15-640 Distributed Systems

Lab3 – Map Reduce Engine Application Programmer Manual

-Amey Ghadigaonkar

7/15/2014

**15 - 640 Distributed Systems**

**Lab 3 – Map Reduce Engine**

**-Amey Ghadigaonkar.**

**Index:**

1. Major Assumptions
2. Description of MR library and I/O library
3. Reference manual for generating K-V pairs
4. **Major Assumptions**

I assume that the code written by the Application Programmers is already tested and working.

**If you want to do error handling inside Mapper/Reducer:**

In case you want to print out errors in your MR programs, you should create a file writer in the Map/Reduce jobs that will write to a specific file on the local system so that error messages are known. This is essential since the MR jobs are run in separate processes.

This framework assumes the following:

1. Input files are text files
2. Each record is on a separate line i.e. separate by “\n”

You can specify what happens to each record in the MR files like Hadoop allows. More about generating K-V pairs later in this document.

1. Description of MR library and I/O library and Reference manual for generating K-V pairs

The MR library and IO library have been merged into compile time units for the convenience of the user.

**For MAP jobs:**

* Implement the mapreduce/Mapper class as shown in examples
* You have to specify an intermediate key-type and an intermediate value type. (for example, in MaxTemperatureMapper.java, the intermediate key type is String and the intermediate value type is Integer)
* Write to the context the types mentioned in the above point

e.g. context.write(new String(year), new Integer(airTemperature));

* You can specify any type of key as long as it is Comparable<?>

**For REDUCE jobs:**

* Implement the mapreduce/Reducer class as shown in examples
* You have to specify an intermediate key-type and an intermediate value type. (for example, in MaxTemperatureMapper.java, the intermediate key type is String and the intermediate value type is Integer)
* These intermediate types should be the same as the ones specified in the MAP phase
* In addition to these intermediate values, you also have to specify output key type and output value type.
* Thus, there are 4 types of parameters to this class, but write the output using only the final K-V types

1. Examples
   1. Temperature MR job (in package temperaturetest)

This example has been derived from: <https://www.youtube.com/watch?v=608lAGNUXs0> and <https://github.com/tomwhite/hadoop-book>

Data can be found at : <ftp://ftp.ncdc.noaa.gov/pub/data/noaa>

* 1. Word count MR job (in package wordcount)

This example will count occurences of a word in a text file.

The K-V pairs for the map stage are thus the <String> which is the actual word and <Integer> which is 1 because we have encountered one occurrence of this string.