Readme File: Set Up and Executing the Resource Manager

1. Introduction to Files

The following files are available at GitHub

1. Project.java
2. ResourceUsage.java
3. Script for stress test (included in this report)

Important Note: All the files need to be placed in one location, as ResourceUsage.java file is being used in Project.java

Project.java holds the code for

a. Creation of VM i.e. virtual machine
b. Placement of VM
c. Migration of the virtual machine (dynamically according to algorithm)

***Note**: Appropriate comments are added to the code for better understanding*

ResourceUsage.java
Dynamically Decide which host to be picked for migration. it returns the host ID

Script for stress. Need to be run manually on every VM.

```
apt-get update && apt-get install stress && stress --cpu 1 --io 4 --vm 2 --vm-bytes 128M
```

***Note**: User guide can be viewed by referring to Appendix 'C' of the project report.*

2. Access the Tesbed

The testbed can be accessed through the gateway using your School of Computing/University username and password:

ssh username@csgate1.leeds.ac.uk

or simply:

ssh csgate1

on the school of computing network.

3. Environment Setup

The example code provided must be compiled and run on **csgate1**.

Assuming you are logged into **csgate1**, to use the OpenNebula Cloud API for Java in your Java program, you need to include the following **org.opennebula.client.jar** file and three **xml-rpc** libraries into your CLASSPATH.

These files are available in **/home/cscloud1_data_a/storage/scskd/nebula/jars/**

- Amend **.bashrc** file (available in your home directory) by inserting at the **end** of the file the following:

  ```
  export
  CLASSPATH=${CLASSPATH}:/home/cscloud1_data_a/storage/scskd/nebula/jars/org.opennebula.client.jar:/home/cscloud1_data_a/storage/scskd/nebula/jars/ws-commons-util-1.0.2.jar:/home/cscloud1_data_a/storage/scskd/nebula/jars/xmlrpc-client-3.1.2.jar:/home/cscloud1_data_a/storage/scskd/nebula/jars/xmlrpc-common-3.1.2.jar
  ```
- Save the file
- Source it by typing **source .bashrc**

Environment is now setup and you should be able to compile Java code.

4. Executing the Code

To compile Java code, use **javac**, e.g.

javac Project.java (this will produce Project.class)

To run a Java class, use **java**, e.g.

java Project