GraphLab 部署过程

1. GraphLab 简单介绍

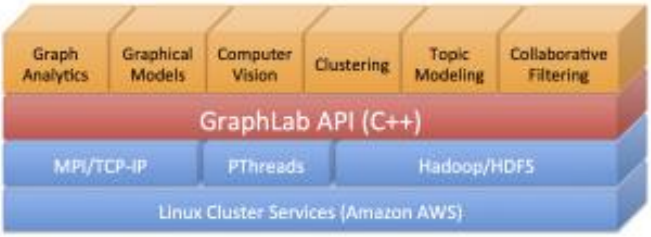
GraphLab 是 CMU（卡耐基梅隆大学）开发的一个以 vertex 为计算单元的大规模图处理系统，是继 google 的 Pregel 之后的第一个开源的大规模图处理系统，它解决了传统 MapReduce 框架对于机器学习应用的处理中最突出的两个问题（频繁迭代计算和大量节点通信）引起的计算效率的问题，与 Haloop，Twister 等基于 MapReduce 批量处理不同的是，它采用 Pregel 的以 vertex 为计算单元，并将机器学习抽象成 GAS（gather，apply，scatter）三个步骤，然后按该抽象模型设计实现算法，事实已经证明该框架对于机器学习这一类跟图处理关系紧密的应用有很好的效果。

我们部署的版本是 **PowerGraph**，是 github 上面的开源项目，链接：https://github.com/jegonzal/PowerGraph，可以直接基于它开发 C++应用。现在 GraphLab 的小组已经成立了公司，对应的产品为 GraphLab-Create,在原有的基础上用 Python 进行封装，还进行了一些优化。GraphLab-Create 并不是免费的，可以加入 Academic Program 免费试用一年。相应地，他们小组也逐渐弃用了 PowerGraph。

这是之前弄的，现在已经没有用这个 ，这个文档什么的都不全，如果是新做图挖掘，建议还是使用官方提供的 Graphlab-Create。

2. 整体部署说明

GraphLab 的这个 PowerGraph 项目包括处于顶层的核心 API、机器学习和数据挖掘的工具包。通过 TCP/IP 进行进程间通信，使用 MPI 来启动和管理 PowerGraph 程序，而且每一个程序都是多线程的。



在安装和编译 GraphLab 源代码之前，需要先安装配置 SSH 免密码登陆和 MPI。因为结点间需要管理远程进程，就必须保证在节点间执行指令

的时候不需要输入密码，所以需要 SSH 免密码登陆，而 MPI 的实现应用（MPI 是一套标准，有很多此标准的实现应用，如 MPICH2）则为并行应用提供消息传递或者相关服务。

3. 部署过程

3.1. 安装 Linux 操作系统

1） 主机环境说明

集群中使用了 3 台主机，每台主机有 CPU 核 4 个，内存 8G，每台主机上安装的操作系统为 Ubuntu14.04 Desktop 64bit，每台主机上面都使用相同的用户名 graphlab。

	型号	操作系统	用户名	主机名	IP 地址
主机 1	HP Z230 Tower Workstation	Ubuntu 14.04 64bit	graphlab	graphlabmaster	192.168.0.5
主机 2	HP Z230 Tower Workstation	Ubuntu 14.04 64bit	graphlab	Graphlabslave1	192.168.0.6
主机 3	HP Z230 Tower Workstation	Ubuntu 14.04 64bit	graphlab	Graphlabslave2	192.168.0.7

可能这个主机名有迷惑作用，会认为主机之间存在主次之分，是因为之前考虑安装 Hadoop 遗留的问题，就是说，GraphLab 的主机结点没有主次之分。

需要注意的是，根据 GraphLab 官方指南，安装 GraphLab 需要使用 64bit 的操作系统。而且根据网上博客内容，最后每台主机的机型完全一致，因为 GraphLab 使用 C++开发，相比 Java 的一处编译多处运行，C++并不具有这种特性。要想让在一台主机上编译的代码能够在其他主机上正确运行，一定要确保所有主机型号一致（但是博主实验时不同机型的主机也可以并行，所以这只是作参考，最好机型一致）。

2） 安装注意事项

建议配置**相同的用户名**，不同的用户名可以配置使用，但是第一是更加麻烦，第二是可能会遇到问题，比如以后想使用网络文件系统 NFS，用户名不同会产生权限之类的问题。

用户的$HOME 目录在文件系统中应该**完全相同**

3.2. 修改主机文件

1） 在每台主机上修改主机名

可能安装系统的时候主机名不是自己想要的，那么就需要修改自己的主机名，如果觉得不影响，可以不作任何修改。

- 修改文件/etc/hostname,把主机名修改为自己想要的名字
- 修改文件/etc/hosts,把原来的主机名改为新的主机名

2） 在每台主机上修改主机文件

需要把其他主机的 IP 和主机名对应关系写在/etc/hosts 文件中，那么集群中的主机就通过主机名互相识别，某一个主机的/etc/hosts 文件内容示例如下：

```
127.0.0.1       localhost
192.168.0.5 graphlabmaster
192.168.0.6  graphlabslave1
192.168.0.7  graphlabslave2

# 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
```

我们配置的集群有三台主机，分别是 graphlabmaster,graphlabslave1 和 graphlabslave2,前面是他们对应的 IP 地址，需要注意的是：默认会有一行内容为 127.0.0.1+主机名，删掉就可以了。

3.3. 配置 ssh 免密码登陆

1) ssh 使用原理

- ssh 采用公钥加密，每一台主机都会有自己的公钥和私钥
- 每一台主机都把自己的公钥分发给其他主机，就是说所有主机都知道集群中其他主机的公钥
- 举一个远程登陆的例子，当 Master 主机通过 SSH 连接 Salve 主机时，Salve 就会生成一个随机数并用 Master 的公钥对随机数进行加密，并发送给 Master。Master 收到加密数之后再用私钥解密，并将解密数回传给 Slave，Slave 确认解密数无误之后就允许 Master 进行连接了。这就是一个公钥认证过程，其间不需要用户手工输入密码

2) 在所有主机中安装 ssh

ubuntu14.04 中默认安装有 Openssh-client,只需要在每个主机中都安装 Openssh-server

```
sudo apt-get install openssh-server
```

3) 生成密钥

正常的过程是所有主机都生成自己的公钥和私钥密钥对，但是为了部署简单，现在一般也都是只需要一台主机生成密钥对，然后把密钥对拷贝至其他主机，所以所有的主机都有相同的密钥对，那么就可以完成认证过程。

在一台主机上生成密钥，生成过程中会提示输入密码，可以输入密码（更加安全），也可以直接跳过，默认的保存目录是~/.ssh/id_rsa location

```
ssh-keygen -t rsa
```

进入.ssh 目录中，添加公钥至授权的 keys

```
cat id_rsa.pub >> authorized_keys
```

4) 修改 ssh 配置文件

打开 ssh 配置文件"/etc/ssh/sshd_config"，取消文件中下列内容的注释”#”

```
    RSAAuthentication yes
    PubkeyAuthentication yes
    AuthorizedKeysFile  %h/.ssh/authorized_keys
```

设置完毕后重启服务，配置才生效

```
service ssh restart
```

设置完毕后，现在还只是在一台主机上面进行的设置，可以自己验证是否配置成功，如果能够登陆本机成功，则配置成功，首先需要配置本地登录成功

```
ssh localhost
```

注意事项：此处很可能不能成功，请检查三个文件目录的权限，$HOME/.ssh/authorized_keys 文件（可以设置为 600），$HOME/.ssh 目录(可以设置为 700)，$HOME 目录（可以设置为 731），这三个目录都只能设置成拥有者有写权限，否则 sshd 不工作，因为 sshd 发现这些目录别人也有写权限的话，它会认为别人也有能力篡改 authorized_keys 文件中的值，那么是不安全的，所以就不会工作。

设置文件权限,如 authorized_keys 的权限设置为 600

```
chmod 600 authorized_keys
```

如果文件权限没有问题，那么便查看自己防火墙 iptables 是否关闭

5) 关闭所有主机的防火墙 iptables

```
sudo ufw disable
```

6) 部署其他主机

刚才在一台主机上面部署了 ssh 并可以本地免密码登陆，现在要实现互相可以免密码登陆。

拷贝.ssh 目录至其他主机，可以使用 scp 命令,如

```
scp –r .ssh graphlab@192.168.0.7:~/
```

这条命令拷贝.ssh 目录至 192.168.0.7 主机中 graphlab 用户的 $HOME 目录下。

参照第 4）步中检查 ssh 的配置文件，重启服务，检查文件夹的权限，然后检查能否互相免密码登陆(按道理是可以的，因为所以的密钥都是相同的，验证是可以成功的)

验证互相之间是不是可以免密码登陆。

3.4. 安装编译 GraphLab

1) 为每台主机更新一下源，保证获取的安装包都是最新的

```
sudo apt-get update
```

2) 为每台主机安装 GraphLab 的依赖库

```
sudo apt-get install gcc g++ build-essential libopenmpi-dev openm
pi-bin default-jdk cmake zlib1g-dev git
```

3) 从 GitHub 上下载 GraphLab(接下来的操作在一台主机上)

选取集群中的一台机器，进入要安装 GraphLab 的目录，使用如下的 git 命令下载 GraphLab

```
git clone https://github.com/jegonzal/PowerGraph.git
```

4) 编译 GraphLab

进入 graphlab 文件夹，使用 graphlab 自带的 configure 脚本配置编译环境

```
cd graphlab
./configure
```

配置成功后会在 graphlab 文件夹内生成 release 和 debug 两个新的目录。这两个目录分别对应不同项目的发行版和测试版，在这两个目录中都可以编译 GraphLab 的所有 Toolkit，分别对应发行版和测试版。编译后发行版与测试版的不同是，发行版在编译过程中程序都做了优化，运行速度更快。

还有一点需要特别指出，GraphLab 不仅提供了分布式大规模图计算模型，而且基于该模型实现了很多实用的工具集，这些工具集可以分成六类：主题建模、图分析、聚类、协同过滤、图模型和计算机视觉。可以根据自己的需要只编译其中的某一类或几类。如果全部编译，第一次编译时会下载很多的库文件，耗费很长时间。我只对其中的图分析工具集比较感兴趣，所以只编译了这一个。同时我也编译了 apps 目录中的相应样例代码。

编译 release 目录下的 apps 子目录：

```
cd release/apps
make -j 3
```

第二行中的参数-j 3 是利用了 make 的并行编译特性，3 指的是同时进行三个编译任务。该数字越大，并行性越高，编译速度越快，但是占用内存也越多。如果该数字过大，会因内存不够用而使编译过程卡住。

编译 release 目录下的 toolkits 中的 graph_analytics：

```
cd release/toolkits/graph_analytics
make -j 3
```

如果希望编译整个 GraphLab，那么可以在 release 目录下运行如下命令：

```
cd release
make -j 3
```

注意事项：编译的过程不一定都顺利，可能会有文件下载失败的问题和权限问题等

- 如果提示 hash not match,则为文件内容不全，下载失败。解决办法：首先检查是否使用代理，网上说代理一般是不能成功的。如果没有使用代理，还是不能成功，可以自己手动下载文件，然后把文件放在指定的目录即可。
- /hadoop/src/hadoop/src/c++/libhdfs/configure: Permission denied，则是文件权限的问题，解决办法：

chmod a+rx /hadoop/src/hadoop/src/c++/libhdfs/configure ，就是给所以用户加上读和执行的权限。

5) 集群测试

在所有结点上创建一个文件，文件名为“machines”，文件里面内容为所有结点的主机名,最好是在一台主机上面新建文件，然后复制至其他主机，保证内容相同。内容示例如下：

```
graphlabmaster
graphlabslave1
graphlabslave2
```

在安装 GraphLab 的主机上面运行如下命令，GraphLab 提供了脚本用于分发编译好的二进制可执行文件到集群中的所有其他机器上。在你下载并编译了 GraphLab 的那台机器上，使用下面的命令来分发二进制可执行文件和相关库文件把一些 GraphLab 文件拷贝至其他主机

```
   cd ~/graphlab/release/toolkits
   ~/graphlab/scripts/mpirsync
   cd ~/graphlab/deps/local
   ~/graphlab/scripts/mpirsync
```

单机测试，在每一台机器上，运行如下的命令来测试分发到每台机器上的二进制可执行程序能否正确运行：

```
    cd ~/graphlab/release/toolkits/graph_analytics/
   ./pagerank --powerlaw=10000
```

分布式测试，在任意一台机器上，运行如下两条命令：

```
   cd ～
mpiexec -n 2 -hostfile machines graphlab/release/toolkits/graph_a
nalytics/pagerank --powerlaw=100000
```

如果上述命令能够正确无误执行，那么 GraphLab 分布式集群运算环境搭建就算完成了。

4. 主要参考资料

[1] 搭建 GraphLab 集群总结
http://www.cnblogs.com/jasonkoo/p/3257517.html

[2] GraphLab PowerGraph v2.2

https://github.com/dato-code/PowerGraph/blob/master/README.md

[3] GraphLab PowerGraph Tutorials

https://github.com/dato-code/PowerGraph/blob/master/TUTORIALS.md#cluster

[4] Setting Up an MPICH2 Cluster in Ubuntu

https://help.ubuntu.com/community/MpichCluster

[5] 一步步教你 Hadoop 多节点集群安装配置

http://www.cnblogs.com/lanxuezaipiao/p/3525554.html

by 王迪