

solr 集群搭建

这里我实现使用4个 solr 服务实现集群，

系统环境为：CentOS-6.7 + jdk-1.8 + tomcat-8.0 + solr-4.10.3 + zookeeper-3.4.10

1. Solr的集群

- a) 需要四个节点
- b) 整个core分成两个分片，每个分片都由两个服务器组成，master和slave
- c) 每个Tomcat有一个solr服务

2. zookeeper集群

- a) zookeeper需要保证高可用，需要搭建集群
- b) zookeeper需要进行投票和选举机制，最少需要三个节点
- c) zookeeper的三个节点，注意修改端口号不一样

1 zookeeper 集群搭建

需要做的步骤如下：

1. 创建 3 个 zookeeper
2. 配置为每个 zookeeper 创建data 文件夹
3. 为每个 zookeeper 创建 myid 文件，并输入内容
4. 修改每个 zookeeper 的配置文件

这里还是使用 shell 编程来实现上述步骤，

zookeeper 压缩包的路径为 /usr/local/src/zookeeper-3.4.10.tar.gz

1 创建 zookeeper-cluster.sh 文件,输入如下内容

```
echo "#####"
echo "#####"
echo "#####      蜂蜜110 ~~~ fmi110      #####"
echo "#####"
echo "#####"

PACKAGE_LOCATION="/usr/local/src/zookeeper-3.4.10.tar.gz"
BASE_DIR="/usr/local/src/solr-cloud"
# zookeeper 解压后得到的文件夹名，要根据实际情况修改该值
ZooKeeperDIR="zookeeper-3.4.10"

# 设置 ip 地址和端口号，需要根据实际情况设置
SERVER_1=10.211.55.8:2882:3882
SERVER_2=10.211.55.8:2883:3883
```

```

SERVER_3=10.211.55.8:2884:3884

mkdir -p $BASE_DIR
# 解压源码包
tar -zxf $PACKAGE_LOCATION -C $BASE_DIR
# 切换到解压目录下
cd $BASE_DIR

# 创建 3 个 zookeeper 并修改配置
for((i=1;i<=3;i++));
do
    mkdir -p zookeeper0${i}/data
    cp -r $ZooKeeperDIR/* zookeeper0${i}/
    # 创建 myid 并添加内容
    echo $i > zookeeper0${i}/data/myid
    mv zookeeper0${i}/conf/zoo_sample.cfg zookeeper0${i}/conf/zoo.cfg

    # 修改配置文件的 dataDir 属性 , dataDir=... 后的路径需要根据实际情况作对应的修改
    sed -i "s/dataDir=.*dataDir=/usr/local/src/solr-
cloud/zookeeper0${i}/data/" zookeeper0${i}/conf/zoo.cfg
    # 修改客户端端口 clientPort
    sed -i "s/clientPort=.*clientPort=318${i}/" zookeeper0${i}/conf/zoo.cfg

    # 追加 ip 地址和端口号
    echo "server.1=$SERVER_1" >> zookeeper0${i}/conf/zoo.cfg
    echo "server.2=$SERVER_2" >> zookeeper0${i}/conf/zoo.cfg
    echo "server.3=$SERVER_3" >> zookeeper0${i}/conf/zoo.cfg

    #echo ">>>> zookeeper$i 配置文件路径 :" /usr/local/src/solr-
cloud/zookeeper0${i}/data

    echo "启动命令为 : $BASE_DIR/zookeeper0${i}/bin/zkServer.sh start "
    echo "查看运行状态命令 : $BASE_DIR/zookeeper0${i}/bin/zkServer.sh status "
    echo "客户端端口号 clientPort=318${i}"
    echo ""
done

echo ">>>> 配置的 ip 地址和端口号为 :"
echo $SERVER_1
echo $SERVER_2
echo $SERVER_3
echo "配置完成.....开始启动 zookeeper..."

$BASE_DIR/zookeeper01/bin/zkServer.sh start
$BASE_DIR/zookeeper02/bin/zkServer.sh start
$BASE_DIR/zookeeper03/bin/zkServer.sh start

#echo "-----查看运行状态-----"

```

```
$BASE_DIR/zookeeper01/bin/zkServer.sh status
$BASE_DIR/zookeeper02/bin/zkServer.sh status
$BASE_DIR/zookeeper03/bin/zkServer.sh status
```

注意 `SERVER_X` 变量指定了暴露的端口号

2 为 zookeeper-cluster.sh 添加可执行权限,并执行

```
[root@centos-6 src]# chmod a+x zookeeper-cluster.sh
[root@centos-6 src]# ./zookeeper-cluster.sh
```

2 安装单机版 solr

略, 参考网上

3 solr 集群搭建

单机版的 solr 安装好,我这里的安装目录是 `/usr/local/src/solr-tomcat`, solrhome 目录路径是 `/usr/local/src/solrhome`

下面开始搭建集群:

1. 复制 4 个单机版的 solr 服务, 修改配置
2. 修改 tomcat 的端口号

tomcat-solr01 的访问端口修改为 8180, tomcat-solr02 的访问端口修改为 8280, 依次类推

3. 修改 solr 应用的 web.xml
4. 新建 solr-cloud 目录, 存放四个 solr 服务的 solrhome 目录, 并修改 solr.xml, 指定端口号

端口号与对应的 tomcat 访问端口号对应

5. 上传 solr 配置文件到 zookeeper

上传的脚本位于 solr 的解压目录下,具体路劲为

```
solr-4.10.3/example/scripts/cloud-scripts/zkcli.sh"
```

`zkcli.sh` 内容如下:

```
#!/usr/bin/env bash
JVM="java"
# Find location of this script
sdir="`dirname \"$0\"`"

if [ ! -d "$sdir/../../solr-webapp/webapp" ]; then
    unzip $sdir/../../webapps/solr.war -d $sdir/../../solr-webapp/webapp
fi

PATH=$JAVA_HOME/bin:$PATH $JVM -
Dlog4j.configuration=file:$sdir/log4j.properties -classpath "$sdir/../../solr-
webapp/webapp/WEB-INF/lib/*:$sdir/../../lib/ext/*" org.apache.solr.cloud.ZkCLI
${1+"$@"}
```

1 创建 solr-cluster.sh 文件

```
[root@centos-6 src]# vim solr-cluster.sh
```

2 在 solr-cluster.sh 文件中输入如下内容并保存退出

```
echo "#####"
echo "#####"
echo "#####      蜂蜜110 ~~~ fmi110      #####"
echo "#####"
echo "#####"

# 将基础目录设置为 /usr/local/src , 程序都在该目录下安装
BASE_DIR="/usr/local/src"
SOLR_CLOUD_HOME="/usr/local/src/solr-cloud"
# 单机版 solr 的安装路径(即 solr 和 tomcat 整合后的 tomcat路径)
SOLR_TOMCAT_HOME="/usr/local/src/solr-tomcat"
# 单机版的 solrhome 的路径
SOLR_HOME="/usr/local/src/solrhome"

# 上传脚本的路径 , 在 solr 的解压包里边
RUN_SCRIPT="/usr/local/src/solr-4.10.3/example/scripts/cloud-scripts/zkcli.sh"
# zookeeper集群 所在主机的 ip 地址和 zk客户端的端口号
ZK_HOST=10.211.55.8:3181,10.211.55.8:3182,10.211.55.8:3183

echo ">>>>>>>>> 单机版的 solr 应用的所在路径为 : " $SOLR_TOMCAT_HOME

# 创建 solr-cloud 目录
mkdir -p $SOLR_CLOUD_HOME
echo "mkdir -p $SOLR_CLOUD_HOME"

# 复制 4个 tomcat 和 4 个 solrhome
for((i=1;i<=4;i++));
```

```

do
    echo ">>>>>>>>> 创建  $SOLR_CLOUD_HOME/tomcat-solr0$i"
    # 1 复制 tomcat
    cp -r $SOLR_TOMCAT_HOME $SOLR_CLOUD_HOME/tomcat-solr0$i

    echo ">>>>>>>>> 修改 tomcat-solr0${i} 端口号 8005 --> 8${i}05"
    echo ">>>>>>>>> 修改 tomcat-solr0${i} 端口号 8080 --> 8${i}80"
    echo ">>>>>>>>> 修改 tomcat-solr0${i} 端口号 8009 --> 8${i}09"
    # 2 修改 server.xml 文件第 15-25 行的 8005 替换为 8105 , 依次类推 8080 -> 8180 ,
    8009 -> 8109
    sed -i "18,26s/8005/8${i}05/g" $SOLR_CLOUD_HOME/tomcat-solr0$i/conf/server.xml
    sed -i "65,73s/8080/8${i}80/g" $SOLR_CLOUD_HOME/tomcat-solr0$i/conf/server.xml
    sed -i "87,95s/8009/8${i}09/g" $SOLR_CLOUD_HOME/tomcat-solr0$i/conf/server.xml

    # 3 复制 solrhome
    cp -r $SOLR_HOME $SOLR_CLOUD_HOME/solrhome0$i

    # 4 修改tomcat 中 solr 应用的 web.xml
    temp="<env-entry-value>\usr\local\src\solr-cloud\solrhome0${i}<\env-
entry-value>"

    sed -i "s/<env-entry-value>.*<\env-entry-value>/$temp/g"
    $SOLR_CLOUD_HOME/tomcat-solr0$i/webapps/solr/WEB-INF/web.xml

    # 5 设置 zookeeper 地址给 tomcat
    echo ""
    echo ">>> zookeeper 地址添加到 catalina.sh 文件 : JAVA_OPTS=\"-
DzkHost=$ZK_HOST\""
    # echo JAVA_OPTS="-DzkHost=$ZK_HOST"    $SOLR_CLOUD_HOME/tomcat-
solr0$i/bin/catalina.sh
    sed -i "s/#JAVA_OPTS=\"\${JAVA_OPTS} -Dorg.* /JAVA_OPTS=\"-DzkHost=$ZK_HOST\"/g"
    $SOLR_CLOUD_HOME/tomcat-solr0$i/bin/catalina.sh

    # 6 修改 solrhome 中的solr.xml , 指定端口
    sed -i "s/8983/8${i}80/g" $SOLR_CLOUD_HOME/solrhome0$i/solr.xml

    # 7 上传 solr 配置文件到 zookeeper
    echo ""
    echo "solrhome0$i 上传 solr 配置文件到 zookeeper..."
    $RUN_SCRIPT -zkhost $ZK_HOST -cmd upconfig -confdir
    $SOLR_CLOUD_HOME/solrhome0$i/collection1/conf -confname myconf
    echo ""
done

```

3 给 solr-cluster.sh 添加可执行权限

```
[root@centos-6 src]# chmod a+x solr-cluster.sh
```

4 运行 solr-cluster.sh

```
[root@centos-6 src]# ./solr-cluster.sh
```

4 验证

启动4个配置好的 tomcat , 然后通过浏览器访问即可任意一个 tomcat 下的 solr 工程

出现这个 说明 solr集群搭建成功

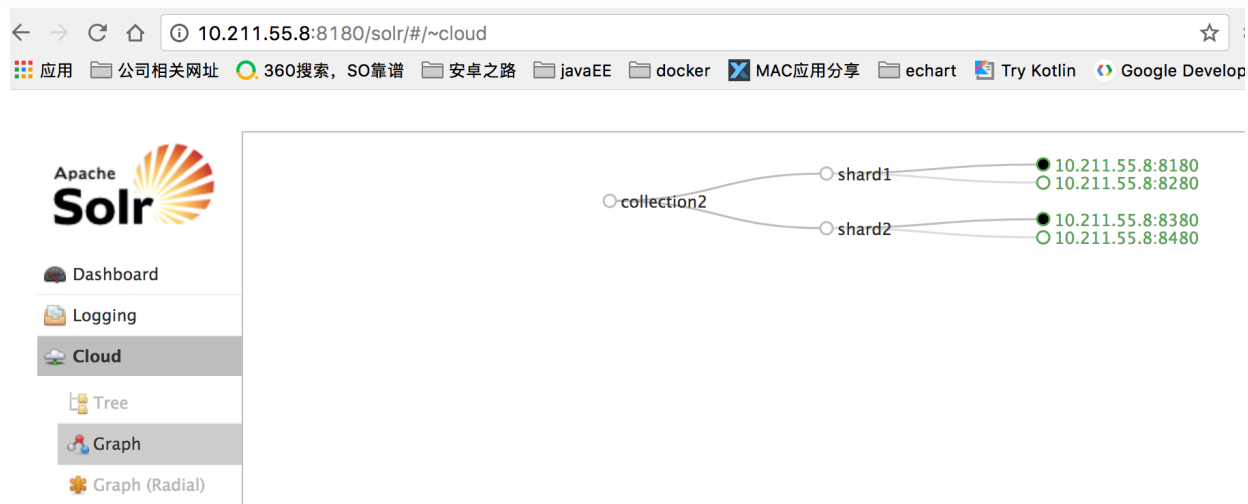
如上图所示,使用上述的上述的配置创建出来的 solr 集群 collection1 只有一个分片 shard1 , 下面我们通过 solr 提供的 api 创建集群 collection2 , 并且分两个分片 : 在浏览器输入下面的地址即可

<http://10.211.55.8:8180/solr/admin/collections?action=CREATE&name=collection2&numShards=2&replicationFactor=2>

删除 collection1 的请求如下 :

<http://10.211.55.8:8180/solr/admin/collections?action=DELETE&name=collection1>

最终效果如下:



如果觉得笔记不错,扫码鼓励下吧,两毛也是爱,O(∩_∩)O~~~~*

