需要下载以下文件：

CDH安装包地址： <http://archive.cloudera.com/cdh5/parcels/latest/>

CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel

CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel.sha

cloudera-manager-centos7-cm5.6.0\_x86\_64.tar.gz

manifest.json

## 准备工作：系统环境搭建

yum install lrzsz wget vim net-tools ntp gcc gcc-c++ python-devel

修改hostname

在每一台服务器上执行vim  /etc/sysconfig/network

HOSTNAME= master1

这两步的作用很关键，如果配置不成功，进行分布式计算的时候有可能找不到主机名

Centos7使用hostnamectl set-hostname master1命令就可以了

修改host（所有主机）/etc/hosts

127.0.0.1 localhost

192.168.21.34 master1 （登陆账号密码root 123456）

192.168.21.40 master2 （登陆账号密码root 123456）

192.168.23.91 slave1 （登陆账号密码root 123456）

192.168.21.7 slave2 （登陆账号密码root 123456）

192.168.23.207 slave3 （登陆账号密码root 123456）

**关闭防火墙:**

systemctl stop firewalld.service

**关闭SELINUX**

setenforce 0 （临时生效）

修改 /etc/selinux/config 下的 SELINUX=disabled （重启后永久生效）

**打通SSH，设置ssh无密码登陆（所有节点）**

在主节点上执行 ssh-keygen -t rsa 一路回车，生成无密码的密钥对。

将公钥添加到认证文件中：

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

并设置authorized\_keys的访问权限：

chmod 600 ~/.ssh/authorized\_keys

scp文件到所有节点：

scp -r ~/.ssh/authorized\_keys root@master2:~/.ssh/

Jdk安装（所有主机）

查看java状况：rpm -qa |grep java

查看jdk状况：rpm -qa |grep jdk

卸载上两步显示的程序可以使用删除

yum -y remove javapackages-tools-3.4.1-5.el7.noarch

安装jdk：将jdk-8-linux-x64.rpm放入任意目录，执行

rpm -ivh jdk-8-linux-x64.rpm

修改环境变量：vim /etc/profile,加入如下几行：

export JAVA\_HOME=/usr/java/jdk1.8.0\_40

export JRE\_HOME=/usr/java/jdk1.8.0\_40/jre

export PATH=$JAVA\_HOME/bin:$JRE\_HOME/bin:$PATH

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

生效：source /etc/profile

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

验证：java -version、echo $JAVA\_HOME

同步到其它服务器一样的目录/usr/java

scp –r /usr/java/jdk1.8.0\_40 root@master1:/usr/java

…

**所有节点配置NTP服务**

[root@localhost ~]# tzselect #按照提示进行选择时区

yum install ntp

vim /etc/ntp.conf

# Use public servers from the pool.ntp.org project.

# Please consider joining the pool (http://www.pool.ntp.org/join.html).

#server 0.centos.pool.ntp.org iburst

#server 1.centos.pool.ntp.org iburst

#server 2.centos.pool.ntp.org iburst

#server 3.centos.pool.ntp.org iburst

server 0.cn.pool.ntp.org

server 1.asia.pool.ntp.org

server 2.asia.pool.ntp.org

#broadcast 192.168.1.255 autokey # broadcast server

#broadcastclient # broadcast client

#broadcast 224.0.1.1 autokey # multicast server

#multicastclient 224.0.1.1 # multicast client

#manycastserver 239.255.254.254 # manycast server

#manycastclient 239.255.254.254 autokey # manycast client

server 127.0.0.1 # local clock

fudge 127.0.0.1 stratum 10

重启

systemctl restart ntpd

systemctl enable  ntpd.service

其他客户端修改添加

server master1

其他时间server注释掉

**安装配置MySql（主节点）**

安装mysql的方法

CentOS 7的yum源中貌似没有正常安装mysql时的mysql-sever文件，需要去官网上下载

|  |
| --- |
| # wget http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm  # rpm -ivh mysql-community-release-el7-5.noarch.rpm  # yum install mysql-community-server |

成功安装之后重启mysql服务

|  |
| --- |
| systemctl restart mysqld.service |
| systemctl enable mysqld.service |

初次安装mysql是root账户是没有密码的

设置密码的方法

mysqladmin -u root password '123456'

创建以下数据库

#hive

create database hive DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

#activity monitor

create database amon DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

# Oozie

create database oozie DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

#授权root用户在主节点拥有所有数据库的访问权限

grant all privileges on \*.\* to 'root'@'master1' identified by '123456' with grant option;

flush privileges;

grant all privileges on \*.\* to 'root'@'localhost' identified by '123456' with grant option;

flush privileges;

grant all privileges on \*.\* to 'root'@'127.0.0.1' identified by '123456' with grant option;

flush privileges;

授权给master2的hive master

grant all privileges on \*.\* to 'root'@'master2' identified by '123456' with grant option;

flush privileges;

grant all privileges on \*.\* to 'root'@'slave1' identified by '123456' with grant option;

flush privileges;

grant all privileges on \*.\* to 'root'@'slave2' identified by '123456' with grant option;

flush privileges;

grant all privileges on \*.\* to 'root'@'slave3' identified by '123456' with grant option;

flush privileges;

**安装Cloudera Manager Server 和Agent**   
主节点解压安装   
cloudera manager的目录默认位置在/opt下，解压：tar xzvf cloudera-manager\*.tar.gz将解压后的cm-5.3.3和cloudera目录放到/opt目录下。   
为Cloudera Manager 5建立数据库   
首先需要去MySql的官网下载JDBC驱动，<http://dev.mysql.com/downloads/connector/j/>，解压后，找到mysql-connector-java-5.1.35-bin.jar，放到/opt/cm-5.6.0/share/cmf/lib/中。   
在主节点初始化CM5的数据库：

/opt/cm-5.6.0/share/cmf/schema/scm\_prepare\_database.sh mysql cm -hlocalhost -uroot -p123456 --scm-host localhost scm scm scm

**Agent配置**

修改/ data/cm-5.6.0/etc/cloudera-scm-agent/config.ini中的server\_host为主节点的主机名。

**同步Agent到其他节点**

scp -r / data/cm-5.6.0 hadoopwww@xxxx:/data/

在所有节点创建cloudera-scm用户

useradd --system --home=/data/cm-5.6.0/run/cloudera-scm-server/ --no-create-home --shell=/bin/false --comment "Cloudera SCM User" cloudera-scm

准备Parcels，用以安装CDH5

将CHD5相关的Parcel包放到主节点的/opt/cloudera/parcel-repo/目录中（parcel-repo需要手动创建）。

相关的文件如下：

[root@master1 cdh5]# cp -r CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel /opt/cloudera/parcel-repo/

[root@master1 cdh5]# cp -r CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel.sha /opt/cloudera/parcel-repo/

[root@master1 cdh5]# cp -r manifest.json /opt/cloudera/parcel-repo/

最后将CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel.sha1，重命名CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel.sha，这点必须注意，否则，系统会重新下载CDH-5.6.0-1.cdh5.6.0.p0.11-el7.parcel文件。

相关启动脚本

通过 /opt/cm-5.6.0/etc/init.d/cloudera-scm-server start 启动服务端。

通过 /opt/cm-5.6.0/etc/init.d/cloudera-scm-agent start 启动Agent服务。

我们启动的其实是个service脚本，需要停止服务将以上的start参数改为stop就可以了，重启

[root@slave3 opt]# /opt/cm-5.6.0/etc/init.d/cloudera-scm-agent start

/opt/cm-5.6.0/etc/init.d/cloudera-scm-agent:行106: pstree: 未找到命令

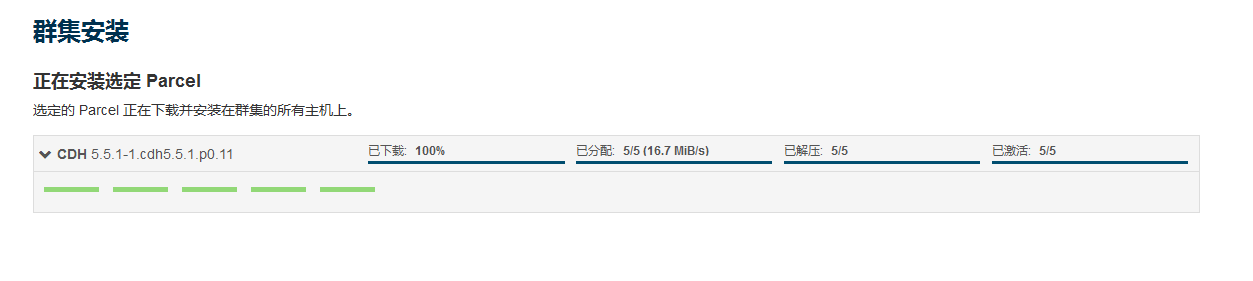
需要安装yum install psmisc

## CDH5的安装配置

Cloudera Manager Server和Agent都启动以后，就可以进行CDH5的安装配置了。

这时可以通过浏览器访问主节点的7180端口测试一下了（由于CM Server的启动需要花点时间，这里可能要等待一会才能访问），默认的用户名和密码均为admin：

**解压安装**





Cloudera 建议将 /proc/sys/vm/swappiness 设置为 0。当前设置为 30。使用 sysctl 命令在运行时更改该设置并编辑 /etc/sysctl.conf 以在重启后保存该设置。您可以继续进行安装，但可能会遇到问题，Cloudera Manager 报告您的主机由于交换运行状况不佳。以下主机受到影响：

#echo 0 > /proc/sys/vm/swappiness 即可解决。

vm.swappiness=0

已启用“透明大页面”，它可能会导致重大的性能问题。版本为“/sys/kernel/mm/transparent\_hugepage”且发行版为“{1}”的 Kernel 已将 enabled 设置为“{2}”，并将 defrag 设置为“{3}”。请运行“echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/defrag”以禁用此设置，然后将同一命令添加到一个 init 脚本中，如 /etc/rc.local，这样当系统重启时就会予以设置。或者，升级到 RHEL 6.5 或更新版本，它们不存在此错误。将会影响到以下主机：

解决:

echo never > /sys/kernel/mm/transparent\_hugepage/defrag

echo 'echo never > /sys/kernel/mm/transparent\_hugepage/defrag' >> /etc/rc.local

终于到安装各个服务的地方了，注意，这里安装Hive的时候可能会报错，因为我们使用了MySql作为hive的元数据存储，hive默认没有带mysql的驱动，通过以下命令拷贝一个就行了：

cp /opt/cm-5.6.0/share/cmf/lib/mysql-connector-java-5.1.38-bin.jar /opt/cloudera/parcels/CDH-5.6.0-1.cdh5.6.0.p0.11/lib/hive/lib/

oozie也需要驱动

cp /opt/cm-5.6.0/share/cmf/lib/mysql-connector-java-5.1.38-bin.jar /var/lib/oozie

记得选择以下这两个，要不hue使用不了

HBASE THIRST

HDFS HTTPFS

一路继续安装下去完成就好了。

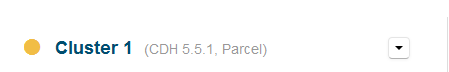
在集群的一台机器上执行以下模拟Pi的示例程序：

sudo -u hdfs hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi 10 100



dfs.journalnode.edits.dir：/dfs/journal

添加服务



Hue时区需要设置成Asia/Shanghai

/var/lib/oozie下面也要mysql jdbc驱动的

HUE依赖库一次过安装可能就不会有下面的一些问题了：

yum install krb5-devel cyrus-sasl-gssapi cyrus-sasl-deve libxml2-devel libxslt-devel mysql mysql-devel openldap-devel python-devel python-simplejson sqlite-devel

HUE报错Couldn't import snappy. Support for snappy compression disabled.

因为缺少snappy,

yum install gcc gcc-c++ python-devel

yum install snappy-devel

wget https://pypi.python.org/packages/source/p/python-snappy/python-snappy-0.5.tar.gz

tar zxvf python-snappy-0.5.tar.gz

python setup.py build

python setup.py install

#python

>import snappy

>a=snappy.compress("hello world!")

>b=snappy.uncompress(a)

>b

不报错就说明安装成功了

Hue启动找不到libxslt.so.1

yum install libxslt

问题：Hive Editor The application won't work without a running HiveServer2.

问题：TTransportException('Could not start SASL: Error in sasl\_client\_start (-4) SASL(-4): no mechanism available: No worthy mechs found',)

都是少上面的依赖库造成，安装了重启就好

由于在主节点上启动了Agent后，又将Agent scp到了其他节点上导致的，首次启动Agent，它会生成一个uuid，这样的话每台机器上的Agent的uuid都是一样的了，就会出现紊乱的情况。  
  
解决方案：  
删除/opt/cm-5.1.3/lib/cloudera-scm-agent/

**删除cd /opt/cm-5.6.0/lib64/cmf/下的python文件，从干净的安装包中同步过去，**

Hue地址：http://192.168.21.40:8888/

Cloudera manager地址：http://192.168.21.34:7180/

**jstorm安装部署**

yum -y install gcc gcc-c++ uuid-devel libuuid-devel libtoolunzip

cd zeromq

./autogen.sh  
   ./configure --without-libsodium

sudo make  
   sudo make install  
   sudo ldconfig  
   cd ..

cd jzmq

./autogen.sh  
    ./configure

touch src/classdist\_noinst.stamp

cd src

javac -d . org/zeromq/\*.java

cd ..

sudo make

sudo make install

sudo ldconfig

解压jstorm到指定目录去

配置conf/storm.yaml

nimbus.childopts: "-Xms2g -Xmx2g -Xmn768m -XX:SurvivorRatio=4 -XX:MaxTenuringThreshold=15 -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70 -XX:+HeapDumpOnOutOfMemoryError -XX:CMSMaxAbortablePrecleanTime=5000"

########### These MUST be filled in for a storm configuration

storm.zookeeper.servers:

- "slave1"

- "slave2"

- "slave3"

storm.zookeeper.root: "/jstorm"

# cluster.name: "default"

#nimbus.host/nimbus.host.start.supervisor is being used by $JSTORM\_HOME/bin/start.sh

#it only support IP, please don't set hostname

# For example

nimbus.host: "192.168.23.229, 192.168.23.235"

#nimbus.host: "localhost"

nimbus.host.start.supervisor: false

# %JSTORM\_HOME% is the jstorm home directory

storm.local.dir: "%JSTORM\_HOME%/data"

# please set absolute path, default path is JSTORM\_HOME/logs

# jstorm.log.dir: "absolute path"

java.library.path: "/usr/local/lib:/opt/local/lib:/usr/lib"

# if supervisor.slots.ports is null,

# the port list will be generated by cpu cores and system memory size

# for example,

# there are cpu\_num = system\_physical\_cpu\_num/supervisor.slots.port.cpu.weight

# there are mem\_num = system\_physical\_memory\_size/(worker.memory.size \* supervisor.slots.port.mem.weight)

# The final port number is min(cpu\_num, mem\_num)

# supervisor.slots.ports.base: 6800

# supervisor.slots.port.cpu.weight: 1.2

# supervisor.slots.port.mem.weight: 0.7

# supervisor.slots.ports: null

supervisor.slots.ports:

- 6800

- 6801

- 6802

- 6803

**配置项：**

#storm.zookeeper.servers: 表示zookeeper 的地址，

#nimbus.host: 表示nimbus的地址

#storm.zookeeper.root: 表示jstorm在zookeeper中的根目录，当多个JStorm共享一个#ZOOKEEPER时，需要设置该选项，默认即为“/jstorm”

#storm.local.dir: 表示jstorm临时数据存放目录，需要保证jstorm程序对该目录有写权限

#java.library.path: zeromq 和java zeromq library的安装目录，默认"/usr/local/lib:/opt/local/lib:/usr/lib"

#supervisor.slots.ports: 表示supervisor 提供的端口slot列表，注意不要和其他端口发生冲突，默认是68xx，而storm的是67xx

#supervisor.disk.slot： 表示提供数据目录，当一台机器有多块磁盘时，可以提供磁盘读写slot，方便有重IO操作的应用。

#topology.enable.classloader: false, 默认关闭classloader，如果应用的jar与jstorm的依赖的jar发生冲突，比如应用使用thrift9，但jstorm使用thrift7时，就需要打开classloader

#nimbus.groupfile.path: 如果需要做资源隔离，比如数据仓库使用多少资源，技术部使用多少资源，无线部门使用多少资源时，就需要打开分组功能， 设置一个配置文件的绝对路径，改配置文件如源码中group\_file.ini所示

#storm.local.dir: jstorm使用的本地临时目录，如果一台机器同时运行storm和jstorm的话， 则不要共用一个目录，必须将二者分离开

**在提交jar的节点上执行（本文中放置在master1，master2）:**

#mkdir ~/.jstorm

#cp -f $JSTORM\_HOME/conf/storm.yaml ~/.jstorm

**安装JStorm web ui**

必须使用tomcat 7.0 或以上版本， 注意不要忘记拷贝 **~/.jstorm/storm.yaml**

web ui 可以和nimbus不在同一个节点

tar -xzf apache-tomcat-7.0.37.tar.gz

cd apache-tomcat-7.0.37

cd webapps

cp $JSTORM\_HOME/jstorm-ui-2.1.0.war ./

mv ROOT ROOT.old

ln -s jstorm-ui-2.1.0 ROOT

cd ../bin

./startup.sh

**启动JStorm**

在nimbus 节点上执行 “nohup jstorm nimbus &”,

查看$JSTORM\_HOME/logs/nimbus.log检查有无错误

在supervisor节点上执行 “nohup jstorm supervisor &”,

查看$JSTORM\_HOME/logs/supervisor.log检查有无错误

访问地址：

Hue地址：http://192.168.21.40:8888/

Cloudera manager地址：<http://192.168.21.34:7180/>

Jstorm地址：http://192.168.21.40:18080/

**用以下命令清理内存**

**echo 1 > /proc/sys/vm/drop\_caches**

清理后内存使用情况再用以下命令看看。   
**free –m**

###########################---

**Impala操作**：

impala-shell -i localhost --quiet

进去shell命令：更多请参考<http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/impala_tutorial.html>

也可以直接在hue后台操作

创建表字段类型： INT,BOOLEAN,DOUBLE,TIMESTAMP,string,float

create table user(id int,name string,age int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
load data inpath '/user/impala’ overwrite into table user;

http://www.cloudera.com/documentation/archive/impala/2-x/2-1-x/topics/impala\_langref\_sql.html  
impala语句文档

**遇到问题解决方案：**

**### 取消hdfs ha或者（迁移Hadoop namenode）时遇到的问题:**

Hive和impala查询表的时候出现

ERROR: AnalysisException: Failed to load metadata for table: 'charge'

更改namenode后，在hive中运行select 依旧指向之前的namenode地址。

具体操作如下：

**1、hive这里使用了mysql来保存metadata**因为这些信息会被写死在hive的DBS和SBS表里，下面是关键的一步：  
进入mysql，执行下面的sql，

update DBS set DB\_LOCATION\_URI=REPLACE(DB\_LOCATION\_URI,'old host','new host');  
update SDS set LOCATION=REPLACE(LOCATION,'old host','new host');

**2、Impala要刷新metastore缓存**

在impala shell中刷新metastore

[localhost:21000] > INVALIDATE METADATA;

**### hbase ERROR: Can't get master address from ZooKeeper**

**还有启动没多久后就挂掉了，原因是因为之前是HA的切换成单点很多数据不同步了，需要清除**

/opt/cloudera/parcels/CDH/lib/hbase/bin

Usage: hbase-cleanup.sh (--cleanZk|--cleanHdfs|--cleanAll|--cleanAcls)

1、[hdfs@master1 bin]$ ./hbase-cleanup.sh –cleanAll

2、[hdfs@master1 bin]$ hdfs dfs rm -r /hbase/\*

**### Oozie提交作业内存太小报内存错误停止运行需要修改配置文件变量**

yarn.scheduler.maximum-allocation-mb

yarn.nodemanager.resource.memory-mb

设置至少2G以上的内存，建议8G内存

**### sqoop2启动后退出发现的原因是**

sqoop2的管理端端口8005和系统一些应用的端口冲突，需要更改

**###sqoop的数据库驱动存放位置，1、2版本的sqoop命令是不一样的，架构也不一样的**

Sqoop的数据库驱动是上传在/var/lib/sqoop/

Sqoop2的数据库驱动是上传在/var/lib/sqoop2/