Move to...

[Mac] 2. How to run Auto-Completion

总结一下这节课项目是两步

第一步是 offline 建立 数据库 我们用两个map reduce 的data pipline 来实现。 第二部是 online显示把数据里面数据读出来显示。
如果实际运用中 第一步 mapreduce 一般数据一天之内更新不会太多 所以mapredcue 一天跑一次。 第二步 online 显示 like 数据大了会比较慢 。 然后可以优化系统版讲的trie树就是其中一种。

获取本机参数，使其能与MapReduce进行交互

$ ifconfig | grep inet | grep broadcast #获取本机IP， 如果有多条结果，可以理解先看这个链接https://en.wikipedia.org/wiki/Ifconfig 理解一下 ifconfig 干了什么事情

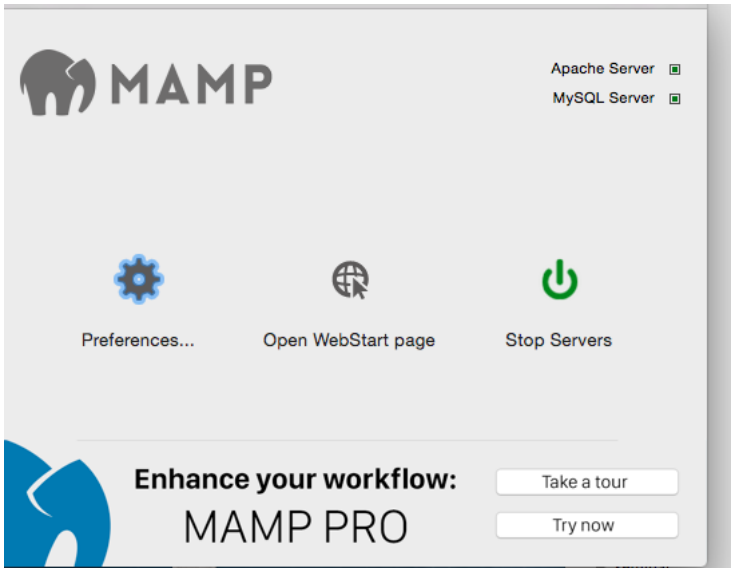
```
hadoop-cluster-docker git:(master)  ifconfig | grep inet | grep broadcast
     inet 192.168.1.5 netmask 0xffffff00 broadcast 192.168.1.255
```

2

Linux Ip 是eth0 开头的

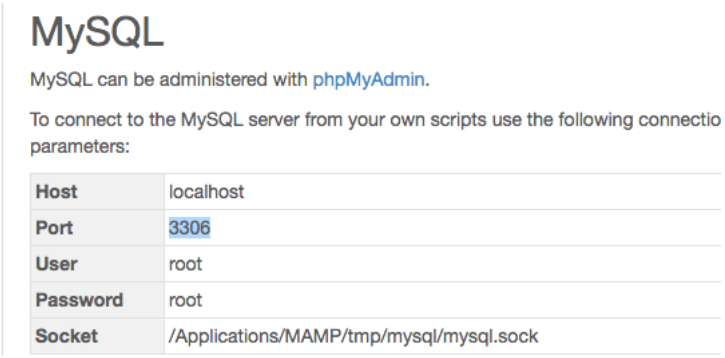
```
eth0      Link encap:Ethernet
          inet addr:10.0.2.15
```

点击Open WebStartPage, 进入XAMP



查看MySQL运行的端口

进入XAMP，查看MySQL的root用户的密码



助教的端口是3306,

助教 root 用户密码也是root

还有一种方式得到端口

就是在mysql 里面输入

$ SHOW VARIABLES WHERE Variable_name = 'port' ;

然后就可以得到端口

配置MySQL，（主要目的是为了创建table，然后可以远程的map reduce 写data）

$ 打开Terminal

$ cd /Applications/MAMP/Library/bin/ $ (mac)

ubuntu 用户在 /usr/local/ammps/sql/bin 这个路径下面

(Windows用命令行进入到这个路径下： {MAMP安装目录}\bin\mysql\bin)

$ ./mysql -uroot -p (Win的话mysql -u root -p)

(先输入机器密码，然后接下来输入刚才得到的password root, 两种密码) （ubuntu密码是mysql）（Win没有机器密码，直接输入上面http://localhost/MAMP/ 中看到的密码即可（一般是root))

```
➜ bin sudo ./mysql -uroot -p
Password:
Sorry, try again.
Password:
Enter password:
```

$ create database test; # 创建 test database

$ use test; # 切换 database (请理解这句话的意思)

$ create table output(starting_phrase VARCHAR(250), following_word VARCHAR(250), count INT);

下面这一句 your-password 替换成刚才得到的密码

$ GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '**your-password** ' WITH GRANT OPTION; //enable remote data transfer

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'password' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
```

$ FLUSH PRIVILEGES;

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

如果是ubuntu / Mac 的同学, 查找是否有 /etc/mysql/my.cnf文件， 若有，查找是否有 bind-address = 127.0.0.1 ，若有， 则用 “#”注释掉， 若没有，就不用去管它

```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address          = 127.0.0.1
#
```

配置Hadoop

打开docker

$./start-container.sh # 启动docker container

```
➜  hadoop-cluster-docker git:(master) ./start-container.sh
Password:
start hadoop-master container...
mkdir: /Users/mabodx/src/: File exists
start hadoop-slave1 container...
start hadoop-slave2 container...
root@hadoop-master:~#
```

$ ./start-hadoop.sh # start hadoop

```
root@hadoop-master:~# ./start-hadoop.sh


Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-hadoop-master.out
```

$ cd src # 进入Hadoop里面的文件夹里面

$ wget https://s3-us-west-2.amazonaws.com/jiuzhang-bigdata/mysql-connector-java-5.1.39-bin.jar # 下载mysql-connector 为了远端链接mysql

下一步在HDFS 里面操作

$ hdfs dfs -mkdir /mysql # 在hdfs 里面创建mysql的文件夹，然后把一些mysql相关的东西进去

$ hdfs dfs -put mysql-connector-java-*.jar /mysql/ #hdfs path to mysql-connector*

下载代码然后修改代码中的参数，使其能够与本机MySQL通信

$ wget https://s3-us-west-2.amazonaws.com/jiuzhang-bigdata/NGram.tar

$ tar -xzvf NGram.tar

$ cd NGram

$ hdfs dfs -mkdir -p input

$ hdfs dfs -rm -r /output

$ hdfs dfs -put bookList/* input/

$ cd src

打开在**Driver**.java文件中修改以下4个参数：

local_ip_address : 192.168.1.5

MySQL_port : 3306

your_password: root

hdfs_path_to_mysql-connector: /mysql/mysql-connector-java-5.1.39-bin.jar

DBConfiguration.configureDB(conf2,

"com.mysql.jdbc.Driver", // driver class

"jdbc:mysql://local_ip_address:MySQL_port/test", // db url

"root", // user name

“your_password”); //password

job2.addArchiveToClassPath(new Path(“hdfs_path_to_mysql-connector”));

下面是助教例子

DBConfiguration.configureDB(conf2,

"com.mysql.jdbc.Driver", // driver class

"jdbc:mysql://192.168.1.5:3306/test", // db url

"root", // user name

“root”); //password

job2.addArchiveToClassPath(new Path(“/mysql/mysql-connector-java-5.1.39-bin.jar”));

运行Auto-Complete

$ hadoop com.sun.tools.javac.Main *.java

$ jar cf ngram.jar *.class

$ hadoop jar ngram.jar Driver input /output 2 3 4

请设置3G以上内存，3 CPU 运行此代码(Windows请至少配置4G内存喂vagrant，

2表示 ngram_size

3 表示 threashold_size，当前单词出现的count<threashold 就会被省略

4 表示 following_word_size

后面三个参数可以随便设置

这个地方会run 两个map reduce， 如果你只看到一个成功了。 第二个没成功，应该是你的ip没有配置正确。

如果你去查看MySQL的database 你会发现数据已经写到了mysql里面啦

用下面语句可以查看

$ select * from output limit 10 ;

看看是否有数据存在 如果没有数据，那就说明没有写到数据里面,检查一下ip地址配对没有

```
mysql> mysql> select * from output LIMIT 10;
+-----------------+----------------+-------+
| starting_phrase | following_word | count |
+-----------------+----------------+-------+
| a               | man            |   263 |
| a               | little         |    90 |
| abash           | d              |     4 |
| abide           | and            |     7 |
| ability         | and            |     6 |
| ability         | in             |     4 |
| ability         | to             |     4 |
| able            | to             |     5 |
| about           | the            |    72 |
| about           | his            |    31 |
+-----------------+----------------+-------+
10 rows in set (0.00 sec)
```

下面是运行Auto-Complete User Interface

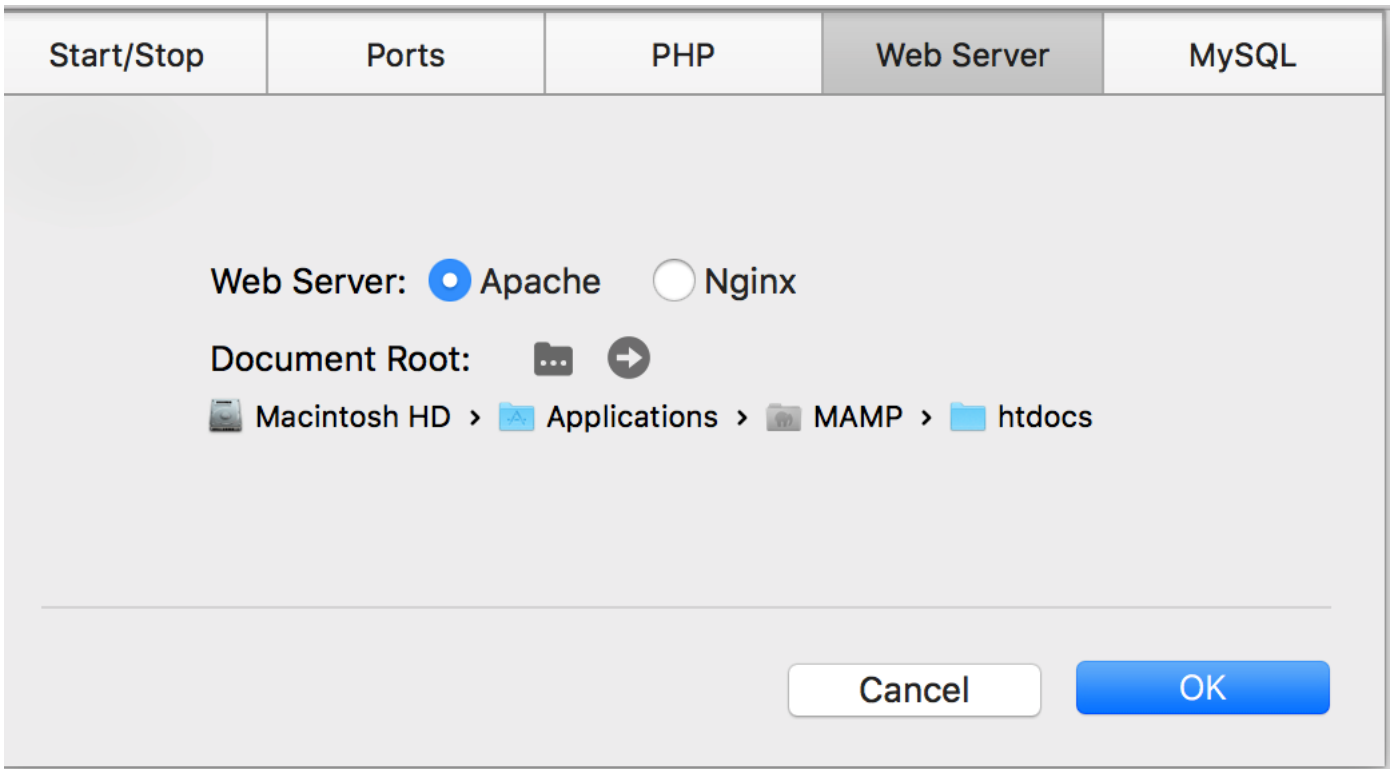
$ wget https://s3-us-west-2.amazonaws.com/jiuzhang-bigdata/autocomplete.tar

下载相关前端代码包（win直接点上面的链接用下载工具下载）

ubuntu 用户放在下面路径下面/usr/local/ammps/www

mac 用户 打开XAMP-> preference -> web server

windows上与Mac相同，然后解压他们到当前目录（tar命令或win下的各种解压软件



将Autocomplete代码包放在web server指定路径下 #如图为/htdocs 路径下

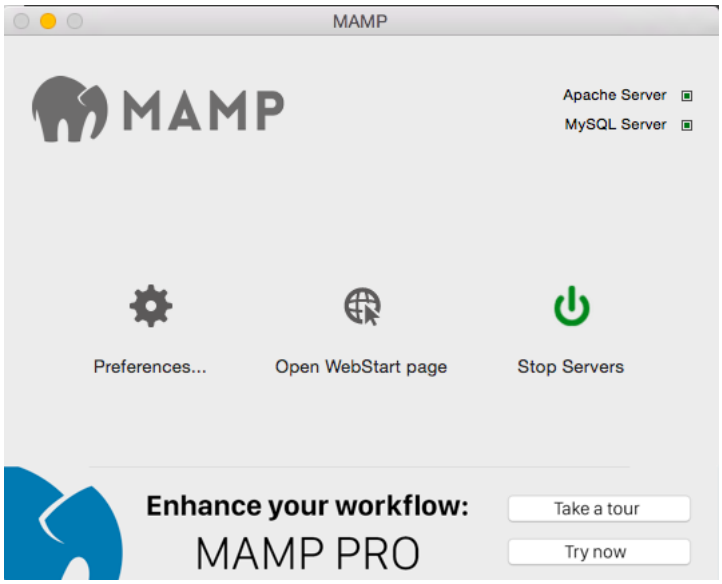
将Autocomplete/ajax_refresh.php 文件一下参数改为本机参数(注意需要指定端口号):

```php
<?php
// PDO connect *********
define ('DBUSER', 'root');
define ('DBPASS','root');
define ('DBNAME','test');

function connect() {
    return new PDO('mysql:host=localhost;dbname=test;port=8889', 'root', 'root', array(PDO::ATTR_ERRMODE => PDO::
        ERRMODE_EXCEPTION, PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
}
```

分别改为本机用户名（默认为root），本机MySQL密码

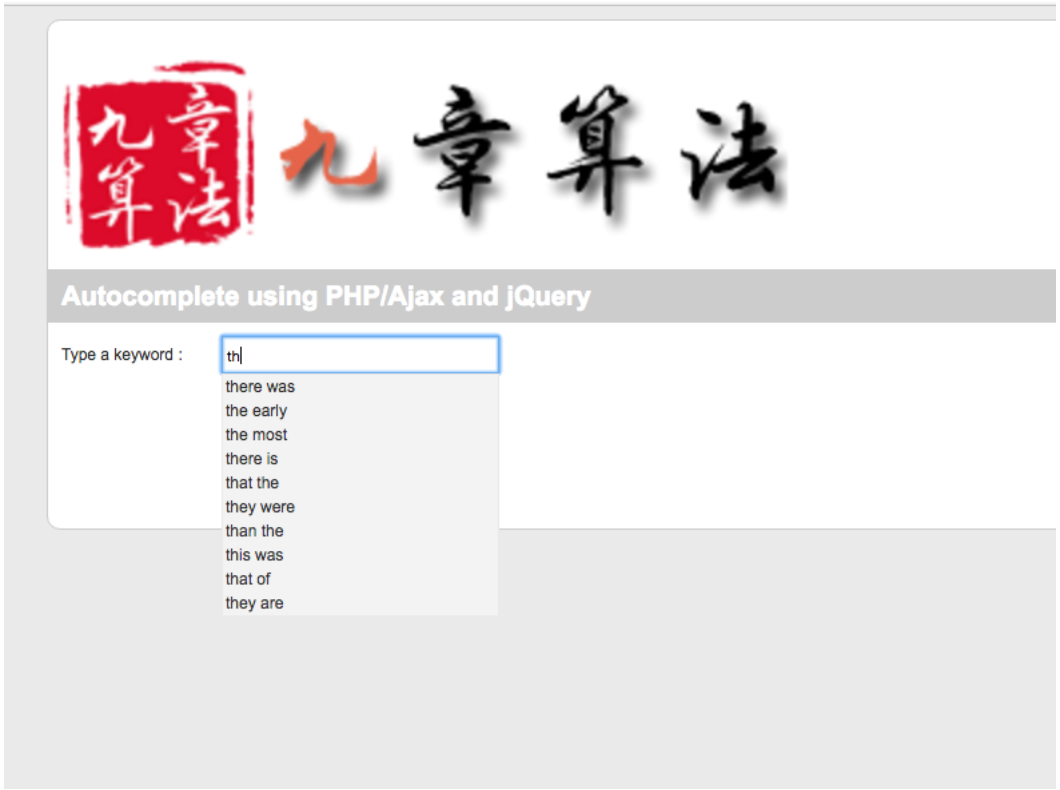
重启server 然后就可以，点击stop servers然后start servers



进入网页运行 localhost:8888(ubuntu可以尝试 localhost:80, Windows用户访问localhost), 选择autocomplete，即可进入输入界面

如果ubuntu 用户 访问 hostlocal

最后成功效果图



常见问题

1. INFO mapreduce.Job: Task Id : attempt_1474070831522_0002_r_000000_1, Status : FAILED

Error: java.io.IOException: null, message from server: "Host 'x.x.x.x' is not allowed to connect to this MySQL server"

解决方法

$ GRANT ALL ON *.* to 'root'@'id_address' IDENTIFIED BY 'your-password'; //enable remote data transfer

$ FLUSH PRIVILEGES;

http://stackoverflow.com/questions/1559955/host-xxx-xx-xxx-xxx-is-not-allowed-to-connect-to-this-mysql-server

2. Communication links failure

```
16/09/18 02:46:32 INFO mapreduce.Job: Task Id : attempt_1474142940242_0007_r_000000_2, Status : FAI
Error: java.io.IOException: Communications link failure
```

原因：IP或者Port错误 导致无法与MySQL链接

解决方法：修改Driver.java 中配置的mysql IP与 Port的参数

3. 对应Ip的问题

```
000_0, Status : FAILED
Error: java.io.IOException: Access denied for user 'root'@'172.18.0.4' (using pa
ssword: YES)
        at org.apache.hadoop.mapreduce.lib.db.DBOutputFormat.getRecordWriter(DBO
utputFormat.java:185)
```

如果遇到没有权限方位172.18.0.4 那就要把对应ip的加到root user里

$ GRANT ALL ON *.* to 'root'@'ip_address' IDENTIFIED BY 'your-password'; //enable remote data transfer

$ GRANT ALL PRIVILEGES ON *.* TO 'root'@'ip_address' WITH GRANT OPTION;

4. Mysql port 变成0的问题

如果有人再遇到mysql port 变成0的情况，那应该是因为这个同学用了Pro版的mamp. 解决办法有两个 1. 在pro版的mysql tab， 在allow network connection那里打钩， 然后选 other computer connection. 2. 下载普通版（没有这个功能）所以就会自然解决。而且pro要付费

5. 如果系统内已有MySQL运行的问题

如果系统内已有mysql运行，并且mysql client在paths 内， 会和mamp中Mysql 冲突，解决办法是，停止原系统中mysql 进程，把原系统中mysql client tool从path中移除。（在mac os 中 遇到的问题，并通过这个方法解决了）

6. Output already exists.

删掉output 就行了

```
root@hadoop-master:~/src/NGram/src# hadoop jar ngram.jar Driver input /output 2
3 4
16/10/23 23:44:48 INFO client.RMProxy: Connecting to ResourceManager at hadoop-m
aster/172.18.0.2:8032
Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException:
Output directory hdfs://hadoop-master:9000/output already exists
        at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSp
ecs(FileOutputFormat.java:146)
        at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java
:266)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitt
er.java:139)
        at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1290)
        at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1287)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:415)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInforma
tion.java:1657)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1287)
        at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1308)
        at Driver.main(Driver.java:38)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
```

212391114192122294311131 13

Sign In

Block Quote
Pull Quote