这也称之为服务发现，比如 **Alibaba** 开源的 **SOA** 框架 **Dubbo** 就采用了 **Zookeeper** 作为服务发现的底层机制)。还有开源的 **Kafka** 队列就采用了 **Zookeeper** 作为 **Cosnumer** 的上下线管理。

二、**zookeeper** 集群安装

1. 下载安装包

<http://apache.fayea.com/zookeeper/zookeeper-3.4.6/>

2. 解压压缩包

```
[root@localhost mysoft]# tar -zxvf zookeeper-3.4.6.tar.gz
[root@localhost mysoft]# ll
total 310526
```

3. 创建 solrcloud 文件夹，将解压的 zookeeper 包拷贝到 solrcloud 文件夹中

```
[root@localhost local]# mkdir solrcloud
[root@localhost local]# pwd
/usr/local
[root@localhost local]#
```

4. 将 zookeeper-3.4.6 拷贝到/usr/local/solrcloud 下，复制三份分别并将目录名改为 zookeeper1、zookeeper2、zookeeper3

```
[root@localhost solrcloud]# cp -r /opt/mysoft/zookeeper-3.4.6 /usr/local/solrcloud/
[root@localhost solrcloud]# mv zookeeper-3.4.6/ zookeeper1
[root@localhost solrcloud]# cp -r zookeeper1/ zookeeper2
[root@localhost solrcloud]# cp -r zookeeper1/ zookeeper3
[root@localhost solrcloud]#
```

5. 进入 zookeeper1 文件夹，创建 data 目录。并在 data 目录中创建一个 myid 文件内容为“1”（echo 1 >> data/myid）。

```
[root@localhost zookeeper1]# mkdir data
[root@localhost zookeeper1]# echo 1 >> data/myid
[root@localhost zookeeper1]# cat data/myid
1
```

6. 进入 conf 文件夹，把 zoo_sample.cfg 改名为 zoo.cfg

```
[root@localhost zookeeper1]# cd conf
[root@localhost conf]# ll
total 12
-rw-r--r--. 1 root root 535 Mar 20 19:38 configuration.xml
-rw-r--r--. 1 root root 2161 Mar 20 19:38 log4j.properties
-rw-r--r--. 1 root root 922 Mar 20 19:38 zoo_sample.cfg
[root@localhost conf]# mv zoo_sample.cfg zoo.cfg
[root@localhost conf]#
```

7. 修改 zoo.cfg

dataDir=/usr/local/solrcloud/zookeeper1/data

clientPort=2181（zookeeper2 中为 2182、zookeeper3 中为 2183）

添加：

server.1=192.168.198.130:2881:3881

server.2=192.168.198.130:2882:3882

server.3=192.168.198.130:2883:3883

```
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sake.
dataDir=/usr/local/solrcloud/zookeeper1/data
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
server.1=192.168.198.130:2881:3881
server.2=192.168.198.130:2882:3882
server.3=192.168.198.130:2883:3883
```

创建的数据目录全路径

客户端连接端口

集群中节点的信息，包括ip地址及投票、选举的端口

8.对 zookeeper2、3 中的设置做第二步至第四步修改。

zookeeper2:

myid 内容为 2

dataDir=/usr/local/solrcloud/zookeeper2/data

clientPort=2182

Zookeeper3:

myid 内容为 3

dataDir=/usr/local/solrcloud/zookeeper3/data

clientPort=2183

9.启动三个 zookeeper，查看状态

启动三个 zookeeper

/usr/local/solrcloud/zookeeper1/bin/zkServer.sh start

/usr/local/solrcloud/zookeeper2/bin/zkServer.sh start

/usr/local/solrcloud/zookeeper3/bin/zkServer.sh start

查看集群状态:

/usr/local/solrcloud/zookeeper1/bin/zkServer.sh status

/usr/local/solrcloud/zookeeper2/bin/zkServer.sh status

/usr/local/solrcloud/zookeeper3/bin/zkServer.sh status

```
[root@localhost solrcloud]# ./zookeeper1/bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/solrcloud/zookeeper1/bin/./conf/zoo.cfg
Mode: follower
[root@localhost solrcloud]# ./zookeeper2/bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/solrcloud/zookeeper2/bin/./conf/zoo.cfg
Mode: leader
[root@localhost solrcloud]# ./zookeeper3/bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/solrcloud/zookeeper3/bin/./conf/zoo.cfg
Mode: follower
[root@localhost solrcloud]#
```

在 CentOS7 上安装 Zookeeper-3.4.9 服务 <http://www.linuxidc.com/Linux/2016-09/135052.htm>

ZooKeeper 官方文档翻译——ZooKeeper Overview 3.4.6 <http://www.linuxidc.com/Linux/2016-12/138025.htm>

CentOS 下 ZooKeeper 3.4.8 集群环境搭建 <http://www.linuxidc.com/Linux/2016-12/137958.htm>

CentOS 7 下 Zookeeper 集群安装 <http://www.linuxidc.com/Linux/2017-01/139733.htm>

ZooKeeper 学习总结 <http://www.linuxidc.com/Linux/2016-07/133179.htm>

Linux 下安装 Zookeeper 集群 <http://www.linuxidc.com/Linux/2017-01/139545.htm>

Zookeeper3.4.6 的安装 <http://www.linuxidc.com/Linux/2015-05/117697.htm>

CentOS 7 下安装 Zookeeper 单机版 <http://www.linuxidc.com/Linux/2015-05/117697.htm>

本文永久更新链接地址: <http://www.linuxidc.com/Linux/2017-03/141400.htm>

欢迎点击这里的链接进入精彩的[Linux公社](http://www.linuxidc.com)网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

[Linux公社](http://www.linuxidc.com)是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：[Ubuntu 专题](#) [Fedora 专题](#) [Android 专题](#) [Oracle 专题](#) [Hadoop 专题](#)
[RedHat 专题](#) [SUSE 专题](#) [红旗 Linux 专题](#) [CentOS 专题](#)



Linux 公社微信公众号：[linuxidc_com](#)

Linuxidc.com

微信扫一扫

订阅专业的最新Linux资讯及开源技术教程。

搜索微信公众号：[linuxidc_com](#)

