

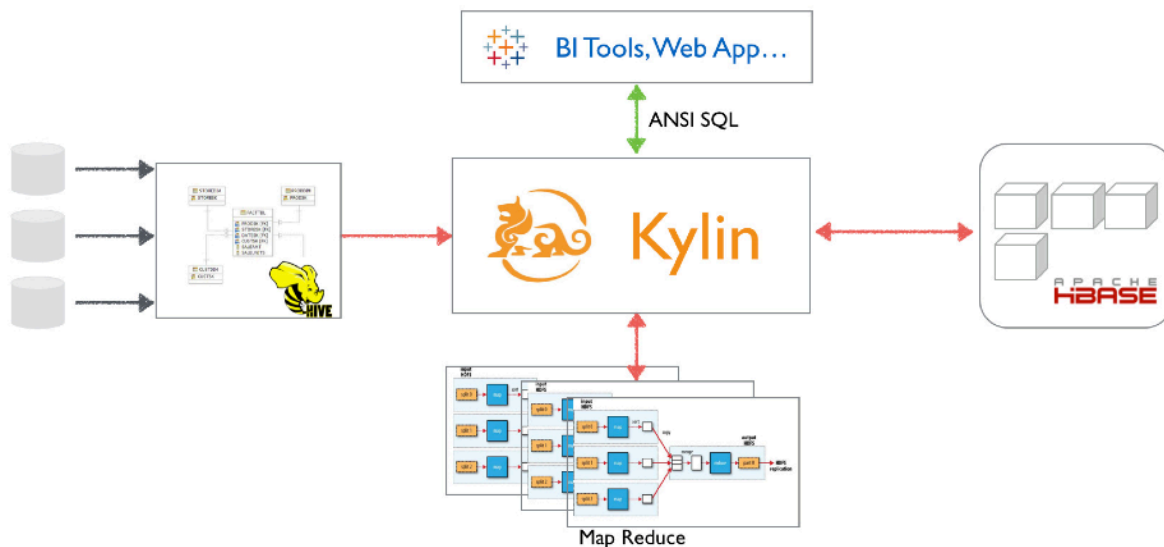
Apache-Kylin数据挖掘初步

Apache-Kylin Data Mining

计科 1402 张崇

2017年1月15日 星期日

指导老师：沈海澜



```
fkylin.py - fkylin - [~/Desktop/fkylin]

Project: fkylin
  fkylin.py
  profile
  README.md
  External Libraries

fkylin.py
65 os.system("tar -zxvf hadoop-2.6.5.tar.gz")
66 os.system("tar -zxvf hbase-1.1.8-bin.tar.gz")
67 os.system("tar -zxvf apache-hive-2.0.1-bin.tar.gz")
68 os.system("tar -zxvf apache-kylin-1.5.3-bin.tar.gz")
69 os.system("mv hadoop-2.6.5.tar.gz hadoop")
70 os.system("mv apache-hive-2.0.1-bin.tar.gz hive")
71 os.system("mv hbase-1.1.8-bin.tar.gz")
72 os.system("mv apache-kylin-1.5.3-bin.tar.gz kylin")
73
74 java_list = os.listdir("/usr/lib/jvm")
75 java_list_arr = []
76 for file in java_list:
77     java_list_arr.append(file)
78     for file in java_list_arr:
79         java_string = "java-1.8.0-openjdk-1.8.0"
80         file_name_slice = file[0:23]
81         if file_name_slice == java_list_arr:
82             java_version = file_name_slice
83
84 if linux_release_version == 1:
85     print "Starting edit environment variable in Centos"
86     os.system("wget http://mirror.bit.edu.cn/apache/kylin/apache-kylin-1.5.3/apache-kylin-1.5.3-bin.tar.gz")
87     os.system("mv /etc/profile /etc/profile.default")
88     os.system("cp profile /etc/")
89     os.system("source /etc/profile")
90
91 if linux_release_version == 2:
92     print "Starting edit environment variable in Ubuntu"
93     os.system("mv ~/.bashrc ~/.bashrc.default")
94     os.system("mv profile .bashrc")
95     os.system("mv .bashrc ~/.")
96     os.system("source ~/.bashrc")
97
98 if linux_release_version == 3:
99     print "Starting edit environment variable in Debain"
100     os.system("mv ~/.bashrc ~/.bashrc.default")
101     os.system("mv profile .bashrc")
102     os.system("mv .bashrc ~/.")
103     os.system("source ~/.bashrc")
104
105 Event Log
68:42 LF: UTF-8: Git: master ?
```

一、apache-kylin的安装配置

测试环境: Centos 6.8

shell: bash

第一次安装后, 使用python写了安装脚本: <https://github.com/rhythm1995/fkylin/blob/master/fkylin.py>。可以自动在centos与debin等发行版部署一个java+hadoop+hbase+hive+kylin的开发环境, 方便之后使用。

手动安装配置如下:

1.安装java

使用yum包管理器安装opendjk1.8版本, yum包管理器使用方便, 是readhat系linux最常用的包管理工具。

更新yum安装源yum update

查看是否已安装java: yum list installed |grep java

卸载已安装的java7: yum -y remove java-1.7.0-openjdk*

查看源中的java: yum -y list java*

安装java8: yum install java-1.8.0-openjdk-devel.x86_64

2.安装hadoop

获取hadoop (从北理工镜像站, 使用wget工具) :

wget <http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz>

解压hadoop:

tar -zxvf hadoop-2.6.5.tar.gz

3.安装hive

获取hive:

wget <http://mirror.bit.edu.cn/apache/hive/hive-2.0.1/apache-hive-2.0.1-bin.tar.gz>

解压hive:

tar -zxvf apache-hive-2.0.1-bin.tar.gz

4.安装hbase

获取hbase:

wget <http://mirror.bit.edu.cn/apache/hbase/1.1.8/hbase-1.1.8-bin.tar.gz>

解压hbase:

tar -zxvf hbase-1.1.8-bin.tar.gz

5.安装hadoop

获取hadoop: `wget http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz`

解压hadoop: `tar -zxvf hadoop-2.6.5.tar.gz`

6.配置环境变量

对主文件目录进行命名:

```
mv hadoop-2.6.5.tar.gz hadoop
mv apache-hive-2.0.1-bin.tar.gz hive
mv hbase-1.1.8-bin.tar.gz
mv apache-kylin-1.5.3-bin.tar.gz kylin
```

用vi编辑器打开环境变量文件并进入编辑模式:

```
vim /etc/profile
```

对该文件进行追加java、hadoop、hbase、hive与kylin的环境变量:

```
# Java home
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-
openjdk-1.8.0.111-2.b15.el7_3.x86_64
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

```
# hadoop home
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export HADOOP_YARN_CONF_DIR=/usr/local/hadoop/etc/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
```

```
# Hbase home
export HBASE_HOME=/usr/local/hbase
export HBASE_CONF_DIR=/usr/local/hbase/conf
export PATH=$PATH:$HBASE_HOME/bin
```

```
# hive home
export HIVE_HOME=/usr/local/hive
export HCAT_HOME=$HIVE_HOME/hcatalog
export HIVE_CONF=$HIVE_HOME/conf
export PATH=$PATH:$HIVE_HOME/bin
```

使环境变量生效:

```
source /etc/profile
```

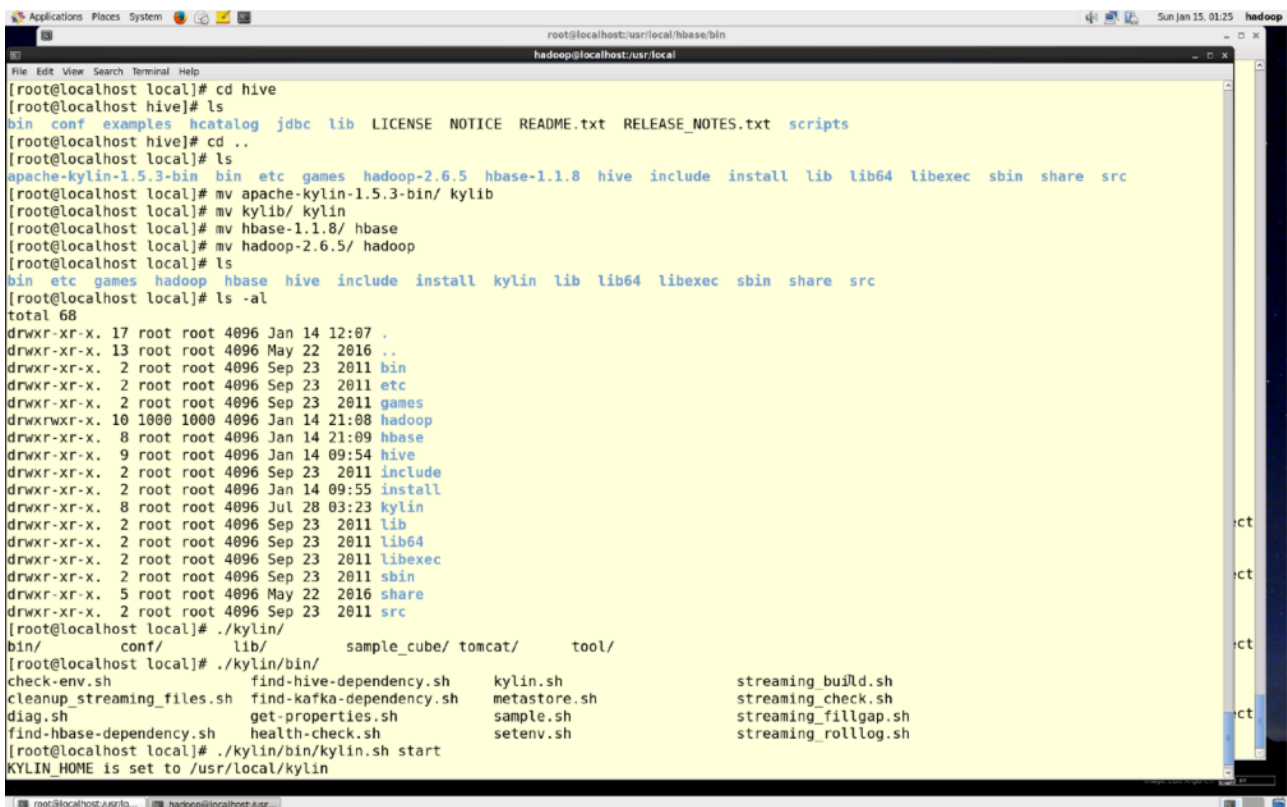
7.启动环境

启动hadoop:

```
./usr/local/hadoop/sbin/start-all.sh
```

启动hbase:
./usr/local/hbase/bin/start-hbase.sh
查看java进程确定java环境完整:
jps
配置kylin.properties :
vim /usr/local/kylin/etc/properties
kylin.rest.servers=192.168.0.222:7070
kyin.server.mode=all
检查环境:
bin/check-env.sh
启动kylin:
bin/kylin.sh start

访问localhost的7070端口即可看到kylin已经成功运行。



```
[root@localhost local]# cd hive
[root@localhost hive]# ls
bin  conf  examples  hcatalog  jdbc  lib  LICENSE  NOTICE  README.txt  RELEASE_NOTES.txt  scripts
[root@localhost hive]# cd ..
[root@localhost local]# ls
apache-kylin-1.5.3-bin  bin  etc  games  hadoop-2.6.5  hbase-1.1.8  hive  include  install  lib  lib64  libexec  sbin  share  src
[root@localhost local]# mv apache-kylin-1.5.3-bin/ kylib
[root@localhost local]# mv kylib/ kylin
[root@localhost local]# mv hbase-1.1.8/ hbase
[root@localhost local]# mv hadoop-2.6.5/ hadoop
[root@localhost local]# ls
bin  etc  games  hadoop  hbase  hive  include  install  kylin  lib  lib64  libexec  sbin  share  src
[root@localhost local]# ls -al
total 68
drwxr-xr-x. 17 root root 4096 Jan 14 12:07 .
drwxr-xr-x. 13 root root 4096 May 22 2016 ..
drwxr-xr-x.  2 root root 4096 Sep 23 2011 bin
drwxr-xr-x.  2 root root 4096 Sep 23 2011 etc
drwxr-xr-x.  2 root root 4096 Sep 23 2011 games
drwxrwxr-x. 10 1000 1000 4096 Jan 14 21:08 hadoop
drwxr-xr-x.  8 root root 4096 Jan 14 21:09 hbase
drwxr-xr-x.  9 root root 4096 Jan 14 09:54 hive
drwxr-xr-x.  2 root root 4096 Sep 23 2011 include
drwxr-xr-x.  2 root root 4096 Jan 14 09:55 install
drwxr-xr-x.  8 root root 4096 Jul 28 03:23 kylin
drwxr-xr-x.  2 root root 4096 Sep 23 2011 lib
drwxr-xr-x.  2 root root 4096 Sep 23 2011 lib64
drwxr-xr-x.  2 root root 4096 Sep 23 2011 libexec
drwxr-xr-x.  2 root root 4096 Sep 23 2011 sbin
drwxr-xr-x.  5 root root 4096 May 22 2016 share
drwxr-xr-x.  2 root root 4096 Sep 23 2011 src
[root@localhost local]# ./kylin/
bin/  conf/  lib/  sample_cube/  tomcat/  tool/
[root@localhost local]# ./kylin/bin/
check-env.sh  find-hive-dependency.sh  kylin.sh  streaming_build.sh
cleanup_streaming_files.sh  find-kafka-dependency.sh  metastore.sh  streaming_check.sh
diag.sh  get-properties.sh  sample.sh  streaming_fillgap.sh
find-hbase-dependency.sh  health-check.sh  setenv.sh  streaming_rollback.sh
[root@localhost local]# ./kylin/bin/kylin.sh start
KYLIN_HOME is set to /usr/local/kylin
```

二、关于安装脚本实现

本来考虑使用shell实现，因为shell编写的程序是批处理，执行最快。但中途解决一些文件字符串数组等操作的时候shell很捉急（主要是我本来就不怎么会shell编程），然后放弃，最后决定用python写，虽然python也不怎么会的（我之前一直写JavaScript的前端和node.js的后端，基本只熟悉JavaScript语言），但考虑到所有linux发行版自带python2的解释器可以省去装其他脚本的runtime的问题，外加python还是比较简单所以用python

Kylin

Query

Cubes

Jobs

Tables

Admin

Help

Welcome, ADMIN

Source Tables

EDW

+

TEST_CAL_DT

TEST_CATEGORY_GROUPINGS

TEST_KYLIN_FACT

TEST_SELLER_TYPE_DIM

TEST_SITES

Table Schema: TEST_KYLIN_FACT

Columns

Extend Information

Columns

Q Filter ...

ID ^	Name ↕	Data Type ↕	Cardinality ↕
1	TRANS_ID	long	
2	CAL_DT	date	
3	LSTG_FORMAT_NAME	string	
4	LEAF_CATEG_ID	int	
5	LSTG_SITE_ID	int	
6	SLR_SEGMENT_CD	short	
7	PRICE	decimal(38,16)	
8	ITEM_COUNT	long	
9	SELLER_ID	long	

Home Page

Google Group

实现了一个自动部署。目前只支持少量发行版，并且只是单节点部署且无配置元数据库，之后有时间我回完善一下。

主要原理就是利用os包调用系统shell命令，以及对文件的读写操作实现，模块概述如下：

1. 获取用户使用的linux发行版

因为每个发行版有部分命令及配置文件不同，所以必须针对每种发行版提供特定的方式。读取系统/etc/issue文件，该文件描述了linux发行版信息，对其就行切片比对名称即可得到准确的发行版版本。

```

etc_issue = open('/etc/issue')
linux_release_etc = etc_issue.readline()
linux_release_key = linux_release_etc[1]
if linux_release_key == 'C':
    linux_release_version = 1
elif linux_release_key == 'U':
    linux_release_version = 2
elif linux_release_key == 'D':
    linux_release_version = 3

```

2. 下载相关源文件

可以通过调用系统命令的wget工具

```

os.system("cd /usr/local")
os.system("wget http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz")

```

```
os.system("wget http://mirror.bit.edu.cn/apache/hbase/1.1.8/
hbase-1.1.8-bin.tar.gz")
os.system("wget http://mirror.bit.edu.cn/apache/hive/hive-2.0.1/
apache-hive-2.0.1-bin.tar.gz")
```

3.调用tar命令解压源文件

```
os.system("tar -zxvf hadoop-2.6.5.tar.gz")
os.system("tar -zxvf hbase-1.1.8-bin.tar.gz")
os.system("tar -zxvf apache-hive-2.0.1-bin.tar.gz")
os.system("apache-kylin-1.5.3-bin.tar.gz")
os.system("mv hadoop-2.6.5.tar.gz hadoop")
os.system("mv apache-hive-2.0.1-bin.tar.gz hive")
os.system("mv hbase-1.1.8-bin.tar.gz")
os.system("mv apache-kylin-1.5.3-bin.tar.gz kylin")
```

4.获取系统java路径，方便配置环境，遍历/usr/lib/jvm下文件存在数组，对其满足java-1.8的版本进行查找

```
java_list = os.listdir("/usr/lib/jvm")
java_list_arr = []
for file in java_list:
    java_list_arr.append(file)
for file in java_list_arr:
    java_string = "java-1.8.0-openjdk-1.8.0"
    file_name_slice = file[0:23]
    if file_name_slice == java_list_arr:
        java_version = file_name_slice
```

5.修改配置环境并运行

```
print "Starting edit environment variable in Ubuntu"
os.system("mv ~/.bashrc ~/.bashrc.default")
os.system("mv profile .bashrc")
os.system("mv .bashrc ~/.")
os.system("source ~/.bashrc")
```

三、对数据挖掘领域的了解

1.数据挖掘需要掌握的知识：

- (1) 以机器学习为主的数据挖掘算法
- (2) java与python为主的数据挖掘常用语言，特别是python的几个数据分析的包的掌握
- (3) hadoop、hbase、hive、kylin、spark、mysql等几个大数据工具、数据库、数据挖掘工具
- (4) 数据结构基础
- (5) linux与shell的使用

2. 我目前的程度

- (1) 了解python, 可以用python进行基本开发
- (2) 熟悉mysql, mongodb等数据库
- (3) 熟悉linux及linux下的开发, 了解shell编程
- (4) 了解一些机器学习算法, 但开发商并不专业
- (5) 对数据可视化比较专业, 熟悉JavaScript

3. 需要学习的内容

- (1) 对基础算法与机器学习算法需要有一定实践, 这是数据挖掘的内功
- (2) 掌握一套数据挖掘框架的工具链
- (3) 熟悉java编程

四、后记

关于报告中未写kylin基础知识, 是因为看到官网文档里已经讲得比较全, 配图解释了架构, 通过demo可以分析文本了解处理过程。

自己在机器学习与java开发上面基本不会, 虽然上过课但就真正从事工程来说还差的比较远。我之前研究和学习的深入的方向是偏全栈的web前端开发, 对web开发方向研究更多, 除上课外从未在java和机器学习上专门研究过, 因为超过四年linux使用经验并且一直运维着几台云服务器, 所以对linux下开发部署运维有初步的理解。如果之后有时间会尝试学习下java与数据挖掘框架的入门, 计划完善下自动部署脚本, 添加多节点集群功能, 有可能的话搞下基于在docker容器下的数据挖掘环境。

感谢沈老师!