GraphLab 安装试用（多台机器）

下面提到的错误是我在安装时出现的，在各自安装的时候还有可能很多未知的错误，到时候最好的办法就是直接去百度= =

还由于自身知识水平有限，可能安装了一些重复的软件，这个在以后的学习中会慢慢积累。

1. 在 root 目录下创建 SSH 信任链接

为方便将两台 PC 主机名改为 station1 and station2

临时修改：

#hostname station1

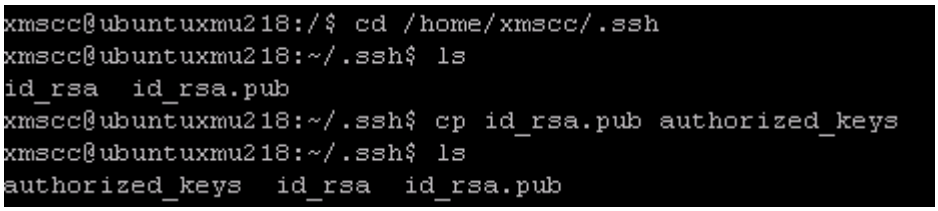
永久修改：

#vi /etc/hosts

把 ip 地址后的主机名改为 station1 即可

1.1 在 station1 生成 SSH 秘钥对

```
xmscc@ubuntuxmu218:/$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/xmscc/.ssh/id_rsa):
Created directory '/home/xmscc/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/xmscc/.ssh/id_rsa.
Your public key has been saved in /home/xmscc/.ssh/id_rsa.pub.
The key fingerprint is:
c3:aa:4d:eb:cc:4d:bb:9d:cc:23:b3:e1:ea:3e:d3:50 xmscc@station1
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|       .E        |
|       .S        |
|      .. .       |
|      ooo        |
|     *o=+*..     |
|    .+X+*==.     |
+-----------------+
```



1.2 建立本身的信任连接

#ssh station1

可能会出现以下出错提示：

ssh: Could not resolve hostname station1: Name or service not known

此时就需要刚刚的永久修改主机名的步骤了：

#vi hosts （sudo 是为了暂时获取 root 权限）

```
请按 ENTER 或其它命令继续
xmscc@station1:/etc$ sudo vi hosts
sudo:  无法解析主机: station1
[sudo] password for xmscc:
xmscc@station1:/etc$ cat hosts
127.0.0.1       localhost
192.168.2.218   station1

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
xmscc@station1:/etc$ cd ..
xmscc@station1:/$ ssh station1
The authenticity of host 'station1 (192.168.2.218)' can't be established.
ECDSA key fingerprint is be:10:d8:c0:28:29:04:93:fc:57:9c:cb:a6:7b:11:fc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'station1,192.168.2.218' (ECDSA) to the list of known
 hosts.
Enter passphrase for key '/home/xmscc/.ssh/id_rsa':
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Aug 11 19:47:14 CST 2014

  System load:  0.0               Processes:           110
  Usage of /:   2.9% of 40.95GB   Users logged in:     1
  Memory usage: 2%                IP address for eth0: 192.168.2.218
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Last login: Mon Aug 11 19:42:17 2014 from 192.168.2.212
xmscc@station1:~$
```

此时，与自身的连接建立好。

1.3 设置 station2 (同样在 root 目录下)

注意：

需要 station1 的 IP 地址

路径也要是自己.ssh 文件安装的地址

#ssh-keygen –t rsa 生成.ssh 文件夹

#scp 172.17.2.1：/root/.ssh/* /root/.ssh 拷贝 station1 上的. ssh 文件

夹覆盖本地的

```
xmscc@station2:/$ scp 192.168.2.218:/home/xmscc/.ssh/* /home/xmscc/.ssh
The authenticity of host '192.168.2.218 (192.168.2.218)' can't be established.
ECDSA key fingerprint is be:10:d8:c0:28:29:04:93:fc:57:9c:cb:a6:7b:11:fc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.218' (ECDSA) to the list of known hosts.
xmscc@192.168.2.218's password:
authorized_keys                               100%  396     0.4KB/s   00:00
id_rsa                                        100% 1766     1.7KB/s   00:00
id_rsa.pub                                    100%  396     0.4KB/s   00:00
known_hosts                                   100%  666     0.7KB/s   00:00
xmscc@station2:/$
```

#scp 172.17.2.1 :/etc/hosts /etc/hosts　拷贝 station1 上的 hosts 文件覆盖本地的

（我操作的时候这里是禁止的，只要保证两个 hosts 中的文件是一样的就可以）

#ssh station1　提示处输入 yes 回车

```
xmscc@station2:/etc$ ssh station1
Enter passphrase for key '/home/xmscc/.ssh/id_rsa':
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Aug 16 12:39:26 CST 2014

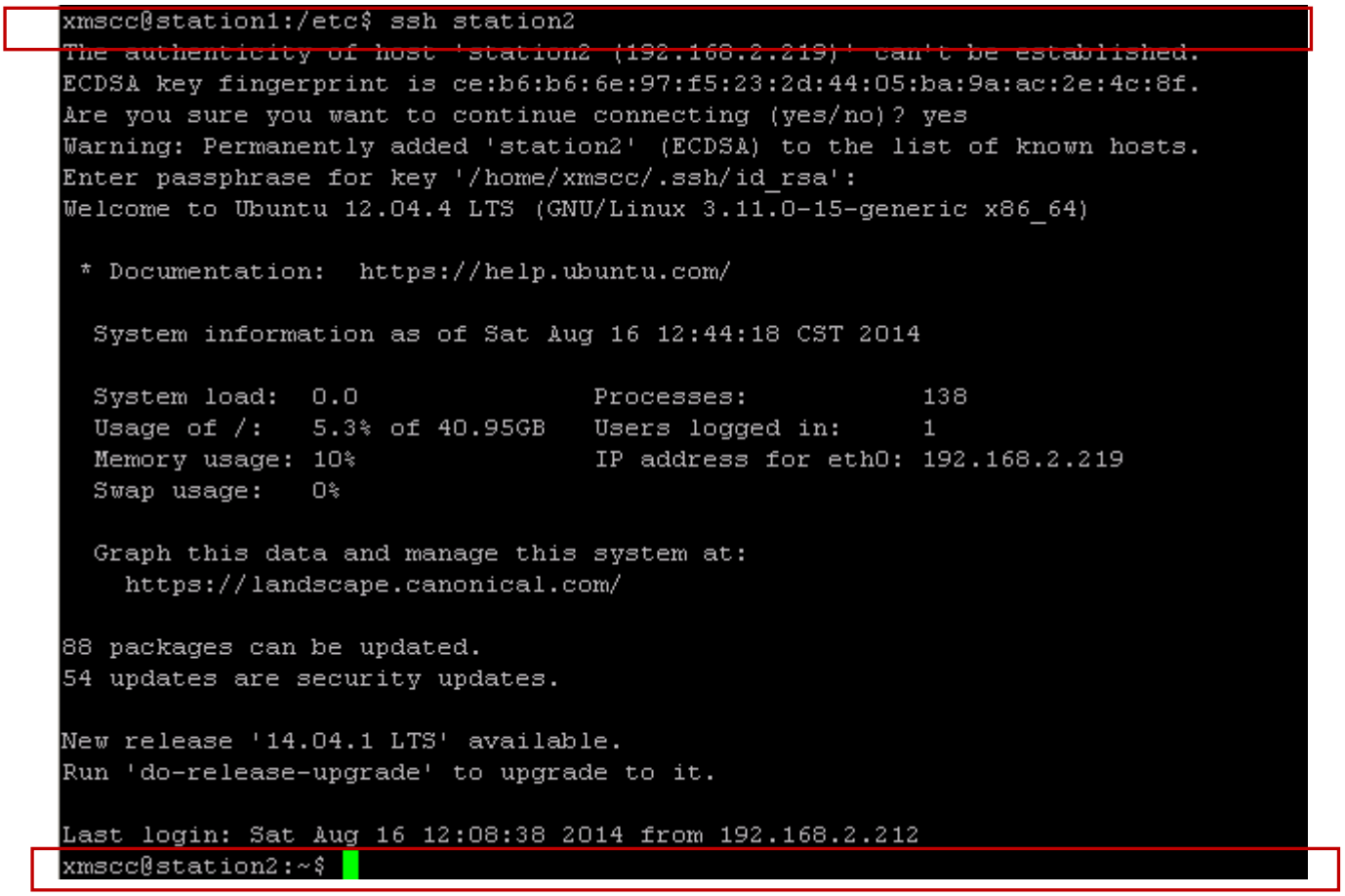
  System load:  0.0               Processes:           110
  Usage of /:   2.9% of 40.95GB   Users logged in:     1
  Memory usage: 2%                IP address for eth0: 192.168.2.218
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

82 packages can be updated.
54 updates are security updates.

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Aug 16 12:08:18 2014 from 192.168.2.212
xmscc@station1:~$
```



设置成功。

1. 安装 mpich

因为是纯净的安装环境，所以先安装 gcc 编译器：

sudo apt-get install gcc

1.1 编译安装

$./configure

$make

$make install

创建配置文件（mpd.conf）

#touch /etc/mpd.conf

#chmod 600 /etc/mpd.conf

#echo secretword=myword >> /etc/mpd.conf

Station1 安装测试

```
root@station1:/etc# mpd &
[2] 29262
[1]   完成                  mpd
root@station1:/etc# mpdtrace
station1
root@station1:/etc#
```

Station2 安装测试

```
root@station2:/home# mpd &
[1] 17549
root@station2:/home# mpdtrace
station2
root@station2:/home# mpdallexit
[1]+  完成                  mpd
root@station2:/home#
```

集群部署：

创建 mpd.hosts 文件

Station1　　//主服务器

Station2　　//子节点

在主机中测试部署是否成功：

```
root@station1:/home/xmscc# mpdboot -n 2 -f mpd.hosts
root@station2's password:
root@station1:/home/xmscc# mpdtrace
station1
station2
root@station1:/home/xmscc#
```

参数-n 2 指定了要起动的机器个数，-f mpd.hosts 指定了通过 mpd.hosts 运行

2. 安装 jdk

```
xmscc@station1:~$ java -version
java version "1.8.0_11"
Java(TM) SE Runtime Environment (build 1.8.0_11-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.11-b03, mixed mode)
```

```
xmscc@station2:~$ java -version
java version "1.8.0_11"
Java(TM) SE Runtime Environment (build 1.8.0_11-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.11-b03, mixed mode)
```

3. 安装 boost 库

   进入到 boost 目录，执行./bootstrap.sh

   修改 tools/build/v2/user-config.jam 在最后一行加 using mpi ; 注意 mpi 和分号之间有空格

   编译 ./b2

   安装 ./b2 install

4. 安装 graphlabapi

   解决依赖 （gcc g++ jdk 已经安装，无须再用命令）

```
apt-get install gcc g++ build-essential libopenmpi-dev default-jdk cmake zlib1g-dev
```

```
[ 41%] No install step for 'boost'
[ 41%] Completed 'boost'
[ 41%] Built target boost
make: *** [all] 错误 2
```

   在这里重新下载了一个包： gperftools-2.2.tar.gz 安装在 graphlabapi/deps/tcmalloc/src 下

   但是在后来运行 kmeans 的时候还会有问题，因此又安装了一个 openmpi-bin 。

   ./configure

```
*******************************************************
*                  Important Note                     *
*                  --------------                     *
*       You do not have to build everything!          *
*                                                     *
* Everything takes a very long time, and a ton of     *
* of memory to build. You can just cd into toolkit    *
* directory you want, and just build that. For        *
* instance, if I just want the release build of       *
* PageRank, I could:                                  *
*                                                     *
*       cd release/toolkits/graph_analytics           *
*       make -j2                                      *
*                                                     *
* Use at most [RAM in GB] parallel builds. The        *
* compilation consumes a lot of RAM. i.e. if you      *
* 4 GB of RAM, do not do more than make -j4           *
*******************************************************
```

按照提示，或者进入到 release/toolkits/clustering 中，make 即可

安装好之后,clustering 中生成了 generate_synthetic &kmeans 等

```
root@station2:/home/xmscc/graphlabapi/release/toolkits/clustering# ls
CMakeFiles            generate_synthetic      Makefile
cmake_install.cmake   graph_laplacian_for_sc  spectral_clustering
CTestTestfile.cmake   kmeans
root@station2:/home/xmscc/graphlabapi/release/toolkits/clustering# ./generate_sy
nthetic
Usage: generate_synthetic [NClusters] [Dimensions] [Ndata]
Center 0 at:  6.18472    -2.33358
Center 1 at: -3.23826    -1.70851
root@station2:/home/xmscc/graphlabapi/release/toolkits/clustering# ls
CMakeFiles            generate_synthetic      Makefile
cmake_install.cmake   graph_laplacian_for_sc  spectral_clustering
CTestTestfile.cmake   kmeans                  synthetic.txt
```

5. 启动 mpi

```
root@station2:/home/xmscc/graphlabapi/release/toolkits/clustering# mpdboot -n 2 -
osts
The authenticity of host 'station1 (192.168.2.218)' can't be established.
ECDSA key fingerprint is be:10:d8:c0:28:29:04:93:fc:57:9c:cb:a6:7b:11:fc.
Are you sure you want to continue connecting (yes/no)? yes
root@station1's password:
root@station2:/home/xmscc/graphlabapi/release/toolkits/clustering#
```

#mpdtrace

启动之后就可以开始运行了。

```
root@station2:/home/xmscc/graphlabapi/release/toolkits/clustering# mpiexec -n 2 .
ic.txt --clusters 2 --output-clusters s1.txt
Computes a K-means clustering of data.

GRAPHLAB_SUBNET_ID/GRAPHLAB_SUBNET_MASK environment variables not defined.
Using default values
Subnet ID: 0.0.0.0
Subnet Mask: 0.0.0.0
Will find first IPv4 non-loopback address matching the subnet
Computes a K-means clustering of data.

GRAPHLAB_SUBNET_ID/GRAPHLAB_SUBNET_MASK environment variables not defined.
Using default values
Subnet ID: 0.0.0.0
Subnet Mask: 0.0.0.0
Will find first IPv4 non-loopback address matching the subnet
INFO:     dc.cpp(init:573): Cluster of 1 instances created.
INFO:     dc.cpp(init:573): Cluster of 1 instances created.
INFO:     distributed_graph.hpp(set_ingress_method:3201): Automatically determine
grid
INFO:     distributed_graph.hpp(load_from_posixfs:2190): Loading graph from file:
INFO:     distributed_ingress_base.hpp(finalize:199): Finalizing Graph...
INFO:     distributed_graph.hpp(set_ingress_method:3201): Automatically determine
grid
INFO:     distributed_ingress_base.hpp(exchange_global_info:546): Graph info:
         nverts: 10000
         nedges: 0
         nreplicas: 10000
         replication factor: 1
Number of datapoints: 10000
INFO:     distributed_graph.hpp(load_from_posixfs:2190): Loading graph from file:
Validating data...Initializing using Kmeans++
Running Kmeans...
Kmeans iteration 1: # points with changed assignments = 247 total cost: 20031.4
Kmeans iteration 2: # points with changed assignments = 24 total cost: 19894.6
Kmeans iteration 3: # points with changed assignments = 2 total cost: 19893.3
Kmeans iteration 4: # points with changed assignments = 1 total cost: 19893.3
Kmeans iteration 5: # points with changed assignments = 0 total cost: 19893.3
Writing Cluster Centers...
INFO:     distributed_ingress_base.hpp(finalize:199): Finalizing Graph...
INFO:     distributed_ingress_base.hpp(exchange_global_info:546): Graph info:
         nverts: 10000
         nedges: 0
         nreplicas: 10000
         replication factor: 1
Number of datapoints: 10000
Validating data...Initializing using Kmeans++
Running Kmeans...
Kmeans iteration 1: # points with changed assignments = 0 total cost: 19917.4
Writing Cluster Centers...
```

参考：

1.MPI 并行计算环境搭建以及集群测试

http://wuyanzan60688.blog.163.com/blog/static/127776163201324520362 23/

2.RHEL5 下利用 MPICH2 配置并行运算集群

3.Ubuntu 下 Boost 库的安装

http://blog.163.com/s591232583@126/blog/static/776848292014713 8435737

4. linux 如何安装 jdk

http://jingyan.baidu.com/article/d621e8dae805272865913fa7.html