

Effective Lock Handling in Stateless Model Checking

Michalis Kokologiannakis, Azalea Raad, Viktor Vafeiadis

We consider our paper’s artifact to be the set of benchmarks we used in the paper, as well as the results we got by running particular versions of model checking tools (GenMC and LAPOR) on the benchmarks set. We do not consider the artifact of the paper to be LAPOR, as it will evolve over time.

While GenMC is available on GitHub, LAPOR has not been merged in that repository yet. We plan to first update LAPOR (perform some optimizations, etc), and then merge it in the repository in the near future.

The artifact consists of a Virtual Machine (VM) containing binaries for GenMC and LAPOR, along with all the benchmarks used in the submitted version of our paper. This hopefully suffices to validate the claims made in the paper.

For any bugs, comments, or feedback regarding GenMC or LAPOR, please do not hesitate to contact us.

1 Getting Started

To verify the state of the artifact please follow the following steps:

1. Download and install VirtualBox, in case it is not already installed in your machine. We have tested the VM with VirtualBox 5.2.26 in Debian GNU/Linux, and VirtualBox 6.0.4 in OSX.
2. Open VirtualBox, and import our VM by clicking "File"->"Import Appliance".
3. After starting the VM, you can log in with the username "user" and password "lapor". Once logged in, shortcuts to the terminal and the file manager can be found under "Activities".
4. To verify that the tools can be run successfully, you can issue the following commands:
 - `genmc -help`
 - `lapor -help`
5. Finally, at `~/oopsla19-benchmarks/benchmarks` you can

find all the benchmarks we used for the tables in our paper.

The above should suffice for a basic functionality test of our artifact. In the next section, we describe how to reproduce the performance claims in the paper, as well as the basic functionalities of each tool and how to run custom test cases with them.

2 Step-by-Step Instructions for Evaluation

2.1 Running the Experiments

In the folder `~/oopsla19-benchmarks` there are some shell scripts and the data we used for the tables of the Evaluation Section of our paper. We note that while the exact execution times will vary depending on the PC hosting the VM, it should be easy to get a general idea about the way the different tools behave and their relative performance.

2.1.1 Reproducing Table 1 (~ 50 seconds)

For this and the following sections, we assume that the working directory is `~/oopsla-benchmarks`. To reproduce the results of Table 1, issue the following command:

```
|| ./get-table1.sh
```

This command will print a \LaTeX table with GenMC on the leftmost columns and LAPOR on the rightmost columns. Each pair of columns represents the number of executions explored by the tool and the execution time that was required, respectively.

2.1.2 Reproducing Table 2 (~ 5-6 minutes)

In order to reproduce most of the results from Table 2 please issue the following command:

```
|| ./get-table2.sh
```

This will produce all entries of that table, apart from the row corresponding to rw(8), since the latter requires more memory than the rest and cripples the VM. Should you want to reproduce it, either comment out L33 in table2-benchmarks.sh and uncomment L34, or issue the following two commands:

```
|| genmc -- -DN=8 benchmarks/synthetic/N-rw/variants/lockN-rw0.c  
|| lapor -- -DN=8 benchmarks/synthetic/N-rw/variants/lockN-rw0.c
```

Note that perhaps an increase on the memory the VM occupies is necessary as well.

2.1.3 Reproducing Table 3 (~ 14 minutes)

Similarly, to reproduce the results from Table 3 issue:

```
|| ./get-table3.sh
```

2.1.4 Reproducing Table 4 (~ 1 hour and 15 minutes)

Finally, to reproduce the results of Table 4 issue:

```
|| ./get-table4.sh
```

2.2 Using the Different Tools

In order to run arbitrary test cases with GenMC or LAPOR, please see the respective subsections below. Please note that the tools assume that programs under test should be:

1. Finite: they should not have infinite loops; if such loops exist the `--unroll` switch or `assume()` statements need to be used, in order to make all executions finite.
2. Data-deterministic: they should not rely on user input, call `rand()`, open files, etc.

2.2.1 Structure of the existing benchmarks

In this paragraph we describe the structure of the benchmarks folder. While this is not strictly necessary for reproducing the results, it could be useful should one wants to run some of our benchmarks individually. All the benchmarks used in the Evaluation Section of our paper are located in the directory `~/oopsla19-benchmarks/benchmarks`.

Apart from the benchmarks located in the folder above, many more benchmarks can be found at `~/genmc/tests`. In that directory and the relevant subdirectories, there is a separate folder for each benchmark, that contains the "core" of the test case, as well as the expected results for the test case, some arguments necessary for the test case to run, etc. However, in order to actually run a test case, one must invoke a tool with one of the test case variants, which are located in a folder named 'variants', in turn located within the respective test case's folder.

For example, to run a simple Store Buffering (SB) test case with LAPOR, please issue:

```
|| lapor ~/genmc/tests/correct/SB/variants/sb0.c
```

2.2.2 Running GenMC

A generic invocation of GenMC looks like the following:

```
|| genmc [OPTIONS] -- [CFLAGS] <file>
```

Where CFLAGS are options that will be passed directly to the C compiler, and OPTIONS include several options that can be passed to GenMC. Amongst these options, the most useful ones are probably the `-unroll=N` switch, which unrolls a loop N times, and the `-wb` and `-mo` options, that enable the WB and the MO variant of GenMC, respectively (default is WB). Lastly, file should be a C file that uses `pthread`s for concurrency.

GenMC is publicly available at <https://github.com/MPI-SWS/genmc>.

2.2.3 Running LAPOR

The usage of LAPOR is similar to that of GenMC, with the only difference being that, currently, LAPOR can only operate under the WB equivalence partitioning.