CSE 7/5337: Information Retrieval and Web Search Spring 2012

Project 2: Indexing (100 points)

Due: 4/1 at midnight per email to mhahsler@lyle.smu.edu

Teams: Individual student or teams of 2.

Tasks:

1. Install Hadoop/MapReduce and set it up for Pseudo-distributed or distributed (e.g. using several PCs or virtual machines on your computer – see http://xebee.xebia.in/2010/10/21/building-a-hadoop-cluster-using-virtualbox/) operation. Move example data to the distributed file system and run an example [0 points].
2. Run your web crawler from project 1 to collect all web pages for the engineering school. **Note:** Make sure that your crawler is polite before you do this! [10 points]
3. Write a MapReduce job to read the compressed web pages (you probably have to separate them into several compressed files so they can be distributed), parse them and create an inverted non-positional index (do the sorting). Use the job on your crawled data. Don't forget to do stemming, normalization and removal of stop words at some point. **Note:** You can find MapReduce code for this online but it is a valuable learning experience if you implement this by yourself starting with the word count example. [30 points]
4. Write code to merge the output of the reduce jobs into a dictionary and posting lists. Use a suitable data structure to represent the dictionary in main memory (optimized for fast search). The postings lists should be stored on disk in a suitable format. E.g. in a single file and the dictionary knows the offset in the file where the list for the term starts. [20 points]
5. Show that 5 simple searches of the form "term1 AND term2" retrieve the correct documents. [10 points]
6. Expand your implementation to create a positional index and test it with 5 positional queries (e.g., Dean AND Orsak within 5 words).[30 points]

Deliverables:

1. Complete code in a compressed archive (zip, tgz, etc)
2. A README file with complete installation, compilation and execution instructions
3. A document with documentation of your code (used software, block or UML diagrams, etc.) and a description for each task.
4. If you worked in a team than you need a document stating who worked on what.