### 分布式环境搭建（适用于工作当中正式环境搭建）

## 集群规划

使用完全分布式，实现namenode高可用，ResourceManager的高可用

集群运行服务规划

|  |  |  |  |
| --- | --- | --- | --- |
|  | node1 | node2 | node3 |
| zookeeper | zk | zk | zk |
| HDFS | JournalNode | JournalNode | JournalNode |
| NameNode | NameNode |  |
| ZKFC | ZKFC |  |
| DataNode | DataNode | DataNode |
| YARN |  | ResourceManager | ResourceManager |
| NodeManager | NodeManager | NodeManager |
| MapReduce |  |  | JobHistoryServer |

## 安装包解压

停止之前的hadoop集群的所有服务，然后重新解压编译后的hadoop压缩包

解压压缩包

node1机器执行以下命令进行解压

|  |
| --- |
| **mkdir** -p **/**opt**/**software  **mkdir** -p **/**opt**/**server  **cd** **/**opt**/**software  **tar** **-**zxvf hadoop-2.7.5.tar.gz -C **/export/server/**  **cd** **/export/server/**hadoop-2.7.5/etc/hadoop |

## 配置文件的修改

以下操作都在node1机器上进行

### 修改core-site.xml

|  |
| --- |
| <configuration>  <!-- 指定NameNode的HA高可用的zk地址 -->  <property>  <name>**ha.zookeeper.quorum**</name>  <value>**node1:2181,node2:2181,node3:2181**</value>  </property>  <!-- 指定HDFS访问的域名地址 -->  <property>  <name>**fs.defaultFS**</name>  <value>**hdfs://ns**</value>  </property>  <!-- 临时文件存储目录 -->  <property>  <name>**hadoop.tmp.dir**</name>  <value>**/export/server/hadoop-2.7.5/data/tmp**</value>  </property>  <!-- 开启hdfs垃圾箱机制，指定垃圾箱中的文件七天之后就彻底删掉  单位为分钟  -->  <property>  <name>**fs.trash.interval**</name>  <value>**10080**</value>  </property>  </configuration> |

### 修改hdfs-site.xml

注意: 删除原有 50070 删除了 secondaryNamenode所有的配置

|  |
| --- |
| <configuration>  <!-- 指定命名空间 -->  <property>  <name>**dfs.nameservices**</name>  <value>**ns**</value>  </property>  <!-- 指定该命名空间下的两个机器作为我们的NameNode -->  <property>  <name>**dfs.ha.namenodes.ns**</name>  <value>**nn1,nn2**</value>  </property>  <!-- 配置第一台服务器的namenode通信地址 -->  <property>  <name>**dfs.namenode.rpc-address.ns.nn1**</name>  <value>**node1:8020**</value>  </property>  <!-- 配置第二台服务器的namenode通信地址 -->  <property>  <name>**dfs.namenode.rpc-address.ns.nn2**</name>  <value>**node2:8020**</value>  </property>  <!-- 所有从节点之间相互通信端口地址 -->  <property>  <name>**dfs.namenode.servicerpc-address.ns.nn1**</name>  <value>**node1:8022**</value>  </property>  <!-- 所有从节点之间相互通信端口地址 -->  <property>  <name>**dfs.namenode.servicerpc-address.ns.nn2**</name>  <value>**node2:8022**</value>  </property>    <!-- 第一台服务器namenode的web访问地址 -->  <property>  <name>**dfs.namenode.http-address.ns.nn1**</name>  <value>**node1:50070**</value>  </property>  <!-- 第二台服务器namenode的web访问地址 -->  <property>  <name>**dfs.namenode.http-address.ns.nn2**</name>  <value>**node2:50070**</value>  </property>    <!-- journalNode的访问地址，注意这个地址一定要配置 -->  <property>  <name>**dfs.namenode.shared.edits.dir**</name>  <value>**qjournal://node1:8485;node2:8485;node3:8485/ns1**</value>  </property>  <!-- 指定故障自动恢复使用的哪个java类 -->  <property>  <name>**dfs.client.failover.proxy.provider.ns**</name>  <value>**org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider**</value>  </property>    <!-- 故障转移使用的哪种通信机制 -->  <property>  <name>**dfs.ha.fencing.methods**</name>  <value>**sshfence**</value>  </property>    <!-- 指定通信使用的公钥 -->  <property>  <name>**dfs.ha.fencing.ssh.private-key-files**</name>  <value>**/root/.ssh/id\_rsa**</value>  </property>  <!-- journalNode数据存放地址 -->  <property>  <name>**dfs.journalnode.edits.dir**</name>  <value>**/export/server/hadoop-2.7.5/hadoopDatas/dfs/jn**</value>  </property>  <!-- 启用自动故障恢复功能 -->  <property>  <name>**dfs.ha.automatic-failover.enabled**</name>  <value>**true**</value>  </property>  <!-- namenode产生的文件存放路径 -->  <property>  <name>**dfs.namenode.name.dir**</name>  <value>**file:///export/server/hadoop-2.7.5/hadoopDatas/dfs/nn/name**</value>  </property>  <!-- edits产生的文件存放路径 -->  <property>  <name>**dfs.namenode.edits.dir**</name>  <value>**file:///export/server/hadoop-2.7.5/hadoopDatas/dfs/nn/edits**</value>  </property>  <!-- dataNode文件存放路径 -->  <property>  <name>**dfs.datanode.data.dir**</name>  <value>**file:///export/server/hadoop-2.7.5/hadoopDatas/dfs/dn**</value>  </property>  <!-- 关闭hdfs的文件权限 -->  <property>  <name>**dfs.permissions**</name>  <value>**false**</value>  </property>  <!-- 指定block文件块的大小 -->  <property>  <name>**dfs.blocksize**</name>  <value>**134217728**</value>  </property>  </configuration> |

### 修改yarn-site.xml，注意node3与node2配置不同

删除原有的: yarn.resourcemanager.hostname 配置

|  |
| --- |
| <configuration>  <!-- Site specific YARN configuration properties -->  <!-- 是否启用日志聚合.应用程序完成后,日志汇总收集每个容器的日志,这些日志移动到文件系统,例如HDFS. -->  <!-- 用户可以通过配置"yarn.nodemanager.remote-app-log-dir"、"yarn.nodemanager.remote-app-log-dir-suffix"来确定日志移动到的位置 -->  <!-- 用户可以通过应用程序时间服务器访问日志 -->  <!-- 启用日志聚合功能，应用程序完成后，收集各个节点的日志到一起便于查看 -->  <property>  <name>**yarn.log-aggregation-enable**</name>  <value>**true**</value>  </property>    <!--开启resource manager HA,默认为false-->  <property>  <name>**yarn.resourcemanager.ha.enabled**</name>  <value>**true**</value>  </property>  <!-- 集群的Id，使用该值确保RM不会做为其它集群的active -->  <property>  <name>**yarn.resourcemanager.cluster-id**</name>  <value>**mycluster**</value>  </property>  <!--配置resource manager 命名-->  <property>  <name>**yarn.resourcemanager.ha.rm-ids**</name>  <value>**rm1,rm2**</value>  </property>  <!-- 配置第一台机器的resourceManager -->  <property>  <name>**yarn.resourcemanager.hostname.rm1**</name>  <value>**node2**</value>  </property>  <!-- 配置第二台机器的resourceManager -->  <property>  <name>**yarn.resourcemanager.hostname.rm2**</name>  <value>**node3**</value>  </property>  <!-- 配置第一台机器的resourceManager通信地址 -->  <property>  <name>**yarn.resourcemanager.address.rm1**</name>  <value>**node2:8032**</value>  </property>  <property>  <name>**yarn.resourcemanager.scheduler.address.rm1**</name>  <value>**node2:8030**</value>  </property>  <property>  <name>**yarn.resourcemanager.resource-tracker.address.rm1**</name>  <value>**node2:8031**</value>  </property>  <property>  <name>**yarn.resourcemanager.admin.address.rm1**</name>  <value>**node2:8033**</value>  </property>  <property>  <name>**yarn.resourcemanager.webapp.address.rm1**</name>  <value>**node2:8088**</value>  </property>  <!-- 配置第二台机器的resourceManager通信地址 -->  <property>  <name>**yarn.resourcemanager.address.rm2**</name>  <value>**node3:8032**</value>  </property>  <property>  <name>**yarn.resourcemanager.scheduler.address.rm2**</name>  <value>**node3:8030**</value>  </property>  <property>  <name>**yarn.resourcemanager.resource-tracker.address.rm2**</name>  <value>**node3:8031**</value>  </property>  <property>  <name>**yarn.resourcemanager.admin.address.rm2**</name>  <value>**node3:8033**</value>  </property>  <property>  <name>**yarn.resourcemanager.webapp.address.rm2**</name>  <value>**node3:8088**</value>  </property>  <!--开启resourcemanager自动恢复功能-->  <property>  <name>**yarn.resourcemanager.recovery.enabled**</name>  <value>**true**</value>  </property>  <!--在node2上配置rm1,在node3上配置rm2,注意：一般都喜欢把配置好的文件远程复制到其它机器上，但这个在YARN的另一个机器上一定要修改，其他机器上不配置此项-->  <property>  <name>**yarn.resourcemanager.ha.id**</name>  <value>**rm1**</value>  <description>**If we want to launch more than one RM in single node, we need this configuration**</description>  </property>    <!--用于持久存储的类。尝试开启-->  <property>  <name>**yarn.resourcemanager.store.class**</name>  <value>**org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore**</value>  </property>  <property>  <name>**yarn.resourcemanager.zk-address**</name>  <value>**node2:2181,node3:2181,node1:2181**</value>  <description>**For multiple zk services, separate them with comma**</description>  </property>  <!--开启resourcemanager故障自动切换，指定机器-->  <property>  <name>**yarn.resourcemanager.ha.automatic-failover.enabled**</name>  <value>**true**</value>  <description>**Enable automatic failover; By default, it is enabled only when HA is enabled.**</description>  </property>  <property>  <name>**yarn.client.failover-proxy-provider**</name>  <value>**org.apache.hadoop.yarn.client.ConfiguredRMFailoverProxyProvider**</value>  </property>  <!-- 允许分配给一个任务最大的CPU核数，默认是8 -->  <property>  <name>**yarn.nodemanager.resource.cpu-vcores**</name>  <value>**2**</value>  </property>  <!-- 每个节点可用内存,单位MB -->  <property>  <name>**yarn.nodemanager.resource.memory-mb**</name>  <value>**2048**</value>  </property>  <!-- 单个任务可申请最少内存，默认1024MB -->  <property>  <name>**yarn.scheduler.minimum-allocation-mb**</name>  <value>**1024**</value>  </property>  <!-- 单个任务可申请最大内存，默认8192MB -->  <property>  <name>**yarn.scheduler.maximum-allocation-mb**</name>  <value>**2048**</value>  </property>  <!--多长时间聚合删除一次日志 此处-->  <property>  <name>**yarn.log-aggregation.retain-seconds**</name>  <value>**2592000**</value><!--30 day-->  </property>  <!--时间在几秒钟内保留用户日志。只适用于如果日志聚合是禁用的-->  <property>  <name>**yarn.nodemanager.log.retain-seconds**</name>  <value>**604800**</value><!--7 day-->  </property>  <!--指定文件压缩类型用于压缩汇总日志-->  <property>  <name>**yarn.nodemanager.log-aggregation.compression-type**</name>  <value>**gz**</value>  </property>  <!-- nodemanager本地文件存储目录-->  <property>  <name>**yarn.nodemanager.local-dirs**</name>  <value>**/export/server/hadoop-2.7.5/yarn/local**</value>  </property>  <!-- resourceManager 保存最大的任务完成个数 -->  <property>  <name>**yarn.resourcemanager.max-completed-applications**</name>  <value>**1000**</value>  </property>  <!-- 逗号隔开的服务列表，列表名称应该只包含a-zA-Z0-9\_,不能以数字开始-->  <property>  <name>**yarn.nodemanager.aux-services**</name>  <value>**mapreduce\_shuffle**</value>  </property>  <!--rm失联后重新链接的时间-->  <property>  <name>**yarn.resourcemanager.connect.retry-interval.ms**</name>  <value>**2000**</value>  </property>  </configuration> |

### 修改mapred-site.xml

|  |
| --- |
| <configuration>  <!--指定运行mapreduce的环境是yarn -->  <property>  <name>**mapreduce.framework.name**</name>  <value>**yarn**</value>  </property>  <!-- MapReduce JobHistory Server IPC host:port -->  <property>  <name>**mapreduce.jobhistory.address**</name>  <value>**node3:10020**</value>  </property>  <!-- MapReduce JobHistory Server Web UI host:port -->  <property>  <name>**mapreduce.jobhistory.webapp.address**</name>  <value>**node03:19888**</value>  </property>  <!-- The directory where MapReduce stores control files.默认 ${hadoop.tmp.dir}/mapred/system -->  <property>  <name>**mapreduce.jobtracker.system.dir**</name>  <value>**/export/server/hadoop-2.7.5/data/system/jobtracker**</value>  </property>  <!-- The amount of memory to request from the scheduler for each map task. 默认 1024-->  <property>  <name>**mapreduce.map.memory.mb**</name>  <value>**1024**</value>  </property>  <!-- <property>  <name>mapreduce.map.java.opts</name>  <value>-Xmx1024m</value>  </property> -->  <!-- The amount of memory to request from the scheduler for each reduce task. 默认 1024-->  <property>  <name>**mapreduce.reduce.memory.mb**</name>  <value>**1024**</value>  </property>  <!-- <property>  <name>mapreduce.reduce.java.opts</name>  <value>-Xmx2048m</value>  </property> -->  <!-- 用于存储文件的缓存内存的总数量，以兆字节为单位。默认情况下，分配给每个合并流1MB，给个合并流应该寻求最小化。默认值100-->  <property>  <name>**mapreduce.task.io.sort.mb**</name>  <value>**100**</value>  </property>    <!-- <property>  <name>mapreduce.jobtracker.handler.count</name>  <value>25</value>  </property>-->  <!-- 整理文件时用于合并的流的数量。这决定了打开的文件句柄的数量。默认值10-->  <property>  <name>**mapreduce.task.io.sort.factor**</name>  <value>**10**</value>  </property>  <!-- 默认的并行传输量由reduce在copy(shuffle)阶段。默认值5-->  <property>  <name>**mapreduce.reduce.shuffle.parallelcopies**</name>  <value>**15**</value>  </property>  <property>  <name>**yarn.app.mapreduce.am.command-opts**</name>  <value>**-Xmx1024m**</value>  </property>  <!-- MR AppMaster所需的内存总量。默认值1536-->  <property>  <name>**yarn.app.mapreduce.am.resource.mb**</name>  <value>**1536**</value>  </property>  <!-- MapReduce存储中间数据文件的本地目录。目录不存在则被忽略。默认值${hadoop.tmp.dir}/mapred/local-->  <property>  <name>**mapreduce.cluster.local.dir**</name>  <value>**/export/server/hadoop-2.7.5/data/system/local**</value>  </property>  </configuration> |

### 修改slaves

|  |
| --- |
| node1  node2  node3 |

### 修改hadoop-env.sh

|  |
| --- |
| export JAVA\_HOME=/export/server/jdk1.8.0\_241 |

## 集群启动过程

将第一台机器的安装包发送到其他机器上

第一台机器执行以下命令：

|  |
| --- |
| **cd** **/export/server**/hadoop-2.7.5/etc/hadoop  scp -r \* node2**:$PWD**  scp -r \* node3**:$PWD** |

更改node3的rm2

第二台机器执行以下命令

|  |
| --- |
| **vim** yarn-site.xml |

<!--在node2上配置rm1,在node3上配置rm2,注意：一般都喜欢把配置好的文件远程复制到其它机器上，

但这个在YARN的另一个机器上一定要修改，其他机器上不配置此项

注意我们现在有两个resourceManager 第二台是rm1 第三台是rm2

这个配置一定要记得去node3上面改好

-->

|  |
| --- |
| <property>  <name>**yarn.resourcemanager.ha.id**</name>  <value>**rm2**</value>  <description>**If we want to launch more than one RM in single node, we need this configuration**</description>  </property> |

##### 启动HDFS过程

node1机器执行以下命令

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  bin**/**hdfs zkfc **-**formatZK  sbin**/**hadoop-daemons.sh start journalnode  bin**/**hdfs namenode **-**initializeSharedEdits **-**force  sbin**/**start-dfs.sh |

node2上面执行

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  bin**/**hdfs namenode **-**bootstrapStandby  sbin**/**hadoop-daemon.sh start namenode |

启动ZKFC

|  |
| --- |
| 在node01机器上面启动zkfc进程  hadoop-daemon.sh start zkfc  在node02机器上面启动zkfc进程  hadoop-daemon.sh start zkfc |

##### 启动yarn过程

node2上执行

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  sbin**/**start-yarn.sh |

node3上面执行

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  sbin**/**start-yarn.sh |

##### 查看resourceManager状态

node2上面执行

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  bin**/**yarn rmadmin **-**getServiceState rm1 |

node3上面执行

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  bin**/**yarn rmadmin **-**getServiceState rm2 |

##### node3启动jobHistory

node3机器执行以下命令启动jobHistory

|  |
| --- |
| **cd** **/export/server/**hadoop-2.7.5  sbin**/**mr-jobhistory-daemon.sh start historyserver |

##### hdfs状态查看

node1机器查看hdfs状态

[http://192.168.88.161:50070/dfshealth.html#tab-overview](http://192.168.52.100:50070/dfshealth.html#tab-overview)

node2机器查看hdfs状态

[http://192.168.88.162:50070/dfshealth.html#tab-overview](http://192.168.52.110:50070/dfshealth.html#tab-overview)

##### yarn集群访问查看

[http://192.168.88.163:8088/cluster](http://node03:8088/cluster)

##### 历史任务浏览界面

页面访问：

[http://192.168.88.163:19888/jobhistory](http://192.168.52.120:19888/jobhistory)