cloudera®
Ask Bigger Questions

CAP - Developing with Spark and Hadoop:

Homework Assignment Guide for Students

Homework: Persist an RDD

Files and Data Used in This Homework:

Exercise Directory: `$DEV1/exercises/spark-persist`

Data files (HDFS):

`/loudacre/weblogs/*`

`/loudacre/accounts/*`

Job Setup Script:

`$DEV1/spark-stages/SparkStages.pyspark`

`$DEV1/spark-stages/SparkStages.scalaspark`

In this Exercise you will explore the performance effect of caching (that is, persisting to memory) an RDD.

1. Make sure the Spark Shell is running. If it isn't, restart it (in local mode with 2 threads) and paste in the job setup code from the (professor VM only) file or the previous exercise.

2. This time to start the job you are going to perform a slightly different action than last time: count the number of user accounts with a total hit count greater than five:

```
pyspark> accounthits\
  .filter(lambda (firstlast,hitcount): hitcount > 5)\
  .count()
```

```
scala> accounthits.filter(pair => pair._2 > 5).count()
```

3. Cache (persist to memory) the RDD by calling `accounthits.persist()`.

4. In your browser, view the Spark Application UI and select the **Storage** tab. At this point, you have marked your RDD to be persisted, but have not yet

performed an action that would cause it to be materialized and persisted, so you will not yet see any persisted RDDs.

5. In the Spark Shell, execute the count again.

6. View the RDD's `toDebugString`. Notice that the output indicates the persistence level selected.

7. Reload the Storage tab in your browser, and this time note that the RDD you persisted is shown. Click on the RDD ID to see details about partitions and persistence.

8. Click on the **Executors** tab and take note of the amount of memory used and available for your one worker node.

   Note that the classroom environment has a single worker node with a small amount of memory allocated, so you may see that not all of the dataset is actually cached in memory. In the real world, for good performance a cluster will have more nodes, each with more memory, so that more of your active data can be cached.

9. Optional: Set the RDD's persistence level to `StorageLevel.DISK_ONLY` and compare the storage report in the Spark Application Web UI. (Hint: Because you have already persisted the RDD at a different level, you will need to `unpersist()` first before you can set a new level.)

This is the end of the Homework