**Impala 2.0安装手册（适用于Ubuntu12.04）**

**注意：SVN提交代码时只提交自己修改过的文件，不要把编译生成的文件也提交，否则硬盘放不下。**

在部署过程中可能会遇到一些问题，详见“部署集群时常见问题及解决方案”部分。

**MySQL Database**

1. sudo apt-get install mysql-server
2. sudo /usr/bin/mysql\_secure\_installation

[...]

Enter current password for root (enter for none):

OK, successfully used password, moving on...

[...]

Set root password? [Y/n] Y

New password:

Re-enter new password:

Remove anonymous users? [Y/n] Y

[...]

Disallow root login remotely? [Y/n] N

[...]

Remove test database and access to it [Y/n] Y

[...]

Reload privilege tables now? [Y/n] Y

All done!

1. sudo apt-get install chkconfig

sudo chkconfig mysql on

1. sudo apt-get install libmysql-java
2. mysql -u root -p

create database amon DEFAULT CHARACTER SET utf8;

grant all on amon.\* TO 'amon'@'%' IDENTIFIED BY 'amon\_password';

create database hive DEFAULT CHARACTER SET utf8;

grant all on hive.\* TO 'hive'@'%' IDENTIFIED BY 'hive\_password';

**Cloudera Manager and CDH**

1. Download and Run Cloudera Manger 5.2.x

<http://www.cloudera.com/content/cloudera/en/downloads/cloudera_manager/cm-5-2-0.html>

（Documentation: <http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/installation.html>）

1. chmod u+x cloudera-manager-installer.bin
2. sudo ./cloudera-manager-installer.bin
3. http://<Server host>:7180，Username：admim，Password：admin
4. Cloudera Express
5. 指定主机时输入别名，如impala、impala-101等(不能是localhost)
6. Install using Parcels
7. CDH 5.2.x
8. Install Zookeeper、HDFS、MapReduce and Hive，and we should use MySQL as databases（default port is 3306）

**SSH**

环境：假设有172.20.1.1~10，共10台机器。

步骤：

1. 安装ssh

sudo apt-get install openssh-server

1. 生成key

ssh-keygen -t rsa

cp ~/.ssh/id\_rsa.pub ~/.ssh/authorized\_keys

chmod go-rwx ~/.ssh/authorized\_keys

ssh-agent $SHELL

ssh-add

1. 拷贝临时key到第一台机器

scp ~/.ssh/authorized\_keys 172.20.1.1:/tmp/1

1. 每台机器均执行1~3步，第3步中的文件名设置/tmp/1~10
2. 组合key（第一台机器上执行）

cd /tmp

cat 1 2 3 4 5 6 7 8 9 10 > authorized\_keys

1. 分发key（第一台机器上执行）

执行以下10条命令:

scp authorized\_keys 172.20.1.1:~/.ssh/

...

scp authorized\_keys 172.20.1.10:~/.ssh/

1. 安装expect（每台机器）

sudo apt-get install expect

1. 制作列表（第一台机器上执行）

cd

vim impala-list

内容为：

172.20.1.1

172.20.1.2

172.20.1.3

172.20.1.4

172.20.1.5

172.20.1.6

172.20.1.7

172.20.1.8

172.20.1.9

172.20.1.10

1. 编写并执行脚本（第一台机器上执行）

vim autossh.expect

内容为：

#!/usr/bin/expect

set f [open impala-list]

while {[gets $f line]>=0} {

spawn ssh $line

set timeout 1

expect {

"(yes/no)?" {send "yes\n";exp\_continue}

"No route to host" {send\_user "failed to connect $line"}

}

send "exit\n"

}

expect eof

close $f

chmod u+x autossh.expect

vim autoscp.expect

内容为：

#!/usr/bin/expect

set f [open impala-list]

while {[gets $f line]>=0} {

spawn scp impala-list $line:~/

set timeout 3

expect {

"(yes/no)?" {send "yes\n";exp\_continue}

}

set timeout 100

spawn scp autossh.expect $line:~/

set timeout 3

expect {

"(yes/no)?" {send "yes\n";exp\_continue}

}

set timeout 100

}

close $f

chmod u+x autoscp.expect

./autoscp.expect

1. 访问各节点进行确认（第一台机器上执行）

vim autocheck.expect

内容为：

#!/usr/bin/expect

set f [open impala-list]

while {[gets $f line]>=0} {

spawn ssh $line ~/autossh.expect

set timeout 3

expect {

"(yes/no)?" {send "yes\n";exp\_continue}

}

}

expect eof

close $f

chmod u+x autocheck.expect

./autocheck.expect

**Impala**

1. sudo apt-get install automake libtool flex bison gcc openssl make cmake doxygen python-dev python-pip bzip2 wget git unzip libevent1-dev subversion g++ clang+ libboost-dev libboost-all-dev libglib2.0-dev libsasl2-dev build-essential libssl-dev libbz2-dev
2. Install LLVM

wget http://llvm.org/releases/3.3/llvm-3.3.src.tar.gz

tar xvzf llvm-3.3.src.tar.gz

cd llvm-3.3.src/tools

svn co http://llvm.org/svn/llvm-project/cfe/tags/RELEASE\_33/final/ clang

cd ../projects

svn co http://llvm.org/svn/llvm-project/compiler-rt/tags/RELEASE\_33/final/ compiler-rt

cd ..

CC=/usr/bin/gcc CXX=/usr/bin/g++ ./configure --with-pic

make -j4 REQUIRES\_RTTI=1

sudo make install

1. Add the following to .bashrc

export JAVA\_HOME=/usr/lib/jvm/j2sdk1.6-oracle

export IMPALA\_HOME=~/impala（Impala项目文件夹）

1. Install Maven

wget http://archive.apache.org/dist/maven/binaries/apache-maven-3.2.2-bin.tar.gz

tar xvf apache-maven-3.2.2-bin.tar.gz && sudo mv apache-maven-3.2.2 /usr/local

Add the following three lines to your .bashrc:

export M2\_HOME=/usr/local/apache-maven-3.2.2

export M2=$M2\_HOME/bin

export PATH=$M2:$PATH

And make sure you pick up the changes either by logging in to a fresh shell or running:

source ~/.bashrc

Confirm by running:

mvn -version

and you should see at least:

Apache Maven 3.2.2...

1. 获取源代码：

git clone https://github.com/cloudera/impala.git

或

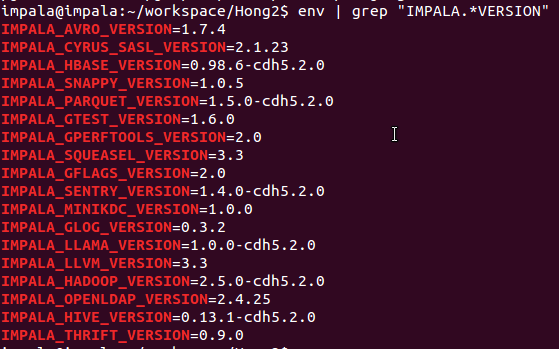
svn co --username XXX https://10.21.31.101/svn/Impala2/trunk

1. Set the Impala environment

cd $IMPALA\_HOME

. bin/impala-config.sh

Confirm your environment looks correct:



1. 清空thirdparty目下各软件的编译信息

以下未特别说明的，均需执行make clean命令：

thrift

gperftools

gflags

glog

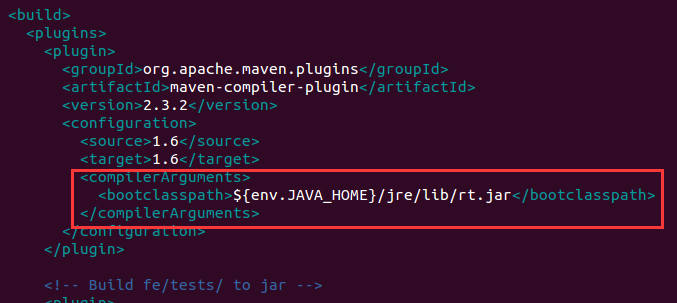
gtest（删CMakeCache.txt，执行./configure后再执行make clean）

snappy

avro-c（删CMakeCache.txt，执行./build.sh clean）

cyrus-sasl

1. 在fe/pom.xml中增加红框部分



1. cd $IMPALA\_HOME

将bin/build\_thirdparty.sh文件中function build\_preamble()函数里的

git clean -dfx

注释掉，并随便加上一行，如：

echo ‘skip git clean’

将buildall.sh中所有的

git clean -Xdf

注释掉。

可从以下选择执行：

1. 编译并跳过测试（初次运行，若报错 执行步骤7清空第三方软件编译文件）：

./buildall.sh -skiptests

1. 重编译并跳过测试，不清除以前编译内容，保持现有Hadoop服务运行（非初次运行，若报错 执行步骤7清空第三方软件编译文件）：

./buildall.sh -skiptests -noclean

1. 只编译前端（在fe目录下）：

mvn package -DskipTests=true

若编译报错，可在mvn后增加-U选项。

1. 修改$IMPALA\_HOME/fe/src/test/resources/core-site.xml文件，将fs.defaultFS对应的值改为：

hdfs://impala:8020

（默认是hdfs://localhost:21000）

1. 在$IMPALA\_HOME下执行

. bin/impala-config.sh

bin/start-statestored.sh &

bin/start-catalogd.sh &

bin/start-impalad.sh（若需单步调试则不执行该命令）

1. 打开一个新的端口，执行$IMPALA\_HOME/bin/impala-shell.sh即可启动命令行工具执行查询等操作。

注：需要统计信息的，在Hive下执行如下命令：

analyze table XXX compute statistics；

查看统计信息（Hive下）：

show extended XXX;

若没有表行数信息，可手动添加（Hive下）：

alter table XXX set tblproperties(‘numRows’=’XX’);

**Easy Test**

hive

create table test(number int);

load data local inpath ‘num.txt’ overwrite into table test;

exit;

impala-shell

（若为虚拟机，且处于未登录状态时，输入connect;

若是集群状态，还需要选择具体连接哪台机器，如connect 172.20.1.2:21000）

refresh test;

select count(\*) from test;

exit;

**部署集群时常见问题及解决方案**

1. **如果先前集群已经安装有Cloudera Manager 4.x版本，需卸载以后重装CM5，可将以下内容复制到uninstall.txt文件中，并重命名为.sh后缀，在系统中运行。该脚本将卸载并删除Cloudera Manager中所有服务和数据，最后两条命令将删除hadoop，hue，yarn，sqoop，ooize，zookeeper的数据，可按需删除。**

sudo rm -Rf /usr/share/cmf /var/lib/cloudera\* /var/cache/yum/cloudera\*

sudo /usr/share/cmf/uninstall-cloudera-manager.sh

sudo service cloudera-scm-server stop

sudo service cloudera-scm-server-db stop

sudo service cloudera-scm-agent stop

sudo apt-get remove cloudera-manager-server

sudo apt-get remove cloudera-manager-server-db-2

sudo service cloudera-scm-agent hard\_stop

sudo /usr/sbin/service cloudera-scm-agent hard\_stop

sudo apt-get purge 'cloudera-manager-\*'

sudo apt-get purge 'cloudera-manager-\*' bigtop-utils bigtop-jsvc bigtop-tomcat hadoop hadoop-hdfs hadoop-httpfs hadoop-mapreduce hadoop-yarn hadoop-client hadoop-0.20-mapreduce hue-plugins hbase hive oozie oozie-client pig zookeeper hue impala impala-shell solr-server

sudo apt-get purge 'cloudera-manager-\*' avro-tools crunch flume-ng hadoop-hdfs-fuse hadoop-hdfs-nfs3 hadoop-httpfs hbase-solr hive-hbase hive-webhcat hue-beeswax hue-hbase hue-impala hue-pig hue-plugins hue-rdbms hue-search hue-spark hue-sqoop hue-zookeeper impala impala-shell kite llama mahout oozie pig pig-udf-datafu search sentry solr-mapreduce spark-python sqoop sqoop2 whirr

sudo rm -Rf /usr/share/cmf /var/lib/cloudera\* /var/cache/yum/cloudera\* /var/log/cloudera\* /var/run/cloudera\*

sudo rm /tmp/.scm\_prepare\_node.lock

sudo rm -Rf /var/lib/flume-ng /var/lib/hadoop\* /var/lib/hue /var/lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop\* /var/lib/zookeeper

sudo rm -Rf /dfs /mapred /yarn

1. **安装CM期间，在cluster中添加host并安装ClouderaManager agent，如果期间abort installation，之后会一直等待获取锁，一段时间后提示失败。**

**解决办法**：在host标签栏，选中该host，然后在上面的action控制按钮中选择，从集群中remove这个host，然后在delete后，重新选择机器安装。

1. **部署hdfs常见问题**
   1. 遇到异常org.apache.hadoop.security.AccessControlException: Permission denied: user=udms, access=udms1234

org.apache.hadoop.security.AccessControlException: Permission denied: user=root, access=WRITE, inode="/user":hdfs:supergroup:drwx at org.apache.hadoop.hdfs.server.namenode.DefaultAuthorizationProvider.checkFsPermission(DefaultAuthorizationProvider.java

**解决办法：**由于提示登陆用户权限不够，无法写/user目录，通过查看hdfs的文件目录，发现/user的最高权限用户为“hdfs”，因此，使用 sudo -u hdfs hadoop fs -chmod 777 /user

* 1. 遇到异常Exception in secureMainjava.lang.RuntimeException: Cannot start datanode because the configured max locked memory size (dfs.datanode.max.locked.memory) of 4294967296 bytes is more than the datanode's available RLIMIT\_MEMLOCK ulimit of 65536 bytes.

**原因：**HDFS的caching机制在CDH5中默认开启，因此在CM5的客户端中需要重启服务，重新调整memlock大小（requires new memlock functionality fromCloudera Manager 5 Agents）。

**解决办法：**

1.停止集群所有的服务，也包括Cloudera Manager服务。

2.依次在每台机器上运行以下命令，重启agent服务：

$sudo service cloudera-scm-agent hard\_restart

3.启动所有的服务。

* 1. 提示无法格式化NameNode

**原因：**/data目录下已经存在/dfs文件夹，产生冲突。

**解决办法：**删除NameNode文件目录下/data/dfs文件夹

* 1. 提示

Initialization failed for Block pool <registering> (Datanode Uuid unassigned) service to udms-101.lab.udms.org/172.21.1.101:8022. Exiting. java.io.IOException: Incompatible clusterIDs in /dfs/dn: namenode clusterID = cluster18; datanode clusterID = cluster16

**原因：**文件目录下已经存在/dfs文件夹，产生冲突。

**解决办法：**删除DataNode文件目录下的/dfs文件夹

1. **部署hive常见问题**
   1. there is no mysql-server on XXX

**解决办法：**检查XXX机器上是否安装有mysql，如没有则安装mysql，

通过netstat –nap|grep 3306查看mysql端口是否在监听，若为listening，则正常，否则重启mysql服务。重新部署hive，如若继续提示没有mysql server，则授权Mysql允许外部机器进行连接，参考4（b）。

* 1. is not allowed to connect to this MySQL serverConnection closed by foreign host

**原因：**MySql服务器不允许外部机器连接。

**解决办法：**

1.GRANT ALL PRIVILEGES ON \*.\* TO ‘hive'@'%' IDENTIFIED BY

'hive' WITH GRANT OPTION;

2.FLUSH PRIVILEGES;

* 1. Unable to connect to database on host 'udms-101.lab.udms.org:3306' from host 'udms-101.lab.udms.org' using the credential provided

**原因：**数据库用户名，密码须正确填写。

**解决办法：**对应于mysql中‘hive’数据库，需要对Cloudera Manager添加hive服务中，将要使用的用户名及密码进行授权（参考4（b））。

1. **如果CM在下载CDH5.2.x包的时候，提示操作系统不兼容，可能是当前操作系统版本过低，（如Ubuntu10.04），则需要将操作系统升级至12.04，操作如下：**

1.修改/etc/apt/source.list的源为12.04版本的源地址；

2.输入以下命令，更新源：

sudo apt-get -y update

sudo apt-get -y upgrade

sudo apt-get -y dist-upgrade

3.更新内核

sudo apt-get install libc6-dev

sudo apt-get install util-linux -f -o APT::Immediate-Configure=0

sudo apt-get install python-minimal -o APT::Immediate-Configure=0

sudo apt-get install passwd -o APT::Immediate-Configure=0

**参考资料**

Installing Impala without Cloudera Manager

<http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/impala_noncm_installation.html>

Post-Installation Configuration for Impala

<http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/impala_config_performance.html#config_performance>

Starting Impala

<http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/impala_processes.html#processes>

Uninstalling Cloudera Manager and Managed Software

<http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_ig_uninstall_cm.html>

Using HDFS Caching with Impala (CDH 5 Only)

<http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v1/latest/Installing-and-Using-Impala/ciiu_perf_hdfs_caching.html>

Configuring Centralized Cache Management in HDFS

<http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_ig_hdfs_caching.html>

Enable LZO compression on Hadoop, Pig and Spark

<http://hsiamin.com/posts/2014/05/03/enable-lzo-compression-on-hadoop-pig-and-spark/>

Troubleshooting Installation and Upgrade Problems

<http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_ig_troubleshooting.html>