日志分析实战之清洗、统计网站信息小教程

作者：pig2
时间：20171017
出处：about 云

目录

1.使用 spark&Scala 分析 Apache 日志

问题导读

1.如何进入 spark shell?
2.spark shell 中如何加载外部文件?
3.spark 中读取文件后做了哪些操作?

about 云日志分析，那么过滤清洗日志。该如何实现。这里参考国外的一篇文章，总结分享给大家。
使用 spark 分析网站访问日志，日志文件包含数十亿行。现在开始研究 spark 使用，他是如何工作的。几年前使用 hadoop，后来发现 spark 也是容易的。
下面是需要注意的：

如果你已经知道如何使用 spark 并想知道如何处理 spark 访问日志记录，我写了这篇短的文章，介绍如何从 Apache 访问日志文件中生成 URL 点击率的排序

安装

spark 安装需要安装 hadoop，并且二者版本要合适。安装可参考下面文章

about 云日志分析项目准备 6：Hadoop、Spark 集群搭建

http://www.aboutyun.com/forum.php?mod=viewthread&tid=20620

进入

[Bash shell] 纯文本查看 复制代码

?

```
./bin/spark-shell
```

可能会出错

[Bash shell] 纯文本查看 复制代码

?

```
java.io.FileNotFoundException: File
file:/data/spark_data/history/event-log does not exist
```

解决办法：

[Bash shell] 纯文本查看 复制代码

?

```
mkdir -p /data/spark_data/history/event-log
```

详细错误如下

[Bash shell] 纯文本查看 复制代码

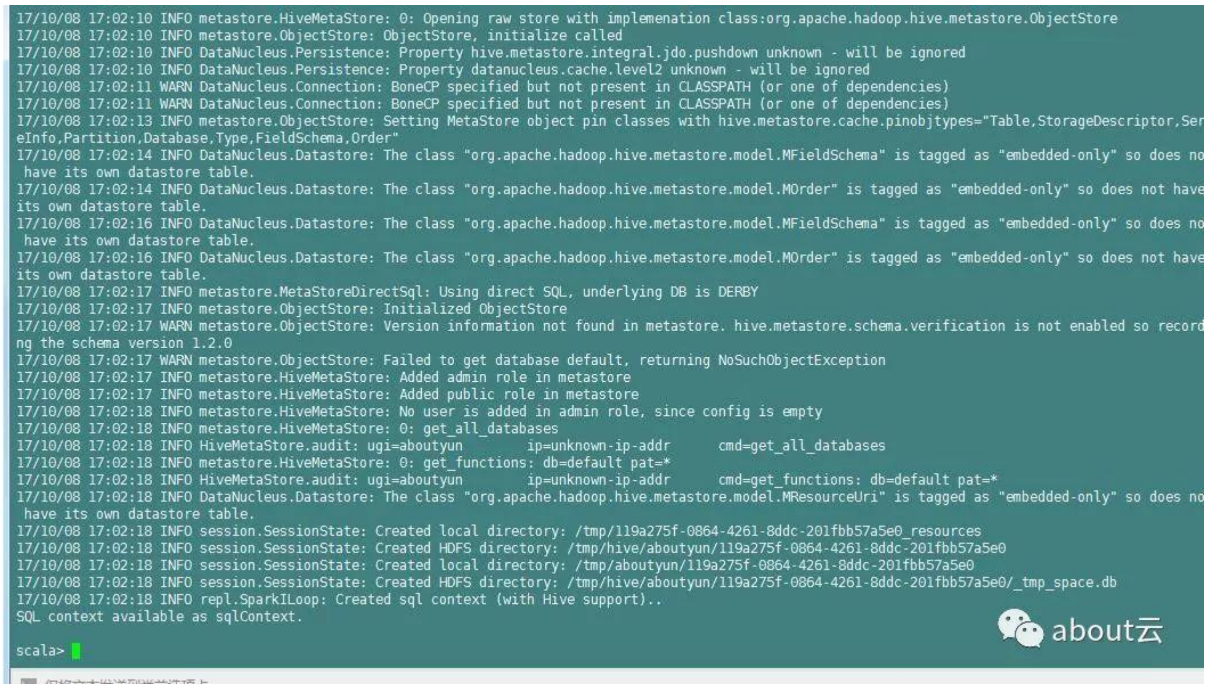
?

```
17/10/08 17:00:23 INFO client.AppClient$ClientEndpoint: Executor updated: app-201
17/10/08 17:00:25 ERROR spark.SparkContext: Error initializing SparkContext.
java.io.FileNotFoundException: File file:/data/spark_data/history/event-log does
            at org.apache.hadoop.fs.RawLocalFileSystem.deprecatedGetFileStatu
            at org.apache.hadoop.fs.RawLocalFileSystem.getFileLinkStatusInter
            at org.apache.hadoop.fs.RawLocalFileSystem.getFileStatus(RawLocal
            at org.apache.hadoop.fs.FilterFileSystem.getFileStatus(FilterFile
            at org.apache.spark.scheduler.EventLoggingListener.start(EventLog
            at org.apache.spark.SparkContext.<init>(SparkContext.scala:549)
            at org.apache.spark.repl.SparkILoop.createSparkContext(SparkILoop
            at $line3.$read$$iwC$$iwC.<init>(<console>:15)
            at $line3.$read$$iwC.<init>(<console>:24)
            at $line3.$read.<init>(<console>:26)
            at $line3.$read$.<init>(<console>:30)
            at $line3.$read$.<clinit>(<console>)
            at $line3.$eval$.<init>(<console>:7)
            at $line3.$eval$.<clinit>(<console>)
```

```
                at $line3.$eval.$print(<console>)
                at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
                at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess
                at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth
                at java.lang.reflect.Method.invoke(Method.java:497)
                at org.apache.spark.repl.SparkIMain$ReadEvalPrint.call(SparkIMair
                at org.apache.spark.repl.SparkIMain$Request.loadAndRun(SparkIMair
                at org.apache.spark.repl.SparkIMain.loadAndRunReq$1(SparkIMain.sc
                at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:87
                at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:81
                at org.apache.spark.repl.SparkILoop.reallyInterpret$1(SparkILoop.
                at org.apache.spark.repl.SparkILoop.interpretStartingWith(SparkIL
                at org.apache.spark.repl.SparkILoop.command(SparkILoop.scala:814)
                at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$
                at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$
                at org.apache.spark.repl.SparkIMain.beQuietDuring(SparkIMain.scal
                at org.apache.spark.repl.SparkILoopInit$class.initializeSpark(Spa
                at org.apache.spark.repl.SparkILoop.initializeSpark(SparkILoop.sc
                at
org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$proce
                at org.apache.spark.repl.SparkILoopInit$class.runThunks(SparkILoo
                at org.apache.spark.repl.SparkILoop.runThunks(SparkILoop.scala:64
                at org.apache.spark.repl.SparkILoopInit$class.postInitialization(
                at org.apache.spark.repl.SparkILoop.postInitialization(SparkILoop
                at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
                at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
                at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
                at scala.tools.nsc.util.ScalaClassLoader$.savingContextLoader(Sca
                at org.apache.spark.repl.SparkILoop.org$apache$spark$repl$SparkIL
                at org.apache.spark.repl.SparkILoop.process(SparkILoop.scala:1059
                at org.apache.spark.repl.Main$.main(Main.scala:31)
                at org.apache.spark.repl.Main.main(Main.scala)
                at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
                at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess
                at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth
                at java.lang.reflect.Method.invoke(Method.java:497)
                at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$S
                at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.s
                at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:
                at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:12
                at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

                at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

```

```
java.lang.NullPointerException
                at org.apache.spark.sql.SQLContext$.createListenerAndUI(SQLContex
                at org.apache.spark.sql.hive.HiveContext.<init>(HiveContext.scala
                at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native 
                at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeCo
                at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Dele
                at java.lang.reflect.Constructor.newInstance(Constructor.java:422
                at org.apache.spark.repl.SparkILoop.createSQLContext(SparkILoop.s
                at $iwC$$iwC.<init>(<console>:15)
                at $iwC.<init>(<console>:24)
                at <init>(<console>:26)
                at .<init>(<console>:30)
                at .<clinit>(<console>)
                at .<init>(<console>:7)
                at .<clinit>(<console>)
                at $print(<console>)
                at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
                at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess
                at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth
                at java.lang.reflect.Method.invoke(Method.java:497)
                at org.apache.spark.repl.SparkIMain$ReadEvalPrint.call(SparkIMain
                at org.apache.spark.repl.SparkIMain$Request.loadAndRun(SparkIMain
                at org.apache.spark.repl.SparkIMain.loadAndRunReq$1(SparkIMain.sc
                at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:87
                at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:81
                at org.apache.spark.repl.SparkILoop.reallyInterpret$1(SparkILoop.
                at org.apache.spark.repl.SparkILoop.interpretStartingWith(SparkIL
                at org.apache.spark.repl.SparkILoop.command(SparkILoop.scala:814)
                at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$
                at org.apache.spark.repl.SparkILoopInit$$anonfun$initializeSpark$
                at org.apache.spark.repl.SparkIMain.beQuietDuring(SparkIMain.scal
                at org.apache.spark.repl.SparkILoopInit$class.initializeSpark(Spa
                at org.apache.spark.repl.SparkILoop.initializeSpark(SparkILoop.sc
                at
org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$proce
                at org.apache.spark.repl.SparkILoopInit$class.runThunks(SparkILoo
                at org.apache.spark.repl.SparkILoop.runThunks(SparkILoop.scala:64
                at org.apache.spark.repl.SparkILoopInit$class.postInitialization
                at org.apache.spark.repl.SparkILoop.postInitialization(SparkILoop
                at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
                at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
                at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
                at scala.tools.nsc.util.ScalaClassLoader$.savingContextLoader(Sca
                at org.apache.spark.repl.SparkILoop.org$apache$spark$repl$SparkIL
                at org.apache.spark.repl.SparkILoop.process(SparkILoop.scala:1059
                at org.apache.spark.repl.Main$.main(Main.scala:31)
                at org.apache.spark.repl.Main.main(Main.scala)
                at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
                at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess
                at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth
                at java.lang.reflect.Method.invoke(Method.java:497)
                at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$S
                at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.s
                at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala
                at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:12
                at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)

<console>:16: error: not found: value sqlContext
                  import sqlContext.implicits._
                         ^
<console>:16: error: not found: value sqlContext
                  import sqlContext.sql
                         ^
```

进入 spark shell



```
17/10/08 17:02:10 INFO metastore.HiveMetaStore: 0: Opening raw store with implemenation class:org.apache.hadoop.hive.metastore.ObjectStore
17/10/08 17:02:10 INFO metastore.ObjectStore: ObjectStore, initialize called
17/10/08 17:02:10 INFO DataNucleus.Persistence: Property hive.metastore.integral.jdo.pushdown unknown - will be ignored
17/10/08 17:02:10 INFO DataNucleus.Persistence: Property datanucleus.cache.level2 unknown - will be ignored
17/10/08 17:02:11 WARN DataNucleus.Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/10/08 17:02:11 WARN DataNucleus.Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/10/08 17:02:13 INFO metastore.ObjectStore: Setting MetaStore object pin classes with hive.metastore.cache.pinobjtypes="Table,StorageDescriptor,Ser
eInfo,Partition,Database,Type,FieldSchema,Order"
17/10/08 17:02:14 INFO DataNucleus.Datastore: The class "org.apache.hadoop.hive.metastore.model.MFieldSchema" is tagged as "embedded-only" so does no
 have its own datastore table.
17/10/08 17:02:14 INFO DataNucleus.Datastore: The class "org.apache.hadoop.hive.metastore.model.MOrder" is tagged as "embedded-only" so does not have
its own datastore table.
17/10/08 17:02:16 INFO DataNucleus.Datastore: The class "org.apache.hadoop.hive.metastore.model.MFieldSchema" is tagged as "embedded-only" so does no
 have its own datastore table.
17/10/08 17:02:16 INFO DataNucleus.Datastore: The class "org.apache.hadoop.hive.metastore.model.MOrder" is tagged as "embedded-only" so does not have
its own datastore table.
17/10/08 17:02:17 INFO metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is DERBY
17/10/08 17:02:17 INFO metastore.ObjectStore: Initialized ObjectStore
17/10/08 17:02:17 WARN metastore.ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so record
ng the schema version 1.2.0
17/10/08 17:02:17 WARN metastore.ObjectStore: Failed to get database default, returning NoSuchObjectException
17/10/08 17:02:17 INFO metastore.HiveMetaStore: Added admin role in metastore
17/10/08 17:02:17 INFO metastore.HiveMetaStore: Added public role in metastore
17/10/08 17:02:18 INFO metastore.HiveMetaStore: No user is added in admin role, since config is empty
17/10/08 17:02:18 INFO metastore.HiveMetaStore: 0: get_all_databases
17/10/08 17:02:18 INFO HiveMetaStore.audit: ugi=aboutyun        ip=unknown-ip-addr      cmd=get_all_databases
17/10/08 17:02:18 INFO metastore.HiveMetaStore: 0: get_functions: db=default pat=*
17/10/08 17:02:18 INFO HiveMetaStore.audit: ugi=aboutyun        ip=unknown-ip-addr      cmd=get_functions: db=default pat=*
17/10/08 17:02:18 INFO DataNucleus.Datastore: The class "org.apache.hadoop.hive.metastore.model.MResourceUri" is tagged as "embedded-only" so does no
 have its own datastore table.
17/10/08 17:02:18 INFO session.SessionState: Created local directory: /tmp/119a275f-0864-4261-8ddc-201fbb57a5e0_resources
17/10/08 17:02:18 INFO session.SessionState: Created HDFS directory: /tmp/hive/aboutyun/119a275f-0864-4261-8ddc-201fbb57a5e0
17/10/08 17:02:18 INFO session.SessionState: Created local directory: /tmp/aboutyun/119a275f-0864-4261-8ddc-201fbb57a5e0
17/10/08 17:02:18 INFO session.SessionState: Created HDFS directory: /tmp/hive/aboutyun/119a275f-0864-4261-8ddc-201fbb57a5e0/_tmp_space.db
17/10/08 17:02:18 INFO repl.SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala>
```

[Bash shell] 纯文本查看 复制代码

?

```
val textFile=sc.textFile("file:///data/spark/README.md")
```

说明：

记得这里如果自己创建的文件可能会读取不到。报错如下

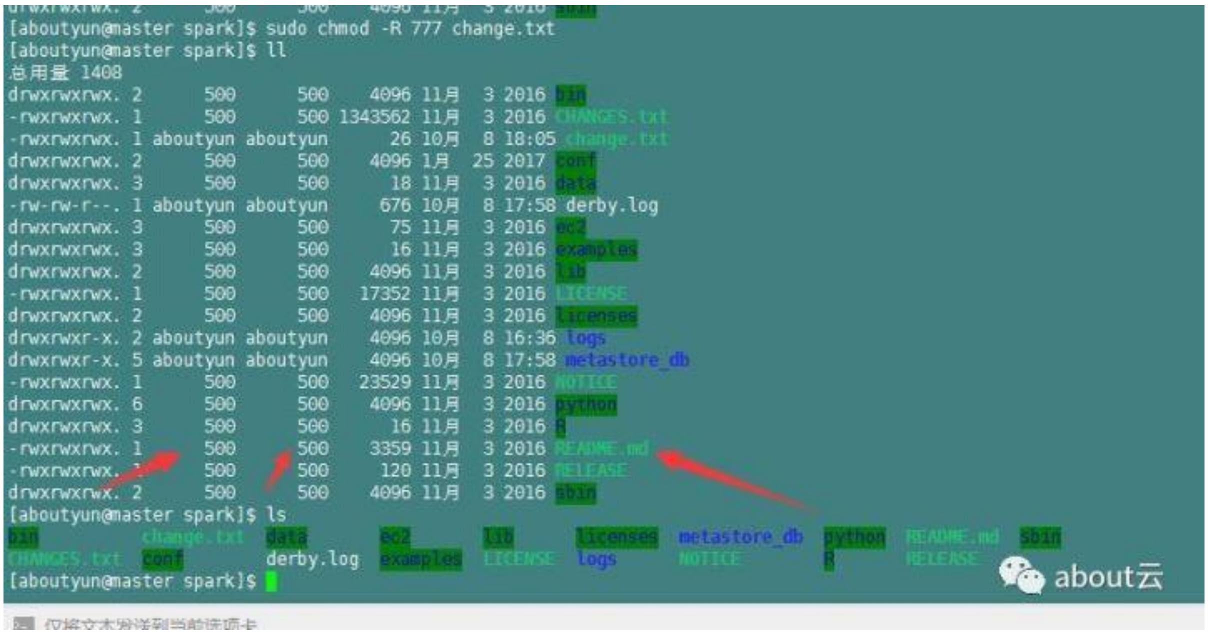
[Bash shell] 纯文本查看 复制代码

?

```
java.io.FileNotFoundException: File file:/data/spark/change.txt does not exist
                at org.apache.hadoop.fs.RawLocalFileSystem.deprecatedGetFileStatu
                at org.apache.hadoop.fs.RawLocalFileSystem.getFileLinkStatusInter
                at org.apache.hadoop.fs.RawLocalFileSystem.getFileStatus(RawLocal
                at org.apache.hadoop.fs.FilterFileSystem.getFileStatus(FilterFile
                at org.apache.hadoop.fs.ChecksumFileSystem$ChecksumFSInputChecker
                at org.apache.hadoop.fs.ChecksumFileSystem.open(ChecksumFileSyste
                at org.apache.hadoop.fs.FileSystem.open(FileSystem.java:766)
                at org.apache.hadoop.mapred.LineRecordReader.<init>(LineRecordRea
                at org.apache.hadoop.mapred.TextInputFormat.getRecordReader(TextI
                at org.apache.spark.rdd.HadoopRDD$$anon$1.<init>(HadoopRDD.scala:
                at org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:211)
                at org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:101)
                at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306
                at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)
                at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD
                at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306
                at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)
                at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala
                at org.apache.spark.scheduler.Task.run(Task.scala:89)
                at org.apache.spark.executor.Executor$TaskRunner.run(Executor.sca
                at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolEx
                at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolE
                at java.lang.Thread.run(Thread.java:745)

Driver stacktrace:
                at
org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$
                at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.
                at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.
                at scala.collection.mutable.ResizableArray$class.foreach(Resizabl
                at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala
                at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGSchedule
                at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSet
                at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSet
                at scala.Option.foreach(Option.scala:236)
```

```
        at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DA
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnRe
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onRece
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onRece
        at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:48
        at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.sc
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:1832)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:1845)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:1858)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:1929)
        at org.apache.spark.rdd.RDD.count(RDD.scala:1157)
        at $iwC$$iwC$$iwC$$iwC$$iwC$$iwC$$iwC$$iwC.<init>(<console>:30)
        at $iwC$$iwC$$iwC$$iwC$$iwC$$iwC$$iwC.<init>(<console>:35)
        at $iwC$$iwC$$iwC$$iwC$$iwC$$iwC.<init>(<console>:37)
        at $iwC$$iwC$$iwC$$iwC$$iwC.<init>(<console>:39)
        at $iwC$$iwC$$iwC$$iwC.<init>(<console>:41)
        at $iwC$$iwC$$iwC.<init>(<console>:43)
        at $iwC$$iwC.<init>(<console>:45)
        at $iwC.<init>(<console>:47)
        at <init>(<console>:49)
        at .<init>(<console>:53)
        at .<clinit>(<console>)
        at .<init>(<console>:7)
        at .<clinit>(<console>)
        at $print(<console>)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.apache.spark.repl.SparkIMain$ReadEvalPrint.call(SparkIMain
        at org.apache.spark.repl.SparkIMain$Request.loadAndRun(SparkIMain
        at org.apache.spark.repl.SparkIMain.loadAndRunReq$1(SparkIMain.sc
        at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:87
        at org.apache.spark.repl.SparkIMain.interpret(SparkIMain.scala:81
        at org.apache.spark.repl.SparkILoop.reallyInterpret$1(SparkILoop.
        at org.apache.spark.repl.SparkILoop.interpretStartingWith(SparkIL
        at org.apache.spark.repl.SparkILoop.command(SparkILoop.scala:814)
        at org.apache.spark.repl.SparkILoop.processLine$1(SparkILoop.scal
        at org.apache.spark.repl.SparkILoop.innerLoop$1(SparkILoop.scala:
        at org.apache.spark.repl.SparkILoop.org$apache$spark$repl$SparkIL
        at
org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$repl$SparkILoop$$proce
        at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
        at org.apache.spark.repl.SparkILoop$$anonfun$org$apache$spark$rep
        at scala.tools.nsc.util.ScalaClassLoader$.savingContextLoader(Sca
        at org.apache.spark.repl.SparkILoop.org$apache$spark$repl$SparkIL
        at org.apache.spark.repl.SparkILoop.process(SparkILoop.scala:1059
        at org.apache.spark.repl.Main$.main(Main.scala:31)
        at org.apache.spark.repl.Main.main(Main.scala)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$S
        at org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.s
        at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:
        at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:12
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
Caused by: java.io.FileNotFoundException: File file:/data/spark/change.txt does n
        at org.apache.hadoop.fs.RawLocalFileSystem.deprecatedGetFileStatu
        at org.apache.hadoop.fs.RawLocalFileSystem.getFileLinkStatusInter
        at org.apache.hadoop.fs.RawLocalFileSystem.getFileStatus(RawLocal
        at org.apache.hadoop.fs.FilterFileSystem.getFileStatus(FilterFile
        at org.apache.hadoop.fs.ChecksumFileSystem$ChecksumFSInputChecker
        at org.apache.hadoop.fs.ChecksumFileSystem.open(ChecksumFileSyste
        at org.apache.hadoop.fs.FileSystem.open(FileSystem.java:766)
        at org.apache.hadoop.mapred.LineRecordReader.<init>(LineRecordRea
        at org.apache.hadoop.mapred.TextInputFormat.getRecordReader(TextI
        at org.apache.spark.rdd.HadoopRDD$$anon$1.<init>(HadoopRDD.scala:
        at org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:211)
        at org.apache.spark.rdd.HadoopRDD.compute(HadoopRDD.scala:101)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:306
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:270)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala
        at org.apache.spark.scheduler.Task.run(Task.scala:89)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.sca
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolEx
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolE
        at java.lang.Thread.run(Thread.java:745)
```

需要是文件权限为 500,才可以读取到。



```
drwxrwxrwx. 2       500       500     4096 11月   3 2016 sbin
[aboutyun@master spark]$ sudo chmod -R 777 change.txt
[aboutyun@master spark]$ ll
总用量 1408
drwxrwxrwx. 2       500       500     4096 11月   3 2016 bin
-rwxrwxrwx. 1       500       500  1343562 11月   3 2016 CHANGES.txt
-rwxrwxrwx. 1 aboutyun aboutyun       26 10月   8 18:05 change.txt
drwxrwxrwx. 2       500       500     4096 1月   25 2017 conf
drwxrwxrwx. 3       500       500       18 11月   3 2016 data
-rw-rw-r--. 1 aboutyun aboutyun      676 10月   8 17:58 derby.log
drwxrwxrwx. 3       500       500       75 11月   3 2016 ec2
drwxrwxrwx. 3       500       500       16 11月   3 2016 examples
drwxrwxrwx. 2       500       500     4096 11月   3 2016 lib
-rwxrwxrwx. 1       500       500    17352 11月   3 2016 LICENSE
drwxrwxrwx. 2       500       500     4096 11月   3 2016 licenses
drwxrwxr-x. 2 aboutyun aboutyun     4096 10月   8 16:36 logs
drwxrwxr-x. 5 aboutyun aboutyun     4096 10月   8 17:58 metastore_db
-rwxrwxrwx. 1       500       500    23529 11月   3 2016 NOTICE
drwxrwxrwx. 6       500       500     4096 11月   3 2016 python
drwxrwxrwx. 3       500       500       16 11月   3 2016 R
-rwxrwxrwx. 1       500       500     3359 11月   3 2016 README.md
-rwxrwxrwx. 1       500       500      120 11月   3 2016 RELEASE
drwxrwxrwx. 2       500       500     4096 11月   3 2016 sbin
[aboutyun@master spark]$ ls
bin          change.txt  data       ec2       lib      licenses  metastore_db  python  README.md  sbin
CHANGES.txt  conf        derby.log  examples  LICENSE  logs      NOTICE        R       RELEASE
[aboutyun@master spark]$
```

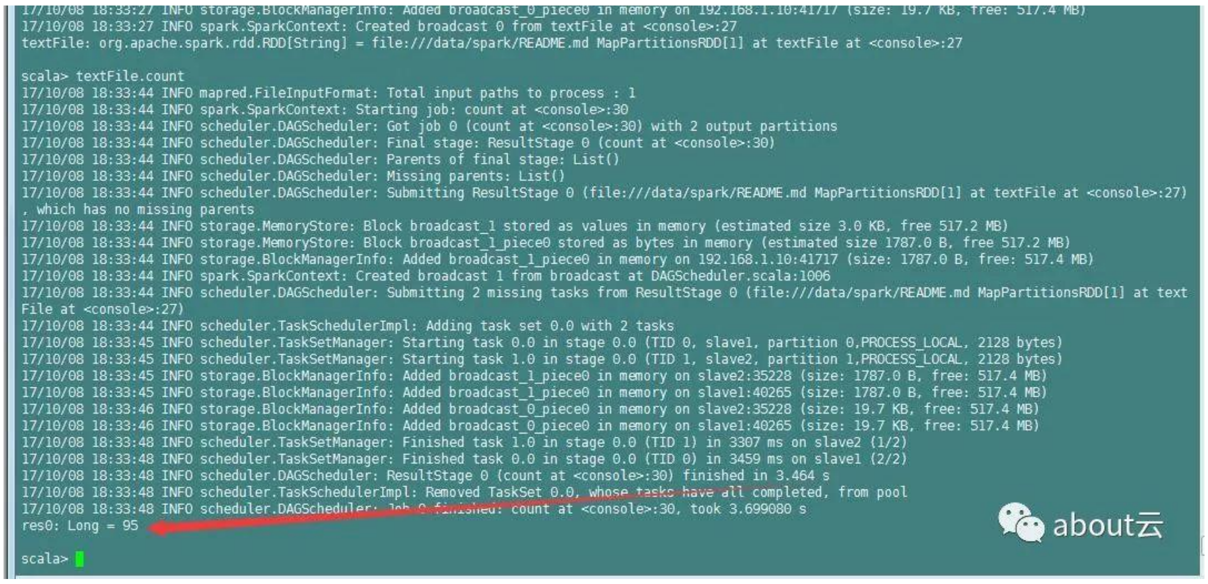
仅将文本发送到当前选项卡

执行

[Bash shell] 纯文本查看 复制代码

?

```
textFile.count
```



```
17/10/08 18:33:27 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on 192.168.1.10:41717 (size: 19.7 KB, free: 517.4 MB)
17/10/08 18:33:27 INFO spark.SparkContext: Created broadcast 0 from textFile at <console>:27
textFile: org.apache.spark.rdd.RDD[String] = file:///data/spark/README.md MapPartitionsRDD[1] at textFile at <console>:27

scala> textFile.count
17/10/08 18:33:44 INFO mapred.FileInputFormat: Total input paths to process : 1
17/10/08 18:33:44 INFO spark.SparkContext: Starting job: count at <console>:30
17/10/08 18:33:44 INFO scheduler.DAGScheduler: Got job 0 (count at <console>:30) with 2 output partitions
17/10/08 18:33:44 INFO scheduler.DAGScheduler: Final stage: ResultStage 0 (count at <console>:30)
17/10/08 18:33:44 INFO scheduler.DAGScheduler: Parents of final stage: List()
17/10/08 18:33:44 INFO scheduler.DAGScheduler: Missing parents: List()
17/10/08 18:33:44 INFO scheduler.DAGScheduler: Submitting ResultStage 0 (file:///data/spark/README.md MapPartitionsRDD[1] at textFile at <console>:27)
, which has no missing parents
17/10/08 18:33:44 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 3.0 KB, free 517.2 MB)
17/10/08 18:33:44 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 1787.0 B, free 517.2 MB)
17/10/08 18:33:44 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on 192.168.1.10:41717 (size: 1787.0 B, free: 517.4 MB)
17/10/08 18:33:44 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1006
17/10/08 18:33:44 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 0 (file:///data/spark/README.md MapPartitionsRDD[1] at text
File at <console>:27)
17/10/08 18:33:44 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 2 tasks
17/10/08 18:33:45 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, slave1, partition 0,PROCESS_LOCAL, 2128 bytes)
17/10/08 18:33:45 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, slave2, partition 1,PROCESS_LOCAL, 2128 bytes)
17/10/08 18:33:45 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on slave2:35228 (size: 1787.0 B, free: 517.4 MB)
17/10/08 18:33:45 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on slave1:40265 (size: 1787.0 B, free: 517.4 MB)
17/10/08 18:33:46 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on slave2:35228 (size: 19.7 KB, free: 517.4 MB)
17/10/08 18:33:46 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on slave1:40265 (size: 19.7 KB, free: 517.4 MB)
17/10/08 18:33:48 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 3307 ms on slave2 (1/2)
17/10/08 18:33:48 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 3459 ms on slave1 (2/2)
17/10/08 18:33:48 INFO scheduler.DAGScheduler: ResultStage 0 (count at <console>:30) finished in 3.464 s
17/10/08 18:33:48 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
17/10/08 18:33:48 INFO scheduler.DAGScheduler: Job 0 finished: count at <console>:30, took 3.699080 s
res0: Long = 95

scala>
```

[Bash shell] 纯文本查看 复制代码

?

```
textFile.first
```

输出如下内容

[Bash shell] 纯文本查看 复制代码

?

```
scala> textFile.first
17/10/08 18:34:23 INFO spark.SparkContext: Starting job: first at <console>:30
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Got job 1 (first at <console>:30) with 1 output partitions
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Final stage: ResultStage 1 (first at <console>:30)
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Parents of final stage: List()
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Missing parents: List()
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Submitting ResultStage 1 (file:///data/spark/README.md MapPartitionsRDD[1] at textFile at <console>:27), which has no missing parents
17/10/08 18:34:23 INFO storage.MemoryStore: Block broadcast_2 stored as values in memory (estimated size 3.1 KB, free 517.2 MB)
17/10/08 18:34:23 INFO storage.MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated size 1843.0 B, free 517.2 MB)
17/10/08 18:34:23 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on 192.168.1.10:41717 (size: 1843.0 B, free: 517.4 MB)
17/10/08 18:34:23 INFO spark.SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:1006
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (file:///data/spark/README.md MapPartitionsRDD[1] at textFile at <console>:27)
17/10/08 18:34:23 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
17/10/08 18:34:23 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 2, slave2, partition 0,PROCESS_LOCAL, 2128 bytes)
17/10/08 18:34:23 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on slave2:35228 (size: 1843.0 B, free: 517.4 MB)
17/10/08 18:34:23 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 2) in 116 ms on slave2 (1/1)
17/10/08 18:34:23 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
17/10/08 18:34:23 INFO scheduler.DAGScheduler: ResultStage 1 (first at <console>:30) finished in 0.117 s
17/10/08 18:34:23 INFO scheduler.DAGScheduler: Job 1 finished: first at
```

```
<console>:30, took 0.161753 s
res1: String = # Apache Spark
```

2.导入日志清洗代码并打包

问题导读

1.通过什么菜单项可以导入源码？
2.打 jar 包需要哪些步骤?
3.如何找到 jar 生成路径？

上一篇：
about 云日志分析实战之清洗日志 1：使用 spark&Scala 分析 Apache 日志
http://www.aboutyun.com/forum.php?mod=viewthread&tid=22856

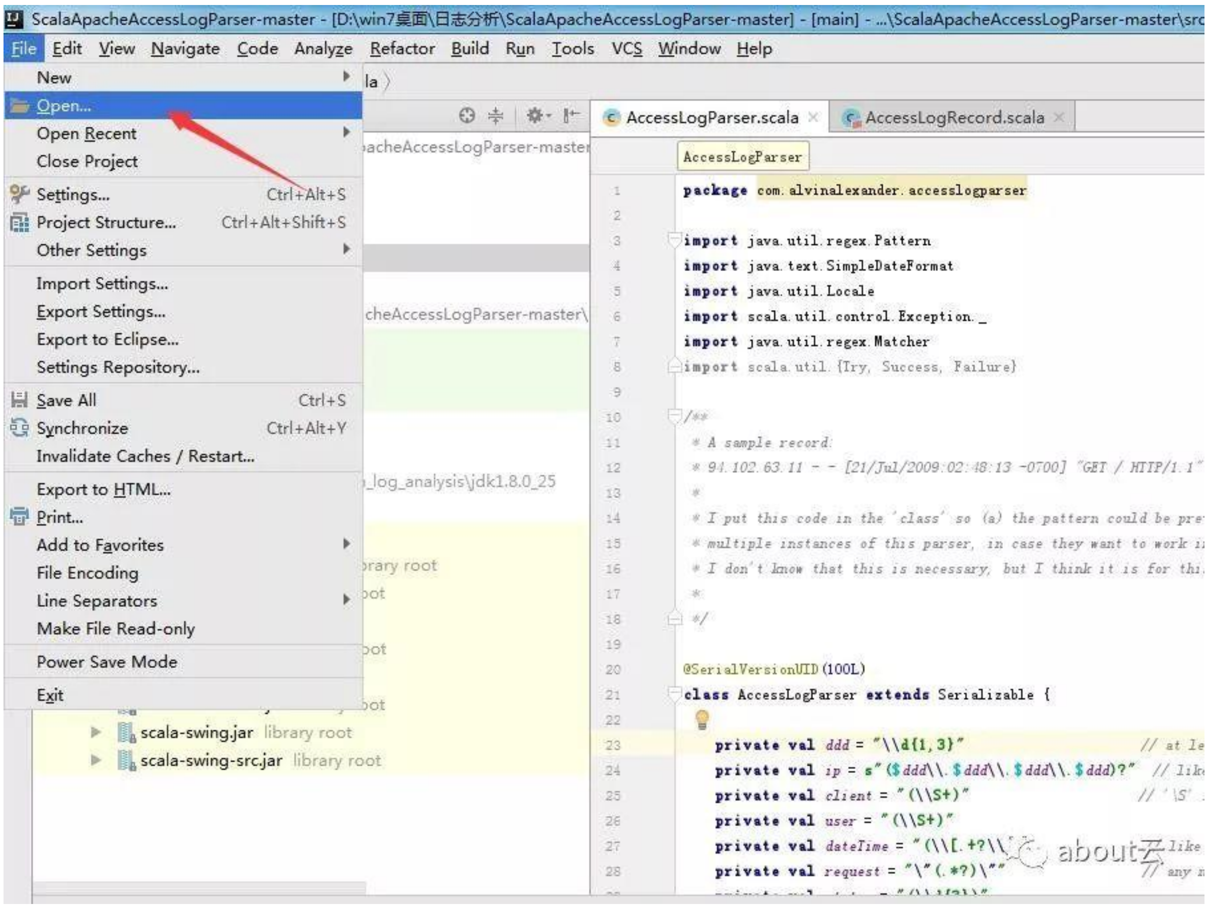
前面测试了一下 spark，准备好环境，下面开始动工源码。分析清洗日志，这里面的代码还是比较复杂的。
对于 iis 日志，可参考
about 云日志分析项目准备 10-3：Spark Local 模式之 Log 文本清洗
http://www.aboutyun.com/forum.php?mod=viewthread&tid=21135
对于 Apache 日志，国外已经实现。源码 git 地址
https://github.com/alvinj/ScalaApacheAccessLogParser

网盘下载地址
链接：http://pan.baidu.com/s/1jIj87wM 密码：p0zd

这里从上面下载下来，然后导入 IntelliJ IDEA ，然后打包。

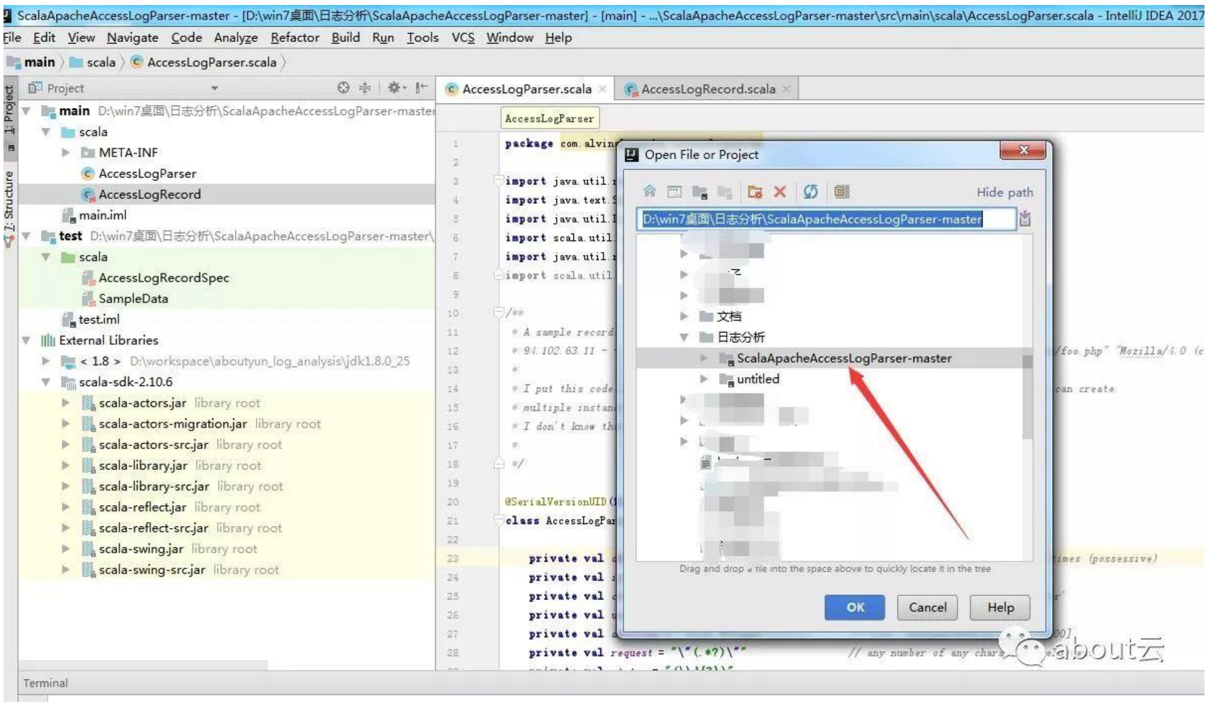
导入源码

首先 file-》open

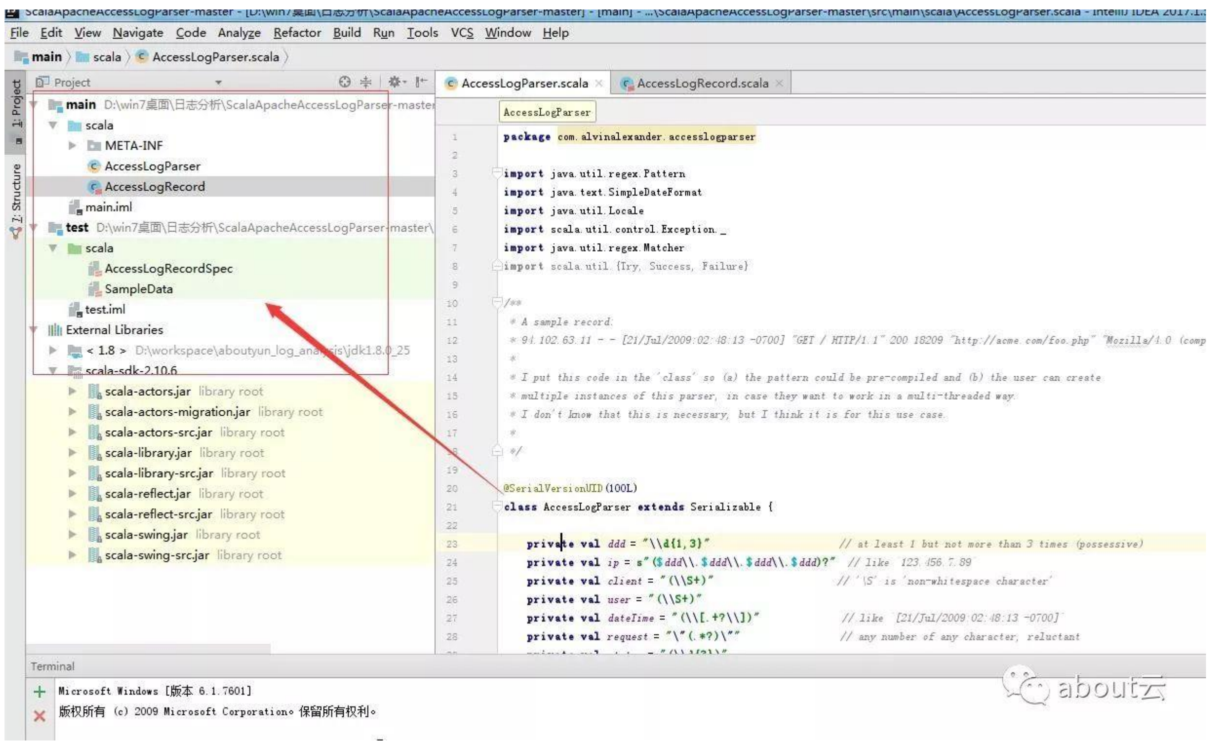


```
package com.alvinalexander.accesslogparser

import java.util.regex.Pattern
import java.text.SimpleDateFormat
import java.util.Locale
import scala.util.control.Exception._
import java.util.regex.Matcher
import scala.util.{Try, Success, Failure}

/**
 * A sample record:
 * 94.102.63.11 - - [21/Jul/2009:02:48:13 -0700] "GET / HTTP/1.1"
 *
 * I put this code in the 'class' so (a) the pattern could be pre
 * multiple instances of this parser, in case they want to work i
 * I don't know that this is necessary, but I think it is for thi
 *
 */

@SerialVersionUID(100L)
class AccessLogParser extends Serializable {

    private val ddd = "\\d{1,3}"                    // at le
    private val ip = s"($ddd\\.$ddd\\.$ddd\\.$ddd)?"  // lik
    private val client = "(\\S+)"                   // '\S'
    private val user = "(\\S+)"
    private val dateTime = "(\\[.+?\\
    private val request = "\"(.*?)\""              // any n
```

选择源码文件


ScalaApacheAccessLogParser-master - [D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master] - [main] - ...\ScalaApacheAccessLogParser-master\src\main\scala\AccessLogParser.scala - IntelliJ IDEA 2017
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
main scala AccessLogParser.scala
Project
main D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master
scala
META-INF
AccessLogParser
AccessLogRecord
main.iml
test D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master\
scala
AccessLogRecordSpec
SampleData
test.iml
External Libraries
< 1.8 > D:\workspace\aboutyun_log_analysis\jdk1.8.0_25
scala-sdk-2.10.6
scala-actors.jar library root
scala-actors-migration.jar library root
scala-actors-src.jar library root
scala-library.jar library root
scala-library-src.jar library root
scala-reflect.jar library root
scala-reflect-src.jar library root
scala-swing.jar library root
scala-swing-src.jar library root
Terminal
Open File or Project
Hide path
D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master
文档
日志分析
ScalaApacheAccessLogParser-master
untitled
Drag and drop a file into the space above to quickly locate it in the tree
OK
Cancel
Help
about云

导入之后看到下面内容


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
main
scala
AccessLogParser.scala
Project
main D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master
scala
META-INF
AccessLogParser
AccessLogRecord
main.iml
test D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master\
scala
AccessLogRecordSpec
SampleData
test.iml
External Libraries
< 1.8 > D:\workspace\aboutyun_log_analysis\jdk1.8.0_25
scala-sdk-2.10.6
scala-actors.jar library root
scala-actors-migration.jar library root
scala-actors-src.jar library root
scala-library.jar library root
scala-library-src.jar library root
scala-reflect.jar library root
scala-reflect-src.jar library root
scala-swing.jar library root
scala-swing-src.jar library root
AccessLogParser.scala
AccessLogRecord.scala
AccessLogParser
Terminal

```
package com.alvinalexander.accesslogparser

import java.util.regex.Pattern
import java.text.SimpleDateFormat
import java.util.Locale
import scala.util.control.Exception._
import java.util.regex.Matcher
import scala.util.{Try, Success, Failure}

/**
 * A sample record:
 * 94.102.63.11 - - [21/Jul/2009:02:48:13 -0700] "GET / HTTP/1.1" 200 18209 "http://acme.com/foo.php" "Mozilla/4.0 (comp
 *
 * I put this code in the 'class' so (a) the pattern could be pre-compiled and (b) the user can create
 * multiple instances of this parser, in case they want to work in a multi-threaded way.
 * I don't know that this is necessary, but I think it is for this use case.
 *
 */

@SerialVersionUID(100L)
class AccessLogParser extends Serializable {

    private val ddd = "\\d{1,3}"                      // at least 1 but not more than 3 times (possessive)
    private val ip = s"($ddd\\.$ddd\\.$ddd\\.$ddd)?"  // like `123.456.7.89`
    private val client = "(\\S+)"                     // '\S' is 'non-whitespace character'
    private val user = "(\\S+)"
    private val dateTime = "(\\[.+?\\])"              // like `[21/Jul/2009:02:48:13 -0700]`
    private val request = "\"(.*?)\""                 // any number of any character, reluctant
```

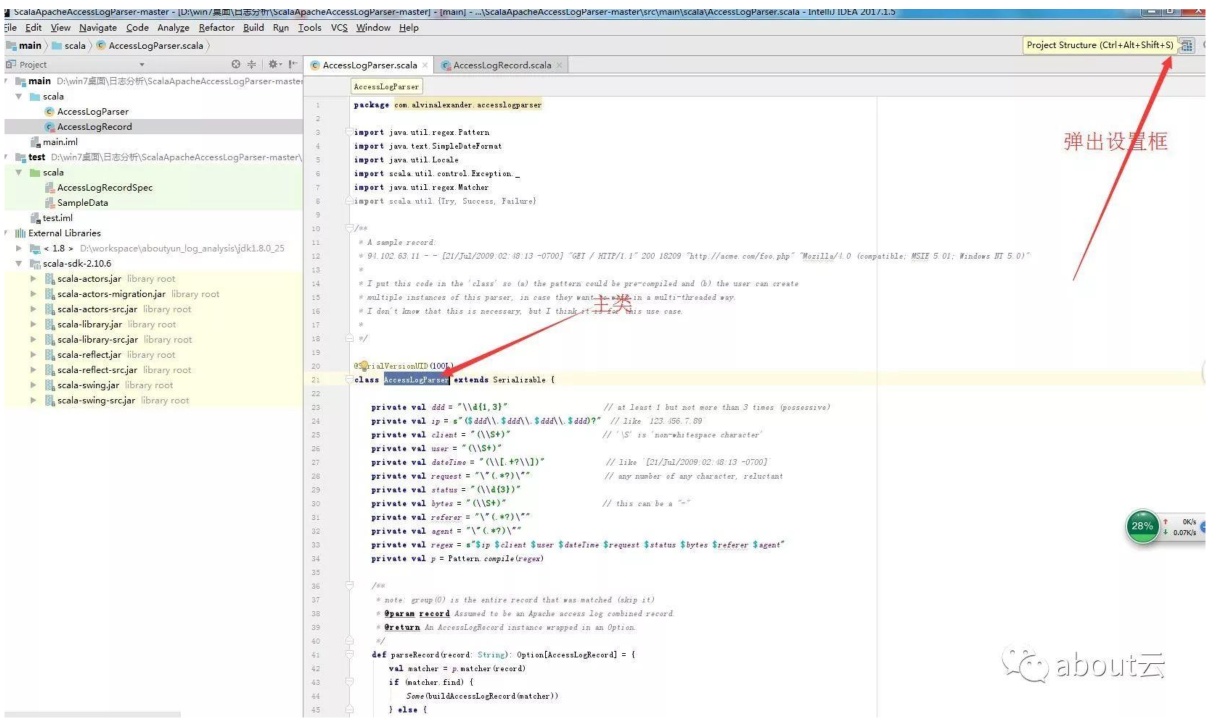
```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。
```

对于 spark 环境不熟悉或则不会操作可参考
spark 开发环境详细教程 1：IntelliJ IDEA 使用详细说明
http://www.aboutyun.com/forum.php?mod=viewthread&tid=22320

spark 开发环境详细教程 3：IntelliJ IDEA 创建项目
http://www.aboutyun.com/forum.php?mod=viewthread&tid=22410

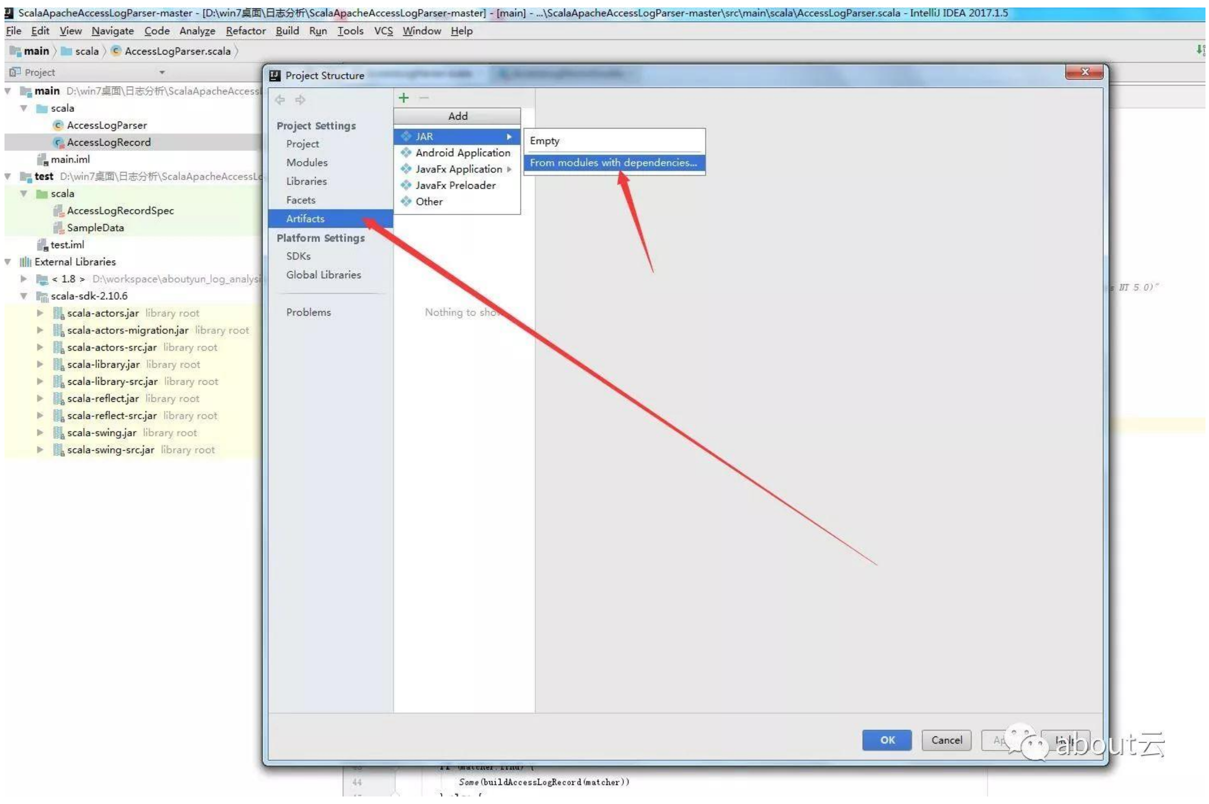
打包

上面我们准备了源码，然后将源码打成 jar 包，供我们项目中使用。
首先打开 project structure,

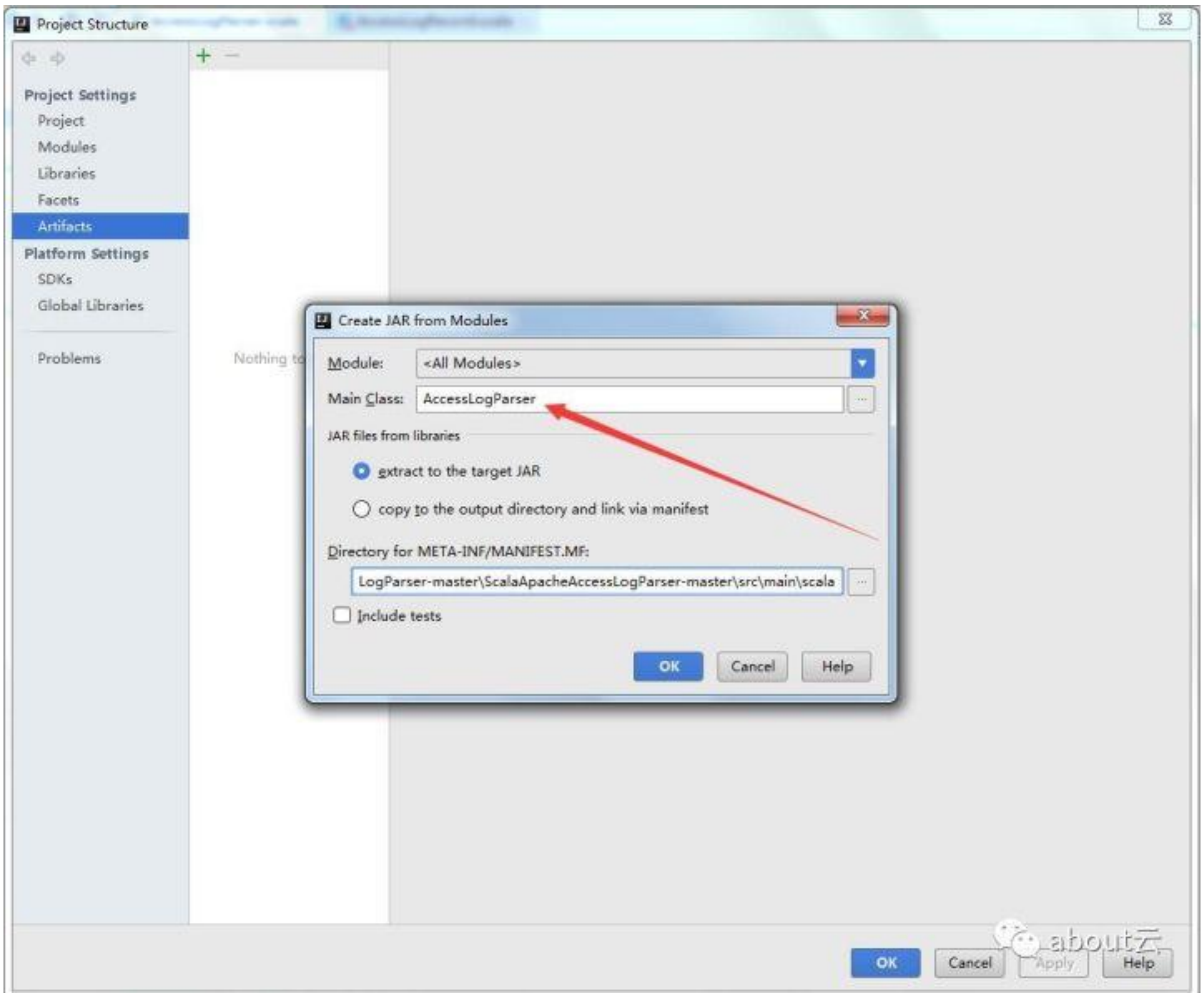


```scala
package com.alvinalexander.accesslogparser

import java.util.regex.Pattern
import java.text.SimpleDateFormat
import java.util.Locale
import scala.util.control.Exception._
import java.util.regex.Matcher
import scala.util.{Try, Success, Failure}

/**
 * A sample record:
 * 94.102.63.11 - - [21/Jul/2009:02:48:13 -0700] "GET / HTTP/1.1" 200 18209 "http://acme.com/foo.php" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"
 *
 * I put this code in the 'class' so (a) the pattern could be pre-compiled and (b) the user can create
 * multiple instances of this parser, in case they want to [illegible] in a multi-threaded way.
 * I don't know that this is necessary, but I think it is for this use case.
 *
 */

@SerialVersionUID(100L)
class AccessLogParser extends Serializable {

    private val ddd = "\\d{1,3}"                      // at least 1 but not more than 3 times (possessive)
    private val ip = s"($ddd\\.$ddd\\.$ddd\\.$ddd)?"  // like `123.456.7.89`
    private val client = "(\\S+)"                     // '\S' is 'non-whitespace character'
    private val user = "(\\S+)"
    private val dateTime = "(\\[.+?\\])"              // like `[21/Jul/2009:02:48:13 -0700]`
    private val request = "\"(.*?)\""                 // any number of any character, reluctant
    private val status = "(\\d{3})"
    private val bytes = "(\\S+)"                      // this can be a "-"
    private val referer = "\"(.*?)\""
    private val agent = "\"(.*?)\""
    private val regex = s"$ip $client $user $dateTime $request $status $bytes $referer $agent"
    private val p = Pattern.compile(regex)

    /**
     * note: group(0) is the entire record that was matched (skip it)
     * @param record Assumed to be an Apache access log combined record.
     * @return An AccessLogRecord instance wrapped in an Option.
     */
    def parseRecord(record: String): Option[AccessLogRecord] = {
        val matcher = p.matcher(record)
        if (matcher.find) {
            Some(buildAccessLogRecord(matcher))
        } else {
```

选择依赖

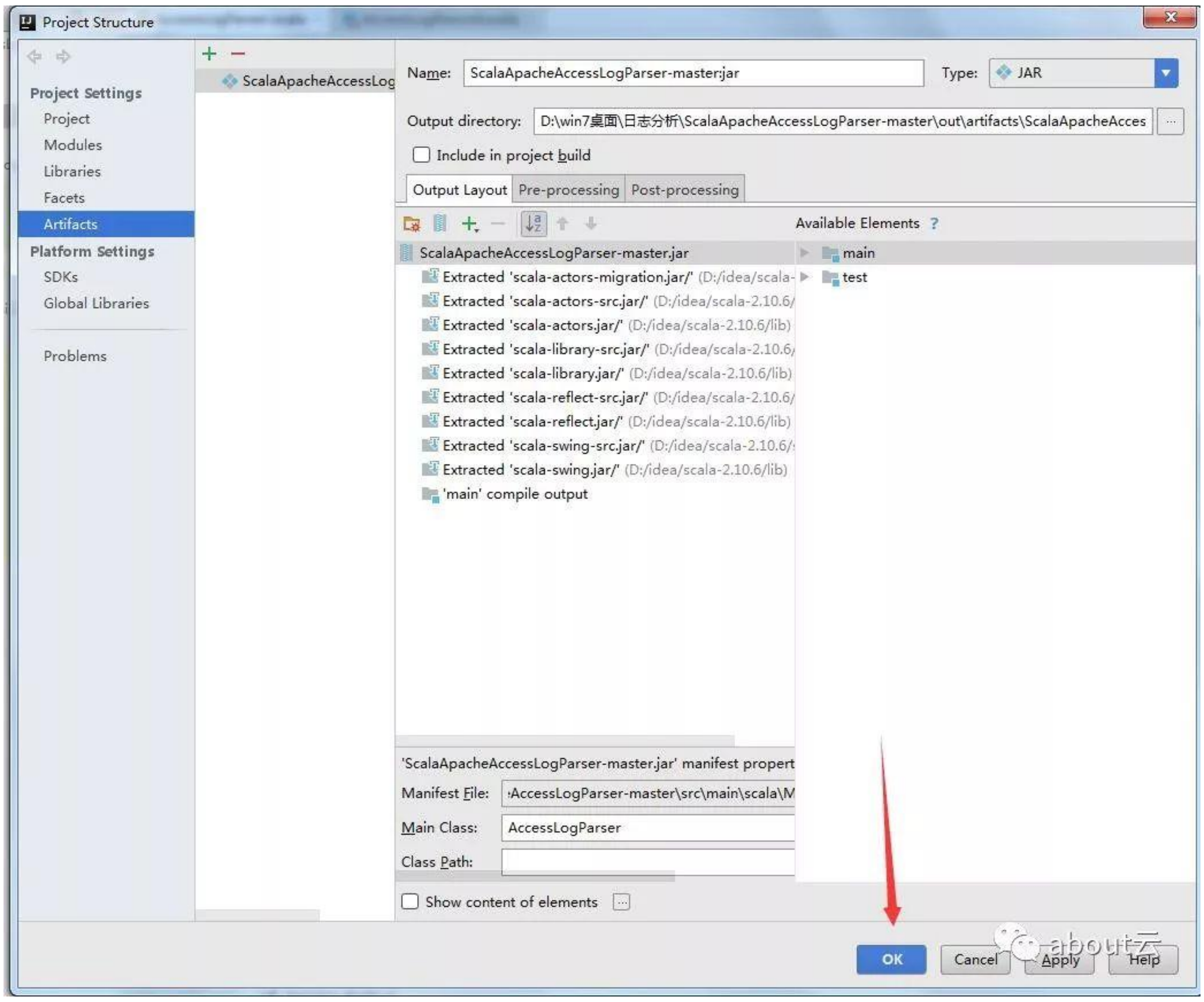


填写主类



Module: <All Modules>
Main Class: AccessLogParser
JAR files from libraries
[✓ extract to the target JAR]
copy to the output directory and link via manifest
Directory for META-INF/MANIFEST.MF:
LogParser-master\ScalaApacheAccessLogParser-master\src\main\scala
Include tests

点击确定



Project Structure

Name: ScalaApacheAccessLogParser-master:jar

Type: JAR

Output directory: D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master\out\artifacts\ScalaApacheAcces

Include in project build

Output Layout | Pre-processing | Post-processing

```
ScalaApacheAccessLogParser-master.jar
  Extracted 'scala-actors-migration.jar/' (D:/idea/scala-
  Extracted 'scala-actors-src.jar/' (D:/idea/scala-2.10.6/
  Extracted 'scala-actors.jar/' (D:/idea/scala-2.10.6/lib)
  Extracted 'scala-library-src.jar/' (D:/idea/scala-2.10.6/
  Extracted 'scala-library.jar/' (D:/idea/scala-2.10.6/lib)
  Extracted 'scala-reflect-src.jar/' (D:/idea/scala-2.10.6/
  Extracted 'scala-reflect.jar/' (D:/idea/scala-2.10.6/lib)
  Extracted 'scala-swing-src.jar/' (D:/idea/scala-2.10.6/
  Extracted 'scala-swing.jar/' (D:/idea/scala-2.10.6/lib)
  'main' compile output
```

'ScalaApacheAccessLogParser-master.jar' manifest propert

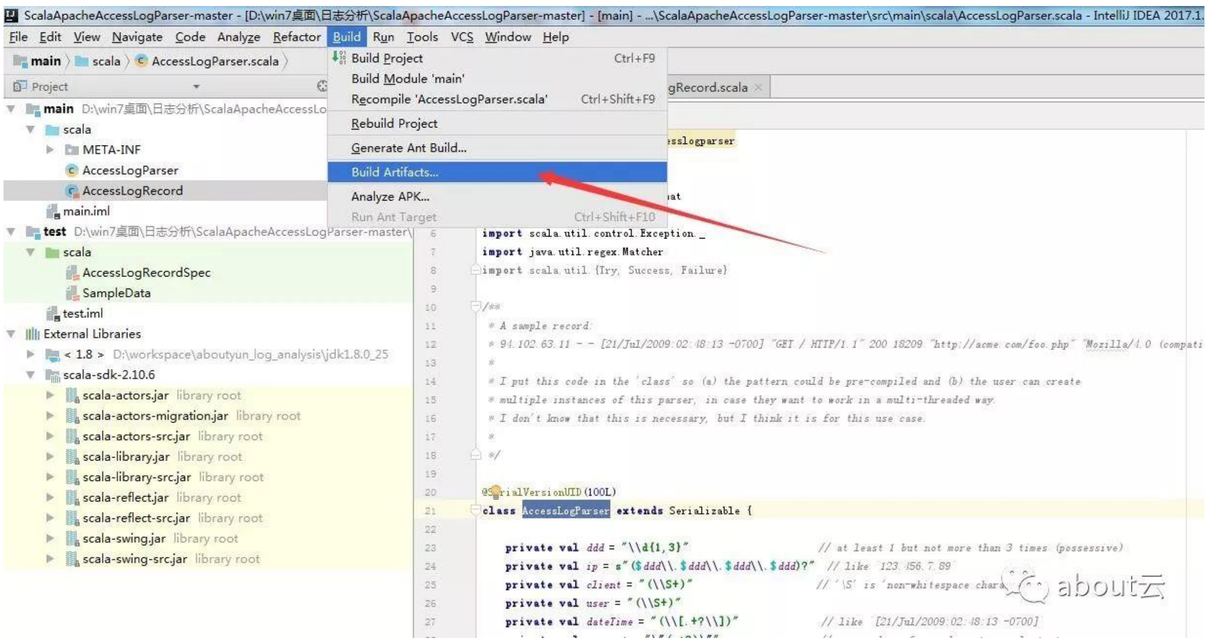
Manifest File: ·AccessLogParser-master\src\main\scala\M

Main Class: AccessLogParser

Class Path:

Show content of elements

选择菜单 Build Artifacts



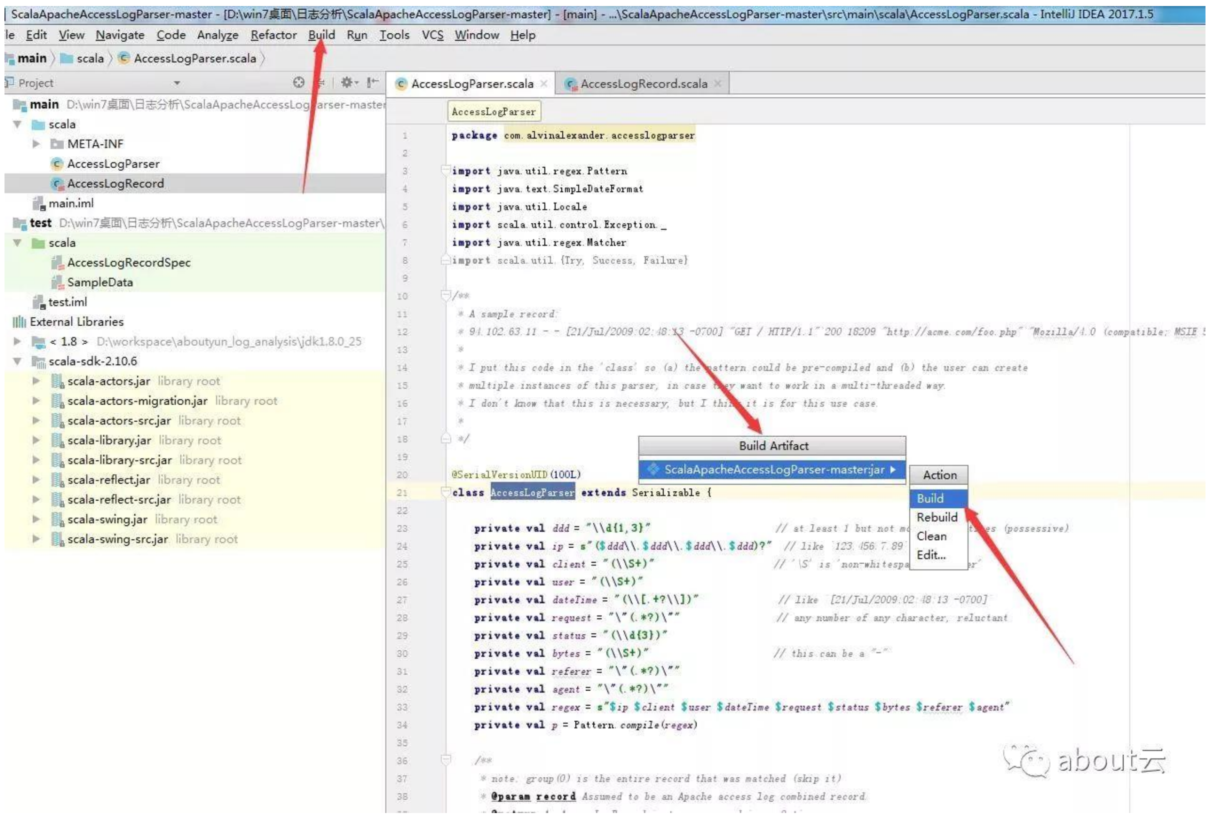
ScalaApacheAccessLogParser-master - [D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master] - [main] - ...\ScalaApacheAccessLogParser-master\src\main\scala\AccessLogParser.scala - IntelliJ IDEA 2017.1.

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
main scala AccessLogParser.scala
Project
Build Project Ctrl+F9
Build Module 'main'
Recompile 'AccessLogParser.scala' Ctrl+Shift+F9
Rebuild Project
Generate Ant Build...
Build Artifacts...
Analyze APK...
Run Ant Target Ctrl+Shift+F10
gRecord.scala

```
main D:\win7桌面\日志分析\ScalaApacheAccessLo
  scala
    META-INF
    AccessLogParser
    AccessLogRecord
  main.iml
test D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master\
  scala
    AccessLogRecordSpec
    SampleData
  test.iml
External Libraries
  < 1.8 > D:\workspace\aboutyun_log_analysis\jdk1.8.0_25
  scala-sdk-2.10.6
    scala-actors.jar library root
    scala-actors-migration.jar library root
    scala-actors-src.jar library root
    scala-library.jar library root
    scala-library-src.jar library root
    scala-reflect.jar library root
    scala-reflect-src.jar library root
    scala-swing.jar library root
    scala-swing-src.jar library root
```

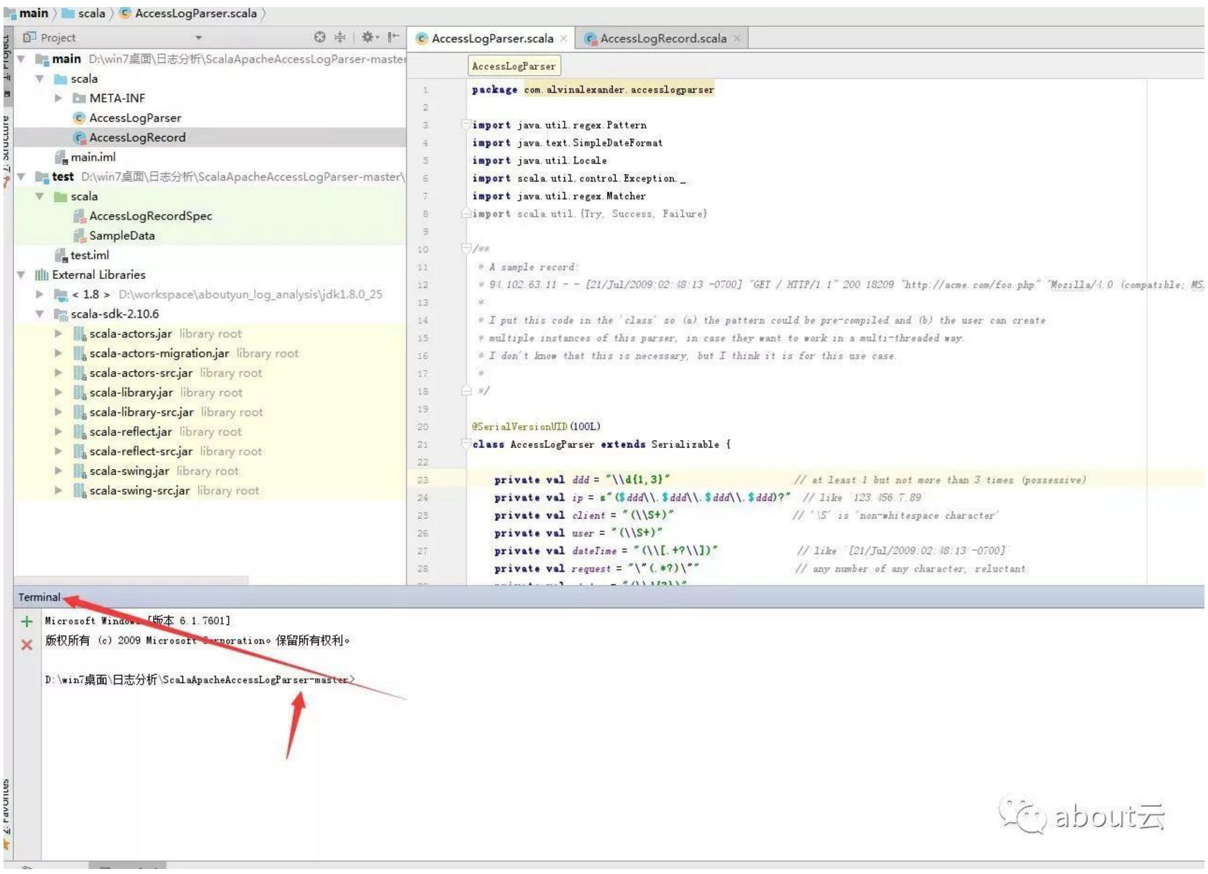
```scala
import scala.util.control.Exception._
import java.util.regex.Matcher
import scala.util.{Try, Success, Failure}

/**
 * A sample record:
 * 94.102.63.11 - - [21/Jul/2009:02:48:13 -0700] "GET / HTTP/1.1" 200 18209 "http://acme.com/foo.php" "Mozilla/4.0 (compati
 *
 * I put this code in the 'class' so (a) the pattern could be pre-compiled and (b) the user can create
 * multiple instances of this parser, in case they want to work in a multi-threaded way.
 * I don't know that this is necessary, but I think it is for this use case.
 *
 */

@SerialVersionUID(100L)
class AccessLogParser extends Serializable {

    private val ddd = "\\d{1,3}"                      // at least 1 but not more than 3 times (possessive)
    private val ip = s"($ddd\\.$ddd\\.$ddd\\.$ddd)?"  // like `123.456.7.89`
    private val client = "(\\S+)"                     // '\S' is 'non-whitespace chara
    private val user = "(\\S+)"
    private val dateTime = "(\\[.+?\\])"              // like `[21/Jul/2009:02:48:13 -0700]`
```

点击 build


ScalaApacheAccessLogParser-master - [D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master] - [main] - ...\ScalaApacheAccessLogParser-master\src\main\scala\AccessLogParser.scala - IntelliJ IDEA 2017.1.5
le Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
main scala AccessLogParser.scala
Project
main D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master
scala
META-INF
AccessLogParser
AccessLogRecord
main.iml
test D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master\
scala
AccessLogRecordSpec
SampleData
test.iml
External Libraries
< 1.8 > D:\workspace\aboutyun_log_analysis\jdk1.8.0_25
scala-sdk-2.10.6
scala-actors.jar library root
scala-actors-migration.jar library root
scala-actors-src.jar library root
scala-library.jar library root
scala-library-src.jar library root
scala-reflect.jar library root
scala-reflect-src.jar library root
scala-swing.jar library root
scala-swing-src.jar library root
AccessLogParser.scala
AccessLogRecord.scala
AccessLogParser
Build Artifact
ScalaApacheAccessLogParser-master:jar
Action
Build
Rebuild
Clean
Edit...
about云

```
package com.alvinalexander.accesslogparser

import java.util.regex.Pattern
import java.text.SimpleDateFormat
import java.util.Locale
import scala.util.control.Exception._
import java.util.regex.Matcher
import scala.util.{Try, Success, Failure}

/**
 * A sample record:
 * 94.102.63.11 - - [21/Jul/2009:02:48:13 -0700] "GET / HTTP/1.1" 200 18209 "http://acme.com/foo.php" "Mozilla/4.0 (compatible; MSIE
 *
 * I put this code in the 'class' so (a) the pattern could be pre-compiled and (b) the user can create
 * multiple instances of this parser, in case they want to work in a multi-threaded way.
 * I don't know that this is necessary, but I think it is for this use case.
 *
 */

@SerialVersionUID(100L)
class AccessLogParser extends Serializable {

    private val ddd = "\\d{1,3}"                      // at least 1 but not mo          tes (possessive)
    private val ip = s"($ddd\\.$ddd\\.$ddd\\.$ddd)?"  // like `123.456.7.89`
    private val client = "(\\S+)"                     // '\S' is 'non-whitespa          er'
    private val user = "(\\S+)"
    private val dateTime = "(\\[.+?\\])"              // like `[21/Jul/2009:02:48:13 -0700]`
    private val request = "\"(.*?)\""                 // any number of any character, reluctant
    private val status = "(\\d{3})"
    private val bytes = "(\\S+)"                      // this can be a "-"
    private val referer = "\"(.*?)\""
    private val agent = "\"(.*?)\""
    private val regex = s"$ip $client $user $dateTime $request $status $bytes $referer $agent"
    private val p = Pattern.compile(regex)

    /**
     * note: group(0) is the entire record that was matched (skip it)
     * @param record Assumed to be an Apache access log combined record.
```

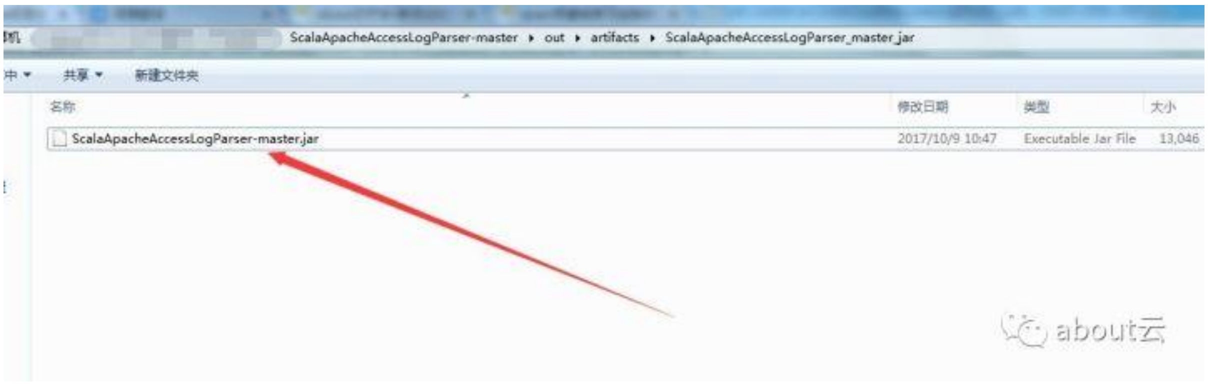
最后生成 jar 包，在 terminal 中会显示输出 jar 包路径


main 〉 scala 〉 AccessLogParser.scala 〉
Project
main D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master
scala
META-INF
AccessLogParser
AccessLogRecord
main.iml
test D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master\
scala
AccessLogRecordSpec
SampleData
test.iml
External Libraries
< 1.8 > D:\workspace\aboutyun_log_analysis\jdk1.8.0_25
scala-sdk-2.10.6
scala-actors.jar library root
scala-actors-migration.jar library root
scala-actors-src.jar library root
scala-library.jar library root
scala-library-src.jar library root
scala-reflect.jar library root
scala-reflect-src.jar library root
scala-swing.jar library root
scala-swing-src.jar library root
AccessLogParser.scala
AccessLogRecord.scala
AccessLogParser
Terminal
about云

```
package com.alvinalexander.accesslogparser

import java.util.regex.Pattern
import java.text.SimpleDateFormat
import java.util.Locale
import scala.util.control.Exception._
import java.util.regex.Matcher
import scala.util.{Try, Success, Failure}

/**
 * A sample record:
 * 94.102.63.11 - - [21/Jul/2009:02:48:13 -0700] "GET / HTTP/1.1" 200 18209 "http://acme.com/foo.php" "Mozilla/4.0 (compatible; MS
 *
 * I put this code in the 'class' so (a) the pattern could be pre-compiled and (b) the user can create
 * multiple instances of this parser, in case they want to work in a multi-threaded way.
 * I don't know that this is necessary, but I think it is for this use case.
 *
 */

@SerialVersionUID(100L)
class AccessLogParser extends Serializable {

    private val ddd = "\\d{1,3}"                      // at least 1 but not more than 3 times (possessive)
    private val ip = s"($ddd\\.$ddd\\.$ddd\\.$ddd)?"  // like `123.456.7.89`
    private val client = "(\\S+)"                     // '\S' is 'non-whitespace character'
    private val user = "(\\S+)"
    private val dateTime = "(\\[.+?\\])"              // like `[21/Jul/2009:02:48:13 -0700]`
    private val request = "\"(.*?)\""                 // any number of any character, reluctant
```

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

D:\win7桌面\日志分析\ScalaApacheAccessLogParser-master>
```

找到生成 jar 包。我们就可以使用了。


ScalaApacheAccessLogParser-master ▸ out ▸ artifacts ▸ ScalaApacheAccessLogParser_master_jar
共享 ▾
新建文件夹
about云

名称	修改日期	类型	大小
ScalaApacheAccessLogParser-master.jar	2017/10/9 10:47	Executable Jar File	13,046

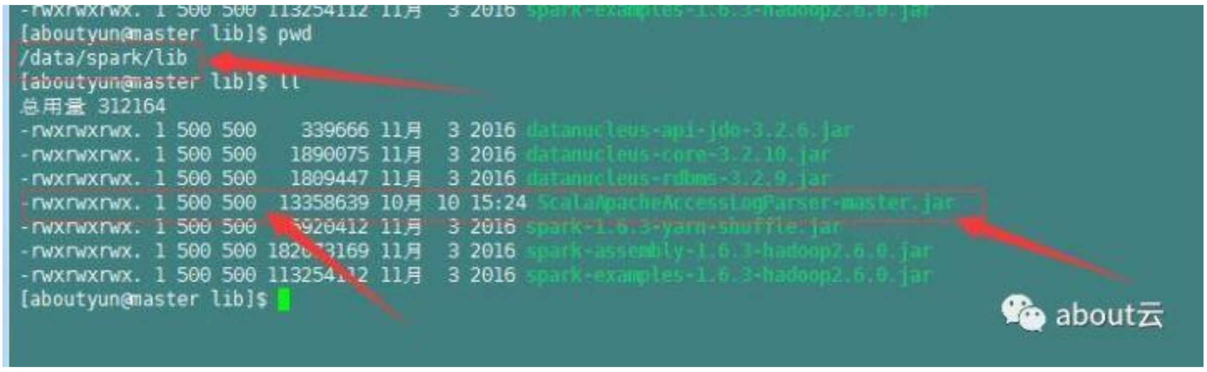
3.如何在 spark shell 中导入自定义包

问题导读

1.自定义包，本文放到哪个路径下面?
2.复制包之后，需要做哪些权限操作?
3.如何验证导入是否成功?

上一篇
about 云日志分析实战之清洗日志 2：导入日志清洗代码并打包
http://www.aboutyun.com/forum.php?mod=viewthread&tid=22862

上一篇文章，生成了包，那么这个包该如何加载到 spark 环境中，并且为我们所使用。那么首先改如何加载这个包。
首先将这个包放到 spark 中的 lib 文件夹下。


```
-rwxrwxrwx. 1 500 500 113254112 11月  3 2016 spark-examples-1.6.3-hadoop2.6.0.jar
[aboutyun@master lib]$ pwd
/data/spark/lib
[aboutyun@master lib]$ ll
总用量 312164
-rwxrwxrwx. 1 500 500    339666 11月  3 2016 datanucleus-api-jdo-3.2.6.jar
-rwxrwxrwx. 1 500 500   1890075 11月  3 2016 datanucleus-core-3.2.10.jar
-rwxrwxrwx. 1 500 500   1809447 11月  3 2016 datanucleus-rdbms-3.2.9.jar
-rwxrwxrwx. 1 500 500  13358639 10月 10 15:24 ScalaApacheAccessLogParser-master.jar
-rwxrwxrwx. 1 500 500  [illegible]5920412 11月  3 2016 spark-1.6.3-yarn-shuffle.jar
-rwxrwxrwx. 1 500 500 182[illegible]3169 11月  3 2016 spark-assembly-1.6.3-hadoop2.6.0.jar
-rwxrwxrwx. 1 500 500 113254112 11月  3 2016 spark-examples-1.6.3-hadoop2.6.0.jar
[aboutyun@master lib]$
```

在复制到 Linux 中，首先需要修改的就是权限。
我们看到用户和组的权限为 500，并且用户，所属组，及其它用户都为满权限，
可以通过下面命令来实现

[Bash shell] 纯文本查看 复制代码

?

```
sudo chown 500:500 ScalaApacheAccessLogParser-master.jar
```

[Bash shell] 纯文本查看 复制代码

?

```
sudo chmod -R a+r ScalaApacheAccessLogParser-master.jar
```

[Bash shell] 纯文本查看 复制代码

?

```
sudo chmod -R a+w ScalaApacheAccessLogParser-master.jar
```

[Bash shell] 纯文本查看 复制代码

?

```
sudo chmod -R a+x ScalaApacheAccessLogParser-master.jar
```

通过上面命令即可实现授权。

授权完毕，接着我们就需要把这个包，加载到 spark shell 环境中。

[Bash shell] 纯文本查看 复制代码

?

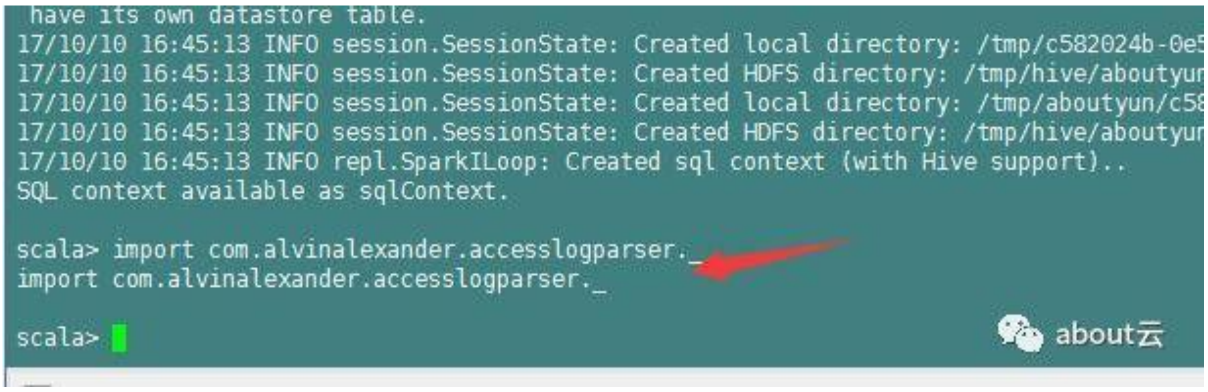
```
./bin/spark-shell --jars lib/ScalaApacheAccessLogParser-master.jar
```

接着我们执行导入 jar 包

[Bash shell] 纯文本查看 复制代码

?

```
import com.alvinalexander.accesslogparser._
```



```
  have its own datastore table.
17/10/10 16:45:13 INFO session.SessionState: Created local directory: /tmp/c582024b-0e5
17/10/10 16:45:13 INFO session.SessionState: Created HDFS directory: /tmp/hive/aboutyun
17/10/10 16:45:13 INFO session.SessionState: Created local directory: /tmp/aboutyun/c58
17/10/10 16:45:13 INFO session.SessionState: Created HDFS directory: /tmp/hive/aboutyun
17/10/10 16:45:13 INFO repl.SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala> import com.alvinalexander.accesslogparser._
import com.alvinalexander.accesslogparser._

scala> 
```

至此我们就可以使用第三方包了。

问题：
同时尝试了比较多的导入方式，没有成功，记录下来共大家借鉴。

[Bash shell] 纯文本查看 复制代码

?

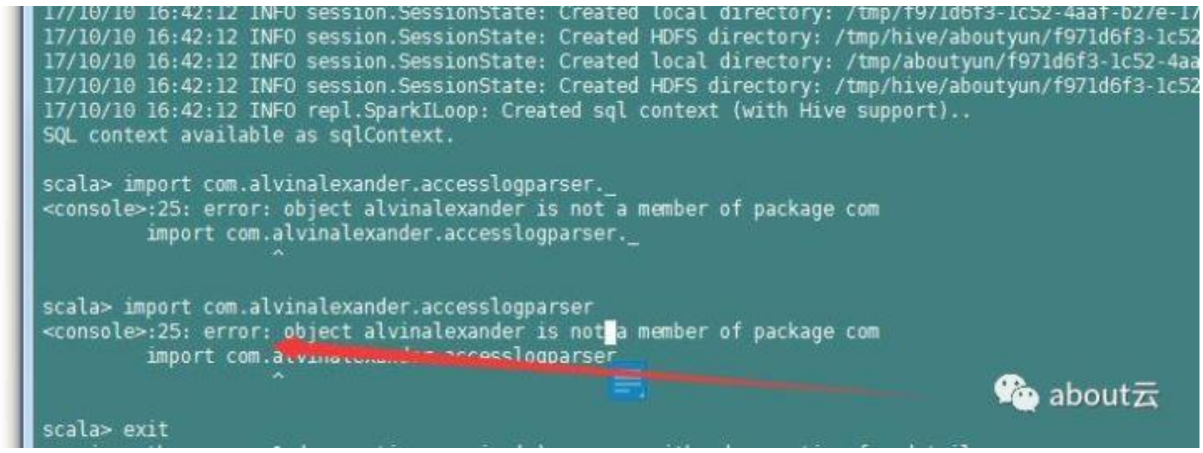
```
./bin/spark-shell –master spark://master:7077 –jars
ScalaApacheAccessLogParser-master.jar
```

[Bash shell] 纯文本查看 复制代码

?

```
MASTER=local[4]
ADD_JARS=/data/spark/lib/AlsApacheLogParser.jar ./bin/spark-shell
```

导入的时候，并不会报错，但是 import 的时候，报错。



```
17/10/10 16:42:12 INFO session.SessionState: Created local directory: /tmp/f971d6f3-1c52-4aaf-b27e-17
17/10/10 16:42:12 INFO session.SessionState: Created HDFS directory: /tmp/hive/aboutyun/f971d6f3-1c52
17/10/10 16:42:12 INFO session.SessionState: Created local directory: /tmp/aboutyun/f971d6f3-1c52-4aa
17/10/10 16:42:12 INFO session.SessionState: Created HDFS directory: /tmp/hive/aboutyun/f971d6f3-1c52
17/10/10 16:42:12 INFO repl.SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala> import com.alvinalexander.accesslogparser._
<console>:25: error: object alvinalexander is not a member of package com
         import com.alvinalexander.accesslogparser._
                    ^

scala> import com.alvinalexander.accesslogparser
<console>:25: error: object alvinalexander is not a member of package com
         import com.alvinalexander.accesslogparser
                    ^

scala> exit
```

4.统计网站相关信息

问题导读

1.如何统计网站总的点击量？
2.如何实现统计不能访问网页的个数?
3.文章中如何定义和使用 Scala 函数的？

上一篇
about 云日志分析实战之清洗日志 3：如何在 spark shell 中导入自定义包

http://www.aboutyun.com/forum.php?mod=viewthread&tid=22881

上一篇，我们已经添加了清洗日志的核心代码，那么剩下的我们就可以统计相关信息，比如最简单的找到不能访问的网页。

导入之后，我们创建 AccessLogParser 实例

[Bash shell] 纯文本查看 复制代码

?

```
val p = new AccessLogParser
```

这个很重要，在后面我们会用到

首先我们需要加载一部分日志样例。

[Bash shell] 纯文本查看 复制代码

?

```
192.168.169.50 - - [17/Feb/2012:10:09:13 +0800] "GET /favicon.ico HTTP/1.1" 404 288 "-" "360se"
192.168.169.50 - - [17/Feb/2012:10:36:26 +0800] "GET / HTTP/1.1" 403 5043 "-" "Mozilla/5.0 (Windows NT 5.1; rv:6.0) Gecko/20100101 Firefox/6.0"
192.168.169.50 - - [17/Feb/2012:10:36:26 +0800] "GET /icons/powered_by_rh.png HTTP/1.1" 200 1213 "http://192.168.55.230/" "Mozilla/5.0 (Windows NT 5.1; rv:6.0) Gecko/20100101 Firefox/6.0"
192.168.169.50 - - [17/Feb/2012:10:09:10 +0800] "GET /icons/powered_by_rh.png HTTP/1.1" 200 1213 "http://192.168.55.230/" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2; 360SE)"
192.168.55.230 - - [24/Feb/2012:09:48:58 +0800] "GET /favicon.ico HTTP/1.1" 404 288 "-" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.169.50 - - [24/Feb/2012:09:45:03 +0800] "GET /server-status HTTP/1.1" 404 290 "-" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2; 360SE)"
192.168.55.230 - - [24/Feb/2012:09:49:02 +0800] "GET / HTTP/1.1" 403 5043 "-" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.55.230 - - [24/Feb/2012:09:49:02 +0800] "GET /icons/apache_pb.gif HTTP/1.1" 200 2326 "http://192.168.55.230/" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.55.230 - - [24/Feb/2012:09:49:02 +0800] "GET /icons/powered_by_rh.png HTTP/1.1" 200 1213 "http://192.168.55.230/" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
```

```
192.168.55.230 - - [24/Feb/2012:09:49:20 +0800] "GET /server-status
HTTP/1.1" 404 290 "-" "Mozilla/5.0 (X11; U; Linux x86_64; en-US;
rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
```

将其保存为 aboutyun.log

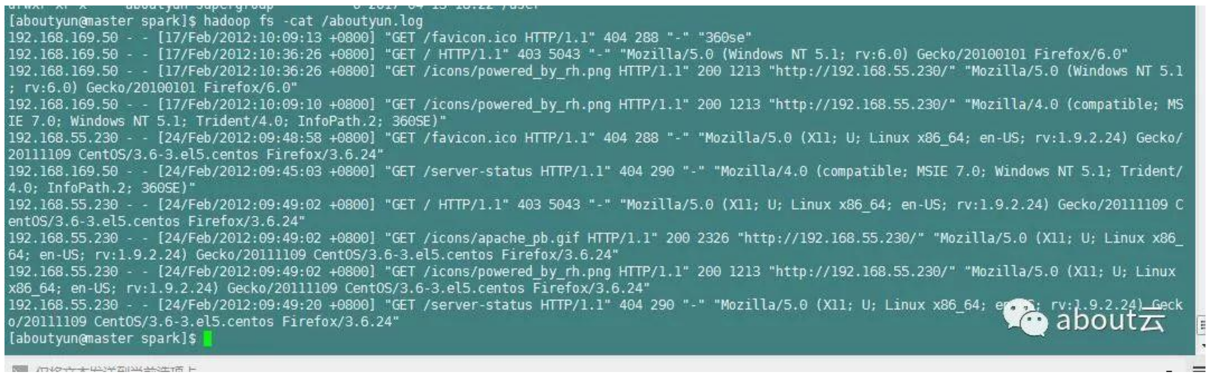
将其上传到 hadoop

[Bash shell] 纯文本查看 复制代码

?

```
hadoop fs -put aboutyun.log /
```

上传成功验证



```
[aboutyun@master spark]$ hadoop fs -cat /aboutyun.log
192.168.169.50 - - [17/Feb/2012:10:09:13 +0800] "GET /favicon.ico HTTP/1.1" 404 288 "-" "360se"
192.168.169.50 - - [17/Feb/2012:10:36:26 +0800] "GET / HTTP/1.1" 403 5043 "-" "Mozilla/5.0 (Windows NT 5.1; rv:6.0) Gecko/20100101 Firefox/6.0"
192.168.169.50 - - [17/Feb/2012:10:36:26 +0800] "GET /icons/powered_by_rh.png HTTP/1.1" 200 1213 "http://192.168.55.230/" "Mozilla/5.0 (Windows NT 5.1
; rv:6.0) Gecko/20100101 Firefox/6.0"
192.168.169.50 - - [17/Feb/2012:10:09:10 +0800] "GET /icons/powered_by_rh.png HTTP/1.1" 200 1213 "http://192.168.55.230/" "Mozilla/4.0 (compatible; MS
IE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2; 360SE)"
192.168.55.230 - - [24/Feb/2012:09:48:58 +0800] "GET /favicon.ico HTTP/1.1" 404 288 "-" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Gecko/
20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.169.50 - - [24/Feb/2012:09:45:03 +0800] "GET /server-status HTTP/1.1" 404 290 "-" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/
4.0; InfoPath.2; 360SE)"
192.168.55.230 - - [24/Feb/2012:09:49:02 +0800] "GET / HTTP/1.1" 403 5043 "-" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Gecko/20111109 C
entOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.55.230 - - [24/Feb/2012:09:49:02 +0800] "GET /icons/apache_pb.gif HTTP/1.1" 200 2326 "http://192.168.55.230/" "Mozilla/5.0 (X11; U; Linux x86_
64; en-US; rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.55.230 - - [24/Feb/2012:09:49:02 +0800] "GET /icons/powered_by_rh.png HTTP/1.1" 200 1213 "http://192.168.55.230/" "Mozilla/5.0 (X11; U; Linux
x86_64; en-US; rv:1.9.2.24) Gecko/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
192.168.55.230 - - [24/Feb/2012:09:49:20 +0800] "GET /server-status HTTP/1.1" 404 290 "-" "Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.24) Geck
o/20111109 CentOS/3.6-3.el5.centos Firefox/3.6.24"
[aboutyun@master spark]$
```

统计网站总的点击量

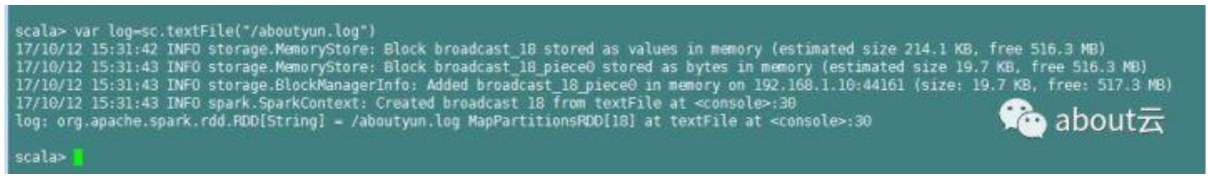
接着我们加载文件。

[Bash shell] 纯文本查看 复制代码

?

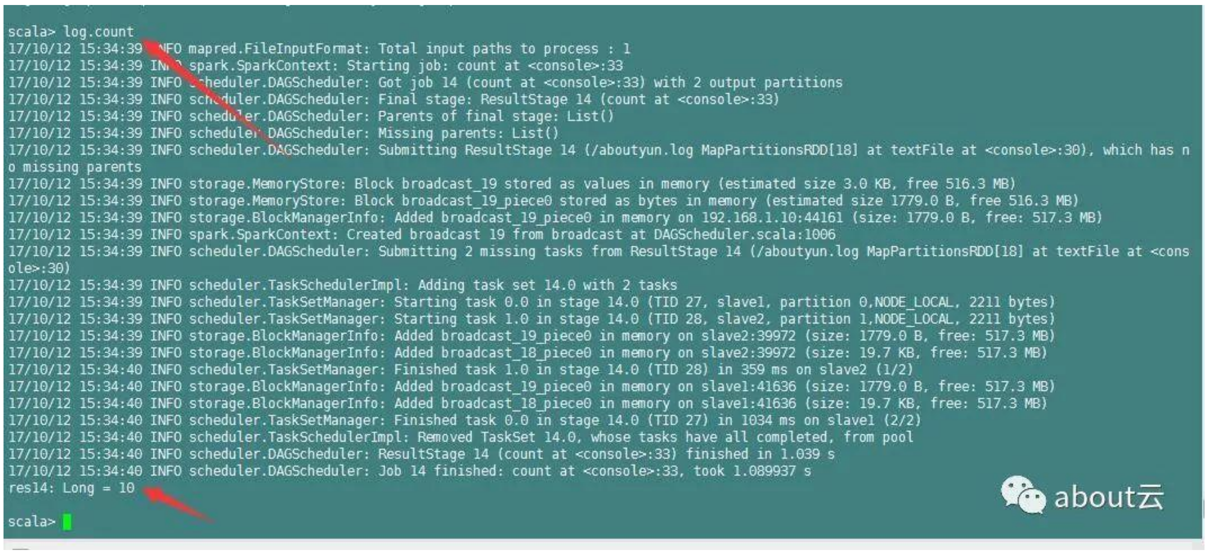
```
var log=sc.textFile("/aboutyun.log")
```

这里 sc 是系统已经初始化的，我们可以直接使用，可以理解为 sparkContext 的实例



```
scala> var log=sc.textFile("/aboutyun.log")
17/10/12 15:31:42 INFO storage.MemoryStore: Block broadcast_18 stored as values in memory (estimated size 214.1 KB, free 516.3 MB)
17/10/12 15:31:43 INFO storage.MemoryStore: Block broadcast_18_piece0 stored as bytes in memory (estimated size 19.7 KB, free 516.3 MB)
17/10/12 15:31:43 INFO storage.BlockManagerInfo: Added broadcast_18_piece0 in memory on 192.168.1.10:44161 (size: 19.7 KB, free: 517.3 MB)
17/10/12 15:31:43 INFO spark.SparkContext: Created broadcast 18 from textFile at <console>:30
log: org.apache.spark.rdd.RDD[String] = /aboutyun.log MapPartitionsRDD[18] at textFile at <console>:30

scala>
```

加载之后，我们统计行数，也可以理解为统计网站总的点击量。这时候我们就看到总点击量为 10



```
scala> log.count
17/10/12 15:34:39 INFO mapred.FileInputFormat: Total input paths to process : 1
17/10/12 15:34:39 INFO spark.SparkContext: Starting job: count at <console>:33
17/10/12 15:34:39 INFO scheduler.DAGScheduler: Got job 14 (count at <console>:33) with 2 output partitions
17/10/12 15:34:39 INFO scheduler.DAGScheduler: Final stage: ResultStage 14 (count at <console>:33)
17/10/12 15:34:39 INFO scheduler.DAGScheduler: Parents of final stage: List()
17/10/12 15:34:39 INFO scheduler.DAGScheduler: Missing parents: List()
17/10/12 15:34:39 INFO scheduler.DAGScheduler: Submitting ResultStage 14 (/aboutyun.log MapPartitionsRDD[18] at textFile at <console>:30), which has n
o missing parents
17/10/12 15:34:39 INFO storage.MemoryStore: Block broadcast_19 stored as values in memory (estimated size 3.0 KB, free 516.3 MB)
17/10/12 15:34:39 INFO storage.MemoryStore: Block broadcast_19_piece0 stored as bytes in memory (estimated size 1779.0 B, free 516.3 MB)
17/10/12 15:34:39 INFO storage.BlockManagerInfo: Added broadcast_19_piece0 in memory on 192.168.1.10:44161 (size: 1779.0 B, free: 517.3 MB)
17/10/12 15:34:39 INFO spark.SparkContext: Created broadcast 19 from broadcast at DAGScheduler.scala:1006
17/10/12 15:34:39 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 14 (/aboutyun.log MapPartitionsRDD[18] at textFile at <cons
ole>:30)
17/10/12 15:34:39 INFO scheduler.TaskSchedulerImpl: Adding task set 14.0 with 2 tasks
17/10/12 15:34:39 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 14.0 (TID 27, slave1, partition 0,NODE_LOCAL, 2211 bytes)
17/10/12 15:34:39 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 14.0 (TID 28, slave2, partition 1,NODE_LOCAL, 2211 bytes)
17/10/12 15:34:39 INFO storage.BlockManagerInfo: Added broadcast_19_piece0 in memory on slave2:39972 (size: 1779.0 B, free: 517.3 MB)
17/10/12 15:34:39 INFO storage.BlockManagerInfo: Added broadcast_18_piece0 in memory on slave2:39972 (size: 19.7 KB, free: 517.3 MB)
17/10/12 15:34:40 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 14.0 (TID 28) in 359 ms on slave2 (1/2)
17/10/12 15:34:40 INFO storage.BlockManagerInfo: Added broadcast_19_piece0 in memory on slave1:41636 (size: 1779.0 B, free: 517.3 MB)
17/10/12 15:34:40 INFO storage.BlockManagerInfo: Added broadcast_18_piece0 in memory on slave1:41636 (size: 19.7 KB, free: 517.3 MB)
17/10/12 15:34:40 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 14.0 (TID 27) in 1034 ms on slave1 (2/2)
17/10/12 15:34:40 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 14.0, whose tasks have all completed, from pool
17/10/12 15:34:40 INFO scheduler.DAGScheduler: ResultStage 14 (count at <console>:33) finished in 1.039 s
17/10/12 15:34:40 INFO scheduler.DAGScheduler: Job 14 finished: count at <console>:33, took 1.089937 s
res14: Long = 10

scala>
```

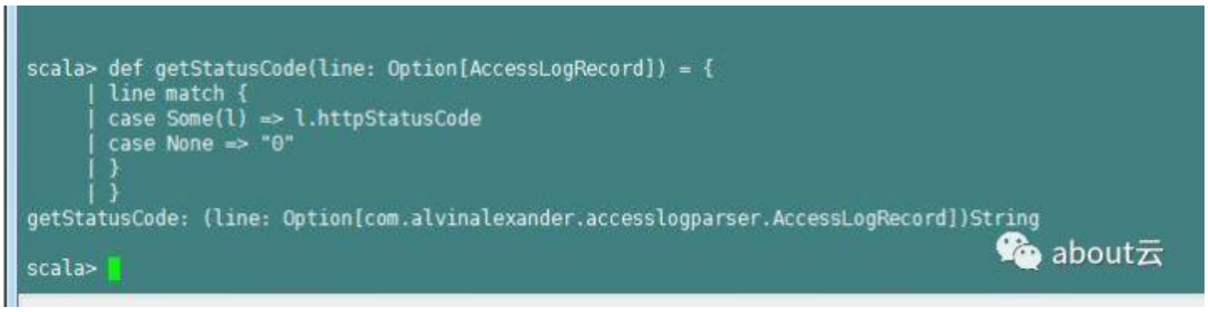
统计网站不能访问网页的数量

首先我们定义一个函数，获取一条记录的 httpStatusCode，也就是返回码

[Scala] 纯文本查看 复制代码

?

```
def getStatusCode(line: Option[AccessLogRecord]) = {
      line match {
            case Some(l) => l.httpStatusCode
            case None => "0"
      }
}
```



```
scala> def getStatusCode(line: Option[AccessLogRecord]) = {
     | line match {
     | case Some(l) => l.httpStatusCode
     | case None => "0"
     | }
     | }
getStatusCode: (line: Option[com.alvinalexander.accesslogparser.AccessLogRecord])String

scala>
```

定义函数之后，我们接着使用

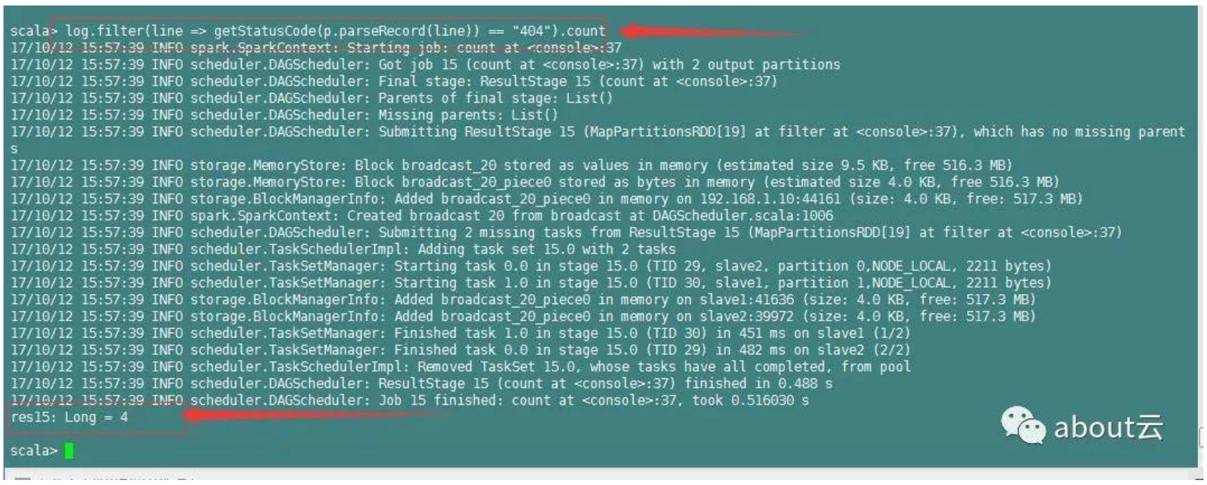
[Bash shell] 纯文本查看 复制代码

?

```
log.filter(line => getStatusCode(p.parseRecord(line)) == "404").count
```

上面的 p 是我们前面定义的对象。

val p = new AccessLogParser，然后调用了 parseRecord 函数。这些都是 jar 包的内容。大家可以详细看看。



```
scala> log.filter(line => getStatusCode(p.parseRecord(line)) == "404").count
17/10/12 15:57:39 INFO spark.SparkContext: Starting job: count at <console>:37
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Got job 15 (count at <console>:37) with 2 output partitions
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Final stage: ResultStage 15 (count at <console>:37)
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Parents of final stage: List()
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Missing parents: List()
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Submitting ResultStage 15 (MapPartitionsRDD[19] at filter at <console>:37), which has no missing parent
s
17/10/12 15:57:39 INFO storage.MemoryStore: Block broadcast_20 stored as values in memory (estimated size 9.5 KB, free 516.3 MB)
17/10/12 15:57:39 INFO storage.MemoryStore: Block broadcast_20_piece0 stored as bytes in memory (estimated size 4.0 KB, free 516.3 MB)
17/10/12 15:57:39 INFO storage.BlockManagerInfo: Added broadcast_20_piece0 in memory on 192.168.1.10:44161 (size: 4.0 KB, free: 517.3 MB)
17/10/12 15:57:39 INFO spark.SparkContext: Created broadcast 20 from broadcast at DAGScheduler.scala:1006
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 15 (MapPartitionsRDD[19] at filter at <console>:37)
17/10/12 15:57:39 INFO scheduler.TaskSchedulerImpl: Adding task set 15.0 with 2 tasks
17/10/12 15:57:39 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 15.0 (TID 29, slave2, partition 0,NODE_LOCAL, 2211 bytes)
17/10/12 15:57:39 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 15.0 (TID 30, slave1, partition 1,NODE_LOCAL, 2211 bytes)
17/10/12 15:57:39 INFO storage.BlockManagerInfo: Added broadcast_20_piece0 in memory on slave1:41636 (size: 4.0 KB, free: 517.3 MB)
17/10/12 15:57:39 INFO storage.BlockManagerInfo: Added broadcast_20_piece0 in memory on slave2:39972 (size: 4.0 KB, free: 517.3 MB)
17/10/12 15:57:39 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 15.0 (TID 30) in 451 ms on slave1 (1/2)
17/10/12 15:57:39 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 15.0 (TID 29) in 482 ms on slave2 (2/2)
17/10/12 15:57:39 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 15.0, whose tasks have all completed, from pool
17/10/12 15:57:39 INFO scheduler.DAGScheduler: ResultStage 15 (count at <console>:37) finished in 0.488 s
17/10/12 15:57:39 INFO scheduler.DAGScheduler: Job 15 finished: count at <console>:37, took 0.516030 s
res15: Long = 4

scala>
```

这样 404 网页的个数就统计出来了。后面我们可以做一些更加复杂的内容

#################

补充说明

1.在统计日志测试的时候，文件一定标准，否则会统计错误，比如日志要换行

2.函数定义

附上所用函数的相关信息

Option and Either

Option 和 Either 都是用来让返回值可以有两个选择

而 Option 是比较简单的版本，两个选择，一定是成功 Some，和失败 None

Option 意味着可能有值 some(x)，也可能没有值(用 None 对象，表示缺失)，典型的例子就是从字典里取值

[Scala] 纯文本查看 复制代码

?

```
val capitals = Map("France" -> "Paris", "Japan" -> "Tokyo")
def show(x: Option[String]) = x match { //Option 类型，可选的 String
```

```
            case Some(s) => s
            case None => "?"
    }
    scala> show(capitals get "France")
    res24: String = Paris
    scala> show(capitals get "North Pole")
    res25: String = ?
```

以前的方式，比如 Java，通过 null 来表示没有取到值，但是有的时候

null 可能作为合法值出现，就需要特殊处理，很麻烦

而 Scala 提供 option 来比较优雅的解决这个问题

Either，更为通用一些，可用自己定义两种选择，直接看个 spark 源码中

的例子，

对于 PutResult 中的 data，有可能是 ByteBuffer 或者 Iterator

而使用的时候，使用 Left 和 Right 来选择到底用哪一个

[Scala] 纯文本查看 复制代码

?

```
private[spark] case class PutResult(size: Long, data: Either[Iterator[_], ByteBuffer])

PutResult(sizeEstimate, Left(values.iterator))PutResult(bytes.limit(), Right(bytes.duplicate()))
```

这里无论 option 或 either 都提高了极好的灵活性，在 Java 中如果要返

回一个有两种可能性的值就比较不那么优雅了

来自：

http://www.cnblogs.com/fxjwind/p/3338829.html

5.实现获取不能访问 url

问题导读

1.在 url 中，如何过滤不需要的内容？
2.如何获取 404 记录并且获取字段？
3.获取不能访问 url 列表的思路是什么？

about 云日志分析实战之清洗日志 4：统计网站相关信息
http://www.aboutyun.com/forum.php?mod=viewthread&tid=22900

上篇文章简单的统计了一些信息，下一步希望找到 404 对应的 url。
思路：
1.获取 request 字段
2.过滤不需要字符
3.实现获取 url，并打印输出

1.创建 getRequest 函数获取 request 字段

[Scala] 纯文本查看 复制代码

?
```scala
// get the `request` field from an access log record
def getRequest(rawAccessLogString: String): Option[String] = {
    val accessLogRecordOption = p.parseRecord(rawAccessLogString)
    accessLogRecordOption match {
        case Some(rec) => Some(rec.request)
        case None => None
    }
}
```

2.创建 extractUriFromRequest 函数

[Scala] 纯文本查看 复制代码

?
```scala
// val request = "GET /foo HTTP/1.0"
```

```
def extractUriFromRequest(requestField: String) = requestField.split(" ")(1)
```

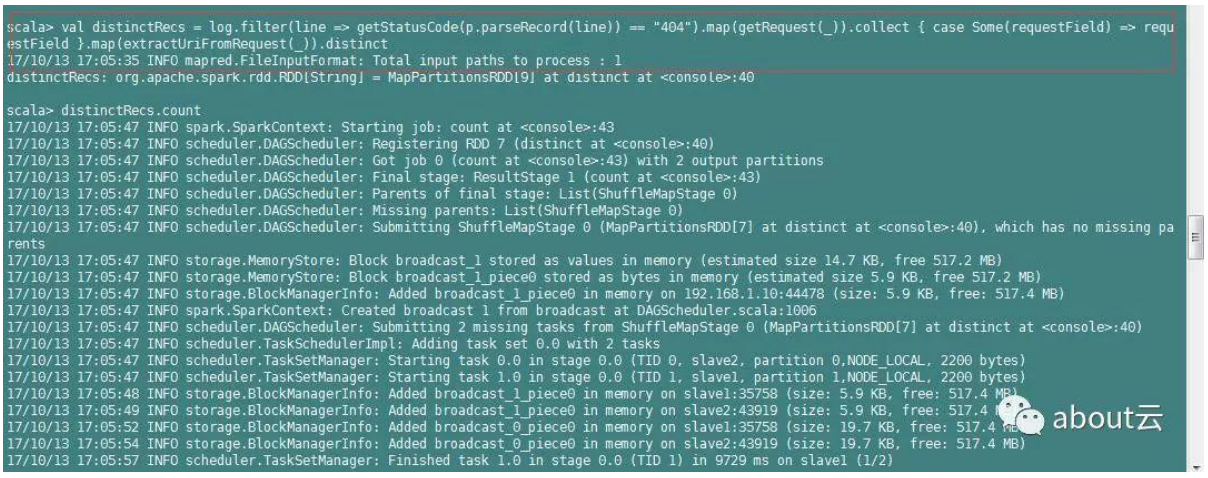
这个目的大家可以猜猜它的作用

获取 404 页面，并且打印出请求的 URL.

[Scala] 纯文本查看 复制代码

?

```
val distinctRecs = log.filter(line => getStatusCode(p.parseRecord(line)) == "404")
                                                              .map(getRequest(_))
                                                              .collect { case
Some(requestField) => requestField }
                                                              .map(extractUriFromRequest(_))
                                                              .distinct
```

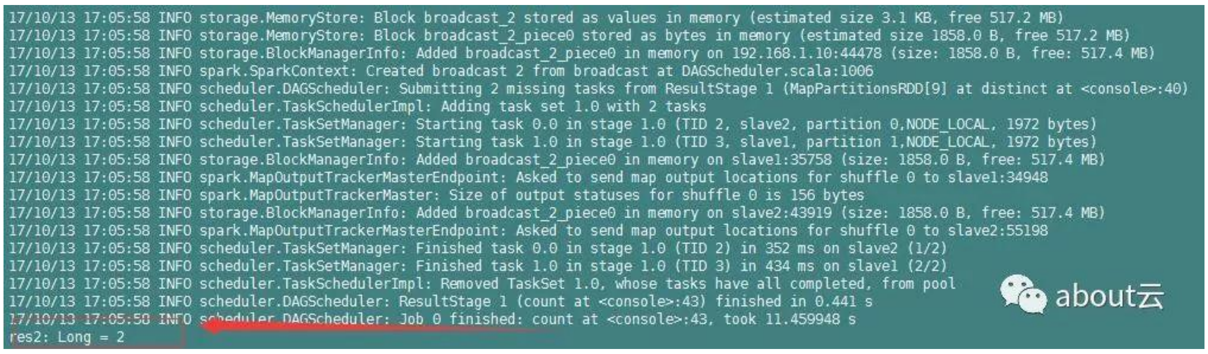


```
scala> val distinctRecs = log.filter(line => getStatusCode(p.parseRecord(line)) == "404").map(getRequest(_)).collect { case Some(requestField) => requ
estField }.map(extractUriFromRequest(_)).distinct
17/10/13 17:05:35 INFO mapred.FileInputFormat: Total input paths to process : 1
distinctRecs: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[9] at distinct at <console>:40

scala> distinctRecs.count
17/10/13 17:05:47 INFO spark.SparkContext: Starting job: count at <console>:43
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Registering RDD 7 (distinct at <console>:40)
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Got job 0 (count at <console>:43) with 2 output partitions
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Final stage: ResultStage 1 (count at <console>:43)
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 0)
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 0)
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 0 (MapPartitionsRDD[7] at distinct at <console>:40), which has no missing pa
rents
17/10/13 17:05:47 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 14.7 KB, free 517.2 MB)
17/10/13 17:05:47 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 5.9 KB, free 517.2 MB)
17/10/13 17:05:47 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on 192.168.1.10:44478 (size: 5.9 KB, free: 517.4 MB)
17/10/13 17:05:47 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1006
17/10/13 17:05:47 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ShuffleMapStage 0 (MapPartitionsRDD[7] at distinct at <console>:40)
17/10/13 17:05:47 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 2 tasks
17/10/13 17:05:47 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, slave2, partition 0,NODE_LOCAL, 2200 bytes)
17/10/13 17:05:47 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, slave1, partition 1,NODE_LOCAL, 2200 bytes)
17/10/13 17:05:48 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on slave1:35758 (size: 5.9 KB, free: 517.4 MB)
17/10/13 17:05:49 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on slave2:43919 (size: 5.9 KB, free: 517.4 MB)
17/10/13 17:05:52 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on slave1:35758 (size: 19.7 KB, free: 517.4 MB)
17/10/13 17:05:54 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on slave2:43919 (size: 19.7 KB, free: 517.4 MB)
17/10/13 17:05:57 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 9729 ms on slave1 (1/2)
```

[Scala] 纯文本查看 复制代码

?

```
distinctRecs.count
```

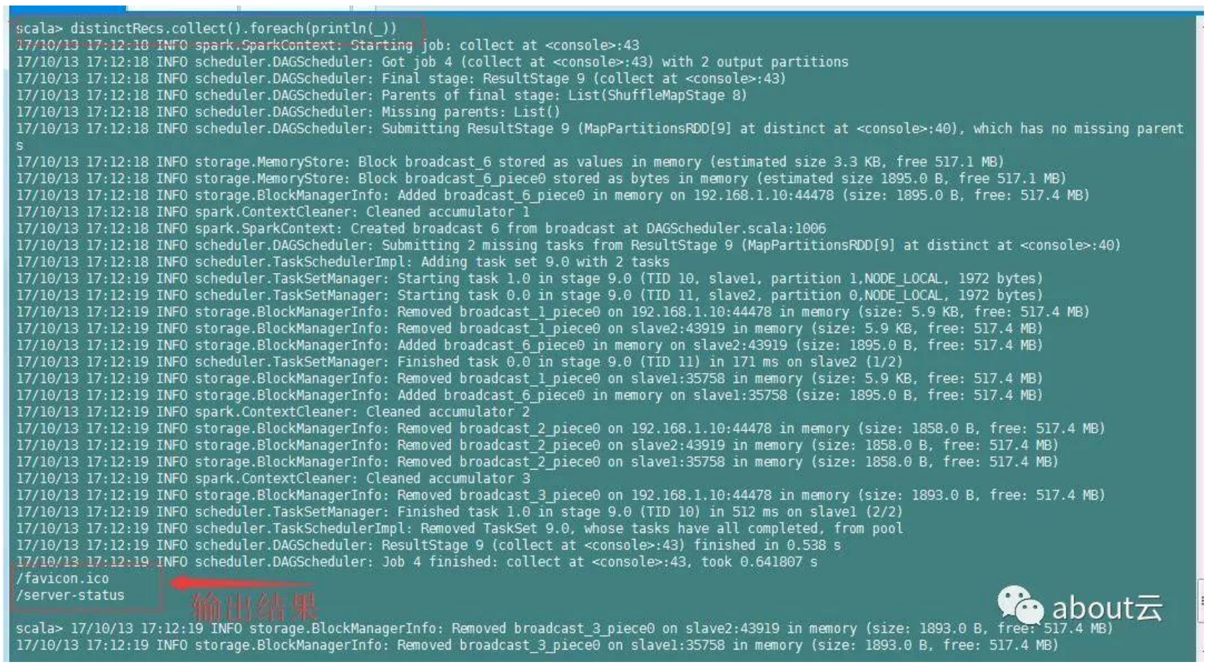


```
17/10/13 17:05:58 INFO storage.MemoryStore: Block broadcast_2 stored as values in memory (estimated size 3.1 KB, free 517.2 MB)
17/10/13 17:05:58 INFO storage.MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated size 1858.0 B, free 517.2 MB)
17/10/13 17:05:58 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on 192.168.1.10:44478 (size: 1858.0 B, free: 517.4 MB)
17/10/13 17:05:58 INFO spark.SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:1006
17/10/13 17:05:58 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 1 (MapPartitionsRDD[9] at distinct at <console>:40)
17/10/13 17:05:58 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 2 tasks
17/10/13 17:05:58 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 2, slave2, partition 0,NODE_LOCAL, 1972 bytes)
17/10/13 17:05:58 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 1.0 (TID 3, slave1, partition 1,NODE_LOCAL, 1972 bytes)
17/10/13 17:05:58 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on slave1:35758 (size: 1858.0 B, free: 517.4 MB)
17/10/13 17:05:58 INFO spark.MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 0 to slave1:34948
17/10/13 17:05:58 INFO spark.MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 156 bytes
17/10/13 17:05:58 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on slave2:43919 (size: 1858.0 B, free: 517.4 MB)
17/10/13 17:05:58 INFO spark.MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 0 to slave2:55198
17/10/13 17:05:58 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 2) in 352 ms on slave2 (1/2)
17/10/13 17:05:58 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 1.0 (TID 3) in 434 ms on slave1 (2/2)
17/10/13 17:05:58 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
17/10/13 17:05:58 INFO scheduler.DAGScheduler: ResultStage 1 (count at <console>:43) finished in 0.441 s
17/10/13 17:05:58 INFO scheduler.DAGScheduler: Job 0 finished: count at <console>:43, took 11.459948 s
res2: Long = 2
```

[Scala] 纯文本查看 复制代码

?

```
distinctRecs.collect().foreach(println(_))
```



```
scala> distinctRecs.collect().foreach(println(_))
17/10/13 17:12:18 INFO spark.SparkContext: Starting job: collect at <console>:43
17/10/13 17:12:18 INFO scheduler.DAGScheduler: Got job 4 (collect at <console>:43) with 2 output partitions
17/10/13 17:12:18 INFO scheduler.DAGScheduler: Final stage: ResultStage 9 (collect at <console>:43)
17/10/13 17:12:18 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 8)
17/10/13 17:12:18 INFO scheduler.DAGScheduler: Missing parents: List()
17/10/13 17:12:18 INFO scheduler.DAGScheduler: Submitting ResultStage 9 (MapPartitionsRDD[9] at distinct at <console>:40), which has no missing parent
s
17/10/13 17:12:18 INFO storage.MemoryStore: Block broadcast_6 stored as values in memory (estimated size 3.3 KB, free 517.1 MB)
17/10/13 17:12:18 INFO storage.MemoryStore: Block broadcast_6_piece0 stored as bytes in memory (estimated size 1895.0 B, free 517.1 MB)
17/10/13 17:12:18 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on 192.168.1.10:44478 (size: 1895.0 B, free: 517.4 MB)
17/10/13 17:12:18 INFO spark.ContextCleaner: Cleaned accumulator 1
17/10/13 17:12:18 INFO spark.SparkContext: Created broadcast 6 from broadcast at DAGScheduler.scala:1006
17/10/13 17:12:18 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 9 (MapPartitionsRDD[9] at distinct at <console>:40)
17/10/13 17:12:18 INFO scheduler.TaskSchedulerImpl: Adding task set 9.0 with 2 tasks
17/10/13 17:12:19 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 9.0 (TID 10, slave1, partition 1,NODE_LOCAL, 1972 bytes)
17/10/13 17:12:19 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 9.0 (TID 11, slave2, partition 0,NODE_LOCAL, 1972 bytes)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_1_piece0 on 192.168.1.10:44478 in memory (size: 5.9 KB, free: 517.4 MB)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_1_piece0 on slave2:43919 in memory (size: 5.9 KB, free: 517.4 MB)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on slave2:43919 (size: 1895.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 9.0 (TID 11) in 171 ms on slave2 (1/2)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_1_piece0 on slave1:35758 in memory (size: 5.9 KB, free: 517.4 MB)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on slave1:35758 (size: 1895.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO spark.ContextCleaner: Cleaned accumulator 2
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_2_piece0 on 192.168.1.10:44478 in memory (size: 1858.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_2_piece0 on slave2:43919 in memory (size: 1858.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_2_piece0 on slave1:35758 in memory (size: 1858.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO spark.ContextCleaner: Cleaned accumulator 3
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_3_piece0 on 192.168.1.10:44478 in memory (size: 1893.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 9.0 (TID 10) in 512 ms on slave1 (2/2)
17/10/13 17:12:19 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
17/10/13 17:12:19 INFO scheduler.DAGScheduler: ResultStage 9 (collect at <console>:43) finished in 0.538 s
17/10/13 17:12:19 INFO scheduler.DAGScheduler: Job 4 finished: collect at <console>:43, took 0.641807 s
/favicon.ico
/server-status

scala> 17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_3_piece0 on slave2:43919 in memory (size: 1893.0 B, free: 517.4 MB)
17/10/13 17:12:19 INFO storage.BlockManagerInfo: Removed broadcast_3_piece0 on slave1:35758 in memory (size: 1893.0 B, free: 517.4 MB)
```

3.获取 url

[Scala] 纯文本查看 复制代码

?

```
val distinctRecs = log.filter(line => getStatusCode(p.parseRecord(line)) == "404")
                                          .map(getRequest(_))
                                          .collect { case Some(requestField) => requestField }
                                          .map(extractUriFromRequest(_))
                                          .distinct
```

通过上面看，其实挺简单。Scala 本身是非常简洁的。

相关说明：

上面看似简单，其实有很多需要说明的

val recs = log.filter(line => getStatusCode(p.parseRecord(line)) == "404").map(getRequest(_))

上面得出 404 对应的 url.getRequest 是上面我们定义的函数
val distinctRecs = log.filter(line => getStatusCode(p.parseRecord(line)) == "404").map(getRequest(_)).distinct
这里多了 distinct 是为了去重，下面是直接打印。
distinctRecs.collect().foreach(println(_))。

对于 extractUriFromRequest，这个主要为过滤我们不想要的内容。如下面，GET 和 HTTP/1.1 都不是我们想要的。所以我们取第二个元素即可。

[Bash shell] 纯文本查看 复制代码

?
```
GET /foo HTTP/1.0
GET /foo HTTP/1.1
```

知识补充：
对于 collect（） 函数，是比较常见的，但是对于下面内容，是什么意思。
collect { case Some(requestField) => requestField }这个作用，类似 map。
##################
更多信息：

在 Scala 中，当我需要对集合的元素进行转换时，自然而然会使用到 map 方法。而当我们在对 tuple 类型的集合或者针对 Map 进行 map 操作时，通常更倾向于在 map 方法中使用 case 语句，这比直接使用_1 与_2 更加可读。例如：

[Scala] 纯文本查看 复制代码

?
```
val languageToCount = Map("Scala" -> 10, "Java" -> 20, "Ruby" -> 5)
languageToCount map { case (_, count) => count + 1 }
```

然而对于上述场景，其实我们也可以使用 collect 方法：

[Scala] 纯文本查看 复制代码

?
```
languageToCount collect { case (_, count) => count + 1 }
```

参考
http://www.jianshu.com/p/fa2ed7ed391e

6.获取 uri 点击量排序并得到最高的 url

问题导读

1.读取日志的过程中，发生异常本文是如何解决的?
2.读取后，如何过滤异常的记录？
3.如何实现统计点击最高的记录？

日志分析实战之清洗日志小实例 5：实现获取不能访问 url
http://www.aboutyun.com/forum.php?mod=viewthread&tid=22911

下面我们开始统计链接的点击量，并做排序。
我们统计记录的时候，为了防止空记录等异常的情况，我们创建一条空记录

[Bash shell] 纯文本查看 复制代码

?

```
val nullObject = AccessLogRecord("", "", "", "", "GET /foo HTTP/1.1", "", "", "", "")
```

下面我们开始找点击量最高的链接。

首先获取我们想要的 uri

[Scala] 纯文本查看 复制代码

?

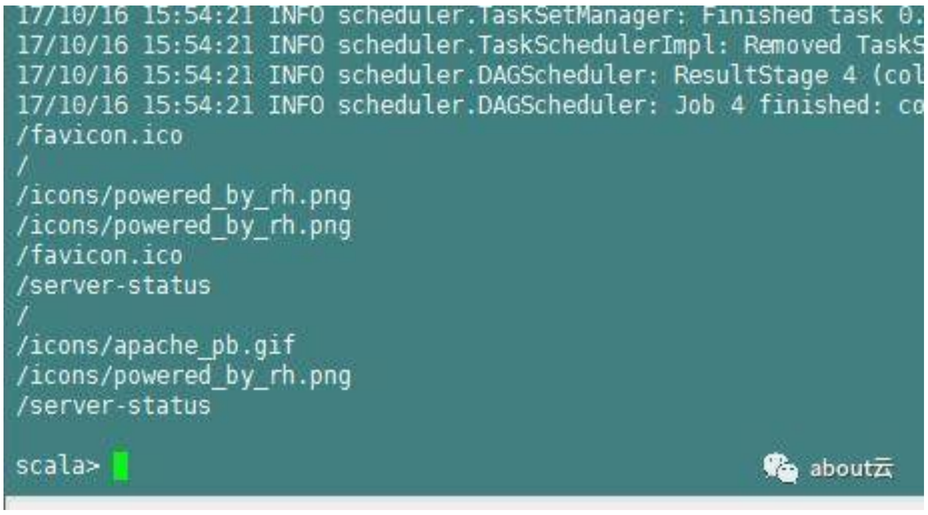
```
val uriCounts = log.map(p.parseRecord(_).getOrElse(nullObject).request)
                                    .map(_.split(" ")(1))
                                    .filter(_ != "/foo")
```

上面的代码做一个简单解释：
p.parseRecord(_)解析记录
p.parseRecord(_).getOrElse(nullObject)如何没有取到值，则使用 nullObject，也就是我们上面定义的对象
p.parseRecord(_).getOrElse(nullObject).request 也就是我们取到 uri
.map(_.split(" ")(1))是取到我们过滤的 url，过滤掉不想要的版本等信息
.filter(_ != "/foo")则是再次过滤掉/foo[也就是空记录]
这样就获取了 uri,然后我们输出

[Scala] 纯文本查看 复制代码

?

```
uriCounts.collect.foreach(print)
```



```
17/10/16 15:54:21 INFO scheduler.TaskSetManager: Finished task 0.
17/10/16 15:54:21 INFO scheduler.TaskSchedulerImpl: Removed TaskS
17/10/16 15:54:21 INFO scheduler.DAGScheduler: ResultStage 4 (col
17/10/16 15:54:21 INFO scheduler.DAGScheduler: Job 4 finished: co
/favicon.ico
/
/icons/powered_by_rh.png
/icons/powered_by_rh.png
/favicon.ico
/server-status
/
/icons/apache_pb.gif
/icons/powered_by_rh.png
/server-status

scala>
```

下面我们统计点击量

[Scala] 纯文本查看 复制代码

?

```
val uriCounts = log.map(p.parseRecord(_).getOrElse(nullObject).request)
                                    .map(_.split(" ")(1))
                                    .map(uri => (uri, 1))
                                    .reduceByKey((a, b) => a + b)
```

rdd 转换为数组

[Scala] 纯文本查看 复制代码

?

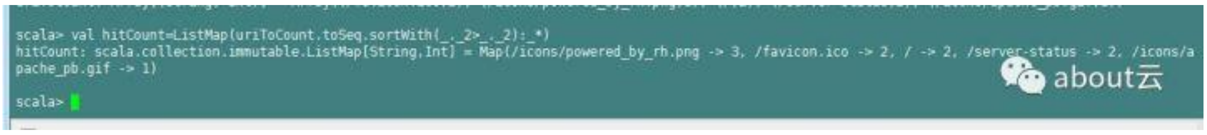
```
val uriToCount = uriCounts.collect
```

数组转换为序列并排序

[Scala] 纯文本查看 复制代码

?

```
import scala.collection.immutable.ListMap
val uriHitCount = ListMap(uriToCount.toSeq.sortWith(_._2 > _._2):_*)
```

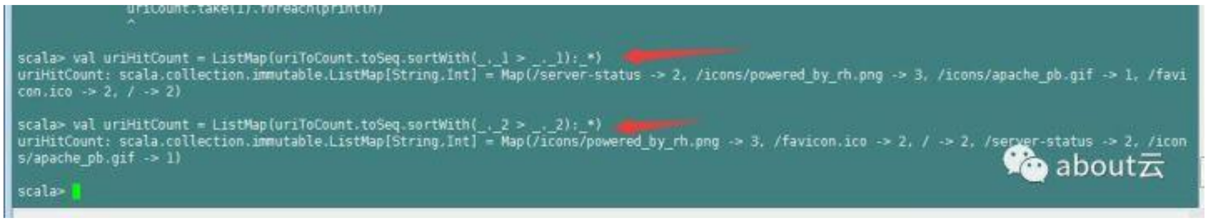


```
scala> val hitCount=ListMap(uriToCount.toSeq.sortWith(_._2>_._2):_*)
hitCount: scala.collection.immutable.ListMap[String,Int] = Map(/icons/powered_by_rh.png -> 3, /favicon.ico -> 2, / -> 2, /server-status -> 2, /icons/a
pache_pb.gif -> 1)

scala> 
```

##############################
这里留下一个问题，如果上面元素不是 2，而是为 sortWith(_._1 > _._1)是对什么排序

[Scala] 纯文本查看 复制代码

?
```
import scala.collection.immutable.ListMap
val uriHitCount = ListMap(uriToCount.toSeq.sortWith(_._1 > _._1):_*)
```
##############################

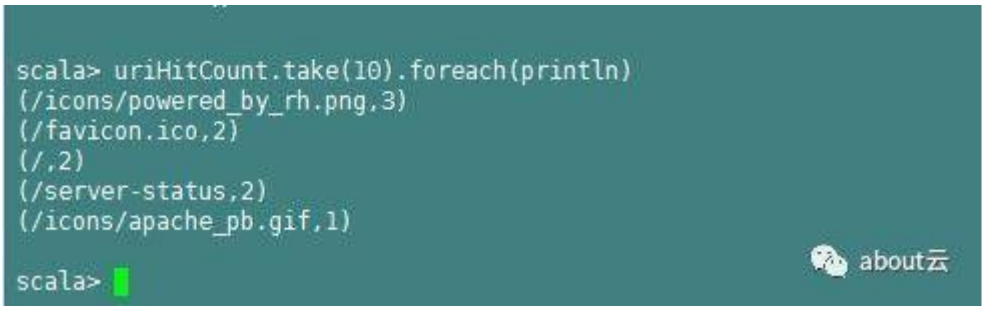


```
              uriCount.take(1).foreach(println)
              ^

scala> val uriHitCount = ListMap(uriToCount.toSeq.sortWith(_._1 > _._1):_*)
uriHitCount: scala.collection.immutable.ListMap[String,Int] = Map(/server-status -> 2, /icons/powered_by_rh.png -> 3, /icons/apache_pb.gif -> 1, /favi
con.ico -> 2, / -> 2)

scala> val uriHitCount = ListMap(uriToCount.toSeq.sortWith(_._2 > _._2):_*)
uriHitCount: scala.collection.immutable.ListMap[String,Int] = Map(/icons/powered_by_rh.png -> 3, /favicon.ico -> 2, / -> 2, /server-status -> 2, /icon
s/apache_pb.gif -> 1)

scala> 
```

输出

[Scala] 纯文本查看 复制代码

?
```
uriHitCount.take(10).foreach(println)
```



```
scala> uriHitCount.take(10).foreach(println)
(/icons/powered_by_rh.png,3)
(/favicon.ico,2)
(/,2)
(/server-status,2)
(/icons/apache_pb.gif,1)

scala> 
```

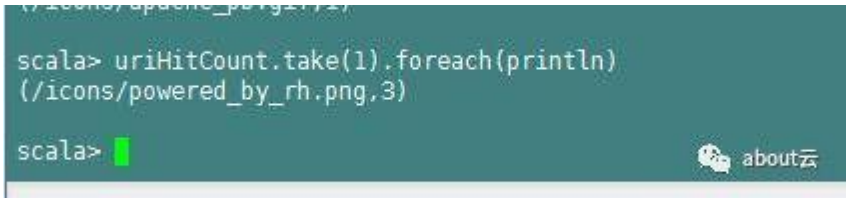
上面便是排序的结果

点击最高的 uri

如果想得出点击最高的 uri

[Scala] 纯文本查看 复制代码

?
```
uriHitCount.take(1).foreach(println)
```



```
scala> uriHitCount.take(1).foreach(println)
(/icons/powered_by_rh.png,3)

scala>
```

~~~~~~

知识补充：
Scala 代码看上去很少，但是内容却是很丰富的。上面用到的相关知识，这里补充，供大家能看懂上面代码

getOrElse:
println(a.get("k1").getOrElse("default")) //根据 key 读取元素，不存在就替换成默认值

在 Spark 中写法是：persons.getOrElse("Spark",1000) //如果 persons 这个 Map 中包含有 Spark，取出它的值，如果没有，值就是 1000。

reduce、reduceByKey

reduce(binary_function)
reduce 将 RDD 中元素前两个传给输入函数，产生一个新的 return 值，新产生的 return 值与 RDD 中下一个元素（第三个元素）组成两个元素，再被传给输入函数，直到最后只有一个值为止。

[Scala] 纯文本查看 复制代码

?
```
val c = sc.parallelize(1 to 10)
c.reduce((x, y) => x + y)//结果 55
```

具体过程，RDD 有 1 2 3 4 5 6 7 8 9 10 个元素，

1+2=3

3+3=6

6+4=10

10+5=15

15+6=21

21+7=28

28+8=36

36+9=45
~~~~~~

45+10=55

reduceByKey(binary_function)
reduceByKey 就是对元素为 KV 对的 RDD 中 Key 相同的元素的 Value 进行 binary_function 的 reduce 操作，因此，Key 相同

的多个元素的值被 reduce 为一个值，然后与原 RDD 中的 Key 组成一个新的 KV 对。

```
val a = sc.parallelize(List((1,2),(1,3),(3,4),(3,6)))
a.reduceByKey((x,y) => x + y).collect
//结果 Array((1,5), (3,10))
```

Seq

Sequence 都有一个预定义的顺序。

```
scala> Seq(1, 1, 2)
res3: Seq[Int] = List(1, 1, 2)
```

(注意返回的结果是一个 List。Seq 是一个 trait；List 是它的一个实现类。Seq 对象是一个工厂对象，正如你所看到

的，它会创建一个 List。)

集合之间可以相互进行转换。

```
def toArray : Array[A]
def toArray [B >: A] (implicit arg0: ClassManifest[B]) : Array[B]
def toBuffer [B >: A] : Buffer[B]
def toIndexedSeq [B >: A] : IndexedSeq[B]
def toIterable : Iterable[A]
def toIterator : Iterator[A]
def toList : List[A]
def toMap [T, U] (implicit ev: <:<[A, (T, U)]) : Map[T, U]
def toSeq : Seq[A]
def toSet [B >: A] : Set[B]
def toStream : Stream[A]
def toString () : String
def toTraversable : Traversable[A]
```

我们可以把一个 Map 转换成一个数组，然后得到一个键值对数组。

```
scala> Map(1 -> 2).toArray
res41: Array[(Int, Int)] = Array((1,2))
```

sortWith

排序操作（sorted, sortWith, sortBy）根据不同的条件对序列元素进行排序。

更多大家可以搜索

后面 about 云会有相关的日志实战视频，会通过 spark sql 等方式来实现。
相关文章链接：
http://www.aboutyun.com/forum.php?mod=group&fid=139
大家感兴趣可加入
日志分析实战群：238250736

更多：
搜索：微信号
wwwaboutyuncom

qq7 群：552029443 spark 学习
qq9 群：634068865 机器学习
qq5 群：432264021 storm 学习