coreservlets.com – Hadoop Course

MapReduce Features

In this exercise, you will develop custom MapReduce counters and compare them with built-in counters. In addition, you will get a chance to practice placing a file on the Distributed Cache as well utilizing a file from the Distributed Cache in Mapper code.

Approx. Time: 60 minutes

Perform

1. Add counters to the job 'mapRed.features.UniqueCounterTool' that reads tokens from HBase and counts up number of occurrences of each distinct value. The result is persisted to another HBase table. Add counters to mapper and reducer that will keep track of number of input and output records. You will need to modify the following classes:

   `mapRed.features.UniqueCounterMapper.java`

   `mapRed.features.UniqueCounterReducer.java`

   Hint: Use the same group name for the counters.

   You should end up with 4 counters, 2 for mapper and 2 for reducer. *Which built-in framework counters correspond to your custom counters?*

2. `mapRed.features.LineSamplerTool` MapReduce job is implemented to take text input and write out sub-set of lines back to HDFS as a single file. Mapper looks at each line and determines whether that line should be sent to a reducer. The pass-through criteria is based on whether that line contains a well-known token. These well-known tokens come from a file on the Distributed Cache. Reducer simply saves all the lines sent from mappers into a single file.

   Your Job is to:

   a. Enhance `mapRed.features.LineSamplerMapper` to load tokens from the text file on the Distributed Cache.

   b. Execute `mapRed.features.LineSamplerTool` and place a text file with words-to-retain on the Distributed Cache

      - Use `$PLAY_AREA/exercises/mapRed/tokensToRetain.txt`
      - Use `/training/data/hamlet.txt` as an input
      - Use `/training/playArea/LineSampler` as output

Solution

1. The solution can be found in the Solutions project

   ```
   mapRed.features.UniqueCounterMapper.java
   mapRed.features.UniqueCounterReducer.java
   mapRed.features.UniqueCounterTool.java
   ```

   The matching built-in counters can be located under ‘Map-Reduce Framework’ group. The counters are ‘Map Input Records’, ‘Map Output Records’, ‘Reduce Input Records’, and ‘Reduce Output Records’

2. The solution can be found in the Solutions project

   ```
   mapRed.features.LineSamplerMapper
   ```

   at execution time you will need to place tokensToRetain.txt on the Distributed Cache:

   ```
   $ yarn jar $PLAY_AREA/Solutions.jar mapRed.features.LineSamplerTool \
        -files $PLAY_AREA/exercises/mapRed/tokensToRetain.txt \
        /training/data/hamlet.txt \
        /training/playArea/LineSampler
   ```