**Clairvoyant**

**Intro to Apache Spark Workshop:**

**Exercise Answers**

**Exercise 1 – Running Spark Jobs**

**Question:**

See Exercises Document

**Java Answer:**

N/A

**Scala Answer:**

N/A

**Python Answer:**

N/A

**Exercise 2 – Access Logs**

**Question:**

Analyze the access.log file and calculate the following:

* Count how many times the “/health” URL was hit
* Get all events that occurred on May 19th 2014 and save them to HDFS

Access log file can be found in two locations:

* In the spark-workshop-data.zip file provided, in the “logs” subdirectory
* In HDFS (on the VM provided) at

/user/cloudera/spark-workshop-data/logs/access.log

**Java Answer:**

JavaRDD<String> accessLogs = sc.textFile("/user/cloudera/spark-workshop-data/logs/access.log");  
JavaRDD<String> accessLogsHealth = accessLogs.filter(new Function<String, Boolean>() {  
 public Boolean call(String s) {  
 return s.contains("/health");  
 }  
});  
System.*out*.println(accessLogsHealth.count());  
//5470  
  
JavaRDD<String> accessLogsMay192014 = accessLogs.filter(new Function<String, Boolean>() {  
 public Boolean call(String s) {  
 return s.contains("Mon, 19 May 2014");  
 }  
});  
accessLogsMay192014.saveAsTextFile("/user/cloudera/spark-workshop-output-data/logs/access-logs-5-19-2014");

**Scala Answer:**

**val** accessLogs = sc.textFile("/user/cloudera/spark-workshop-data/logs/access.log")  
accessLogs.filter(\_.contains("/health")).count()  
//res0: Long = 5470  
  
accessLogs.filter(\_.contains("Mon, 19 May 2014")).saveAsTextFile("/user/cloudera/spark-workshop-output-data/logs/access-logs-5-19-2014")

**Python Answer:**

accessLogs = sc.textFile("/user/cloudera/spark-workshop-data/logs/access.log")  
accessLogs.filter(**lambda** x: "/health" **in** x).count()  
#5470  
accessLogs.filter(**lambda** x: "Mon, 19 May 2014" **in** x).saveAsTextFile("/user/cloudera/spark-workshop-output-data/logs/access-logs-5-19-2014")

**Exercise 3 – Joining Datasets**

**Question:**

Using the README.md and CHANGES.txt, find out how many time the word “Spark” shows up in both of the files together by following the bellow steps:

1. Create RDD’s to filter each file for the keyword “Spark”
2. Perform a WordCount on each of the resulting datasets so the results are (K, V) pairs of (word, count)
3. Join the two RDDs

Files can be found in two locations:

* In the spark-workshop-data.zip file provided, in the “spark” subdirectory
* In HDFS (on the VM provided) at

/user/cloudera/spark-workshop-data/spark/

**Java Answer:**

JavaRDD<String> readme = sc.textFile("/user/cloudera/spark-workshop-data/spark/README.md");  
JavaPairRDD<String, Integer> readmeWordCount = readme.flatMap(new FlatMapFunction<String, String>() {  
 public Iterable<String> call(String s) { return Arrays.*asList*(s.split(" ")); }  
}).filter(new Function<String, Boolean>() {  
 public Boolean call(String s) throws Exception {  
 return s.equals("Spark");  
 }  
}).mapToPair(new PairFunction<String, String, Integer>() {  
 public Tuple2<String, Integer> call(String s) {  
 return new Tuple2<String, Integer>(s, 1);  
 }  
}).reduceByKey(new Function2<Integer, Integer, Integer>() {  
 public Integer call(Integer a, Integer b) { return a + b; }  
});  
  
JavaRDD<String> changes = sc.textFile("/user/cloudera/spark-workshop-data/spark/CHANGES.txt");  
JavaPairRDD<String, Integer> changesWordCount = changes.flatMap(new FlatMapFunction<String, String>() {  
 public Iterable<String> call(String s) { return Arrays.*asList*(s.split(" ")); }  
}).filter(new Function<String, Boolean>() {  
 public Boolean call(String s) throws Exception {  
 return s.equals("Spark");  
 }  
}).mapToPair(new PairFunction<String, String, Integer>() {  
 public Tuple2<String, Integer> call(String s) {  
 return new Tuple2<String, Integer>(s, 1);  
 }  
}).reduceByKey(new Function2<Integer, Integer, Integer>() {  
 public Integer call(Integer a, Integer b) { return a + b; }  
});  
  
System.*out*.println(readmeWordCount.join(changesWordCount).collect());

**Scala Answer:**

**val** readme = sc.textFile("/user/cloudera/spark-workshop-data/spark/README.md")  
**val** readmeWordCount = readme.flatMap(line => line.split(" ")).filter(\_.equals("Spark")).map(word => (word, 1)).reduceByKey(\_ + \_)  
  
**val** changes = sc.textFile("/user/cloudera/spark-workshop-data/spark/CHANGES.txt")  
**val** changesWordCount = changes.flatMap(line => line.split(" ")).filter(\_.equals("Spark")).map(word => (word, 1)).reduceByKey(\_ + \_)  
  
readmeWordCount.join(changesWordCount).collect()  
//res0: Array[(String, (Int, Int))] = Array((Spark,(12,101)))

**Python Answer:**

readme = sc.textFile("/user/cloudera/spark-workshop-data/spark/README.md")  
readmeWordCount = readme.flatMap(**lambda** line: line.split(" ")).filter(**lambda** word: word == "Spark").map(**lambda** word: (word, 1)).reduceByKey(**lambda** a, b: a + b)  
  
changes = sc.textFile("/user/cloudera/spark-workshop-data/spark/CHANGES.txt")  
changesWordCount = changes.flatMap(**lambda** line: line.split(" ")).filter(**lambda** word: word == "Spark").map(**lambda** word: (word, 1)).reduceByKey(**lambda** a, b: a + b)  
  
readmeWordCount.join(changesWordCount).collect()  
# [(u'Spark', (12, 101))]

**Exercise 4 – Shared Variables (VERIFY)**

**Question:**

**Java Answer:**

**Scala Answer:**

**Python Answer:**