

第一部分.安装 hdfs 集群的具体步骤:

一、首先需要准备 N 台 linux 服务器

先准备 3 台虚拟机: 1 个 namenode 节点 + 2 个 datanode 节点

二、修改各台机器的主机名和 ip 地址

主机名: hdp-01 对应的 ip 地址: 192.168.33.61

主机名: hdp-02 对应的 ip 地址: 192.168.33.62

主机名: hdp-03 对应的 ip 地址: 192.168.33.63

主机名: hdp-04 对应的 ip 地址: 192.168.33.64

三、从 windows 中用 MobaXterm_Personal_10.软件进行远程连接

在 windows 中将各台 linux 机器的主机名配置到的 windows 的本地域名映射文件中:

c:/windows/system32/drivers/etc/hosts

```
192.168.33.61 hdp-01
192.168.33.62 hdp-02
192.168.33.63 hdp-03
192.168.33.64 hdp-04
```

四、配置 linux 服务器的基础软件环境

- 防火墙

关闭防火墙: `service iptables stop`

关闭防火墙自启: `chkconfig iptables off`

- 安装 jdk: (hadoop 体系中的各软件都是 java 开发的)

1) 打开 sftp 窗口, 然后将 jdk 压缩包拖入 sftp 窗口

2) 然后在 linux 中将 jdk 压缩包解压到 /root/apps 下

3) 配置环境变量: `JAVA_HOME` `PATH`

`vi /etc/profile` 在文件的最后, 加入:

```
export JAVA_HOME=/root/apps/jdk1.8.0_60
export PATH=$PATH:$JAVA_HOME/bin
```

4) 修改完成后, 记得 `source /etc/profile` 使配置生效

5) 检验: 在任意目录下输入命令: `java -version` 看是否成功执行

6) 将安装好的 jdk 目录用 `scp` 命令拷贝到其他机器

7) 将/etc/profile 配置文件也用 scp 命令拷贝到其他机器并分别执行 source 命令

● 集群内主机的域名映射配置

在 hdp-01（任意一台虚拟机的名字）上，vi /etc/hosts

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.*  ***
192.168.1.*  ***
192.168.1.*  ***
192.168.1.*  ***
```

然后，将 hosts 文件拷贝到集群中的所有其他机器上

```
scp /etc/hosts ***:/etc/
```

```
scp /etc/hosts ***:/etc/
```

```
scp /etc/hosts ***:/etc/
```

如果用的是 VMware 虚拟机，则使用以下方法：

补充提示

如果在执行 scp 命令的时候，提示没有 scp 命令，则可以配置一个本地 yum 源来安装

1、先在虚拟机中配置 cdrom 为一个 centos 的安装镜像 iso 文件

2、在 linux 系统中将光驱挂在到文件系统中（某个目录）

3、mkdir /mnt/cdrom

4、mount -t iso9660 -o loop /dev/cdrom /mnt/cdrom

5、检验挂载是否成功：ls /mnt/cdrom

6、3、配置 yum 的仓库地址配置文件

7、yum 的仓库地址配置文件目录：/etc/yum.repos.d

8、先将自带的仓库地址配置文件批量更名：

```
[root@hdp-01 ~]# cd /etc/yum.repos.d/
[root@hdp-01 yum.repos.d]# ll
total 24
-rw-r--r--. 1 root root 1991 Aug  4 2015 CentOS-Base.repo
-rw-r--r--. 1 root root  647 Aug  4 2015 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  289 Aug  4 2015 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  630 Aug  4 2015 CentOS-Media.repo
-rw-r--r--. 1 root root 6259 Aug  4 2015 CentOS-Vault.repo
[root@hdp-01 yum.repos.d]# rename .repo .repo.bak *
[root@hdp-01 yum.repos.d]# ll
total 24
-rw-r--r--. 1 root root 1991 Aug  4 2015 CentOS-Base.repo.bak
-rw-r--r--. 1 root root  647 Aug  4 2015 CentOS-Debuginfo.repo.bak
-rw-r--r--. 1 root root  289 Aug  4 2015 CentOS-fasttrack.repo.bak
-rw-r--r--. 1 root root  630 Aug  4 2015 CentOS-Media.repo.bak
-rw-r--r--. 1 root root 6259 Aug  4 2015 CentOS-Vault.repo.bak
```

9、然后，拷贝一个出来进行修改

```
[root@hdp-01 yum.repos.d]# cp CentOS-Media.repo.bak dvd.repo
[root@hdp-01 yum.repos.d]# ll
total 28
-rw-r--r--. 1 root root 1991 Aug  4 2015 CentOS-Base.repo.bak
-rw-r--r--. 1 root root  647 Aug  4 2015 CentOS-Debuginfo.repo.bak
-rw-r--r--. 1 root root  289 Aug  4 2015 CentOS-fasttrack.repo.bak
-rw-r--r--. 1 root root  630 Aug  4 2015 CentOS-Media.repo.bak
-rw-r--r--. 1 root root 6259 Aug  4 2015 CentOS-Vault.repo.bak
-rw-r--r--. 1 root root  630 Aug 12 20:04 dvd.repo
[root@hdp-01 yum.repos.d]# vi dvd.repo
```

```
[c6-media]
name=CentOS-$releasever - Media
baseurl=file:///mnt/cdrom
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
~
```

10、修改完配置文件后，再安装 scp 命令：

11、yum install openssh-clients -y

如果是华为云服务器，则直接挂在华为云提供的 centos 镜像，步骤如下

以 root 用户登录弹性云服务器。

执行以下命令，备份 CentOS-Base.repo 文件。

```
mkdir -p /etc/yum.repos.d/repo_bak/
```

```
mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/repo_bak/
```

使用 curl 命令下载对应版本的 CentOS-Base.repo 文件，并将其放入/etc/yum.repos.d/目录。

针对华为云当前支持的 CentOS 镜像源版本，使用的 curl 命令分别如下：

CentOS 6

```
curl -o /etc/yum.repos.d/CentOS-Base.repo
```

```
http://mirrors.myhuaweicloud.com/repo/CentOS-Base-6.repo
```

CentOS 7

```
curl -o /etc/yum.repos.d/CentOS-Base.repo
```

```
http://mirrors.myhuaweicloud.com/repo/CentOS-Base-7.repo
```

执行以下命令，生成缓存。

```
yum makecache
```

说明：

如果执行命令 **yum makecache** 后，回显信息为“Another app is currently holding the yum lock”，则需执行命令 **rm -rf /var/run/yum.pid** 删除 yum.pid 文件，然后再执行命令 **yum makecache** 生成缓存。

如果执行命令 **yum makecache** 后，回显信息为“Error: Cannot retrieve metalink for repository: epel. Please verify its path and try again”，则需执行命令 **yum --disablerepo=epel -y update ca-certificates** 更新证书，临时禁用 epel 源。

说明

如果执行命令 **yum makecache** 后，回显信息为“Another app is currently holding the yum lock”，则需执行命令 **rm -rf /var/run/yum.pid** 删除 yum.pid 文件，然后再执行命令 **yum makecache** 生成缓存。

如果执行命令 **yum makecache** 后，回显信息为“Error: Cannot retrieve metalink for repository: epel. Please verify its path and try again”，则需执行命令 **yum --disablerepo=epel -y update ca-certificates** 更新证书，临时禁用 epel 源。

五、安装 hdfs 集群

1、上传 hadoop 安装包到 hdp-01

2、修改配置文件

要点	核心配置参数：
提示	<ul style="list-style-type: none">1) 指定 hadoop 的默认文件系统为：hdfs2) 指定 hdfs 的namenode 节点为哪台机器3) 指定 namenode 软件存储元数据的本地目录4) 指定 datanode 软件存放文件块的本地目录

hadoop 的配置文件在：/root/apps/hadoop 安装目录/etc/hadoop/

1) 修改 hadoop-env.sh

```
export JAVA_HOME=/root/apps/jdk1.8.0_60
```

2) 修改 core-site.xml

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://hdp-01:9000</value>
</property>
</configuration>
```

3) 修改 hdfs-site.xml

```
<configuration>
<property>
<name>dfs.namenode.name.dir</name>
<value>/root/dfs/name</value>
</property>

<property>
<name>dfs.datanode.data.dir</name>
<value>/root/dfs/data</value>
</property>

</configuration>
```

4) 拷贝整个 hadoop 安装目录到其他机器

```
scp -r /root/apps/hadoop-2.8.0 hdp-02:/root/apps/  
scp -r /root/apps/hadoop-2.8.0 hdp-03:/root/apps/  
scp -r /root/apps/hadoop-2.8.0 hdp-04:/root/apps/
```

5) 启动 HDFS

所谓的启动 HDFS，就是在对的机器上启动对的软件

要 点 提 示 ：	<p>要运行 <i>hadoop</i> 的命令，需要在 <i>linux</i> 环境中配置 <i>HADOOP_HOME</i> 和 <i>PATH</i> 环境变量</p> <pre>vi /etc/profile export JAVA_HOME=/root/apps/jdk1.8.0_60 export HADOOP_HOME=/root/apps/hadoop-2.8.0 export PATH=\$PATH:\$JAVA_HOME/bin:\$HADOOP_HOME/bin:\$HADOOP_HOME/sbin</pre>
-----------------------	--

首先，初始化 **namenode** 的元数据目录

要在 **hdp-01** 上执行 *hadoop* 的一个命令来初始化 **namenode** 的元数据存储目录

```
hadoop namenode -format
```

- 创建一个全新的元数据存储目录
- 生成记录元数据的文件 *fsimage*
- 生成集群的相关标识：如：集群 id——*clusterID*

然后，启动 **namenode** 进程（在 **hdp-01** 上）

```
hadoop-daemon.sh start namenode
```

启动完后，首先用 *jps* 查看一下 **namenode** 的进程是否存在

然后，在 **windows** 中用浏览器访问 **namenode** 提供的 **web** 端口：50070

```
http://hdp-01:50070
```

然后，启动众 **datanode** 们（在任意地方）

```
hadoop-daemon.sh start datanode
```

6) 用自动批量启动脚本来启动 HDFS

- 1) 先配置 **hdp-01** 到集群中所有机器（包含自己）的免密登陆
- 2) 配完免密后，可以执行一次 `ssh 0.0.0.0`
- 3) 修改 *hadoop* 安装目录中 */etc/hadoop/slaves*（把需要启动 **datanode** 进程的节点列入）

```
hdp-01  
hdp-02
```

hdp-03

hdp-04

- 4) 在 hdp-01 上用脚本: **start-dfs.sh** 来自动启动整个集群
- 5) 如果要停止, 则用脚本: **stop-dfs.sh**

第二部分. 安装 yarn 集群

yarn 集群中有两个角色:

主节点: Resource Manager 1 台

从节点: Node Manager N 台

Resource Manager 一般安装在一台专门的机器上

Node Manager 应该与 HDFS 中的 data node 重叠在一起

修改配置文件:

yarn-site.xml

```
<property>
<name>yarn.resourcemanager.hostname</name>
<value>hdp-04</value>
</property>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
```

然后复制¹到每一台机器上

然后在 hdp-04 上, 修改 hadoop 的 **slaves** 文件², 列入要启动 nodemanager 的机器

然后将 hdp-04 到所有机器的免密³登陆配置好

然后, 就可以用脚本启动 yarn 集群:

sbin/start-yarn.sh

停止:

sbin/stop-yarn.sh

启动完成后, 可以在 windows 上用浏览器访问 resourcemanager 的 web 端口:

<http://hdp-04:8088>

看 resource manager 是否认出了所有的 node manager 节点

第三部分. zookeeper 集群搭建

上传安装包，解压

修改 conf/zoo.cfg

```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sake.
dataDir=/root/zkdata
# the port at which the clients will connect
clientPort=2181
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
server.1=hdp-01:2888:3888
server.2=hdp-02:2888:3888
server.3=hdp-03:2888:3888
```

配置文件修改完后，将安装包拷贝给 hdp-02 和 hdp-03

接着，到 hdp-01 上，新建数据目录/root/zkdata，并在目录中生成一个文件 myid，内容为 1
接着，到 hdp-02 上，新建数据目录/root/zkdata，并在目录中生成一个文件 myid，内容为 2
接着，到 hdp-03 上，新建数据目录/root/zkdata，并在目录中生成一个文件 myid，内容为 3
启动 zookeeper 集群：zookeeper 集群启动时可写个脚本，运行脚本后，便不需要一台一台的启动了。

脚本请自行编写。

第四部分.hive 安装

一．最简安装：用内嵌 derby 作为元数据库

准备工作：安装 hive 的机器上应该有 HADOOP 环境（安装目录，HADOOP_HOME 环境变量）

安装：直接解压一个 hive 安装包即可

此时，安装的这个 hive 实例使用其内嵌的 derby 数据库作为记录元数据的数据库

此模式不便于让团队成员之间共享协作

二. 标准安装：将 mysql 作为元数据库

mysql 安装

① 上传 mysql 安装包

② 解压：

```
[root@mylove ~]# tar -xvf MySQL-5.6.26-1.linux_glibc2.5.x86_64.rpm-bundle.tar
```

③ 安装 mysql 的 server 包

```
[root@mylove ~]# rpm -ivh MySQL-server-5.6.26-1.linux_glibc2.5.x86_64.rpm
```

依赖报错：

缺 perl

```
yum install perl
```

安装完 perl 后，继续重新安装 mysql-server

（可以配置一个本地 yum 源进行安装：

1、先在 vmware 中给这台虚拟机连接一个光盘镜像

2、挂在光驱到一个指定目录：mount -t iso9660 -o loop /dev/cdrom /mnt/cdrom

3、将 yum 的配置文件中 baseURL 指向/mnt/cdrom

）

```
[root@mylove ~]# rpm -ivh MySQL-server-5.6.26-1.linux_glibc2.5.x86_64.rpm
```

又出错：包冲突 conflict with

移除老版本的冲突包：mysql-libs-5.1.73-3.el6_5.x86_64

```
[root@mylove ~]# rpm -e mysql-libs-5.1.73-3.el6_5.x86_64 --nodeps
```

继续重新安装 mysql-server

```
[root@mylove ~]# rpm -ivh MySQL-server-5.6.26-1.linux_glibc2.5.x86_64.rpm
```

成功后，注意提示：里面有初始密码及如何改密码的信息

初始密码：/root/.mysql_secret

改密码脚本：/usr/bin/mysql_secure_installation

④ 安装 mysql 的客户端包：

```
[root@mylove ~]# rpm -ivh MySQL-client-5.6.26-1.linux_glibc2.5.x86_64.rpm
```

⑤ 启动 mysql 的服务端：

```
[root@mylove ~]# service mysql start
```

Starting MySQL. SUCCESS!

⑥ 修改 root 的初始密码：

```
[root@mylove ~]# /usr/bin/mysql_secure_installation 按提示
```

⑦ 测试：

用 mysql 命令行客户端登陆 mysql 服务器看能否成功

```
[root@mylove ~]# mysql -uroot -proot
```

```
mysql> show databases;
```

⑧ 给 root 用户授予从任何机器上登陆 mysql 服务器的权限：

```
mysql> grant all privileges on *.* to 'root'@'%' identified by '你的密码' with grant option;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.00 sec)
```

注意点：要让 mysql 可以远程登录访问

最直接测试方法：从 windows 上用 Navicat 去连接，能连，则可以，不能连，则要去 mysql 的机器上用命令行客户端进行授权：

在 mysql 的机器上,启动命令行客户端：

```
mysql -uroot -proot
```

```
mysql>grant all privileges on *.* to 'root'@'%' identified by 'root 的密码' with grant option;  
mysql>flush privileges;
```

hive 的元数据库配置

```
vi conf/hive-site.xml
```

```
<configuration>  
<property>  
<name>javax.jdo.option.ConnectionURL</name>  
<value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</value>  
<description>JDBC connect string for a JDBC metastore</description>  
</property>  
  
<property>  
<name>javax.jdo.option.ConnectionDriverName</name>  
<value>com.mysql.jdbc.Driver</value>  
<description>Driver class name for a JDBC metastore</description>  
</property>  
  
<property>  
<name>javax.jdo.option.ConnectionUserName</name>  
<value>root</value>  
<description>username to use against metastore database</description>  
</property>  
  
<property>  
<name>javax.jdo.option.ConnectionPassword</name>  
<value>root</value>  
<description>password to use against metastore database</description>  
</property>  
</configuration>
```

- 2、上传一个 mysql 的驱动 jar 包到 hive 的安装目录的 lib 中
 - 3、配置 HADOOP_HOME 和 HIVE_HOME 到系统环境变量中：/etc/profile
 - 4、source /etc/profile
 - 5、hive 启动测试
- 然后用命令启动 hive 交互界面：
- ```
[root@hdp20-04 ~]# hive
```

### 三. hive 使用方式

#### 最基本使用方式:

启动一个 hive 交互 shell

bin/hive

hive>

设置一些基本参数, 让 hive 使用起来更便捷, 比如:

1、让提示符显示当前库:

hive>set hive.cli.print.current.db=true;

2、显示查询结果时显示字段名称:

hive>set hive.cli.print.header=true;

但是这样设置只对当前会话有效, 重启 hive 会话后就失效, 解决办法:

在 linux 的当前用户目录中, 编辑一个.hiverc 文件, 将参数写入其中:

vi .hiverc

```
set hive.cli.print.header=true;
set hive.cli.print.current.db=true;
```

### 四. 启动 hive 服务使用

启动 hive 的服务:

```
[root@hdp20-04 hive-1.2.1]# bin/hiveserver2 -hiveconf hive.root.logger=DEBUG,console
```

上述启动, 会将这个服务启动在前台, 如果要启动在后台, 则命令如下:

nohup bin/hiveserver2 1>/dev/null 2>&1 &

启动成功后, 可以在别的节点上用 beeline 去连接

❖ 方式 (1)

[root@hdp20-04 hive-1.2.1]# bin/beeline 回车, 进入 beeline 的命令界面

输入命令连接 hiveserver2

beeline> !connect jdbc:hive2://mini1:10000

(hadoop01 是 hiveserver2 所启动的那台主机名, 端口默认是 10000)

❖ 方式 (2)

启动时直接连接:

**bin/beeline -u jdbc:hive2://mini1:10000 -n root**

接下来就可以做正常 sql 查询了

### 五. 脚本化运行

大量的 hive 查询任务, 如果用交互式 shell 来进行输入的话, 显然效率及其低下, 因此, 生产中更多的是使用脚本化运行机制:

该机制的核心点是: hive 可以用一次性命令的方式来执行给定的 hql 语句

```
[root@hdp20-04 ~]# hive -e "insert into table t_dest select * from t_src;"
```

然后，进一步，可以将上述命令写入 shell 脚本中，以便于脚本化运行 hive 任务，并控制、调度众多 hive 任务，示例如下：

```
vi t_order_etl.sh
```

```
#!/bin/bash
hive -e "select * from db_order.t_order"
hive -e "select * from default.t_user"
hql="create table default.t_bash as select * from db_order.t_order"
hive -e "$hql"
```

如果要执行的 hql 语句特别复杂，那么，可以把 hql 语句写入一个文件：

```
vi x.hql
```

```
select * from db_order.t_order;
select count(1) from db_order.t_user;
```

然后，用 `hive -f /root/x.hql` 来执行

## 第五部分. 安装 HBASE

HBASE 是一个分布式系统

其中有一个管理角色： HMaster(一般 2 台，一台 active，一台 backup)

其他的数据节点角色： HRegionServer(很多台，看数据容量)

### 安装准备：

首先，要有一个 HDFS 集群，并正常运行； regionserver 应该跟 hdfs 中的 datanode 在一起

其次，还需要一个 zookeeper 集群，并正常运行

然后，安装 HBASE

角色分配如下：

Hdp01: namenode datanode regionserver hmaster zookeeper

Hdp02: datanode regionserver zookeeper

Hdp03: datanode regionserver zookeeper

### 安装步骤：

#### 1. 安装 zookeeper

自行查阅第三部分

## 2. 安装 hbase

解压 hbase 安装包

修改 hbase-env.sh

```
export JAVA_HOME=/root/apps/jdk1.7.0_67
export HBASE_MANAGES_ZK=false
```

修改 hbase-site.xml

```
<configuration>

 <!-- 指定 hbase 在 HDFS 上存储的路径 -->

 <property>
 <name>hbase.rootdir</name>
 <value>hdfs://hdp01:9000/hbase</value>
 </property>

 <!-- 指定 hbase 是分布式的 -->

 <property>
 <name>hbase.cluster.distributed</name>
 <value>true</value>
 </property>

 <!-- 指定 zk 的地址，多个用“,”分割 -->

 <property>
 <name>hbase.zookeeper.quorum</name>
 <value>hdp01:2181,hdp02:2181,hdp03:2181</value>
 </property>
</configuration>
```

修改 regionservers

```
hdp01
hdp02
hdp03
```

## 3. 启动 hbase 集群:

bin/start-hbase.sh

启动完后，还可以在集群中找任意一台机器启动一个备用的 master

bin/hbase-daemon.sh start master

新启的这个 master 会处于 backup 状态

## 4. 启动 hbase 的命令行客户端

bin/hbase shell

Hbase> list // 查看表

Hbase> status // 查看集群状态

Hbase> version // 查看集群版本

# 第六部分.Flume 的安装部署

1、Flume 的安装非常简单，只需要解压即可，当然，前提是已有 hadoop 环境  
上传安装包到数据源所在节点上

然后解压 tar -zxvf apache-flume-1.6.0-bin.tar.gz

然后进入 flume 的目录，修改 conf 下的 flume-env.sh，在里面配置 JAVA\_HOME

2、根据数据采集的需求**配置采集方案**，描述在配置文件中(文件名可任意自定义)

3、**指定采集方案配置文件**，在相应的节点上启动 flume agent

先用一个最简单的例子来测试一下程序环境是否正常：

1、先在 flume 的 conf 目录下新建一个配置文件（采集方案）

vi netcat-logger.properties

# 定义这个 agent 中各组件的名字

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# 描述和配置 source 组件：r1

a1.sources.r1.type = netcat

a1.sources.r1.bind = localhost

a1.sources.r1.port = 44444

# 描述和配置 sink 组件：k1

a1.sinks.k1.type = logger

# 描述和配置 channel 组件，此处使用是内存缓存的方式

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

# 描述和配置 source channel sink 之间的连接关系

```
a1.sources.r1.channels = c1
```

```
a1.sinks.k1.channel = c1
```

2、启动 agent 去采集数据

```
bin/flume-ng agent -c conf -f conf/netcat-logger.conf -n a1 -Dflume.root.logger=INFO,console
```