CReST: The Cloud Research Simulation Toolkit

CReST Quickstart

The source code for CReST is available via SVN on sourceforge. Build the source code using the ant build script build-CReST.xml.

```
$ ant –buildfile build-CReST.xml
```

This creates a distribution directory dist/ containing three folders:

1. CReST-app: containing executable CReST jar and start scripts
2. CReST-code: containing a copy of CReST source code
3. Javadocs: containing CReST javadocs documentation

To run CReST, move to the CReST-app directory and run the example start script:

```
$ ./example-start-crest.sh
```

This script contains the following command:

```
$ java –Xmx1000m –Xdock:name="CReST" –jar CReST.jar –c
resources/config/example_1dc_1.xml.gz –p
resources/config/prop/example_params_1.properties
```

This command will start an instance of CReST in graphical interactive mode. For non-graphical batch mode, use the command line flag '-nogui' or '-ng'. The required '-c' flag denotes the datacentre configuration file to run and the optional (but recommended) '-p' flag denotes the CReST simulation properties file to run.

The CReST GUI will open when the script is run. The configuration file can be changed by using the file > open dialog box. To start CReST, do Run > Run Simulator (or the short-cut F5). This starts the simulation running. To stop the simulation, do Run > Stop Simulator (or the short-cut q).

CReST Configuration

CReST Builder – Datacentre Configuration Files

In order to run, CReST requires a datacentre configuration file. These files contain an XML description of the physical properties of the datacentre, including the location and description of all servers. Configuration files can be generated and edited using the CReST Builder GUI application. To create a new configuration file, run CReSTBuilder using the shell script:

```
$ startCReSTBuilder.sh
```

Select “New Configuration” and give it a name, e.g., “my_dc_config”. When the configuration is complete, it will be saved with extension .xml.gz

CReST Simulation Properties Files

Simulation properties can be input into CReST using simple text files that must have extension ‘.properties’. See file: docs/user-guide-properties-file.txt for a list of properties that can be altered for the subscription/middleware module. Some example ‘.properties’ files can be found in the folder: resources/config/prop/. Simulation properties files override parameter values in CReST configuration files, providing an easy way for users to vary parameter values across runs, without creating many different datacentre configuration files.

CReST User Events Files

User defined events can be configured for CReST using user events files. Examples of these can be found in the resources/event/ directory. Events determine the fail time and fail type of individual hardware.