HFT Simulation System: Connection and Setup

Elaine Wah
ewah@umich.edu

Updated: June 2013

1 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 me.

To access the repository, you will need to generate a public RSA key (you may need to install OpenSSH first):

1. `cd` into the (hidden) `.ssh` directory in your home folder (e.g., `\home\ewah\.ssh`)
2. Run the following command:

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

3. 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.
4. Enter a non-empty passphrase when prompted.
5. Email me the `id_rsa.pub` file.
6. Once your key is authorized by me, you will be able to connect.

Before you can access the repository, I will need to authorize your RSA key. Once I've let you know that this is done, you will be able to access the repository. To clone the repository, first `cd` into whichever directory in which you'd like the repository to be copied. Then enter the following command to create a folder `hft` with the repository contents:

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

2 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.

1. Download and unzip "Eclipse for Java Developers" (`http://www.eclipse.org/downloads/`)

2. Start up Eclipse. For your workspace, select the directory that contains the `hft` folder cloned from the repository.

   For example, if the source code is located within `\home\ewah\projects\hft\`, then the Eclipse workspace should be `\home\ewah\projects`.

3. From the File menu, select New Project. Select "Java Project from Existing Ant Buildfile."

4. Browse for the Ant buildfile; select the `build.xml` file in the top level directory of the repository. Leave the project name unchanged and click Finish.

5. At this point, the project will be imported automatically. Remove the JRE System Library from the build path (via the right-click menu). Delete `${classpath}`.

6. Right-click the project in the Project Explorer View and select Properties. To set up libraries correctly:

   (a) Click *Java Build Path* in the left panel, then select Libraries (at the top). Remove the JRE System Library and `${classpath}`.
   Click "Add Library" and add a JRE System Library (the workspace default one will be fine).

   (b) Still within *Java Build Path*, select Source. At the very bottom, change the default output folder to `hft/build`.

7. In the Properties window, to set up run/debug configurations:

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

   (b) At the top, change the name of the launch if you'd like. To select the correct main class, click "Search..." and select `SystemManager`.

   (c) Select the *Arguments* tab. To keep the repository organized, run simulations within the `simulations` directory.

   > The HFT simulator takes in two input arguments.
   > **Argument 1** is the location of the configuration file (named `simulation_spec.json`).
   > **Argument 2** is an arbitrary number. For debugging purposes this can be set to whatever you'd like; it is the number appended to the name of generated observation files to tell the simulation run it is associated with.

   (d) Within the "Program Arguments" window, type the input arguments.
   Example: `simulations\test 2` indicates that the configuration file is stored within the subfolder `test` within `simulations`.
   Note: Prior to this you will need to create a subfolder within `simulations` and copy the appropriate `simulation_spec.json` file into the subfolder. There is an example configuration file in `simulations` which you can copy into your subfolder (in this example, the subfolder is named `test`).

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