Universität Zürich[UZH]

Institut für Informatik

Lecture **Distributed Systems**
Fall 2010

Prof. Abraham Bernstein, Ph.D.

ASSIGNMENT II - MAP REDUCE (10 POINTS)

Due date: Nov. 23, 2010, 14:00 (CET)

Rules

- Assumed programming language is *Java*.
- Code that is handed in and does not compile will NOT be graded. So please make sure to test your implementation properly.
- Assignments have to be solved individually. It is ok and also desired to discuss problems with peers, whereas copying code is not. As a result, plagiarism will lead to 0 points for the particular assignment for both parties.
- The due date is a hard deadline. E-mails that arrive after this deadline will be discarded and therefore the contained solution not graded.

All the above rules are final and no matter for further discussions!

The purpose of this assignment is to get familiar and gain practical experience with the MapReduce programming model. MapReduce is used by Google for processing large data sets (terabytes of data).

You will build your assignment on top of the Hadoop software platform. Hadoop is an open-source version of Map Reduce written in Java. For your assignment you are required to use Hadoop as a local node on your machine and solve the two tasks below.

Now your tasks:

1. Write a simple map-reduce program using Hadoop to count the number of words that appear in the following dataset (the dataset consists of multiple documents). Use the stop word list to avoid stop words: http://www.textfixer.com/resources/common-english-words-with-contractions.txt. (**4 Points**)

WORDS DATASET: http://www.ifi.uzh.ch/ddis/fileadmin/teaching/FAll10/DistSystems10/assignments/Shakespeare.zip

2. Write the map and reduce methods to determine the average ratings of movies. The input consists of a series of lines, each containing a movie number, user number, rating, and a timestamp:

UserID::MovieID::Rating::Timestamp

The map should emit movie number and list of rating, and reduce should return for each movie number a list of average rating as Double, and number of ratings as Integer.

(**4 Points**)

MOVIES DATASET: http://www.ifi.uzh.ch/ddis/fileadmin/teaching/FAll10/DistSystems10/assignments/ratings.dat

3. Create a short documentation in which you briefly describe your implementation, such that somebody who has not seen your code can understand what you did. (**2 Points**)

The whole documentation should not be longer than 1-2 pages.

Grading:

Grading will be based on

a) the correctness of your code, i.e. does it solve the given task? and appropriate error handling.

b) readability/structure of your code (including appropriate comments).

c) clarity of your documentation, i.e. does it really describe what you implemented and how well can it be understood by somebody who has not written or read the code.

What to hand in and how:

- Create a zip file named <your_student_id>_<first name>_<last name>_A2.zip (e.g. 1234567_John_Doe_A2.zip).

- This zip file should contain two source code folders (for the two exercises) and also your documentation as doc_a2.pdf file.

- The structure of zip:

    - /src_1

    - /src_2

    - doc_a2.pdf

    - README <= optional see below

For each part please include also the output you got on the two datasets; it can be two text files which contain the frequency of words for the first one and the average ratings for the second one.

IMPORTANT: do not include Hadoop dependencies, unless you are using extra libraries that are not part of the standard requirements to run the assignment. The limit of the zip file should not exceed 500 KB. If it does please contact Cosmin Basca (*basca|at|ifi.uzh.ch*) or Floarea Serban (*serban|at|ifi.uzh.ch*).

In case your code requires any special treatment to compile, you have to enclose a README describing the necessary steps.

Send this zip archive on time via email to:

Cosmin Basca (*basca|at|ifi.uzh.ch*)

Floarea Serban (*serban|at|ifi.uzh.ch*).

The email subject should start with ***[DS 2010]*** .

Helpful reading

Map Reduce paper

http://labs.google.com/papers/mapreduce-osdi04.pdf

http://labs.google.com/papers/mapreduce.html

More about Map Reduce

http://wiki.apache.org/hadoop/HadoopMapReduce

http://en.wikipedia.org/wiki/MapReduce

Map Reduce Hadoop implementation

http://wiki.apache.org/hadoop/

http://hadoop.apache.org/common/docs/r0.21.0/single_node_setup.html#Local

http://www.infosci.cornell.edu/hadoop/mac.html