jcstress

The Java Concurrency Stress tests (`jcstress`) is an experimental harness and a suite of tests to aid the research in the correctness of concurrency support in the JVM, class libraries, and hardware.

Quick start

There are two scenarios for jcstress use.

Scenario 1. Creating and running a small batch of concurrency tests.

For this, it is more convenient to create a standalone project with jcstress build dependencies. You should have Maven installed. **You will need JDK 8+ to build and run the test project.** There is a public Maven archetype that can be used to bootstrap the simple testing project. It can be invoked in non-interactive mode with:

```
$ mvn archetype:generate \
 -DinteractiveMode=false \
 -DarchetypeGroupId=org.openjdk.jcstress \
 -DarchetypeArtifactId=jcstress-java-test-archetype \
 -DgroupId=org.sample \
 -DartifactId=test \
 -Dversion=1.0
```

Or in interactive, which will prompt for project properties:

```
$ mvn archetype:generate \
 -DarchetypeGroupId=org.openjdk.jcstress \
 -DarchetypeArtifactId=jcstress-java-test-archetype
```

After this, change into the project directory, build it, and the runnable JAR would await.

```
$ cd test
$ mvn clean install
$ java -jar target/jcstress.jar
```

Run the JAR with `-h` to get the help. Maven projects can be opened in your IDE of choice.

Scenario 2. Building and running the entire suite of existing tests

You should have Mercurial and Maven installed to check out and build the tests. **You will need JDK 9+ to compile all the tests**, while most tests are runnable on JDK 8+. The suite can reference the APIs from future releases, as the jcstress harness will fail gracefully on API mismatches, and the mismatched tests will be just skipped.

Checkout, build and run the suite:

```
$ hg clone http://hg.openjdk.java.net/code-tools/jcstress/ jcstress
$ cd jcstress/
$ mvn clean install
$ java -jar tests-all/target/jcstress.jar
```

You may want to build and run some test collections separately, like this:

```
$ mvn clean install -pl tests-custom -am
$ java -jar tests-custom/target/jcstress.jar
```

Run the JAR with `-h` to get the help.

Caveats

- The harness and tests are heavily experimental, and can change without notice as we understand the methodology better, fix issues, and improve the test reliability and correctness

- Most of the tests are probabilistic, and require substantial time to catch all the cases. It is highly recommended to run tests longer to get reliable results.

- Most of the tests require at least 2 online CPUs. Low CPU count machines could also use these tests, but harness will force yielding there.

- Test failure does not immediately mean the implementation bug. The usual suspects are the bugs in test infrastructure, test grading error, bugs in hardware, or something else. Share your results, discuss them, we will figure out what's wrong. Discuss the result on the relevant mailing lists first. Two usual options are:
    - concurrency-interest@cs.oswego.edu: general discussion on concurrency
    - jcstress-dev@openjdk.java.net: to discuss jcstress issues

Developing tests

If you want to develop a test, these steps are useful:

Read and understand jcstress-samples. They come in three groups: **APISample_*** describe the API, **JMMSample_*** describe the basics of Java Memory Model, **ConcurrencySample_*** show the interesting concurrent behaviors of standard library. There is a runnable JAR available at Maven Central: jcstress-samples-0.1-full.jar (change the version accordingly to get the most recent build). Please consider contributing the interesting tests back, for greater public good! We follow the OpenJDK contribution policy for patches.

Review jcstress API Javadocs. Understand the properties and conditions that are guaranteed for those APIs. If you need some other test interface/harness support, please don't hesitate to raise the issue and describe the scenario you want to test. You are encouraged to provide the thorough explanation why particular test outcome is acceptable/forbidden/special. Even though harness will print the debug output into the console if no description is given.

Project layout

- **jcstress-core**: jcstress infrastructure itself. Any jcstress-driven project should depend on this module. If you have the standalone jcstress tests, you may depend on this module alone.
- **jcstress-test-gen**: Generator which builds the autogenerated tests in the suite.
- **jcstress-java-test-archetype**: Maven archetype sources.
- **tests-chapter-***: Generated suite tests. The build of this module will invoke the test generator, and compile the generated tests.
- **tests-custom**: Custom tests, not coming from the generator. You can use this module to add your own custom tests. Use the JAR for this module to only run the custom tests.
- **tests-all**: Aggregates all the tests. The build of this module will merge the tests available in other modules.

Links

- Source repository
- Mailing list