

COP5615 Fall 2013

Project 4 Distributed Debugging (Phase I)

Team Members:

1. Monica Mavoori(UFID – 46309228)
2. Rohini Kar(UFID – 67308301)

Command line arguments given:

numNodes topology algorithm

where ,

numNodes is the number of nodes in the system

topology can be either of {full, 2DGrid}

algorithm can be {Gossip}

As an example,

Run the program as following

```
scalac Project4.scala
```

```
scala Project4 10 2DGrid gossip
```

where number of nodes is 10, topology is 2DGrid and algorithm is gossip.

Working Of Code:

A part of the 2nd project (Gossip simulator) has been reused for performing the logging here in the 4th project. The code, on running, produces a log file for the Master node and a log file, one for each of the worker actors. Topologies for “Full” and “2DGrid” ones and algorithm for “Gossip” have been used to account for different scenarios that may arise. A node has also been selected randomly to behave as a failure node. Logging starts with the Master node and logs are produced for the execution of each of the actors. Conditions are checked if the user provides inaccurate arguments while running the program,

for example, providing the wrong number of arguments or providing correct numbers of inappropriate arguments. For these situations, the program does not run and the logs can be analysed for the causes.

The logs are generated in folder Logs. This folder must be present in the workspace along with the folders src, bin, so that program runs without any error of not finding folder Logs.

Integration of logging with actors

At first, the number of arguments provided by the user is verified and appropriate information provided back to him. Logging starts when the Master is first initiated. The master actor checks if the number of nodes is a perfect square in case of “2DGrid” topology and also checks if proper arguments have been provided while running the program. A random failure node is selected and made sure that this node does not participate in the gossip simulator. All these execution information of the master actor are simultaneously written down into the log “Boss.txt”. The Master node selects a random node for starting the gossip algorithm and it is made sure that this node is not the failure node because otherwise the program will not work. This condition, if it is satisfied or not, can also be verified using the log of the Master actor.

In the log file for each of the worker actors, information is stored regarding all the gossip messages sent by which actor to which other actor so as to help analyse the entire execution in case of an undesired circumstance.

Convergence for the algorithm is checked in the Master actor and appropriate messages printed in its log.

For easy and accurate debugging, a “Datestamp” and “Time stamp” is associated with each and every message written in the logs.

Log files and interpretation

There is one log file called “Boss” for the Master actor and logs called “node0 log”, “node1 log”.....”noden log”, one for each of the worker actors. The “Boss” log file has 4 columns “**DATESTAMP**”, “**TIMESTAMP**”, “**MESSAGE**” and “**NODE SELECTED**”. Each worker actor log file consists of 5 columns “**DATESTAMP**”, “**TIMESTAMP**”, “**MESSAGE**”, “**SOURCE NODE**” and “**DESTINATION NODE**”. The “**DATESTAMP**” and “**TIMESTAMP**” fields represent the date and time on which the particular

message in the “**MESSAGE**” field was written into the log respectively, and the “**MESSAGE**” field contains the appropriate program execution message for the user. The “**NODE SELECTED**” field in the “Boss” log file denotes the node under consideration by the Master actor and the “**SOURCE NODE**” and “**DESTINATION NODE**” fields in the worker actor log files represent the source from which the gossip message is being sent and the destination actor which is supposed to receive it respectively.