Setting up synchrobench (linux)

1 Prerequisites

To compile the source codes you will need to have installed: ant, make and Java 8.

```
$ sudo apt-get install ant
```

```
$ sudo apt-get install make
```

Check for a jdk version 8 (or 1.8) available:

```
$ apt-cache search openjdk
```

In the following examples we use the package openjdk-8-jdk-headless[1].

```
$ sudo apt-get install openjdk-8-jdk-headless
```

2 Build synchrobench files

Clone the sycnhrobench repository:

```
$ git clone https://github.com/gramoli/synchrobench.git
```

Go to synchrobench/java and open the Makefile. In line 31, modify the path to rt.jar with the appropriate from your machine.

```
30 ifeq ($(OS_NAME), Linux)
31    JDK_RT ?= /usr/lib/jvm/java-1.7.0/jre/lib/rt.jar
32 endif
```

You can search the appropriate path (e.g. /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/rt.jar) for java 8 using the find command.

Open the terminal at the synchrobench/java directory, execute the line below and type the number for the jdk 8 option.

```
$ sudo update-alternatives --config javac
```

This will modify the java version in the current environment to the selected one. Type java -version to verify the current used version is 8 (or 1.8). Use make to compile the files. Depending on the computer you might need to execute (to preserve environment variables):

```
$ sudo -E make
```

[1] In case you could not find any jdk 8 available using apt, you can search it in a repository for your linux distribution and install it using dpkg.

3 Running the Benchmark

Once the build is successful, you can already verify the performance of built in abstractions. The following example uses the CoarseGrainedListBasedSet class:

```
$ java -cp bin contention.benchmark.Test -b
linkedlists.lockbased.CoarseGrainedListBasedSet -d 2000 -t 4
-u 10 -i 10000 -r 20000 -W 0
```

Here -d is the duration (in ms), -t is the number of threads, -u is the update ratio (10% in the example), -i is the list size, which is recommended to be half the size of -r, the range. -W is the warm up time.

3.1 Running in a remote machine

We will use use one of the machines at `https://lames.enst.fr/`, recall that in order to access remote machines you will need to use a vpn (`https://eole.telecom-paris.fr/vos-services/services-numeriques/connexions-aux-reseaux`).

Select a machine (e.g. lame21, check if the machine you are using has java installed) and copy the bin folder.

```
$ scp -r \bin user@lame21.enst.fr:/home/infres/user/BenchBin
```

Access the remote machine using ssh and run the benchmark.

```
$ ssh user@lame21/enst.fr
$ java -cp BenchBin contention.benchmark.Test -b
linkedlists.lockbased.CoarseGrainedListBasedSet -d 2000 -t 10
-u 10 -i 10000 -r 20000 -W 0
```