**GraphLab部署过程**

1. GraphLab简单介绍

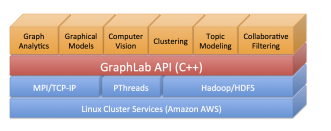
GraphLab是CMU（卡耐基梅隆大学）开发的一个以vertex为计算单元的大规模图处理系统，是继google的Pregel之后的第一个开源的大规模图处理系统，它解决了传统MapReduce框架对于机器学习应用的处理中最突出的两个问题（频繁迭代计算和大量节点通信）引起的计算效率的问题，与Haloop，Twister等基于MapReduce批量处理不同的是，它采用Pregel的以vertex为计算单元，并将机器学习抽象成GAS（gather，apply，scatter）三个步骤，然后按该抽象模型设计实现算法，事实已经证明该框架对于机器学习这一类跟图处理关系紧密的应用有很好的效果。

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。

1. 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。

1. 部署过程
   1. 安装Linux操作系统
2. 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）。

1. 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**

* 1. 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

1. 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。

* 1. 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

1. 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

1. 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

1. 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

1. 关闭所有主机的防火墙iptables

sudo ufw disable

1. 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。

* 1. 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

1. 为每台主机安装GraphLab的依赖库

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

1. 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

1. 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。

1. 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\_analytics/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。

1. 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