

Configuration Manual

Student Name: David Tynan

Student Number: 20153104

Project Title: Latency Assessment on Inclusion of a FEC Orchestrator Capable of Invoking Serverless Functions

Instructions

Running of the project source code requires installation of Java. Java 1.8 specifically was used throughout source code development so this version should be used if possible. To avoid JVM memory issues it is important to increase memory allocation size for device Java Virtual Machine by adjusting the Java VM argument options to include `-Xmx1400m`, which is further detailed in the stages below:

1. Install the Eclipse Integrated Development Environment from:
<https://www.eclipse.org/downloads/packages/>
2. Retrieve the project source code and import it into an Eclipse workspace either by:
 - a. Downloading the submitted zipped source code folder from Moodle and extract its content when download has completed. From within an Eclipse Workspace select File -> Import -> Maven -> Existing Maven Projects -> Navigate to the folder containing the extracted source code and select to import it.
 - b. Open the Eclipse workspace directory within a CMD/Terminal window and type in `git clone https://github.com/djtyan/researchProject`, excluding the quotation marks. This command will retrieve the source code from the GitHub repository it is hosted in.
3. With the source code imported to the Eclipse workspace, right click the pom.xml file in the root directory and select Maven -> Update Project. This will ensure all project dependencies get installed.
4. Right click the file located in the Project Explorer directory of fecsimulator -> src/main/java -> fecsimulator -> HospitalSimulation.java and select the run configurations option. From the window that opens select the Arguments tab, located beside the Main tab, and type in `-Xmx1400m`, excluding the quotation marks, in the VM arguments text area. This will increase the JVM memory allocation size which is required for running larger scale simulations.
5. Once again right click the HospitalSimulation.java file, mentioned in the previous step, and select the Run As -> Java Application option. This will start the simulation execution.
6. At the start of the simulation execution, an input will be required to determine how many hospital wings to include in the simulation. To avoid JVM memory usage errors, input a number between 1 and 5. The next stage of the simulation can be executed with two different approaches, outlined below:
 - a. To run the simulation without integrating serverless functions enter `"n"`, excluding the quotation marks, in the Eclipse console window when prompted if wanted to integrate serverless functions This will start the execution of a simulation which used only the simulated resources which have been implemented in the local FEC environment and after a few minutes results of the simulation will be printed in the console.

- b. To run the simulation with serverless function integration enter “y”, excluding the quotation marks, in the Eclipse console window, when prompted to integrate serverless functions or not. This will then prompt for AWS access key, AWS secret access key and AWS session token credentials to use for interacting with the AWS services which have been integrated into the simulation. Credential values can be found by logging into an NCI account using the [https://ncirl.awsapps.com/start#/
website and selecting the Command line or programmatic access option. When each credential is entered after being prompted for its input, the simulation will execute with serverless functions. This can take quite some time the simulation results will be printed in the console when the execution completes.](https://ncirl.awsapps.com/start#/)