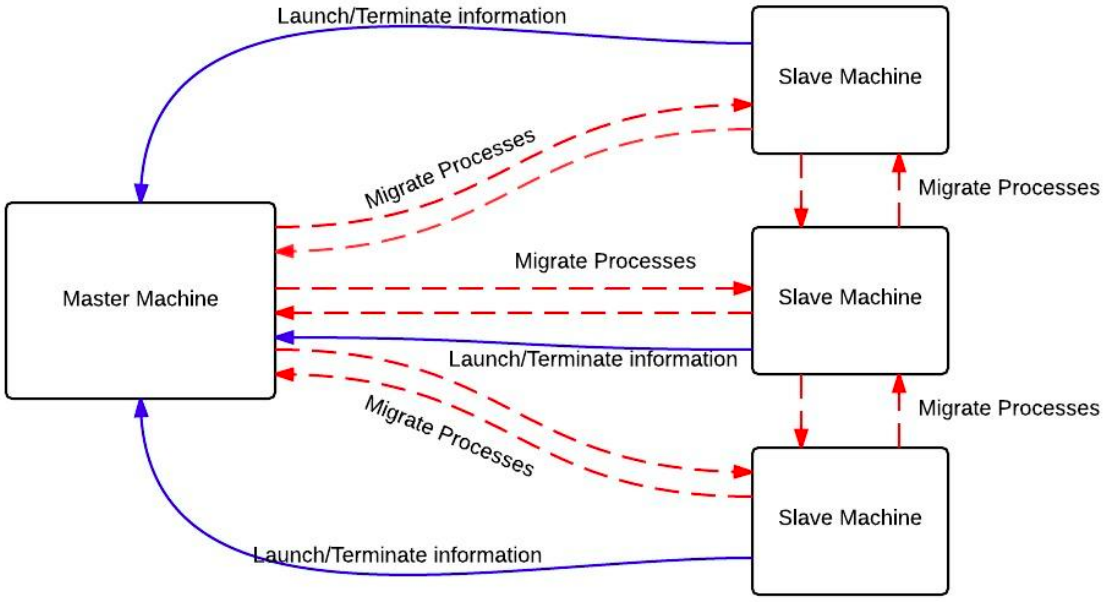
# 15-640 Distributed System Project1 Report

Name: Zhengxiong Zhang
AndrewID: zhengxiz

(My teammate dropped the course. So, I need to find another teammate before project 2...)

**1. Clearly explain your design and illustrate its use, being sure to highlight any special features or abilities**




1.1 Introduction

My design for project1 can migrate processes between all machines (including both master machine and slave machine), using the hostname. My design support transactional I/O, e.g. all processes can continually read and write files on AFS. In my framework, each machine (master or slave) have its own tiny shell. So, you can input command on each machine to launch/migrate processes.

1.2 Framework

I built one java project for both master and slave machine. If you want to deploy a master machine, then you can input argument “master”. If you want to deploy a slave machine, you should input arguments “slave <hostname of master machine>”. There are three parts of this project: MigratableProcesses, ProcessManager and Transactional I/O. In migratable processes, I built MigratableProcess interface, and designed two classes (ReplaceProcess and WordCountProcess) using this interface. In ProcessManager, I designed a process manager monitor. So, we can input command to launch process, migrate process and list all processes on all machines. In Transactional I/O, I wrote two stream classes to facilitate migrating processes with open files.

1.3 Abilities

help : Print help information. Describe all the commands.
des : Display all the machines with all running processes (ID).
ls : Show all the running processes (ID) in this machine.
quit : Quit this machine.
test : Run the test case.
run <processName> <args> : Run a process in the machine.
mig <processID> <hostname> : Migrage the process with <processID> to another machine(hostname).

**2. Describe the portions of the design that are correctly implemented, that have bugs, and that remain unimplemented.**

2.1 Implemented Design

(1) Both master and slave machines have their own tiny shells. They can input command to launch, migrate and list processes.
(2) Master machine can list all processes on master and slave machines. Slave machine can only list all processes on its own.
(3) Implement two migratable classes using MigratableProcess interface (ReplaceProcess and WordCountProcess). They can continually deal with files after migrating.

2.2 Remain Unimplemented Design

(1) Implement a Message Class. Each time machines migrate processes or transfer launching/terminating information, we can only send a message object containing everything we need.
(2) Master machine can control all slave machines. Now, master can only see slaves' processes list, but master cannot control slaves. I want to let master send messages controlling slaves to launch, migrate and terminate processes.

**3. Tell us how to cleanly build, deploy, and run your project.**

I write shell scripts to do it.

* : You should change parameters in the file.

(1) * deploy.sh : deploy my project on AFS **private/MigratableProcess** folder. Also, if you want to test my project, you should **change** the **user** and **host parameter** in the **deploy.sh**.

(2) runMaster.sh : launch master process on the machine.

(3) * runSlave.sh : launch slave process on the machine. You should **change** the **master hostname** parameter in the runSlave.sh.

**4. Highlight any dependencies and software or system requirements.**

The server use port 12323.

There are three required packages in jar folder: com.google.common_1.0.0.201004262004.jar, javassist-3.8.0.GA.jar, and reflections-0.9.9-RC1-uberjar.jar.

**5. Tell us how to run and test your framework with your two examples.**

(1) Change **andrewID** in **deploy.sh**. Then, run **sh deploy.sh**. (copy the project to your own **private/MigratableProcess** folder)

(2) You should deploy master machine first, and then deploy some slave machines. You should run runMaster.sh on a machine (hostname: ghc01.ghc.andrew.cmu.edu). Then, on some other machines, you should input **master hostname (for example** ghc01.ghc.andrew.cmu.edu**)** in **runSlave.sh**.

(3) You can just input **test** in master machine tiny shell command. Then, you can see some simple tests.

(4) Or, you can do some tests by yourself. Input "help" for help.

mig <processID> <hostname> : Migrage the process with <processID> to another machine(hostname).
run <processName> <args>  : Run a process in the machine.

run GrepProcess <queryString> <inputFile> <outputFile>
e.g. run GrepProcess 5 ./testCase/grepTest ./testCase/grepTest.out

run ReplaceProcess <regexString> <replacementString> <inputFile> <outputFile>
e.g. run ReplaceProcess 0 32123 ./testCase/replaceTest ./testCase/replaceTest.out

run WordCountProcess <inputFile> <outputFile>
e.g. run WordCountProcess ./testCase/wordTest ./testCase/wordTest.out