Setup

Elaine Wah
ewah@umich.edu

Erik Brinkman
erik.brinkman@umich.edu

Updated: March 25, 2015

1 Initial Setup

Before you get a copy of the code you need to make sure that you have the following things installed on your machine. If you have any trouble installing them contact me.

Open SSH: To connect to the remote repository. You can verify you have it if `which ssh` returns something

Git: To get a copy of the source code. `which git`

Java 6: We are using Java 6 for a number of reasons. If `vava -version` says `1.6` somewhere then you have Java 1.6 installed, but you may have it even if that says `1.7` or `1.8.`

Ant: To compile Java. If you have Java you probably have Apache Ant already.

2 Connecting to the repository

The repository for the HFT simulation system is hosted on `hft.eecs.umich.edu`. If you need an account on this machine in order to run simulations, please email Elaine. To access the repository, you will need to generate a public RSA key:

1. Generate a public/private RSA key pair with the following command:

   ```
   ssh-keygen -t rsa -C "<youremailhere>"
   ```

2. Press enter when prompted for the name to save under. Do not change the name from `id_RSA.pub`, and do not change the location where it is saved.

3. Enter a non-empty passphrase when prompted.

4. Email Elaine the `.ssh/id_RSA.pub` file located in your home directory.

5. After you receive notification that your key has been authorized to connect, proceed to the next step.

6. Create a directory to hold all of your HFT-related work, and then enter it by running the following (or similar) command:

```
cd
mkdir hft_project
cd hft_project
```

7. Clone the git repository to get a copy of the source code

```
git clone git@hft.eecs.umich.edu:hft.git
```

3 Setting up Eclipse

It will be easiest to debug any new code added to the simulation system using an IDE. The following instructions are for Eclipse, but any other IDE for Java will suffice. Note that instructions are based on the Eclipse for Linux, and may differ depending on your OS. For the rest of the instructions we will assume that the source code is in `~/hft_project/hft/hft-sim`, which is the path for the HFT simulation system project. If you checked out the code somewhere else, make the necessary changes.

1. Download and unzip "Eclipse for Java Developers" (eclipse.org/downloads)
2. Start up Eclipse. For your workspace, select the path to the `hft` folder cloned from the repository (e.g. `~/hft_project/hft`).
3. Set Eclipse to auto-refresh the workspace
   (a) Select Window from the top bar and go down to Preferences
   (b) Expand `General` and then click on `Workspace`
   (c) Check `Refresh using native hooks or polling` and `Refresh on access`
   (d) Click OK
4. From the File menu, select New » Project...
5. Select `Java Project from Existing Ant Buildfile`.
6. Browse for the Ant buildfile; select the `build.xml` file located in `hft-sim`, the simulation system directory (`~/hft_project/hft/hft-sim/build.xml`). Leave the project name unchanged and click Finish.
7. At this point, the project will be imported automatically.
8. Fix the project build path:
   (a) Right click on the `hft-sim` project and select Properties.
   (b) Click Java Build Path in the left panel, then select Libraries (at the top). Remove the JRE_LIB Library.
   (c) Click `Add Library` and add a JRE System Library (make sure you add a Java 6 library!).
   (d) Select Source at the top (next to libraries).

(e) Click Add Folder and select `test`.

(f) Select `Output folder` under hft-sim/src and then click edit.

(g) Select `Project's default output folder` and then OK to close the window.

(h) Click OK and close the window.

9. Setup JUnit

   (a) Click on `Window » Preferences`.

   (b) On the left hand side, expand `Java` and click on `JUnit`.

   (c) Check "add '-ea' to VM arguments when creating a new JUnit launch configuration" to enable assertions.

10. Verify that the tests are configured correctly

    (a) Right click on the `test` folder in the left hand nav bar, and then select `Run As » JUnit Test`.

    (b) A panel should come up indicating that the tests are running. If everything is setup correctly, there should be no errors.

11. Before getting setup to run the simulator, you need to create a directory in which to run your simulations. To keep the repository organized, run all of your simulations within the `~/hft_project/hft/hft-sim/simulations` directory.

    (a) Create a folder in the `hft-sim` folder called `simulations`.

    (b) Create a folder within simulations (e.g. `simulations/test`).

    (c) Copy `docs/simulation_spec.json` into the folder you just created.

12. In the Properties window set up Run/Debug configurations:

    (a) Right click on the project again and select Properties.

    (b) Select `Run/Debug Settings` in the left panel. Add a new launch configuration (Java Application).

    (c) At the top, change the name to `Run Simulator.` To select the correct main class, click `Search...` and select `SystemManager`.

    (d) Select the Arguments tab and enter something like the following into the `Program Arguments` window:

    ```
    simulations/test 1
    ```

    The first argument is the path to the simulation directory that you created above. This can be any directory as long as it has a `simulation_spec.json` file in it.

    The second argument represents what observation number to use. In this case we're specifying run 1, but it can be any arbitrary number. This number is appended to the name of generated observation files.

13. Save your settings and you should now be able to run the simulator in Eclipse!

14. Test that everything worked:
    (a) Click the down arrow next to the green arrow Run icon and select `Run Configurations...`
    (b) Select `Run Simulator` from the left hand column and click Run.
    (c) Check that there's an observation file called `simulations/test/observation1.json`, and a log file in `simulations/test/logs`.

4 Setting up EclEmma Code Coverage Tool in Eclipse

To install EclEmma, you first need to have the Eclipse Marketplace Client installed. If you see `Eclipse Marketplace...` under the `Help` menu, then the Eclipse Marketplace Client is installed, and you can skip to section 4.2.

4.1 Installing the Eclipse Marketplace Client

1. Go to `Help » Install New Software...`
2. Click the drop down arrow next to `Work with:` and select `-All Available Sites-`
3. In the search box that says `type filter text` enter `marketplace`. This may lag a lot, eclipse is just poorly threaded
4. You should see a listing called `Marketplace Client`. Select the checkbox next to it, and then click `Next >` at the bottom
5. The next page should that you are installing the Marketplace Client. Click `Next >` again
6. Read and accept the license agreement. Then click `Finish`. Eclipse should begin downloading the marketplace
7. After the Marketplace client is installed, Eclipse should prompt you to restart Eclipse. When you're ready, click `Yes`

4.2 Installing EclEmma

1. Go to `Help » Eclipse Marketplace...`
2. Type `eclemma` in the searchbox next to `Find:` and hit `Return`
3. You should see a single entry called `EclEmma Java Code Coverage`. Click the install button next to it
4. You should be prompted with a confirm dialogue. Click `Next >`
5. Read and accept the license agreement. Then click `Finish`
6. After installation, you should be prompted to restart Eclipse. When you're ready, click `Yes`

4.3 Using EclEmma

Traditionally with Eclipse, you can right click on a JUnit test, or a package with JUnit tests in it, and select `Run As » JUnit Test`. Once EclEmma is installed, if you right click on a JUnit test, or a package containing JUnit tests, you can go to `Coverage As » JUnit Test`. This will run all of the JUnit tests and show code coverage.

After running coverage, there are two main sources of coverage information. You should see a View called `Coverage` which lists the coverage of every folder in the project. In addition, each line in Eclipse will be highlighted Green, Yellow, or Red indicating Covered, Partially Covered, or Not Covered respectively.

To remove the line highlighting from your source and test files, simple click the black `X` icon in the coverage View called `Remove Active Session`.