分布式列族数据库 HBase

一、实训说明

本次实训，主要是分布式列族数据库 hbase 搭建，HBase – Hadoop Database，是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统。

二、实训环境

1）已经完成配置的 Hadoop 完全分布式文件系统。
2）已经配置完成的 Zookeeper 集群模式环境。
3）集群中包括三个节点，并且可以保证网络互通。
4）使用软件：
Hbase：hbase-1.2.1-bin.tar.gz
下载地址（版本更新路径会导致地址不存在）：
http://archive.apache.org/dist/hbase/2.1.0/hbase-2.1.0-bin.tar.gz

三、实训内容

1. 安装 ntp 服务（所有节点）

1）安装 ntp

```
[hadoop@master ~]$ sudo yum install ntp -y
```

```
[hadoop@master ~]$ yum install ntp -y
Loaded plugins: fastestmirror
You need to be root to perform this command.
[hadoop@master ~]$ sudo yum install ntp -y
[sudo] password for hadoop:
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
epel/x86_64/metalink                          | 7.2 kB     00:00
 * base: mirror.bit.edu.cn
 * epel: mirrors.njupt.edu.cn
 * extras: mirror.bit.edu.cn
 * updates: mirror.bit.edu.cn
base                                          | 3.6 kB     00:00
epel                                          | 5.3 kB     00:00
extras                                        | 3.4 kB     00:00
mysql-connectors-community                    | 2.5 kB     00:00
mysql-tools-community                         | 2.5 kB     00:00
mysql57-community                             | 2.5 kB     00:00
updates                                       | 3.4 kB     00:00
(1/2): epel/x86_64/primary_db                   | 6.8 MB   00:00
(2/2): epel/x86_64/updateinfo                   | 998 kB   00:01
```

2）开启 ntp 服务并设置开机自启动

```
[hadoop@master ~]$ sudo systemctl start ntpd

[hadoop@master ~]$ sudo systemctl enable ntpd
```

```
[hadoop@master ~]$ sudo systemctl start ntpd
[hadoop@master ~]$ sudo systemctl enable ntpd
Created symlink from /etc/systemd/system/multi-user.target.wants/nt
pd.service to /usr/lib/systemd/system/ntpd.service.
[hadoop@master ~]$
```

3）将系统时区设置为上海

```
[hadoop@master ~]$ sudo cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

```
[hadoop@master ~]$ sudo cp /usr/share/zoneinfo/Asia/Shanghai /etc/l
ocaltime
[sudo] password for hadoop:
[hadoop@master ~]$
```

4）查看各节点的时间是否一致

master 节点：

```
[hadoop@master ~]$ date
```

```
[hadoop@master ~]$ date
Mon Jul 29 11:39:33 CST 2019
[hadoop@master ~]$
```

slave1 节点：

```
[hadoop@slave1 ~]$ date
```

```
[hadoop@slave1 ~]$ date
Mon Jul 29 11:39:33 CST 2019
[hadoop@slave1 ~]$
```

slave2 节点：

```
[hadoop@slave2 ~]$ date
```

```
[hadoop@slave2 ~]$ date
Mon Jul 29 11:39:33 CST 2019
[hadoop@slave2 ~]$
```

一定要确保 3 个节点的时间是一致的，不然待会启动 hbse 会出错

2. 安装 Hbase

1）解压安装包（master 节点）

```
[hadoop@master ~]$ sudo tar -zxvf /home/package/hbase-2.1.0-bin.tar.gz -C /usr/
```

2）重命名安装路径（master 节点）

```
[hadoop@master ~]$ sudo mv /usr/hbase-2.1.0/ /usr/hbase
```

3）添加环境变量（所有节点）

```
[hadoop@master ~]$ sudo vim /etc/profile
```

在配置文件中添加以下环境变量信息

```
export HBASE_HOME=/usr/hbase

export PATH=$PATH:$HBASE_HOME/bin
```

4）使环境变量生效（所有节点）

```
export HBASE_HOME=/usr/hbase
export PATH=$PATH:$HBASE_HOME/bin
```

```
[hadoop@master ~]$ source /etc/profile
```

5）配置 hbase-env.sh 配置文件（master 节点）

```
[hadoop@master ~]$ vim $HBASE_HOME/conf/hbase-env.sh
```

修改配置文件中的 JAVA_HOME、HBASE_CLASSPATH、HBASE_MANAGES_ZK 的值

```
# into the startup scripts (bin/hbase, etc.)

# The java implementation to use.  Java 1.8+ required.
export JAVA_HOME=/usr/java/jdk1.8.0_201

# Extra Java CLASSPATH elements.  Optional.
export HBASE_CLASSPATH=/usr/hadoop/etc/hadoop
```

```
# Tell HBase whether it should manage it's own instance of ZooKeeper
 or not.
export HBASE_MANAGES_ZK=false

# The default log rolling policy is RFA, where the log file is rolle
d as per the size defined for the
```

说明：export HBASE_MANAGES_ZK=false #值为 true 使用 hbase 自带的 zookeeper,值为 false 使用在 Hadoop 上装的 zookeeper

6）配置 hbase-site.xml 配置文件（master 节点）

```
[hadoop@master ~]$ vim $HBASE_HOME/conf/hbase-site.xml
```

在配置文件中添加以下内容：

```
        <property>
                <name>hbase.rootdir</name>
                <value>hdfs://master:9000/hbase</value>
        </property>
        <property>
                <name>hbase.master</name>
                <value>60000</value>
        </property>
        <property>
                <name>hbase.zookeeper.property.clientPort</name>
                <value>2181</value>
        </property>
        <property>
                <name>zookeeper.session.timeout</name>
                <value>120000</value>
        </property>
        <property>
                <name>hbase.zookeeper.quorum</name>
                <value>master,slave1,slave2</value>
```

```
	</property>
	<property>
		<name>hbase.tmp.dir</name>
		<value>/usr/hbase/tmp</value>
	</property>
	<property>
		<name>hbase.cluster.distributed</name>
		<value>true</value>
	</property>
	<property>
		<name>hbase.unsafe.stream.capability.enforce</name>
		<value>false</value>
	</property>
```

```
<configuration>
        <property>
                <name>hbase.rootdir</name>
                <value>hdfs://master:9000/hbase</value>
        </property>
        <property>
                <name>hbase.master</name>
                <value>60000</value>
        </property>
        <property>
                <name>hbase.zookeeper.property.clientPort</nam
e>
                <value>2181</value>
        </property>
        <property>
                <name>zookeeper.session.timeout</name>
                <value>120000</value>
        </property>
        <property>
                <name>hbase.zookeeper.quorum</name>
                <value>master,slave1,slave2</value>
        </property>
```

```
        <property>
                <name>hbase.zookeeper.quorum</name>
                <value>master,slave1,slave2</value>
        </property>
        <property>
                <name>hbase.tmp.dir</name>
                <value>/usr/hbase/tmp</value>
        </property>
        <property>
                <name>hbase.cluster.distributed</name>
                <value>true</value>
        </property>
        <property>
                <name>hbase.unsafe.stream.capability.enforce</
name>
                <value>false</value>
        </property>
</configuration>
```

7）配置 regionservers 配置文件（master 节点）

```
[hadoop@master ~]$ vim $HBASE_HOME/conf/regionservers
```

在配置文件中加入 regionservers 节点的主机名

```
slave1
slave2
~
```

8）将 htrace-core-3.1.0-incubating.jar 复制到 lib 目录下

```
[hadoop@master
hbase]$ cp ./lib/client-facing-thirdparty/htrace-core-3.1.0-incubating.jar ./l
ib
```

9）创建 hbase.tmp.dir 目录（master 节点）

```
[hadoop@master ~]$ sudo mkdir /usr/hbase/tmp
```

10）将文件分发到 slave1 和 slave2

```
[hadoop@master ~]$ sudo scp -r /usr/hbase/ slave1:/usr
[hadoop@master ~]$ sudo scp -r /usr/hbase/ slave2:/usr
```

11）修改 hbase 目录权限（所有节点）

```
[hadoop@master ~]$ sudo chown -R hadoop:hadoop /usr/hbase/
```

3. 验证测试

1）启动 hbase（先启动 hadoop，然后启动 zookeeper，最后启动 hbase）

```
[hadoop@master ~]$ start-hbase.sh
```

```
[hadoop@master ~]$ start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
running master, logging to /usr/hbase/logs/hbase-hadoop-master-master.out
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
slave2: running regionserver, logging to /usr/hbase/bin/../logs/hbase-hadoop-regionserver-slave2.out
slave1: running regionserver, logging to /usr/hbase/bin/../logs/hbase-hadoop-regionserver-slave1.out
```

2）查看每个节点的守护进程

master：

```
[hadoop@master ~]$ jps
```

```
[hadoop@master hbase]$ jps
26880 SecondaryNameNode
26641 NameNode
15810 QuorumPeerMain
27684 HMaster
27912 Jps
27117 ResourceManager
[hadoop@master hbase]$
```

slave1：

```
[hadoop@slave1 ~]$ jps
```

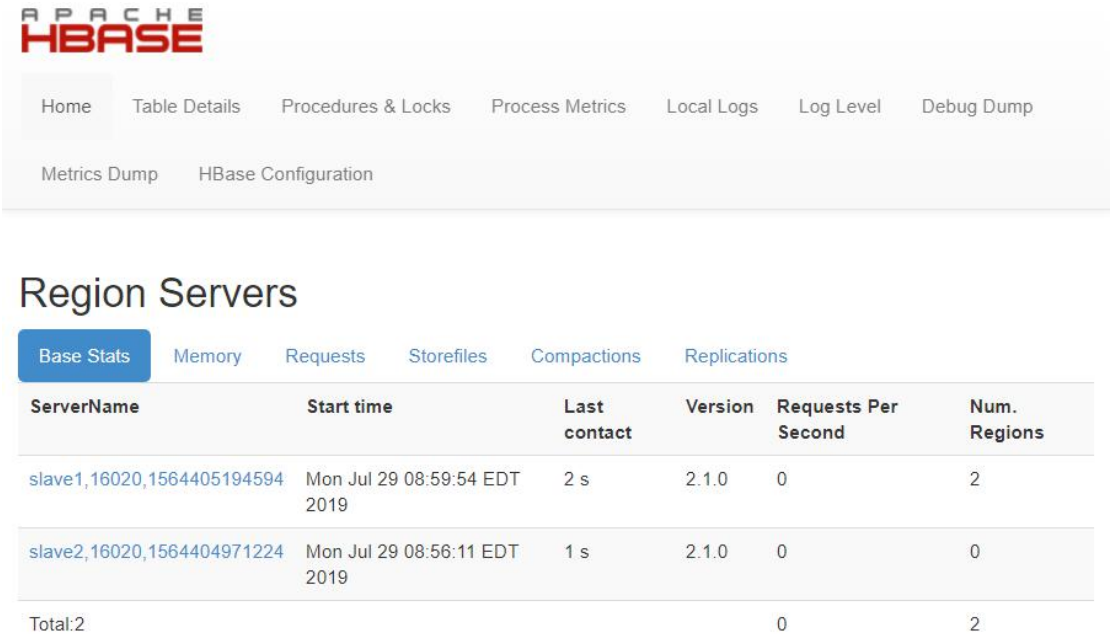
```
[hadoop@slave1 ~]$ jps
11094 Jps
10615 DataNode
10919 HRegionServer
8329 QuorumPeerMain
10734 NodeManager
[hadoop@slave1 ~]$
```

slave2：

```
[hadoop@slave2 ~]$ jps
```

```
[hadoop@slave2 ~]$ jps
3136 NodeManager
9124 HRegionServer
3017 DataNode
9229 Jps
8191 QuorumPeerMain
[hadoop@slave2 ~]$
```

3）在浏览器打开 master:16010，查看 hbase 的状态



4）关闭 hbase

```
[hadoop@master hbase]$ stop-hbase.sh
```

```
[hadoop@master hbase]$ stop-hbase.sh
stopping hbase.........
```