（默认安装JDK和scala）

（HA模式需要安装zookeeper，所以前置条件为zookeeper已搭建成功）

**1.配置环境变量**

JAVA\_HOME=/opt/client/JDK/jdk

SCALA\_HOME=/opt/client/Scala/scala

HADOOP\_HOME=/opt/client/Hadoop/hadoop

HADOOP\_PID\_DIR=/opt/client/Hadoop/pids

HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

HADOOP\_OPTS="$HADOOP\_OPTS -Djava.library.path=$HADOOP\_HOME/lib/native"

HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

HADOOP\_COMMON\_HOME=$HADOOP\_HOME

HADOOP\_HDFS\_HOME=$HADOOP\_HOME

YARN\_HOME=$HADOOP\_HOME

HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

HDFS\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native

ZOO\_HOME=/opt/client/Zookeeper/zookeeper

ZOO\_LOG\_DIR=$ZOO\_HOME/logs

PATH=$PATH:/opt/client/JDK/jdk/bin:/opt/client/Scala/scala/bin:/opt/client/Hadoop/hadoo p/bin:/opt/client/Hadoop/hadoop/sbin:$ZOO\_HOME/bin

**2.创建相应目录**

mkdir -p /opt/client/Hadoop/{pids,storage}

mkdir -p /opt/client/Hadoop/storage/{hdfs,tmp,journal}

mkdir -p /opt/client/Hadoop/storage/tmp/nodemanager/{local,remote,logs}

mkdir -p /opt/hzgc/client/Hadoop/storage/hdfs/{name,data}

**3.修改配置文件**

【core-site.xml】

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://hacluster</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/opt/client/Hadoop/storage/tmp</value>

</property>

<property>

<name>ha.zookeeper.quorum</name>

<value>192.168.2.145:2181,192.168.2.160:2181,192.168.2.146:2181,192.168.2.152:2181</value>

</property>

<property>

<name>ha.zookeeper.session-timeout.ms</name>

<value>2000</value>

</property>

<property>

<name>fs.trash.interval</name>

<value>4320</value>

</property>

<property>

<name>hadoop.http.staticuser.use</name>

<value>root</value>

</property>

<property>

<name>hadoop.proxyuser.hadoop.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hadoop.groups</name>

<value>\*</value>

</property>

<property>

<name>hadoop.native.lib</name>

<value>true</value>

</property>

</configuration>

【hdfs-site.xml】

<configuration>

<property>

<name>dfs.namenode.name.dir</name>

<value>/opt/client/Hadoop/storage/hdfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/opt/client/Hadoop/storage/hdfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.blocksize</name>

<value>67108864</value>

</property>

<property>

<name>dfs.datanode.du.reserved</name>

<value>10737418240</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

<property>

<name>dfs.permissions</name>

<value>true</value>

</property>

<property>

<name>dfs.permissions.enabled</name>

<value>true</value>

</property>

<property>

<name>dfs.nameservices</name>

<value>hacluster</value>

</property>

<property>

<name>dfs.ha.namenodes.hacluster</name>

<value>nn1,nn2</value>

</property>

<property>

<name>dfs.namenode.rpc-address.hacluster.nn1</name>

<value>s100:8020</value>

</property>

<property>

<name>dfs.namenode.rpc-address.hacluster.nn2</name>

<value>s101:8020</value>

</property>

<property>

<name>dfs.namenode.servicerpc-address.hacluster.nn1</name>

<value>s100:53310</value>

</property>

<property>

<name>dfs.namenode.servicerpc-address.hacluster.nn2</name>

<value>s101:53310</value>

</property>

<property>

<name>dfs.namenode.http-address.hacluster.nn1</name>

<value>s100:50070</value><!-- 该处不建议占掉8080端口-->

</property>

<property>

<name>dfs.namenode.http-address.hacluster.nn2</name>

<value>s101:50070</value>

</property>

<property>

<name>dfs.datanode.http.address</name>

<value>0.0.0.0:50070</value>

</property>

<property>

<name>dfs.namenode.shared.edits.dir</name>

<value>qjournal://s100:8485;s101:8485;s102:8485;s103:8485/hacluster</value>

</property>

<property>

<name>dfs.client.failover.proxy.provider.hacluster</name>

<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>

</property>

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence</value>

</property>

<property>

<name>dfs.ha.fencing.ssh.private-key-files</name>

<value>/root/.ssh/id\_rsa</value>

</property>

<property>

<name>dfs.ha.fencing.ssh.connect-timeout</name>

<value>30000</value>

</property>

<property>

<name>dfs.journalnode.edits.dir</name>

<value>/opt/client/Hadoop/storage/hdfs/journal</value>

</property>

<property>

<name>dfs.ha.automatic-failover.enabled</name>

<value>true</value>

</property>

<property>

<name>ha.failover-controller.cli-check.rpc-timeout.ms</name>

<value>60000</value>

</property>

<property>

<name>ipc.client.connect.timeout</name>

<value>60000</value>

</property>

<property>

<name>dfs.image.transfer.bandwidthPerSec</name>

<value>41943040</value>

</property>

<property>

<name>dfs.namenode.accesstime.precision</name>

<value>3600000</value>

</property>

<property>

<name>dfs.datanode.max.transfer.threads</name>

<value>4096</value>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>s103:50090</value>

</property>

</configuration>

【yarn-site.xml】注意下面配置中的注释信息

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>s100:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>s100:8031</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>s100:8032</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>s100:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>s100:80</value>

</property>

<property>

<name>yarn.nodemanager.hostname</name>

<value>s100</value> <!-- 每个slave应该对应自己的hostName-->

<description>the nodemanagers bind to this port</description>

</property>

<property>

<name>yarn.nodemanager.webapp.address</name>

<value>${yarn.nodemanager.hostname}:80</value>

</property>

<property>

<name>yarn.nodemanager.address</name>

<value>${yarn.nodemanager.hostname}:8034</value>

</property>

<property>

<name>yarn.nodemanager.local-dirs</name>

<value>${hadoop.tmp.dir}/nodemanager/local</value>

</property>

<property>

<name>yarn.nodemanager.remote-app-log-dir</name>

<value>${hadoop.tmp.dir}/nodemanager/remote</value>

</property>

<property>

<name>yarn.nodemanager.log-dirs</name>

<value>${hadoop.tmp.dir}/nodemanager/logs</value>

</property>

</configuration>

【mapred-site.xml】

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>s100:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>s100:19888</value>

</property>

<property>

<name>mapreduce.application.classpath</name>

<value>

/opt/client/Hadoop/hadoop/etc/hadoop,

/opt/client/Hadoop/hadoop/share/hadoop/common/\*,

/opt/client/Hadoop/hadoop/share/hadoop/common/lib/\*,

/opt/client/Hadoop/hadoop/share/hadoop/hdfs/\*,

/opt/client/Hadoop/hadoop/share/hadoop/hdfs/lib/\*,

/opt/client/Hadoop/hadoop/share/hadoop/mapreduce/\*,

/opt/client/Hadoop/hadoop/share/hadoop/mapreduce/lib/\*,

/opt/client/Hadoop/hadoop/share/hadoop/yarn/\*,

/opt/client/Hadoop/hadoop/share/hadoop/yarn/lib/\*

</value>

</property>

</configuration>

【slaves】

建议不要将主备两个节点当做子节点

**4.集群启动**

【hdfs】

1. 在namenode1上执行，创建命名空间

$>hdfs zkfc -formatZK

2. 格式化主NameNode节点（主节点）

$>hdfs namenode -format

3. 格式备NameNode节点（备节点）

 $>hdfs namenode -bootstrapStandby

4.通过脚本启动namenode以及datanode（启动后需通过日志检查各个节点的启动情况， 有可能出现jps查看已启动实际未启动的状况）

$>start-dfs.sh

【yarn】

1.启动yarn

$>start-yarn

**5.测试Yarn是否可用**

1.$> hdfs dfs -put/usr/local/hadoop/etc/hadoop/yarn-site.xml /tmp

2.hadoop jar ./hadoop-mapreduce-examples-2.2.0.jar wordcount /tmp/yarn-site.xml /mytest

6.HDF的HA功能测试



