##### 前提准备：

软件：/data/apps/

数据： /data/hdfs-data

机器：192.16.153.12  192.16.153.11 192.16.153.10

有sudo权限的用户

hdpusr

hdpusr

##### 集群规划：

192.16.153.10 node1 nn dn nm hive

192.16.153.11 node2 dn nm snm rm

192.16.153.12 node3 dn nm mysql

在etc/hosts文件中不要将127.0.0.1与hostname做映射 否则后面会出现NM去连接NN的9000端口时出问题

##### 版本信息:

jdk1.8

scala2.11.8

hadoop2.6.4

spark2.1.1&2.4

hive-2.0.1

mysql-5.7.24

##### 修改主机名：

1. uname -a 查看hostname

2. hostname newname 修改下，让hostname立刻生效。

3. vi /etc/hosts 修改原hostname为 newname

4. vi /etc/sysconfig/network 修改原hostname为

5. newname , reboot重启后也生效

##### 建立免密登录:

cd /home/hdpusr/.ssh

ssh-keygen -t rsa

scp -r id\_rsa.pub hdpusr@node3:/home/hdpusr/.ssh/authorized\_keys\_from\_node1 #将id\_rsa.pub发往被登录的服务器

cat authorized\_keys\_from\_node1 >> authorized\_keys #在被登录的服务器合并秘钥

失败->检测.ssh目录以及文件权限 执行：chmod 700 .ssh/ chmod 700 .ssh/\*

ssh localhost的配置上面的操作一致将自己生成的id\_rsa.pub追加入authorized\_keys文件中即可

##### 环境变量配置:

export JAVA\_HOME=/data/apps/jdk

export SCALA\_HOME=/data/apps/scala

export HADOOP\_HOME=/data/apps/hadoop

export SPARK\_HOME=/data/apps/spark

export HIVE\_HOME=/data/apps/hive

export PATH=$JAVA\_HOME/bin:$SCALA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$SPARK\_HOME/bin:$HIVE\_HOME/bin:$PATH

##### hadoop集群安装：

参数配置:

hadoop-env.sh:

export HADOOP\_LOG\_DIR=/data/logs/hadoop-log/

core-site.xml:

<property>

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

<value>/data/apps/hadoop/tmp/</value>

<description>A base for other temporary directories.</description>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://node1</value>

</property>

<!-- file system properties -->

<property>

<name>fs.default.name</name>

<value>hdfs://node1:9000</value>

</property>

hdfs-site.xml:

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

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

<value>node2:50090</value>

</property>

<property>

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

    <value>file://data/hdfs-data/</value>

</property>

mapred-site.xml:

<!-- 指定mr运行在yarn上 -->

<property>

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

<value>yarn</value>

</property>

yarn-site.xml:

<property>

<name>yarn.resourcemanager.hostname</name>

<value>node2</value>

</property>

<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>

<description>Whether to enable log aggregation</description>

<name>yarn.log-aggregation-enable</name>

<value>true</value>

</property>

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>22528</value>

<description>Usable memory</description>

</property>

<property>

<name>yarn.nodemanager.resource.cpu-vcores</name>

<value>22</value>

<description>Number of virtual CPUs that can be used</description>

</property>

<property>

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

<value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>

</property>

<property>

<name>yarn.scheduler.fair.allocation.file</name>

<value>/data/apps/hadoop/etc/hadoop/fair-scheduler.xml</value>

</property>

<property>

<name>yarn.scheduler.fair.preemption</name>

<value>true</value>

</property>

<property>

<name>yarn.scheduler.fair.user-as-default-queue</name>

<value>true</value>

<description>default is True</description>

</property>

<property>

<name>yarn.scheduler.fair.allow-undeclared-pools</name>

<value>false</value>

<description>default is True</description>

</property>

yarn-env.sh

export YARN\_LOG\_DIR=/data/logs/hadoop-log/

hadoop namenode -format

Namenode和ResourceManger如果不是同一台机器，不能在NameNode上启动 yarn，应该在ResouceManager所在的机器上启动yarn

刷新yarn队列的配额:

yarn rmadmin –refreshQueues

##### spark on yarn集群安装

spark-defaults.conf：

spark.serializer org.apache.spark.serializer.KryoSerializer

spark.driver.memory 1g

spark.executor.extraJavaOptions -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"

spark-env.sh：

export JAVA\_HOME==/data/apps/jdk

export SCALA\_HOME=/data/apps/scala

export HADOOP\_HOME=/data/apps/hadoop

export HADOOP\_CONF\_DIR=/data/apps/hadoop/etc/hadoop

export SPARK\_MASTER\_PORT=7077

export SPARK\_EXECUTOR\_CORES=2

export SPARK\_EXECUTOR\_MEMORY=1024m

export SPARK\_EXECUTOR\_INSTANCES=1

log4j.properties:

spark 日志级别设置为WARN

/data/apps/spark/conf/log4j.properties

##### 查看web界面

在node2机器上执行:

w3m http://node2:8088

w3m http://node1:50070

##### yarn的rest api：

--集群信息

http://node2:8088/ws/v1/cluster/info

--资源信息

http://node2:8088/ws/v1/cluster/metrics

--调度

http://node2:8088/ws/v1/cluster/scheduler

--应用程序

http://node2:8088/ws/v1/cluster/apps

--应用程序状态

http://node2:8088/ws/v1/cluster/appstatistics?states=accepted,running,finished&applicationTypes=mapreduce

--节点信息

http://node2:8088/ws/v1/cluster/nodes

curl --compressed -H "Accept: application/json" -X GET "http://node2:8088/ws/v1/cluster/scheduler"

##### hive安装

hive-env.sh

HADOOP\_HOME=/data/apps/hadoop

export HIVE\_CONF\_DIR=/data/apps/hive/conf

export HIVE\_AUX\_JARS\_PATH=/data/apps/hadoop/lib

hive-site配置

<property>

<name>hive.metastore.uris</name>

<value>thrift://node1:9083</value>

</property>

<property>

<name>system:java.io.tmpdir</name>

<value>/tmp/hive/java</value>

</property>

<property>

<name>system:user.name</name>

<value>hdpusr</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://node3:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=true</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>root</value>

</property>

添加Mysql依赖包:

cp ~/mysql-connector-java-5.1.39.jar /data/apps/hive/lib/

#对数据库进行初始化

schematool -initSchema -dbType mysql

#启动metastore:

./bin/hive --service metastore 1>/dev/null 2>&1 &

#将mysql驱动包和hive-site.xml复制到spark的jars和conf目录下

scp -r mysql-connector-java-5.1.39.jar hdpusr@node2:/data/apps/spark/jars/

scp -r conf/hive-site.xml hdpusr@node2:/data/apps/spark/conf/

##### mysql安装

卸载系统自带的Mariadb

rpm -qa|grep mariadb //查询出已安装的mariadb

rpm -e --nodeps 文件名 //卸载 ， 文件名为使用rpm -qa|grep mariadb 命令查出的所有文件

rm /etc/my.cnf

1.解压：

mysql-5.7.24-linux-glibc2.12-x86\_64.tar

继续解压：

tar -zxvf mysql-5.7.24-linux-glibc2.12-x86\_64.tar.gz

2.安装和初始化MySQL数据库：

bin/mysqld --initialize --user=hdpusr --basedir=/data/apps/mysql/ --datadir=/data/apps/mysql/data/

在mysql目录下拷贝mysql.server文件到 /etc/init.d/mysqld

执行命令：cp -a ./support-files/mysql.server /etc/init.d/mysqld

编辑/etc/init.d/mysqld 添加数据库安装目录，和数据存放的地方

basedir这个目录是数据库的根目录 也就是安装目录(注意:是安装目录)

datadir这个目录是数据存放的地方

vim /etc/init.d/mysqld

删除系统默认的MySQL配置文件/etc/my.cnf

rm -rf /etc/my.cnf

3.初始化MySQL

./mysqld\_safe --user=hdpusr &

4.启动MySQL

/etc/init.d/mysqld restart

5.初始化密码设置

查看初始化密码:

在bin执行以下命令:cat /root/.mysql\_secret

\_yPyalNtgdig

修改初始化密码：

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

/etc/init.d/mysqld stop

./mysqld\_safe --user=hdpusr --skip-grant-tables --skip-networking &

mysql -u root mysql

UPDATE user SET authentication\_string=PASSWORD('root') where USER='root';

flush privileges;

ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';

6.添加远程访问

use mysql;

update user set host = '%' where user = 'root';

##### Spark on yarn client模式出现ClosedChannelException：

查看nodemanager的日志得知:

C:\Users\v_starhe\AppData\Local\Temp\企业微信截图_15474613733164.png

意思是说container使用的虚拟内存超过了设置的2.1G

虚拟内存是从yarn-site.xml中配置计算来的，yarn.scheduler.minimum-allocation-mb  \* yarn.nodemanager.vmem-pmem-ratio = 虚拟内存的总量，如果需要的虚拟内存总量超过这个计算所得的数值，就会出发 Killing container；

此处 我的yarn.scheduler.minimum-allocation-mb值没设置，默认为1G，yarn.nodemanager.vmem-pmem-ratio也没设置，默认为2.1，因此，就有了以上的日志，用了1g里的360M物理内存，用了2.1G里的2.4G虚拟内存。

通过yarn-site.xml文件修改默认值:

<property>

<name>yarn.scheduler.maximum-allocation-mb</name>

<value>9216</value>

<discription>每个任务最多可用内存,单位MB,默认8182MB</discription>

</property>

<property>

<name>yarn.scheduler.minimum-allocation-mb</name>

<value>4000</value>

<discription>每个任务最shao可用内存</discription>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>4.1</value>

</property>

查看物理cpu个数：

cat /proc/cpuinfo | grep "physical id" | sort | uniq|wc -l

查看逻辑cpu的个数：

cat /proc/cpuinfo | grep "processor" |wc -l

查看cpu是几核：

cat /proc/cpuinfo | grep "cores"|uniq