Information Retrieval

María Carrasco Rodríguez (16874129)
Fabian Lindenberg (74076658)
Lea Voget (45869178)

Assignment 6

University of California Irvine

March 18, 2011

1

Extra Credit

Yes, there are at least two solutions to this problem.

1. **Using one single MapReduce job:**

   Given one line of the input log file, the map phase separates the host name and the corresponding visit count and emits the key-value pair $\langle host, count\rangle$. If necessary, the URL can be parsed using regular expressions to retrieve the host name.

   The reduce task gets one particular host and a *list* of counts (since there can be several entries in the input file with the same host name) as input. By iterating over the list, the counts can be summed up. The reduce task then emits the pair $\langle host, totalCount\rangle$.

   As a result of the MapReduce job we get several output files, one per reducer. We can use the UNIX commands `cat`, `sort`, and `head` to first concatenate all output files, then sort the result, and retrieve the first ten lines. This will yield the ten hosts with the top 10 highest visit counts.

2. **Using two MapReduce jobs and distributed sorting:**

   Assuming very large output files, it might be desirable to distribute the sorting, as well. Instead of using the UNIX programs on one single machine, we would write a second MapReduce job that works on the output of the first one.

   Given one line of an output file, the map task of the second job identifies the visit count as the sorting key. It emits $\langle totalCount, host\rangle$.

   Thanks to the sorting of these intermediate keys performed by the MapReduce framework, the reduce task just acts as an identity function and emits all pairs unchanged.

   As a result of the second job, we obtain output files sorted according to the visit counts. One of these files will contain the top ten hosts with the top 10 highest visit counts.