

Project #3: Parallel Programming Using MapReduce/Hadoop

Objectives:

- Understand the Hadoop Infrastructure on the Research Computing Clusters
- Run a simple MapReduce example
- Develop a simple MapReduce program

Due date:

Phase 1: February 17, via email to anda@cse.

Phase 2: March 3rd, via Digital Drop in BlackBoard.

Note: Updates for clarification still possible. Send me any questions you might have.

Phase 1: Understand the Hadoop Infrastructure on the Research Computing Cluster

Steps:

- 1) Request an account with Research Computing by going to this page: <https://rc.usf.edu/signup/signup.php> . List me as your faculty sponsor. My NetID is a i i.
- 2) Configure your environment. For this:
 - a. create a file `~/ .modules` (in your home directory). Insert the following line in your file:

```
module load util/modules
```
 - b. Run the following commands to put hadoop in your environment:

```
module add apps/jdk/1.6.0_10.x86_64 apps/hadoop/0.19.0
module initadd apps/jdk/1.6.0_10.x86_64 apps/hadoop/0.19.0
```
- 3) Unzip and untar the file I sent you by email. If you have not received it, download it from here: <http://www.cse.usf.edu/~anda/PDS/hadoop-job.tar.gz>
- 4) Understand what files you have: a hadoop submission file, the jar for the wordcount MapReduce program and a test file. Try to understand `submit.sh`, as this is the file you will need to modify to run your MapReduce implementation.
- 5) Submit a job with

```
qsub submit.sh
```

 - Please note that there is helpful documentation available on the Research Computing web site. In particular, see: <https://rc.usf.edu/trac/doc/wiki/SoftwarePortal> for a mini Hadoop tutorial. Consult the Hadoop documentation here to better understand the software infrastructure:
<http://public.yahoo.com/gogate/hadoop-tutorial/start-tutorial.html>
http://hadoop.apache.org/core/docs/current/mapred_tutorial.html
<http://hadoop.apache.org/core/docs/current/index.html>
- 6) To see the state of the job queue, run

qstat

- 7) When the job finishes, you will get numerous files and directories as output. Some will tell you how and where the Hadoop daemons run, others will give you the results and logs of your program executing. Try to make sense of this information to answer the following questions:
 - a. What happens if two users will submit MapReduce jobs simultaneously? How will they share the JobTracker?
 - b. Where can you control the number of jobs your MapReduce job is running?
 - c. Time your WordCount MapReduce job for multiple numbers of slaves running on the same file. How does it vary?
 - d. Where is the output of the MapReduce job written? Where is it specified where to be written?
 - e. The Hadoop tutorial speaks about a configuration file in XML being submitted together with the MapReduce code. Where is that?
- 8) The following questions invite you to think of MapReduce rather than its implementation (Hadoop):
 - a. Give an example of a MapReduce problem not listed in the MapReduce paper or in the Hadoop reading. In your example, what are the map and reduce functions (including inputs and outputs)?
 - b. What part of the MapReduce implementation do you find most interesting? Why?
 - c. Give an example of a distributable problem that should not be solved with MapReduce. What are the limitations of MapReduce that make it ill-suited for your task?

Phase 2: Implement Distributed Sort with MapReduce

Write a MapReduce job to sort words stored in a huge file. You can take as input any large text files, or try something more involved if you feel inspired by the Sort Benchmark Challenge (<http://www.hpl.hp.com/hosted/sortbenchmark/>).

Deliverables:

- A document that answers the questions in Phase 1. There will be no grade assigned to this portion of the assignment, it is a best effort from my part to prevent you from postponing thinking of this problem until it's too late. You are welcome and encouraged to work with other colleagues BUT you are responsible for understanding the system yourself.
- A tar file with your MapReduce code with your solution to Phase 2, your submit.sh to help me run it, a test file, an output sample, and a brief document that describes your solution.
- I will grade your submission during a meeting with each of you when you'll give me a demo of your program and be able to answer my questions on your implementation and the running of the Hadoop.