

林禹臣 5140309507

December 27, 2015 (Week 16)

电类工程导论C 实验报告

——Hadoop 基础

目录页

1. 引言
2. 实验环境
3. 实验原理
4. 实验过程
 - 4.1. 通过 homebrew 安装 hadoop
 - 4.2. 配置 OS X 下的 sshd 服务
 - 4.3. 修改各种配置文件
 - 4.4. 解决 DataNode 无法启动
 - 4.5. word count 实验
 - 4.6. 圆周率 π 计算实验
5. 实验总结
6. 参考

1.引言

在这个实验里，我完成了在 OS X 10.11.2 上搭建hadoop 2.5.2环境，解决了很多在 OS X 上才有出现的问题，完成了一个 mapreduce 的样例，通过测试得到关于计算 π 时的精确度和时间与 map 和 sample 的数量关系。

2.实验环境

OS X 10.11.2 + hadoop 2.5.2 + java 1.6.0

3.实验原理

3.1 Hadoop

Hadoop 是一个能够对大量数据进行分布式处理的框架，最主要目的是让用户可以在不了解分布式底层细节的情况下，简易地开发分布式程序，从而充分利用服务器集群的威力来进行高速运算和存储。

Hadoop 最重要的两个部分是 HDFS 和 MapReduce

3.2 HDFS

HDFS = Hadoop Distributed File System

是一个分布式文件系统，特点是具有高容错性，可以以流的方式访问文件系统中的数据。

3.3 MapReduce

MapReduce 广义上指的是一种变成模型。Map 和 Reduce 是两个操作。Map 是映射，Reduce 是归约。这个思想是从函数式编程中借鉴过来的。用途是进行大规模数据集的并行运算。最重要的特点之一是其鲁棒性。

Map 函数接受一个键值对，然后产生一组中间键值对。

MapReduce 框架将 map 函数产生的中间键值对里Key 相同的值传给一个 Reduce 函数。

Reduce 函数接受一个键，以及相关的一组值，将这组值进行计算产生一个数据规模更小的值，通常只有一个值。

4.实验过程

4.1. 通过 homebrew 安装 hadoop

OS X 本身并不提供类似 apt-get 的包管理器。但是也有第三方做的非常好的，比如 homebrew。使用 homebrew 可以直接运行命令 brew install hadoop25 来安装 hadoop2.5.2。

4.2. 配置 OS X 下的 sshd 服务

由于我们需要 ssh 自己本地，所以需要开启 sshd 服务从而实现把localhost 当做一个 openssl-server 来进行链接。但是 OS X 默认是不开启 sshd，需要手动进行配置：

1.编辑/etc/sshd_config文件，注释掉

```
#ForceCommand /usr/local/bin/ssh_session
```

2.启动sshd服务：

```
sudo launchctl load -w /System/Library/LaunchDaemons/ssh.plist
```

3.停止sshd服务：

```
sudo launchctl unload -w /System/Library/LaunchDaemons/ssh.plist
```

4查看是否启动：

```
sudo launchctl list | grep ssh
```

如果看到下面的输出表示成功启动了：- 0 com.openssh.sshd

5.最后ssh localhost 成功

4.3.修改各种配置文件

这一步基本和下发的pdf所陈述一致，但是针对 mac 还需要进行修改两个地方，第一个是在hadoop-env.sh 中修改

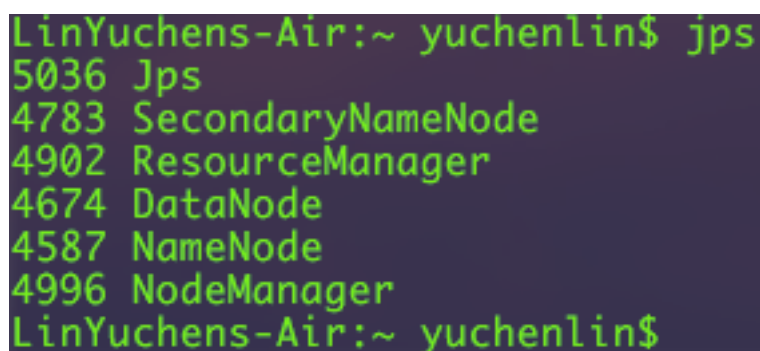
```
export HADOOP_OPTS="-Djava.security.krb5.realm=OX.AC.UK -  
Djava.security.krb5.kdc=kdc0.ox.ac.uk:kdc1.ox.ac.uk"
```

否则会出现一些关于 Realm 的报错信息。

第二个地方是修改 mydata 文件夹所在路径，这个是由于 OS X 目录结构和 linux 不一样导致的。

4.4.解决DataNode 无法启动

由于之间多次进行 hdfs namenode -format 导致 slave 和 master 热的 clusterID 出现了不一致的情况，这时会出现 datanode 无法启动的问题，这时需要手动修改 mydata 的文件夹中的 Version 文件，从而实现统一 clusterID。解决了这些问题之后，终于配置成功：

A terminal window with a dark background and green text. It shows the command 'jps' being executed, which lists several Hadoop processes: Jps, SecondaryNameNode, ResourceManager, DataNode, NameNode, and NodeManager, each with its corresponding PID. The prompt indicates the user is 'yuchenlin' on a machine named 'LinYuchens-Air'.

```
LinYuchens-Air:~ yuchenlin$ jps  
5036 Jps  
4783 SecondaryNameNode  
4902 ResourceManager  
4674 DataNode  
4587 NameNode  
4996 NodeManager  
LinYuchens-Air:~ yuchenlin$
```

4.5. word count 实验

接下来我按照 ppt 的指示进行了第一个测试，那就是 word count 实验。但是一开始就遇到了一个问题，ppt 中直接利用了

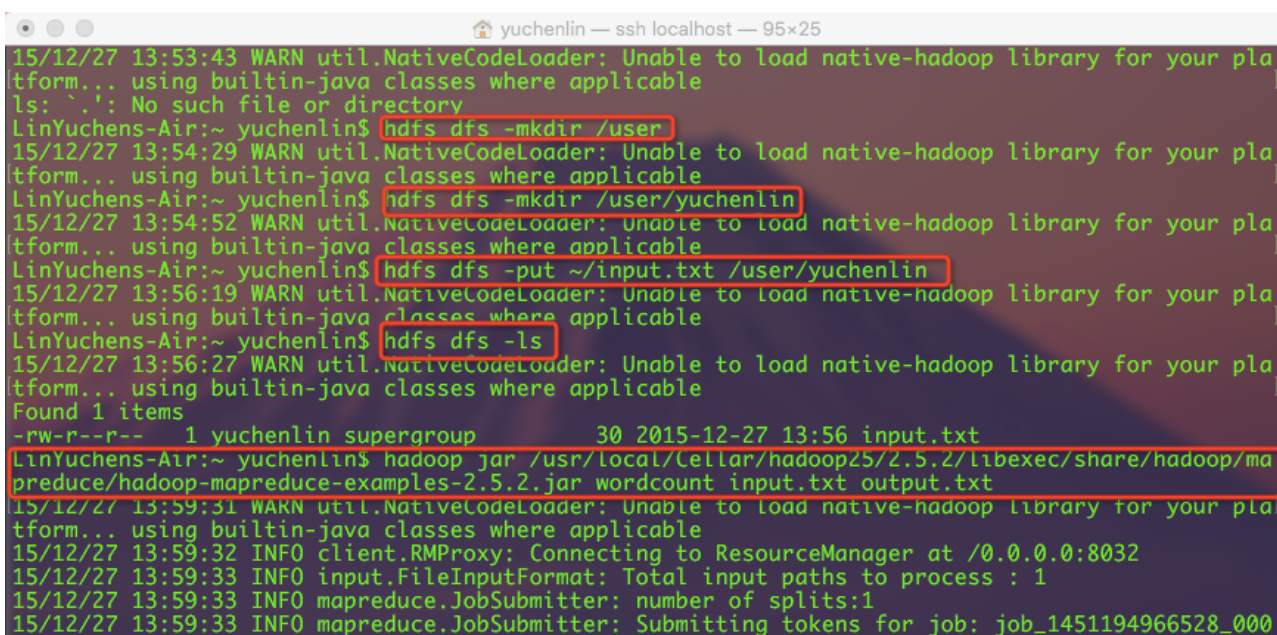
`hdfs dfs -copyFromLocal` 本地文件路径 服务器文件夹路径

这一命令，我发现我这里使用的时候并不会自动不存在的创建文件夹。而且发现 `hdfs dfs -ls` 这个命令也会报错。

通过搜索得知，`hdfs dfs ls` 这个命令只有当建立好了 `/user/yuchenlin/` 这个文件夹之后才可以自动定位到这里，否则必须指定目录。

所以我先用了 `hdfs dfs -mkdir` 来创建了文件夹，再成功的利用了 `put` 命令向服务器上传了文件。这里的 `put` 和 `copyFromLocal` 命令效果一致。区别在于 `put` 还可以通过 `stdin` 输入。

接下来就可以执行我们的 mapreduce 样例了。



```

15/12/27 13:53:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
ls: '.': No such file or directory
LinYuchens-Air:~ yuchenlin$ hdfs dfs -mkdir /user
15/12/27 13:54:29 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
LinYuchens-Air:~ yuchenlin$ hdfs dfs -mkdir /user/yuchenlin
15/12/27 13:54:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
LinYuchens-Air:~ yuchenlin$ hdfs dfs -put ~/input.txt /user/yuchenlin
15/12/27 13:56:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
LinYuchens-Air:~ yuchenlin$ hdfs dfs -ls
15/12/27 13:56:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--  1 yuchenlin supergroup          30 2015-12-27 13:56 input.txt
LinYuchens-Air:~ yuchenlin$ hadoop jar /usr/local/Cellar/hadoop25/2.5.2/libexec/share/hadoop/ma
preduce/hadoop-mapreduce-examples-2.5.2.jar wordcount input.txt output.txt
15/12/27 13:59:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
15/12/27 13:59:32 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
15/12/27 13:59:33 INFO input.FileInputFormat: Total input paths to process : 1
15/12/27 13:59:33 INFO mapreduce.JobSubmitter: number of splits:1
15/12/27 13:59:33 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1451194966528_000
  
```


对比 1 2 3 行，我们可以发现当 Sample 固定，Map 的数量增多的时候，时间增长，准确率在提高。

对比 1 4，还有 3 5 可以发现，当 Map 固定，Sample 数量增多的时候，时间稍有减少，准确率也在提高。

所以可以归纳猜测，提高 Map 和 Sample 都会增大精确性，但是提高 Map 会大量增加耗时。

所以我继续做了几次实验来提高准确性。

Number of Maps	Number of samples	Time(s)	π
2	10	23.431	3.800000
5	10	29.652	3.280000
10	10	43.79	3.200000
2	100	21.469	3.120000
10	100	38.683	3.148000
30	1000	89.672	3.141466666
30	2000	90.35	3.14200000
35	1000	101.444	3.14125714
100	10000	398.377	3.14159264920

可以发现当最后一行，Map=100，Sample=10000的时候我们可以得到非常精确的结果了。但是代价就是耗时太多了。

而且我发现了一个有趣的现象，那就是对比红色框中的第1和第2行可以发现在这个时候 Map 都是30，Sample 从1000提高到2000的时候 Time 是增多了1秒的，这个和之前猜测得到的结论稍有不同。

我觉得这个可能是每次实验耗时的计算并不是在完全相同的环境下进行造成的。

5.实验总结

在这次实验中，我有了对 Hadoop 的基本了解，具备了基本的 hadoop 使用能力，对 HDFS 和 MapReduce 有一定的了解，但是还没有进行自己编写可以运行的以 MapReduce 为模型的代码，会在以后的业余时间自行学习。

6.参考

<http://www.chinahadoop.cn/group/5/thread/32>

http://blog.sina.com.cn/s/blog_6d932f2a0101fsxn.html

<http://blog.csdn.net/xbwer/article/details/35614679>

非常感谢助教和老师的耐心指导和讲解。

林禹臣

yuchenlin@sjtu.edu.cn

5140309507

2015.12.27