Jupyter_Notebook对接FusionInsight

安装Jupyter notebook

Jupyter notebook的安装依赖于Python，且涉及到许多工具的依赖包，相互之间还存在版本依赖关系，比较麻烦，通常可以直接安装Anaconda包，里面包含了Python、Jupyter Notebook，以及众多的科学工具包，这里我们直接安装Anaconda

- 从Anaconda官网下载并安装Anaconda2-4.4

  ```
  wget https://repo.continuum.io/archive/Anaconda2-4.4.0-Linux-x86_64.sh
  bash Anaconda2-4.4.0-Linux-x86_64.sh
  ```

- 生成Jupyter notebook的配置文件

  ```
  jupyter notebook --generate-config --allow-root
  ```

- 修改Jupyter notebook的配置IPc.NotebookApp.ip为本机IP地址

  ```
  vi /root/.jupyter/jupyter_notebook_config.py
  ```

- 启动Jupyter notebook::

  ```
  jupyter notebook --allow-root
  ```

- 出现如下提示表示Jupyter notebook启动成功

  ```
  [I 15:53:46.918 NotebookApp] Serving notebooks from local directory: /opt
  [I 15:53:46.918 NotebookApp] 0 active kernels
  [I 15:53:46.918 NotebookApp] The Jupyter Notebook is running at: http://172.21.33.122:8888/?
  token=f0494a2274cba1a6098ef21c417af2f3c49df872c6b34938
  [I 15:53:46.918 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
  [W 15:53:46.919 NotebookApp] No web browser found: could not locate runnable browser.
  [C 15:53:46.919 NotebookApp]

      Copy/paste this URL into your browser when you connect for the first time,
      to login with a token:
          http://172.21.33.122:8888/?token=f0494a2274cba1a6098ef21c417af2f3c49df872c6b34938
  ```

- 使用 `Ctrl+C` 可以退出Jupyter notebook

安装FusionInsight Client

- 参考FusionInsight的产品文档完成Linux下的FusionInsight客户端的安装，安装到 `/opt/hadoopclient` 目录

完成Kerberos认证

- 使用sparkuser进行Kerberos认证(sparkuser为FusionInsight中创建的拥有Spark访问权限的人机用户)

  ```
  cd /opt/hadoopclient/
  source bigdata_env
  kinit sparkuser
  ```

导入ipython相关环境变量

- 执行以下命令导入环境变量，或者将下面两行添加到 `/opt/hadoopclient/bigdata_env文件` ，后续source bigdata_env时可以自动将环境变量导入

  ```
  export PYSPARK_DRIVER_PYTHON="ipython"
  export PYSPARK_DRIVER_PYTHON_OPTS="notebook --allow-root"
  ```

Jupyter notebook中使用pyspark进行分析

- 执行pyspark会自动启动Jupyter notebook

```
[root@test01 opt]# pyspark
[TerminalIPythonApp] WARNING | Subcommand `ipython notebook` is deprecated and will be removed in future versions.
[TerminalIPythonApp] WARNING | You likely want to use `jupyter notebook` in the future
[I 16:24:20.802 NotebookApp] The port 8888 is already in use, trying another port.
[I 16:24:20.809 NotebookApp] Serving notebooks from local directory: /opt
[I 16:24:20.809 NotebookApp] 0 active kernels
[I 16:24:20.809 NotebookApp] The Jupyter Notebook is running at: http://172.21.33.121:8889/?
token=a951f440e47d932b1782fd97383c3dc935d468799a3c36c6
[I 16:24:20.809 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 16:24:20.810 NotebookApp] No web browser found: could not locate runnable browser.
[C 16:24:20.810 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://172.21.33.121:8889/?token=a951f440e47d932b1782fd97383c3dc935d468799a3c36c6
```

- 打开上述链接，可以进行数据分析

```
wget http://s3-us-west-2.amazonaws.com/sparkr-data/flights.csv
```

```
Sys.setenv(SPARK_HOME="/opt/hadoopclient/Spark/spark")
.libPaths(c(file.path(Sys.getenv("SPARK_HOME"), "R","lib"), .libPaths()))
library(SparkR)
library(magrittr)
sc <- sparkR.init(master = "yarn-client", sparkPackages = "com.databricks:spark-csv_2.10-1.2.0")
sqlContext <- sparkRSQL.init(sc)
flightsDF <- read.df(sqlContext, "/user/sparkuser/flights.csv", source = "com.databricks.spark.csv", header = "true")
destDF <- select(flightsDF, "dest", "cancelled")
groupBy(flightsDF, flightsDF$date) %>%
summarize(avg(flightsDF$dep_delay), avg(flightsDF$arr_delay)) -> dailyDelayDF
head(dailyDelayDF)
```

```
wget http://files.grouplens.org/datasets/movielens/ml-100k/u.user
```

```
%pylab inline
user_data = sc.textFile("ml-100k/u.user")
user_fields = user_data.map(lambda line: line.split("|"))
num_users = user_fields.map(lambda fields: fields[0]).count()
num_genders = user_fields.map(lambda fields: fields[2]).distinct().count()
num_occupations = user_fields.map(lambda fields: fields[3]).distinct().count()
num_zipcodes = user_fields.map(lambda fields: fields[4]).distinct().count()
print "Users: %d, genders: %d, occupations: %d, ZIP codes: %d" % (num_users, num_genders, num_occupations, num_zipcodes)

ages = user_fields.map(lambda x: int(x[1])).collect()
hist(ages, bins=20, color='lightblue', normed=True)
fig = matplotlib.pyplot.gcf()
fig.set_size_inches(16, 10)
```

Jupyter notebook中使用R语言进行分析

TBD