Zeppelin_Demo_final

```
%sh echo $SPARK_HOME
```

FINISHED

```
/data/apps/spark-2.1.0-bin-hadoop2.7/
```

Took 0 sec. Last updated by user2181 at April 18 2017, 3:01:31 PM.

```
%sh spark-submit --version
```

FINISHED

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.1.0
      /_/

Using Scala version 2.11.8, Java HotSpot(TM) 64-Bit Server VM, 1.8.0_92
Branch
Compiled by user jenkins on 2016-12-16T02:04:48Z
Revision
Url
Type --help for more information.
```

Took 4 sec. Last updated by user2181 at April 18 2017, 3:01:38 PM.

```
%sh yarn node -list
```

FINISHED

```
17/04/18 16:35:20 INFO client.RMProxy: Connecting to ResourceManager at c252-109.wrangler.tac
c.utexas.edu/129.114.58.152:8032
Total Nodes:3
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
c252-112.wrangler.tacc.utexas.edu:57753         RUNNING c252-112.wrangler.tacc.utexas.edu:8042
                                   0
c252-110.wrangler.tacc.utexas.edu:44787         RUNNING c252-110.wrangler.tacc.utexas.edu:8042
                                   0
c252-111.wrangler.tacc.utexas.edu:44124         RUNNING c252-111.wrangler.tacc.utexas.edu:8042
                                   0
```

Took 7 sec. Last updated by user2181 at April 18 2017, 4:35:23 PM.

```
%sh
hadoop fs -put /work/00791/xwj/DMS/hadoop-training/stopwords.txt .
hadoop fs -put /work/00791/xwj/DMS/hadoop-training/book.txt .

hadoop fs -ls
```

FINISHED

```
put: `stopwords.txt': File exists
put: `book.txt': File exists
Found 5 items
drwxr-xr-x   - rhuang hadoop          0 2017-04-19 14:30 .sparkStaging
-rw-r--r--   2 rhuang hadoop     400115 2017-04-15 11:14 book.txt
drwxr-xr-x   - rhuang hadoop          0 2017-04-19 15:03 data
drwxr-xr-x   - rhuang hadoop          0 2017-04-19 15:31 output-streaming-py
-rw-r--r--   2 rhuang hadoop       1914 2017-04-15 11:14 stopwords.txt
```

Took 30 sec. Last updated by user2181 at April 19 2017, 3:53:28 PM.

```
%spark
import org.apache.spark.rdd.RDD
val textFile = sc.textFile("book.txt")
```

FINISHED

```
import org.apache.spark.rdd.RDD
textFile: org.apache.spark.rdd.RDD[String] = book.txt MapPartitionsRDD[98] at textFile at <con
sole>:28
```

Took 2 sec. Last updated by user2181 at April 18 2017, 3:06:00 PM.

```
%sh yarn node -list
```

FINISHED

```
17/04/18 15:06:47 INFO client.RMProxy: Connecting to ResourceManager at c252-109.wrangler.tac
c.utexas.edu/129.114.58.152:8032
Total Nodes:3
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
c252-112.wrangler.tacc.utexas.edu:57753         RUNNING c252-112.wrangler.tacc.utexas.edu:8042
                                   4
c252-110.wrangler.tacc.utexas.edu:44787         RUNNING c252-110.wrangler.tacc.utexas.edu:8042
                                   5
c252-111.wrangler.tacc.utexas.edu:44124         RUNNING c252-111.wrangler.tacc.utexas.edu:8042
                                   4
```

Took 7 sec. Last updated by user2181 at April 18 2017, 3:06:50 PM.

```
textFile.count() // Number of items in this RDD%spark
```

FINISHED

```
res0: Long = 7454
```

Took 2 sec. Last updated by user2181 at April 18 2017, 3:07:00 PM.

```
textFile.first() // First item in this RDD
```

FINISHED

```
res1: String = The Project Gutenberg EBook of The Hand of Providence, by J. H. Ward
```

Took 2 sec. Last updated by user2181 at April 18 2017, 3:07:04 PM.

```
textFile.filter(line => line.contains("Providence")).count()
```

FINISHED

```
res2: Long = 16
```

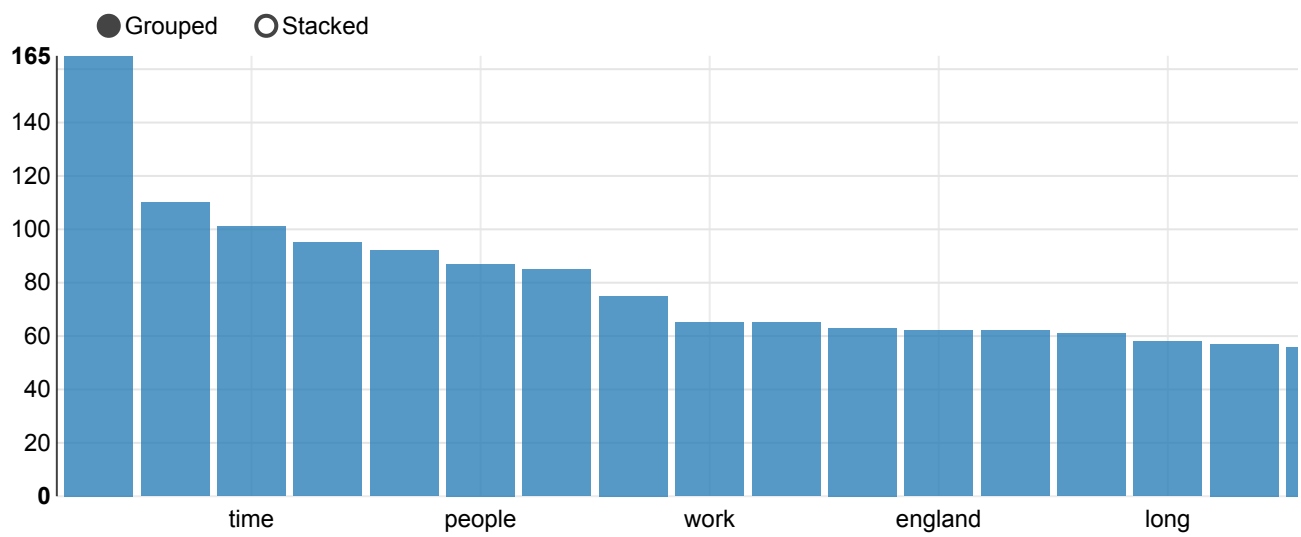
Took 2 sec. Last updated by user2181 at April 18 2017, 3:07:09 PM.

```
val stopWords = sc.textFile("stopwords.txt")
val stopWordSet = stopWords.collect.toSet
val stopWordSetBC = sc.broadcast(stopWordSet)

import org.apache.spark.sql.Row
//textFile.flatMap(_.toLowerCase.split(" ")).subtract(stopWords).take(100)
val wordCounts =  textFile.flatMap(_.toLowerCase.split(" ")).filter( w => !stopWordSetBC.valu

val top50 = wordCounts.sortBy(_._2,ascending=false).map{case (word:String,count:Int) => {word
//top50.mkString("\n")
print("%table Word\t Count\n" + top50.mkString("\n"))
```

FINISHED

```
stopWords: org.apache.spark.rdd.RDD[String] = stopwords.txt MapPartitionsRDD[101] at textFile
 at <console>:28
stopWordSet: scala.collection.immutable.Set[String] = Set(serious, latterly, down, side, moreo
ver, please, ourselves, behind, for, find, further, mill, due, any, wherein, across, twenty, n
ame, this, in, move, itse", have, your, off, once, are, is, his, why, too, among, everyone, sh
ow, empty, already, nobody, less, am, hence, system, than, four, fire, anyhow, three, whereby,
 con, twelve, throughout, but, whether, below, co, mine, becomes, eleven, what, would, althoug
h, elsewhere, another, front, if, hereby, own, neither, bottom, up, etc, so, our, per, therei
n, must, beforehand, keep, do, all, him, had, somehow, re, onto, nor, every, herein, full, bef
ore, afterwards, somewhere, whither, else, namely, us, it, whereupon, two, thence, a, herse",
 sometimes, became, though, within, as, because...
stopWordSetBC: org.apache.spark.broadcast.Broadcast[scala.collection.immutable.Set[String]] =
 Broadcast(46)
import org.apache.spark.sql.Row
wordCounts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[105] at reduceByKey at <cons
ole>:37
top50: Array[String] = Array(great      165, men        110, time       101, years      95, ne
```

settings

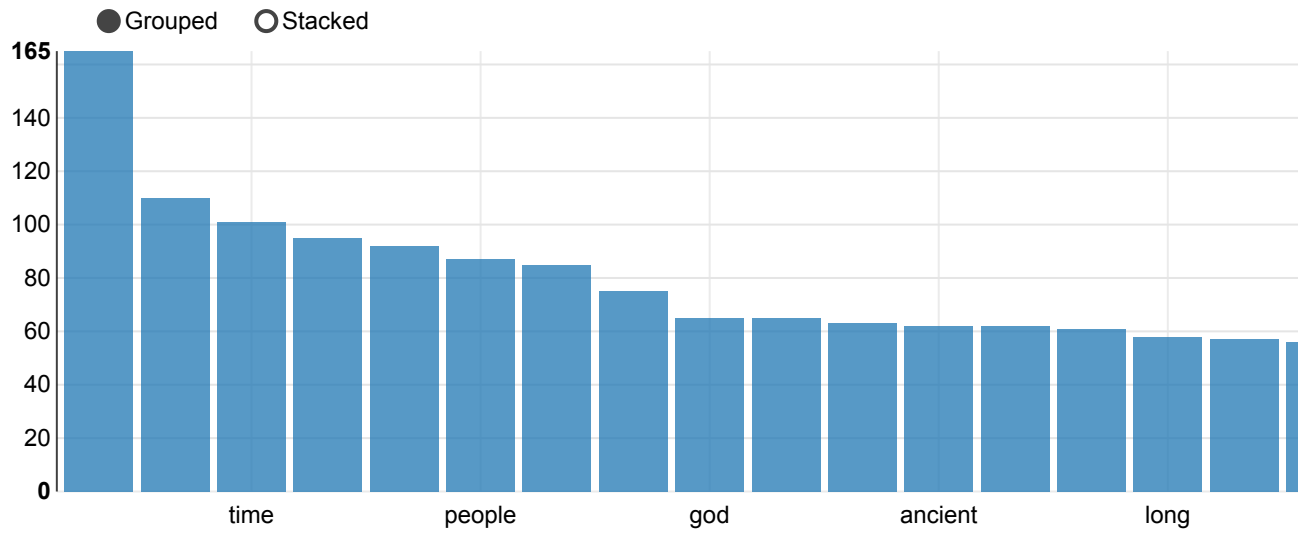


Took 25 sec. Last updated by user2181 at April 18 2017, 3:07:36 PM.

```
%pyspark
textFile = sc.textFile("book.txt")

stopWords = sc.textFile("stopwords.txt")


wordCounts = textFile.flatMap(lambda line: line.lower().split()).subtract(stopWords).map(lamb
top50 = wordCounts.sortBy(lambda a: a[1],ascending=False).map(lambda x: x[0]+ "\t" + str(x[1]

print("%table Word\t Count\n" + "\n".join(top50))
```

FINISHED

settings



Took 3 min 37 sec. Last updated by user2181 at April 19 2017, 2:33:16 PM. (outdated)

```
%sh
module list
```

FINISHED

```
Currently Loaded Modules:
  1) TACC-paths      4) intel/15.0.3   7) TACC
  2) Linux           5) mvapich2/2.1   8) RstatsPackages/3.2.1
  3) cluster-paths   6) cluster        9) Rstats/3.2.1
```

Took 1 sec. Last updated by user2181 at April 18 2017, 3:08:04 PM.

```
%sh
echo $LD_LIBRARY_PATH
# copy and paste the below and set spark.executor.extraLibraryPath in the interpreter spark s
```

FINISHED

```
/opt/apps/intel15/mvapich2_2_1/RstatsPackages/3.2.1/jags-3.4.0/lib64/JAGS/modules-3:/opt/apps/
intel15/mvapich2_2_1/RstatsPackages/3.2.1/jags-3.4.0/lib64:/opt/apps/intel15/mvapich2_2_1/Rsta
tsPackages/3.2.1/proj-4.7.0/lib:/opt/apps/intel15/mvapich2_2_1/RstatsPackages/3.2.1/protobuf-
4.7.1/lib:/opt/apps/intel15/mvapich2_2_1/RstatsPackages/3.2.1/gdal-1.9.2/lib:/opt/apps/intel1
5/mvapich2_2_1/RstatsPackages/3.2.1/nlopt-2.4.2/lib:/opt/apps/gcc/4.9.1/lib64:/opt/apps/gcc/4.
9.1/lib:/opt/apps/intel15/mvapich2_2_1/Rstats/3.2.1/lib64/R/lib:/opt/apps/intel15/mvapich2/2.
1/lib:/opt/apps/intel15/mvapich2/2.1/lib/shared:/opt/apps/intel/15/composer_xe_2015.3.187/mpir
t/lib/intel64:/opt/apps/intel/15/composer_xe_2015.3.187/ipp/lib/intel64:/opt/apps/intel/15/com
```

```
poser_xe_2015.3.187/mkl/lib/intel64:/opt/apps/intel/15/composer_xe_2015.3.187/tbb/lib/intel6
4:/opt/apps/intel/15/composer_xe_2015.3.187/tbb/lib/intel64/gcc4.4:/opt/apps/intel/15/composer
_xe_2015.3.187/compiler/lib/intel64
```

Took 0 sec. Last updated by user2181 at April 19 2017, 3:33:38 PM. (outdated)

```
%spark.r                                                                    FINISHED

#detach("package:dplyr", unload=TRUE)

people <- read.df(sprintf("file:%s/examples/src/main/resources/people.json",Sys.getenv('SPARK_
head(people)
printSchema(people)

# From Hive tables
sql("CREATE TABLE IF NOT EXISTS src (key INT, value STRING)")
input = sprintf("file:%s/examples/src/main/resources/kv1.txt",Sys.getenv('SPARK_HOME'))
sql(sprintf("LOAD DATA LOCAL INPATH '%s' INTO TABLE src",input))
# Queries can be expressed in HiveQL.
results <- sql("FROM src SELECT key, value")
printSchema(results)
# results is now a SparkDataFrame
dim(results)
results

schema <- structType(structField("key", "integer"), structField("value", "string"),
                     structField("key2", "double"))

df1 <- dapply(results, function(x) { x <- cbind(x, x$key * 2) }, schema)
head(df1)
```

```
age    name
1  NA Michael
2  30    Andy
3  19  Justin
root
 |- age: long (nullable = true)
 |- name: string (nullable = true)
SparkDataFrame[]
SparkDataFrame[]
root
 |- key: integer (nullable = true)
 |- value: string (nullable = true)
[1] 26000     2
SparkDataFrame[key:int, value:string]
  key   value key2
1 238 val_238  476
2  86  val_86  172
3 311 val_311  622
```

Took 10 sec. Last updated by user2181 at April 18 2017, 3:08:16 PM.

```
%spark.r                                                                    FINISHED
```

```
### Running SQL Queries from SparkR
people <- read.df(sprintf("file:%s/examples/src/main/resources/people.json",Sys.getenv('SPARK_
head(people)

# Register this SparkDataFrame as a temporary view.
createOrReplaceTempView(people, "people")

# SQL statements can be run by using the sql method
teenagers <- sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
head(teenagers)
```

```
age    name
1  NA Michael
2  30    Andy
3  19  Justin
    name
1 Justin
```

Took 2 sec. Last updated by user2181 at April 18 2017, 3:08:22 PM.

```
%spark.r
# Create the SparkDataFrame
df <- as.DataFrame(faithful)

# Get basic information about the SparkDataFrame
df
dim(df)

# Select only the "eruptions" column
head(select(df, df$eruptions))

# You can also pass in column name as strings
head(select(df, "eruptions"))

# Filter the SparkDataFrame to only retain rows with wait times shorter than 50 mins
head(filter(df, df$waiting < 50))
```

FINISHED

```
SparkDataFrame[eruptions:double, waiting:double]
[1] 272   2
  eruptions
1     3.600
2     1.800
3     3.333
4     2.283
5     4.533
6     2.883
  eruptions
1     3.600
2     1.800
3     3.333
4     2.283
5     4.533
6     2.883
  eruptions waiting
1     1.750      47
```

Took 11 sec. Last updated by user2181 at April 18 2017, 3:08:37 PM.

```
%spark.r
# We use the `n` operator to count the number of times each waiting time appears
head(summarize(groupBy(df, df$waiting), count = n(df$waiting)))

# We can also sort the output from the aggregation to get the most common waiting times
waiting_counts <- summarize(groupBy(df, df$waiting), count = n(df$waiting))
head(arrange(waiting_counts, desc(waiting_counts$count)))
```

FINISHED

```
waiting count
1      70     4
2      67     1
3      69     2
4      88     6
5      49     5
6      64     4
  waiting count
1      78    15
2      83    14
3      81    13
4      77    12
5      82    12
6      79    10
```

Took 11 sec. Last updated by user2181 at April 18 2017, 3:08:54 PM.

```
%spark.r
##### Run a given function on a large dataset using dapply or dapplyCollect
# Convert waiting time from hours to seconds.
# Note that we can apply UDF to DataFrame.
schema <- structType(structField("eruptions", "double"), structField("waiting", "double"),
                     structField("waiting_secs", "double"))
df1 <- dapply(df, function(x) { x <- cbind(x, x$waiting * 60) }, schema)
head(collect(df1))
```

FINISHED

```
eruptions waiting waiting_secs
1     3.600      79         4740
2     1.800      54         3240
3     3.333      74         4440
4     2.283      62         3720
5     4.533      85         5100
6     2.883      55         3300
```

Took 2 sec. Last updated by user2181 at April 18 2017, 3:09:00 PM.

```
%spark.r
#### Applying User-Defined Function
#### Run a given function on a large dataset grouping by input column(s) and using gapply or g
# Determine six waiting times with the largest eruption time in minutes.
schema <- structType(structField("waiting", "double"), structField("max_eruption", "double"))
result <- gapply(
    df,
    "waiting",
    function(key, x) {
```

FINISHED

```
        y <- data.frame(key, max(x$eruptions))
    },
    schema)
head(collect(arrange(result, "max_eruption", decreasing = TRUE)))
```

```
waiting max_eruption
1      96        5.100
2      76        5.067
3      77        5.033
4      88        5.000
5      86        4.933
6      82        4.900
```

Took 6 sec. Last updated by user2181 at April 18 2017, 3:09:16 PM.

```
%spark.r                                                                    FINISHED
### Run local R functions distributed using spark.lapply
# Perform distributed training of multiple models with spark.lapply. Here, we pass
# a read-only list of arguments which specifies family the generalized linear model should be
families <- c("gaussian", "poisson")
train <- function(family) {
  model <- glm(Sepal.Length ~ Sepal.Width + Species, iris, family = family)
  summary(model)
}
# Return a list of model's summaries
model.summaries <- spark.lapply(families, train)

# Print the summary of each model
print(model.summaries)
```

```
[[1]]
Call:
glm(formula = Sepal.Length ~ Sepal.Width + Species, family = family,
    data = iris)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.30711  -0.25713  -0.05325   0.19542   1.41253
Coefficients:
                  Estimate Std. Error t value Pr(&gt;|t|)
(Intercept)         2.2514     0.3698   6.089 9.57e-09 ***
Sepal.Width         0.8036     0.1063   7.557 4.19e-12 ***
Speciesversicolor   1.4587     0.1121  13.012  &lt; 2e-16 ***
Speciesvirginica    1.9468     0.1000  19.465  &lt; 2e-16 ***
Signif. codes:  0 '**' 0.001 '' 0.01 '' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for gaussian family taken to be 0.1918059)
Null deviance: 102.168  on 149  degrees of freedom
Residual deviance:  28.004  on 146  degrees of freedom
AIC: 183.94
```

Took 1 sec. Last updated by user2181 at April 18 2017, 3:09:20 PM.

```
%sh                                                                         FINISHED
# run python work count hadoop example
cd /work/00791/xwj/DMS/hadoop-training/hadoop-streaming-py/
hadoop fs -mkdir data
hadoop fs -put /data/03076/rhuang/training_dataset/book.txt data
```

```
hadoop fs -rm -r  output-streaming-py
export HADOOP_STREAMING=/usr/lib/hadoop-mapreduce/hadoop-streaming.jar
source wordcount.sh
# the last two lines of errors message not important, success anyway
```

```
mkdir: `data': File exists
put: `data/book.txt': File exists
Deleted output-streaming-py
17/04/19 15:29:53 WARN streaming.StreamJob: -file option is deprecated, please use generic opt
ion -files instead.
packageJobJar: [mapper.py, reducer.py] [/usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-cdh5.
8.2.jar] /tmp/streamjob7307957412870837712.jar tmpDir=null
17/04/19 15:29:58 INFO client.RMProxy: Connecting to ResourceManager at c252-109.wrangler.tac
c.utexas.edu/129.114.58.152:8032
17/04/19 15:29:59 INFO client.RMProxy: Connecting to ResourceManager at c252-109.wrangler.tac
c.utexas.edu/129.114.58.152:8032
17/04/19 15:30:03 INFO mapred.FileInputFormat: Total input paths to process : 1
17/04/19 15:30:03 INFO mapreduce.JobSubmitter: number of splits:4
17/04/19 15:30:03 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use
 mapreduce.job.maps
17/04/19 15:30:03 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead,
 use mapreduce.job.reduces
17/04/19 15:30:03 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1492270400644_00
```

```
Paragraph received a SIGTERM
ExitValue: 143
```

Took 1 min 49 sec. Last updated by user2181 at April 19 2017, 3:31:09 PM. (outdated)

```
%sh
hadoop fs -cat output-streaming-py/*|head -n 20
```

FINISHED

```
"A      3
"Araby  1
"Beyond 1
"But    2
"By     1
"Chastity,"     1
"Clothe 1
"Common 1
"Do     1
"Does   1
"For    2
"Given  1
"Golden 1
"Great  2
"Humble 1
"I      8
"Impart 1
"In     2
```

Took 10 sec. Last updated by user2181 at April 19 2017, 3:32:07 PM.

```
%sh
```

READY