

D6.2.2: Microbenchmarking Software Package - Installation Guide

Paolo Bertasi, Università di Padova (UNIPD)

1 Introduction

This is a brief guide to the installation of the **Microbenchmarking Software Package** (Deliverable 6.2.2 – UNIPD). The suite of microbenchmarks is a stand-alone application and runs on the AEOLUS distributed testbed maintained by the University of Paderborn

2 Installation on a generic set of machines

In order to have the microbenchmarking suite running, its jar file `Benchmarking.jar` has to be copied onto every peer. Then the suite can be invoked either in slave or in master mode. (The suite can also be invoked to establish relay or rendezvous peers.) The following is a quick step-by-step guide on how to install the suite on UNIX/Linux machines.

1. Copy the suite to the all the target machines, e.g.

```
scp Benchmarking.jar username@machine.dei.unipd.it:~
```

2. copy the JXTA and AEOLUS library to the target machines, e.g.

```
scp -r jxtalib_directory username@machine.dei.unipd.it:~
```

3. connect to the selected machines to start the slave peers, e.g.

```
ssh username@machine.dei.unipd.it  
java - jar Benchmarking.jar edge
```

4. connect to the master peer to start the test, e.g.

```
java -jar Benchmarking.jar ma cpu 6 10 1000 cycles 1000000 15979
```

5. wait for the results

By default, the suite loads the list of rendezvous and relay peers from `http://verona.dei.unipd.it/ jxta`

These are the current lists in the files:

`rdvs.txt:`

`tcp://147.162.96.109:9701`

`relay.txt:`

`tcp://147.162.96.121:9701`

3 Installation onto the AEOLUS testbed

The installation is influenced by the way the testbed is implemented. In order to have the suite running, the jar file has to be uploaded to the testbed through the web interface, then a set of nodes has to be started and the operations described in the previous section are to be performed.

4 Command line guide

These are the self-explanatory available command lines associated to the services offered by the suite. The prefix of every line is clearly:

```
java -jar Benchmarkink.jar ma
```

for the generic version, while it is

```
java -cp Benchmarking.jar aeolus.unipd.acg.testbed.BenchBedMaster ma
```

to run tests on the AEOLUS testbed.

```
cpu NUM_PEERS NUM_REP MAX_TIME IS_TIME_OR_CYCLES TIME_OR_CYCLES
```

```
START_VALUE IS_FLOATING IS_TRUSTED
```

```
bandwidth NUM_PEERS NUM_REP MAX_TIME DATA_SIZE DYNAMICITY
```

scatter NUM_PEERS NUM_REP MAX_TIME DATA_SIZE DYNAMICITY

gather NUM_PEERS NUM_REP MAX_TIME DATA_SIZE DYNAMICITY

all2all NUM_PEERS NUM_REP MAX_TIME DATA_SIZE DYNAMICITY

We refer the reader to Deliverable D6.2.2. for a full explanation on the different microbenchmarks and their parameters.