

Benchmark archive for the SSP'16 paper H. Eichelberger, A. Sass, K. Schmid: From Reproducibility Problems to Improvements: A journey

Files in this file set:

- analysis.r: The modified R script used for creating the figures in the SSP'16 paper.
- benchmark.docx: This file.
- Benchmark.pdf: The PDF of this file
- clonezilla-live-2.4.9-17-amd64.zip: The clonezilla versions used to create images from the two machines used for reproduction and experimentation for the SSP'16 paper.
- clonezilla-live-2.4.9-17-i686.zip: The clonezilla versions used to create images from the two machines used for reproduction and experimentation for the SSP'16 paper.
- dell.zip: The small files of the clonezilla image for the Dell setup. Uncompressing the gz images shall provide you access to the actual installation used during the experiments. The entire clonezilla image requires also
 - o sdb2.ext4-ptcl-img.gz.aa
 - o sdb3.ext2-ptcl-img.gz.aa
 - o sdb3.ext2-ptcl-img.gz.ab
 - o sdb3.ext2-ptcl-img.gz.ac
 - o sdb3.ext2-ptcl-img.gz.ad
 - o sdb3.ext2-ptcl-img.gz.aePlease note that sdb2.ext4-ptcl-img.gz.aa is part of dell.zip. The password for testuser is available on request.
- hp.zip: The entire clonezilla image for the HP setup. Uncompressing the gz images shall provide you access to the actual installation used during the experiments. The password for testuser is available on request.
- initial_ds.zip: The MooBench measurements for both machines, Dell and HP without modifications to SPASS-meter (binary jar is also in the clonezilla image).
- MooBench.zip: The modified MooBench version used for the experiments as described in the SSP'16 paper.
- Result_ds.zip: The MooBench measurements for both machines, Dell and HP with modifications to SPASS-meter.
- Server.zip: The TCP/IP server implementation used for the file/network experiments in the MSc thesis of Aike Sass.
- Spass-meter.zip: The modified SPASS-meter versions including the last patch by Aike.

How to run the experiment (on the Dell image, in HP directly in /home/testuser):

- base dir is /home/testuser/aike
- There are 3 different run-Scripts. All these scripts call the corresponding benchmarkXY.sh

Name	Description	SPASS-meter Ressource	MooBench-MonitoredApplication	MooBench-Version	Additional notes
run.sh	standard benchmark	CPU_TIME, MEMORY	MonitoredClassSimple	MooBench.jar	
run_FileIO.sh	file IO benchmark	FILE IO	MonitoredClassSimpleFileIO	MooBench_fileNetIO.jar	
run_NetIO.sh	network IO benchmark	NET IO	MonitoredClassSimpleNetIO	MooBench_fileNetIO.jar	make sure the Server.jar is running on the Debian system; Moobench uses a hard coded IP "147.172.177.234"

The main difference in the corresponding benchmarkXY.sh is the configured parameters for the Moobench-MonitoredApplication and the used MooBench-Version.

General steps to run the benchmark:

1. Upload the SPASS-meter jars into lib/linux and link them to linux
2. Open lib dir and edit the config.xml so that it satisfies your needs and link it to config.xml
2. Run "run.sh"
3. When the benchmark finished download the tmp dir
4. Use the r-script to plot the results

Additional notes:

MooBench is configured to collect results for memory and time. You can change these settings by configuring a different BenchmarkingThread. To do this you need to edit Benchmark.java at line 71. Instead of using BenchmarkingThreadMemoryAndNano you can use BenchmarkingThreadNano. After that you will need to generate a new Moobench.jar and upload this jar onto the server. Make sure the scripts are using this new jar.

JMX getCpuThreadTime bypass:

See gearsBridge and gearsBridgeJ projects in SPASS-meter.zip

- gearsBridgeJ: de.uni_hildesheim.sse.system.deflt.ThreadDataGatherer was extended by the native method getCpuThreadTime0 used by the relevant methods. Here not all methods were modified. This is part of a more recent commit to github.
- gearsBrige:
 - o de_uni_hildesheim_sse_system_DataGatherer.h defines the signature of getCpuThreadTime0
 - o de_uni_hildesheim_sse_system_DataGatherer.cc implements the call to the JVM internal method via the internal JVM interface (jmm_interface). jmm_interface is obtained during library loading in JNI_OnLoad from the JVM.