

CARNEGIE MELLON UNIVERSITY

15640: Project 3 Administrator Guide

Building a Map-Reduce facility

Yang Sun (Andrew ID: yksun) | Yuan Gu (Andrew ID: yuangu)

4/13/2013

Table of Contents

Deployment 2

Start Processes 2

Management Tools..... 3

Compile and Preset 3

Deployment

As suggested in the project description, we implemented the configuration file to help administrator to deploy the facility. The configuration file must at least contain the following properties:

- **max_maps**: This property specifies the maximum number of maps a task can take. The value must be a positive integer.
- **max_reduces**: This property specifies the maximum number of reduces a task can take. The value must be a positive integer.
- **master**: This property specifies the host address of the master controller. The value must be a valid IP address or anything that DNS can locate.
- **master_port**: This property specifies the host port of the master controller. The value must be an available port.
- **clients**: This property specifies all clients ID, delimited by comma (without any spaces). The client ID must match with the arguments provided when the administrator starts the client processes. An example of this property would be **"clients=client_1,client_2,client_3"**.
- **<clientID>**: After specifying all involved clients ID, the administrator must define the host address of this client. The format would be something like **"client_1=unix13.andrew.cmu.edu"**. The number of such property must match with the number of clients defined in "clients".
- **<clientID_port>**: After specifying all involved clients ID, the administrator must define the host port of this client. The format would be something like **"client_1_port=10087"**. The number of such property must match with the number of clients defined in "clients".

The example configuration file is given in **Configuration/config.properties**. The administrator may choose to modify on top of it. Any time the administrator modifies this file, the MapReduce facility has to restart to make the changes effective.

Start Processes

There are two types of processes need to start:

- **Master**: To start the master process, the administrator must call **java Master.Master <Properties File>**, where **<Properties File>** is the path to the property file described in the above section.
- **Client**: To start the client process, the administrator must call **java Client.Client <Client ID> <Properties File>**, where **<Client ID>** is the identification of the current client host, it must match with that the administrator defined in the property file.

Management Tools

After both the master and client processes start, the administrator may issue the following commands:

- **start:** This will turn on the background MapReduce facility. All the following commands except quit require the facility to be running. The Screen should print “Start Successfully” if the facility starts without any problem.
- **stop:** This will shut down the MapReduce facility. The administrator can start the facility again after it shuts down. Similarly, the Screen should print “Stop Successfully” if the facility stops without any problem. Sometimes, the process may take up to 5 seconds to finish, because it waits for the status checker to finish the one cycle of querying.
- **monitor:** This will shows the current running tasks on the current host. Note that the master controller will always return 0 due to the design that it doesn’t get involved to execute tasks.
- **submit <jarFile> <MapClass> <InputFile> <ReduceClass> <OutputFolder>:** This will allow the administrator to submit a MapReduce job to the facility. The <jarFile> is the jar achieve file that consists of the specified <MapClass> and <ReduceClass>. <MapClass> and <ReduceClass> are just the class name so that the facility can locate them in the jar file. The <InputFile> is the file path the map should process. Note that the input file has to be the pre-processed fixed length input records. For details about how to pre-process it, please refer to the example code in Examples.StringOutputWrite. The <OutputFolder> is the folder path where the reducer should dump the result to. Note that the reducer output file has to be the post-processed so that it can be human-readable. For details about how to post-process it, please refer to the example code in Examples.StringInputRead.
- **quit:** This will end the current process. If the current host is master, then the entire process will end.

Compile and Preset

We have provided a Makefile that ease the process to compile and preset the environment. For the submission package, the Makefile contains the following options:

- **make all:** It will compile all the source codes.
- **make master:** It will start the master process, the master must run on **unix1.andrew.cmu.edu** according to the config.properties provided.
- **make client_1:** It will start the client_1 process, the client_1 must run on **unix13.andrew.cmu.edu** according to the config.properties provided.
- **make client_2:** It will start the client_2 process, the client_2 must run on **unix11.andrew.cmu.edu** according to the config.properties provided.
- **make client_3:** It will start the client_3 process, the client_3 must run on **unix14.andrew.cmu.edu** according to the config.properties provided.

- **make clean:** It will clear out all .class files.