

Spark

```
%spark2.spark
import scala.io.Source
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import java.io.PrintWriter;
```

FINISHED

```
import scala.io.Source
import org.apache.hadoop.conf.Configuration
import org.apache.hadoop.fs.FileSystem
import org.apache.hadoop.fs.Path
import java.io.PrintWriter
import scala.io.Source
import org.apache.hadoop.conf.Configuration
import org.apache.hadoop.fs.FileSystem
import org.apache.hadoop.fs.Path
import java.io.PrintWriter
```

Took 1 min 50 sec. Last updated by anonymous at February 26 2020, 3:36:00 PM.

```
%spark2.spark
// sc.stop()
```

FINISHED

Took 2 sec. Last updated by anonymous at February 26 2020, 2:03:03 PM. (outdated)

```
%spark2.spark
```

FINISHED

```
spark.version
```

```
res1: String = 2.3.1.3.0.1.0-187
```

```
res1: String = 2.3.1.3.0.1.0-187
```

Took 1 sec. Last updated by anonymous at February 26 2020, 3:37:31 PM.

```
%spark2.pyspark
```

FINISHED

```
from pyspark import SparkConf, SparkContext
```

Took 4 sec. Last updated by anonymous at February 26 2020, 3:37:56 PM.

```
%sh
cd data/hdfs-files && ls -al
```

FINISHED

```
total 10496
drwxr-xr-x 3 zeppelin hadoop    4096 Feb 26 17:47 .
drwxr-xr-x 4 zeppelin hadoop    4096 Feb 26 17:48 ..
-rw-r--r-- 1 zeppelin hadoop    4826 Feb 26 17:47 access_log_small.txt
-rw-r--r-- 1 zeppelin hadoop 10631573 Feb 26 17:47 access_log.txt
-rw-r--r-- 1 zeppelin hadoop    1373 Feb 26 17:47 CassandraSpark.py
```

Spark

```
-rw-r--r-- 1 zeppelin hadoop      162 Feb 26 17:47 ~$doopCourseTips.docx
-rw-r--r-- 1 zeppelin hadoop      566 Feb 26 17:47 example.conf
-rw-r--r-- 1 zeppelin hadoop      911 Feb 26 17:47 flumelogs.conf
-rw-r--r-- 1 zeppelin hadoop      821 Feb 26 17:47 HBaseExamples.py
-rw-r--r-- 1 zeppelin hadoop      310 Feb 26 17:47 hbase.pig
-rw-r--r-- 1 zeppelin hadoop      199 Feb 26 17:47 job.properties
-rw-r--r-- 1 zeppelin hadoop     1617 Feb 26 17:47 LowestRatedMovieDataFrame.py
-rw-r--r-- 1 zeppelin hadoop     1815 Feb 26 17:47 LowestRatedMovieSpark.py
-rw-r--r-- 1 zeppelin hadoop     1742 Feb 26 17:47 LowestRatedPopularMovieDataFrame.py
-rw-r--r-- 1 zeppelin hadoop     1925 Feb 26 17:47 LowestRatedPopularMovieSpark.py
drwxr-xr-x 2 zeppelin hadoop     4096 Feb 26 17:47 ml-100k
```

Took 4 sec. Last updated by anonymous at February 26 2020, 3:38:04 PM.

```
%pyspark
def loadMovieNames():
    movieNames = {}
    with open("data/hdfs-files/ml-100k/u.item") as f:
        for line in f:
            fields = line.split('|')
            movieNames[int(fields[0])] = fields[1]
    return movieNames
```

FINISHED

Took 0 sec. Last updated by anonymous at February 26 2020, 4:24:55 PM.

```
%pyspark
def parseInput(line):
    fields = line.split()
    return (int(fields[1]), (float(fields[2]), 1.0))
```

FINISHED

Took 0 sec. Last updated by anonymous at February 26 2020, 2:03:30 PM.

```
%spark2.spark
import org.apache.spark._
import org.apache.spark.SparkContext._
import org.apache.log4j._
import org.apache.spark.SparkConf
```

FINISHED

```
import org.apache.spark._
import org.apache.spark.SparkContext._
import org.apache.log4j._
import org.apache.spark.SparkConf
import org.apache.spark._
import org.apache.spark.SparkContext._
import org.apache.log4j._
import org.apache.spark.SparkConf
```

Took 1 sec. Last updated by anonymous at February 26 2020, 3:38:14 PM.

```
%spark2.spark
sc.stop()
```

FINISHED

Took 2 sec. Last updated by anonymous at February 26 2020, 3:39:50 PM.

Spark

```
%spark2.spark
sc.getConf.getAll.foreach(println)

(spark.history.kerberos.keytab,none)
(spark.eventLog.enabled,true)
(spark.history.ui.port,18081)
(spark.driver.memory,512M)
(spark.driver.extraLibraryPath,/usr/hdp/current/hadoop-client/lib/native:/usr/hdp/current/hadoop-client/lib/native/Linux-amd64-64)
(spark.history.fs.cleaner.interval,7d)
(spark.shuffle.io.serverThreads,128)
(spark.executor.extraLibraryPath,/usr/hdp/current/hadoop-client/lib/native:/usr/hdp/current/hadoop-client/lib/native/Linux-amd64-64)
(spark.sql.statistics.fallBackToHdfs,true)
(spark.shuffle.file.buffer,1m)
(spark.history.provider,org.apache.spark.deploy.history.FsHistoryProvider)
(spark.sql.hive.convertMetastoreOrc,true)
(spark.yarn.historyServer.address,sandbox-hdp.hortonworks.com:18081)
(spark.master,local[*])
(spark.submit.deployMode,client)
('Toy Story (1995)', 'GoldenEye (1995)', 'Four Rooms (1995)', 'Get Shorty (1995)', 'Copcat (1995)', 'Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)', 'Twelve Monkeys (1995)', 'Babe (1995)', 'Dead Man Walking (1995)', 'Richard III (1995)')
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:36:36 PM.

```
%spark2.spark
val conf = new SparkConf().setAppName("WorstMoviesZeppelin").setMaster("local[*]")
val sc = new SparkContext(conf)
```

FINISHED

```
conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@15655646
sc: org.apache.spark.SparkContext = org.apache.spark.SparkContext@584e1eb8
conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@15655646
sc: org.apache.spark.SparkContext = org.apache.spark.SparkContext@584e1eb8
```

Took 2 sec. Last updated by anonymous at February 26 2020, 3:41:10 PM.

```
%pyspark
movieNames.values()[:10]
```

FINISHED

```
['Toy Story (1995)', 'GoldenEye (1995)', 'Four Rooms (1995)', 'Get Shorty (1995)', 'Copcat (1995)', 'Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)', 'Twelve Monkeys (1995)', 'Babe (1995)', 'Dead Man Walking (1995)', 'Richard III (1995)']
['Toy Story (1995)', 'GoldenEye (1995)', 'Four Rooms (1995)', 'Get Shorty (1995)', 'Copcat (1995)', 'Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)', 'Twelve Monkeys (1995)', 'Babe (1995)', 'Dead Man Walking (1995)', 'Richard III (1995)']
```

Took 0 sec. Last updated by anonymous at February 26 2020, 2:03:51 PM.

```
%spark2.spark
val lines = sc.textFile("hdfs:///user/maria_dev/ml-100k/u.data")
```

FINISHED

```
lines: org.apache.spark.rdd.RDD[String] = hdfs:///user/maria_dev/ml-100k/u.data MapPartitionsRDD[1]
at textFile at <console>:41
lines: org.apache.spark.rdd.RDD[String] = hdfs:///user/maria_dev/ml-100k/u.data MapPartitionsRDD[1]
at textFile at <console>:41
```

Took 4 sec. Last updated by anonymous at February 26 2020, 3:41:35 PM.

Spark

```
%spark2.spark
lines.take(5).foreach(println)
```

FINISHED

```
196      242      3      881250949
186      302      3      891717742
22       377      1      878887116
244      51       2      880606923
166      346      1      886397596
196      242      3      881250949
186      302      3      891717742
22       377      1      878887116
244      51       2      880606923
166      346      1      886397596
```

Took 7 sec. Last updated by anonymous at February 26 2020, 3:41:46 PM.

```
%spark2.spark
val moviesSplit = lines.map(movieline => (movieline.split("\t")))
```

FINISHED

```
moviesSplit: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[2] at map at <console>:43
moviesSplit: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[2] at map at <console>:43
```

Took 3 sec. Last updated by anonymous at February 26 2020, 3:41:54 PM.

```
%spark2.spark
val movieRating = moviesSplit.map(x => (x(1).toInt , (x(2).toFloat, 1)))
```

FINISHED

```
movieRating: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = MapPartitionsRDD[3] at map at <console>:45
movieRating: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = MapPartitionsRDD[3] at map at <console>:45
```

Took 3 sec. Last updated by anonymous at February 26 2020, 3:42:00 PM.

```
%spark2.spark
movieRating.take(5).foreach(println)
```

FINISHED

```
(242,(3.0,1))
(302,(3.0,1))
(377,(1.0,1))
(51,(2.0,1))
(346,(1.0,1))
(242,(3.0,1))
(302,(3.0,1))
(377,(1.0,1))
(51,(2.0,1))
(346,(1.0,1))
```

Took 2 sec. Last updated by anonymous at February 26 2020, 3:42:06 PM.

```
%spark2.spark
```

FINISHED

Spark

```
testx: org.apache.spark.rdd.RDD[(Int, Int)] = ParallelCollectionRDD[4] at parallelize at <console>:4
```

```
testx: org.apache.spark.rdd.RDD[(Int, Int)] = ParallelCollectionRDD[4] at parallelize at <console>:4
1
```

Took 5 sec. Last updated by anonymous at February 26 2020, 3:42:20 PM.

```
%spark2.spark
val xRating = testx.map(x => (x._1, ((x._2).toFloat, 1)))
```

FINISHED

```
xRating: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = MapPartitionsRDD[5] at map at <console>:43
```

```
xRating: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = MapPartitionsRDD[5] at map at <console>:43
```

Took 2 sec. Last updated by anonymous at February 26 2020, 3:42:25 PM.

```
%spark2.spark
val list_elements = xRating.collect().take(4)
// xRating.take(4).foreach(println(_._1))
```

FINISHED

```
list_elements: Array[(Int, (Float, Int))] = Array((10,(5.0,1)), (20,(5.0,1)), (10,(6.0,1)), (20,(6.0,1)))
```

```
list_elements: Array[(Int, (Float, Int))] = Array((10,(5.0,1)), (20,(5.0,1)), (10,(6.0,1)), (20,(6.0,1)))
```

Took 2 sec. Last updated by anonymous at February 26 2020, 3:42:32 PM.

```
%spark2.spark
for (element <- list_elements
    if element._1 == 10 ) {
    println(element)
}
```

FINISHED

```
(10,(5.0,1))
```

```
(10,(6.0,1))
```

```
(10,(5.0,1))
```

```
(10,(6.0,1))
```

Took 2 sec. Last updated by anonymous at February 26 2020, 3:45:49 PM.

```
%spark2.spark
val testy = xRating.reduceByKey( (x, y) => (x._1 + y._1, x._2 + y._2))
```

FINISHED

```
testy: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = ShuffledRDD[6] at reduceByKey at <console>:45
```

```
testy: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = ShuffledRDD[6] at reduceByKey at <console>:45
```

Took 3 sec. Last updated by anonymous at February 26 2020, 3:46:01 PM.

```
%spark2.spark
testy.take(4).foreach(println)
```

FINISHED

```
(20,(11.0,2))
(10,(11.0,2))
(20,(11.0,2))
(10,(11.0,2))
```

Spark

Took 3 sec. Last updated by anonymous at February 26 2020, 3:46:10 PM.

```
%spark2.spark
val list_elements2 = testy.collect().take(4)
```

FINISHED

```
list_elements2: Array[(Int, (Float, Int))] = Array((20,(11.0,2)), (10,(11.0,2)))
list_elements2: Array[(Int, (Float, Int))] = Array((20,(11.0,2)), (10,(11.0,2)))
```

Took 1 sec. Last updated by anonymous at February 26 2020, 3:46:17 PM.

```
%spark2.spark
for (element <- list_elements2) {
  println(element._2._2)
}
```

FINISHED

```
2
2
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:28:10 PM.

```
%spark2.spark
val averageX = testy.mapValues(totalY => totalY._1 / totalY._2)
```

FINISHED

```
averageX: org.apache.spark.rdd.RDD[(Int, Float)] = MapPartitionsRDD[23] at mapValues at <console>:78
averageX: org.apache.spark.rdd.RDD[(Int, Float)] = MapPartitionsRDD[23] at mapValues at <console>:78
```

Took 5 sec. Last updated by anonymous at February 26 2020, 3:11:36 PM.

```
%spark2.spark
averageX.take(4).foreach(println)
```

FINISHED

```
(20,5.5)
(10,5.5)
(20,5.5)
(10,5.5)
```

Took 5 sec. Last updated by anonymous at February 26 2020, 3:11:44 PM.

```
%spark2.spark
val movieRatingsCounts = movieRating.reduceByKey( (x, y) => (x._1 + y._1 , x._2 + y._2 ))
```

FINISHED

```
movieRatingsCounts: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = ShuffledRDD[7] at reduceByKey at
<console>:47
movieRatingsCounts: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = ShuffledRDD[7] at reduceByKey at
<console>:47
```

Took 2 sec. Last updated by anonymous at February 26 2020, 3:46:29 PM.

```
%spark2.spark
movieRatingsCounts.take(5).foreach(println)
```

FINISHED

Spark

```
(454,(46.0,16))
(1084,(81.0,21))
(1410,(11.0,4))
(772,(150.0,49))
(752,(129.0,39))
(454,(46.0,16))
(1084,(81.0,21))
(1410,(11.0,4))
(772,(150.0,49))
(752,(129.0,39))
```

Took 8 sec. Last updated by anonymous at February 26 2020, 2:56:00 PM.

```
%spark2.spark
val movieRatingsFiltered = movieRatingsCounts.filter(x => x._2._2 > 10)
```

FINISHED

```
movieRatingsFiltered: org.apache.spark.rdd.RDD[(Int, (Float, Int))] = MapPartitionsRDD[59] at filter
at <console>:56
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:31:08 PM.

```
%spark2.spark
val averageRatings = movieRatingsFiltered.mapValues(totalAndCount => totalAndCount._1 / totalAndCou
```

FINISHED

```
averageRatings: org.apache.spark.rdd.RDD[(Int, Float)] = MapPartitionsRDD[60] at mapValues at <conso
le>:58
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:31:13 PM.

```
%spark2.spark
averageRatings.take(4).foreach(println)
```

FINISHED

```
(454,2.875)
(1084,3.857143)
(772,3.0612245)
(752,3.3076923)
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:30:15 PM.

```
%spark2.spark
val sortedMovies = averageRatings.sortBy(x => - x._2)
```

FINISHED

```
sortedMovies: org.apache.spark.rdd.RDD[(Int, Float)] = MapPartitionsRDD[65] at sortBy at <console>:6
0
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:31:21 PM.

```
%spark2.spark
sortedMovies.take(10).foreach(println)
```

FINISHED

Spark

```
(408,4.491071)
(318,4.466443)
(169,4.4661016)
(483,4.45679)
(147,4.47761)
(64,4.4452295)
(603,4.38756)
(12,4.385768)
(50,4.3584905)
(178,4.344)
```

Took 2 sec. Last updated by anonymous at February 26 2020, 5:31:28 PM.

```
%spark2.spark
for (movie <- sortedMovies
  if movie._1 == 50 ) {
  println(movie)
}
```

FINISHED

warning: there was one deprecation warning; re-run with -deprecation for details
(50,4.3584905)

Took 2 sec. Last updated by anonymous at February 26 2020, 5:29:06 PM.

```
%spark2.spark
sortedMovies.lookup(50)
```

FINISHED

res46: Seq[Float] = WrappedArray(4.3584905)

Took 1 sec. Last updated by anonymous at February 26 2020, 5:29:11 PM.

```
%spark2.spark
import java.nio.charset.CodingErrorAction
import scala.io.Codec

implicit val codec = Codec("UTF-8")
codec.onMalformedInput(CodingErrorAction.REPLACE)
codec.onUnmappableCharacter(CodingErrorAction.REPLACE)
```

FINISHED

```
import java.nio.charset.CodingErrorAction
import scala.io.Codec
codec: scala.io.Codec = UTF-8
res23: codec.type = UTF-8
import java.nio.charset.CodingErrorAction
import scala.io.Codec
codec: scala.io.Codec = UTF-8
res23: codec.type = UTF-8
```

Took 1 sec. Last updated by anonymous at February 26 2020, 4:08:27 PM.

```
%spark2.spark
```

FINISHED

```
val movieNamesRaw = sc.textFile("hdfs:///user/maria_dev/ml-100k/u.item")
```

movieNamesRaw: org.apache.spark.rdd.RDD[String] = hdfs:///user/maria_dev/ml-100k/u.item MapPartition

Spark

mark Last updated by anonymous at February 26 2020, 4:11:00 PM.

```
%spark2.spark
movieNamesRaw.take(3).foreach(println)
```

FINISHED

1 Toy Story (1995) 01-Jan-1995 http://us.imdb.com/M/title-exact?Toy%20Story%20(1995) 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 GoldenEye (1995) 01-Jan-1995 http://us.imdb.com/M/title-exact?GoldenEye%20(1995) 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
3 Four Rooms (1995) 01-Jan-1995 http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
1 Toy Story (1995) 01-Jan-1995 http://us.imdb.com/M/title-exact?Toy%20Story%20(1995) 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 GoldenEye (1995) 01-Jan-1995 http://us.imdb.com/M/title-exact?GoldenEye%20(1995) 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
3 Four Rooms (1995) 01-Jan-1995 http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

Took 2 sec. Last updated by anonymous at February 26 2020, 4:11:07 PM.

```
%spark2.spark
import scala.io.Source
import java.nio.charset.CodingErrorAction
import scala.io.Codec
```

FINISHED

```
def loadMovieNames() : Map[Int, String] = {

  // Handle character encoding issues:
  implicit val codec = Codec("UTF-8")
  codec.onMalformedInput(CodingErrorAction.REPLACE)
  codec.onUnmappableCharacter(CodingErrorAction.REPLACE)

  // Create a Map of Ints to Strings, and populate it from u.item.
  var movieNames:Map[Int, String] = Map()

  val lines = Source.fromFile("data/hdfs-files/ml-100k/u.item").getLines()
  for (line <- lines) {
    var fields = line.split('|')
    if (fields.length > 1) {
      movieNames += (fields(0).toInt -> fields(1))
    }
  }

  return movieNames
}
```

```
import scala.io.Source
import java.nio.charset.CodingErrorAction
import scala.io.Codec
loadMovieNames: ()Map[Int,String]
```

Took 2 sec. Last updated by anonymous at February 26 2020, 5:19:44 PM.

Spark

```
%spark2.spark
val nameDict = loadMovieNames()
```

```
nameDict: Map[Int,String] = Map(645 -> Paris Is Burning (1990), 892 -> Flubber (1997), 69 -> Forrest
Gump (1994), 1322 -> Metisse (Café au Lait) (1993), 1665 -> Brother's Kiss, A (1997), 1036 -> Drop
Dead (1991), 1586 -> Lashou shentan (1992), 1501 -> Prisoner of the Mountains (Kavkazsky Plenni
k) (1996), 809 -> Rising Sun (1993), 1337 -> Larger Than Life (1996), 1411 -> Barbarella (1968), 629
-> Victor/Victoria (1982), 1024 -> Mrs. Dalloway (1997), 1469 -> Tom and Huck (1995), 365 -> Powder
(1995), 1369 -> Forbidden Christ, The (Cristo proibito, Il) (1950), 138 -> D3: The Mighty Ducks (199
6), 1190 -> That Old Feeling (1997), 1168 -> Little Buddha (1993), 760 -> Screemers (1995), 101 -> H
eavy Metal (1981), 1454 -> Angel and the Badman (1947), 1633 -> Koldum klaka (Cold Fever) (19
9...
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:20:00 PM.

```
%spark2.spark
nameDict(50)
```

FINISHED

```
res37: String = Star Wars (1977)
```

Took 1 sec. Last updated by anonymous at February 26 2020, 5:20:05 PM.

```
%spark2.spark
val results = sortedMovies.collect()
```

FINISHED

```
results: Array[(Int, Float)] = Array((1500,5.0), (814,5.0), (1122,5.0), (1536,5.0), (1293,5.0), (118
9,5.0), (1201,5.0), (1467,5.0), (1653,5.0), (1599,5.0), (1449,4.625), (1398,4.5), (1642,4.5), (1594,
4.5), (119,4.5), (408,4.491071), (318,4.466443), (169,4.4661016), (483,4.45679), (114,4.447761), (6
4,4.4452295), (603,4.38756), (12,4.385768), (50,4.3584905), (178,4.344), (1191,4.3333335), (513,4.33
33335), (1639,4.3333335), (134,4.292929), (963,4.292683), (427,4.2922373), (357,4.2916665), (98,4.28
97434), (480,4.2849164), (127,4.283293), (285,4.265432), (272,4.262626), (251,4.2608695), (657,4.259
542), (474,4.2525773), (174,4.252381), (479,4.2513967), (1524,4.25), (1064,4.25), (1125,4.25), (313,
4.245714), (511,4.231214), (641,4.212121), (484,4.210145), (172,4.2043595), (515,4.20398), (488,
4....
```

Took 2 sec. Last updated by anonymous at February 26 2020, 5:23:31 PM.

```
%spark2.spark
for (result <- results) {
  val movieId = result._1
  println(nameDict(movieId) , result._2)
}
```

FINISHED

```
(Santa with Muscles (1996),5.0)
(Great Day in Harlem, A (1994),5.0)
(They Made Me a Criminal (1939),5.0)
(Aiqing wansui (1994),5.0)
(Star Kid (1997),5.0)
(Prefontaine (1997),5.0)
(Marlene Dietrich: Shadow and Light (1996) ,5.0)
(Saint of Fort Washington, The (1993),5.0)
(Entertaining Angels: The Dorothy Day Story (1996),5.0)
(Someone Else's America (1995),5.0)
(Pather Panchali (1955),4.625)
(Anna (1996),4.5)
```

(Some Mother's Son (1996),4.5)
(Everest (1998),4.5)
(Maya Lin: A Strong Clear Vision (1994),4.5)
(Close Shave, A (1995),4.491071)
(Some Mother's List (1993),4.466443)

Spark

Took 1 sec. Last updated by anonymous at February 26 2020, 5:31:40 PM.

%spark2.spark

READY

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark

Spark
