FASE 2023 Artifact

NOTE: this README is available also in HTML or PDF format.

Summary

This artifact contains a prototype of a runtime monitoring middleware called VAMOS as discussed in the FASE 2023 paper *VAMOS: Middleware for Best-Effort Third-Party Monitoring* by Marek Chalupa, Stefanie Muroya Lei, Fabian Muehlboeck, and Thomas A. Henzinger. The artifact consist of the software, benchmarks discussed in the paper, and a set of scripts to run experiments and present the results.

Hardware Requirements

For all benchmarks to make sense, the artifact requires a reasonably modern (~past 5? years) x86-64 processor with multiple cores (ideally at least 6 cores).

Using a bundled docker image

[TBD]

Building the artifact

To build and run the artifact run this command from the top-level directory:

```
docker build . -t vamos:fase
```

If you are on a new enough Linux system, you may use this command to get faster builds:

```
DOCKER_BUILDKIT=1 docker build . -f Dockerfile.buildkit -t vamos:fase
```

To run the built image, use:

```
docker run -ti vamos:fase
```

Once in the docker container, continue with the test instructions below.

Test Instructions

We have prepared script to run two versions of experiments: short and full experiments. Short experiments run just a few minutes, but the results may slightly diverge from the results in the paper. You can run short experiments with:

```
./short-experiments.sh
```

Full experiments are run with

```
./full-experiments.sh
```

Full experiments can run for several hours and should reproduce numbers from the paper. Note that running full experiments will overwrite results of short experiments.

Once (short or full) experiments are finished, you can generate plots in the PDF format as follows:

[TBD]

You can also generate the plots from the original data that we measured. The data are in [TBD] and you can generate the plots by [TBD].

Running just some experiments

To run just some experiments, you can comment out lines with experiments in `{short,full}-experiments.sh` scripts and rerun these scripts or you can follow what these script do:

First, pick if you want short of full experiments and according to that, copy `scripts/setup-short.sh` or `scripts/setup-full.sh` into `setup.sh` (we are in the top-level directory now):

```
cp scripts/setup-short.sh setup.sh   # short experiments
cp scripts/setup-full.sh setup.sh    # full experiments
```

Then, go into directories with experiments and run `make experiments` in the directory, for example:

```
cd scalability
make experiments
```

Available experiment directories are `scalability`, `primes`, `bank`, `bank-tessla`, and `dataraces`. The only exception from this pattern are primes-tessla experiments that are run from the `primes` directory using `make experiments-tessla`:

```
cd primes
make experiments-tessla
```