在Max OSX上安装Hadoop

这学期刚好选修了大数据课程，需要使用Hadoop。由于自己使用的电脑刚好是Mac OSX,这里给出如何在Mac OSX上面安装和使用Hadoop

**安装homebrew**

习惯使用Ubuntu的同学，一定很喜欢Ubuntu系统apt-get的软件安装方式。Mac上也有类似这样的包管理器，利用Homebrew即可。  
[Homebrew的官方网站](http://brew.sh/)  
安装Homebrew的方法：

|  |
| --- |
| /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" |

需要注意的是：  
Homebrew安装的软件都集中在/usr/local/Cellar里面

**ssh登录本地**

|  |
| --- |
| ssh-keygen -t rsa -P "" cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys |

这样就可以生成ssh公钥，接下来进行测试登录本地是否成功

|  |
| --- |
| ssh localhost |

登录成功显示结果如下：

|  |
| --- |
| Last login: Mon Feb 29 18:29:55 2016 from ::1 |

**安装Hadoop**

输入以下代码，自动安装hadoop

|  |
| --- |
| brew install hadoop |

安装过程会提示重要的信息，如下：

|  |
| --- |
| $JAVA\_HOME has been set to be the output of:  /usr/libexec/java\_home |

在Mac中，我们可以终端输入：/usr/libexec/java\_home来获取JAVA\_HOME的路径  
Hadoop的安装需要配置JAVA\_HOME，用 brew安装，就已经帮我们配置好了。

**测试Hadoop是否安装成功**

Hadoop有三种安装模式：单机模式，伪分布式模式，分布式模式  
分布式模式需要在多台电脑上面测试，这里只测试前面两种，即单机模式和伪分布式模式

**测试单机模式**

这里使用Hadoop附带的示例来检验单机模式是否运行正常。  
通过Homebrew安装的Hadoop，附带的示例在路径/usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/mapreduce  
单词计数wordcount是最简单也是最能体现MapReduce思想的程序之一，可以称为MapReduce版”Hello World”，单词计数主要完成功能是：统计一系列文本文件中每个单词出现的次数.

* 创建input目录和output目录  
  input作为输入目录，output目录作为输出目录

|  |
| --- |
| cd /usr/local/Cellar/hadoop/2.7.1/ mkdir input  mkdir output |

* 在input文件夹中创建两个测试文件file1.txt和file2.txt

|  |
| --- |
| cd input echo 'hello world' > file1.txt echo 'hello hadoop' > file2.txt |

* 运行示例检测单机模式

|  |
| --- |
| hadoop jar ./libexec/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount ./input ./output |

* 查看结果

|  |
| --- |
| more output/part-r-00000 |

显示结果如下：

* hadoop 1
* hello 2
* world 1

**测试伪分布式模式**

测试为分布模式前，需要修改相关的配置文件，把之前的单机模式修改成伪分布式模式

* 修改hadoop-env.sh  
  文件地址：/usr/local/Cellar/hadoop/2.7.1/libexec/etc/hadoop/hadoop-env.sh  
  注释原来的HADOOP\_OPOTS

|  |
| --- |
| #export HADOOP\_OPTS="$HADOOP\_OPTS -Djava.net.preferIPv4Stack=true" |

* 修改为

|  |
| --- |
| export HADOOP\_OPTS="$HADOOP\_OPTS -Djava.net.preferIPv4Stack=true -Djava.security.krb5.realm= -Djava.security.krb5.kdc=" |

* 修改Core-site.xml  
  文件地址：/usr/local/Cellar/hadoop/2.7.1/libexec/etc/hadoop/core-site.xml  
  把原来的

|  |
| --- |
| <configuration> </configuration> |

* 修改为

|  |
| --- |
| <configuration>  <property>  <name>hadoop.tmp.dir</name>  <value>/usr/local/Cellar/hadoop/hdfs/tmp</value>  <description>A base for other temporary directories</description>  </property>  <property>  <name>fs.default.name</name>  <value>hdfs://localhost:9000</value>  </property> </configuration> |

* fs.default.name 保存了NameNode的位置，HDFS和MapReduce组件都需要用到它，这就是它出现在core-site.xml 文件中而不是 hdfs-site.xml文件中的原因
* 修改mapred-site.xml.template  
  文件地址：/usr/local/Cellar/hadoop/2.7.1/libexec/etc/hadoop/mapred-site.xml.template  
  把原来的

|  |
| --- |
| <configuration> </configuration> |

* 修改为

|  |
| --- |
| <configuration>  <property>  <name>mapred.job.tracker</name>  <value>localhost:9010</value>  </property> </configuration> |

* 变量mapred.job.tracker 保存了JobTracker的位置，因为只有MapReduce组件需要知道这个位置，所以它出现在mapred-site.xml文件中。
* 修改hdfs-site.xml  
  文件地址：/usr/local/Cellar/hadoop/2.7.1/libexec/etc/hadoop/hdfs-site.xml  
  把原来的

|  |
| --- |
| <configuration> </configuration> |

* 修改为

|  |
| --- |
| <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property> </configuration> |

* 变量dfs.replication指定了每个HDFS数据库的复制次数。 通常为3, 由于我们只有一台主机和一个伪分布式模式的DataNode，将此值修改为1。
* 运行

|  |
| --- |
| hadoop namenode -format ./sbin/start-all.sh |

* 运行成功后，用浏览器访问：[Hdfs的Web界面](localhost:50070" \t "_blank),可以查看相关的信息
* 用示例测试
  + 估计圆周率PI的值

|  |
| --- |
| hadoop jar ./libexec/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar pi 2 5 |

* + 运行结果如下：

|  |
| --- |
| Estimated value of Pi is 3.60000000000000000000 |

* + wordcount统计数据  
    把原来用于单机模式的测试input文件上传到hdfs中

|  |
| --- |
| hadoop fs -put ./input input |

* + 运行wordcount

|  |
| --- |
| hadoop jar ./libexec/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount input output |

* + 查看output的结果文件

|  |
| --- |
| hadoop fs -tail output/part-r-00000 |

* + 得到的结果是：
    - hadoop 1
    - hello 2
    - world 1
* 关闭伪分布式

|  |
| --- |
| ./sbin/stop-all.sh |