Simultaneous Multithreading and Hard Real Time: Can it be Safe? (Artifact)

Sims Hill Osborne
University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina U.S.A.
http://shosborn.web.unc.edu/
shosborn@cs.unc.edu

Joshua J. Bakita
University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina U.S.A.
https://jbakita.me/
jbakita@cs.unc.edu

James H. Anderson
University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina U.S.A.
http://jamesanderson.web.unc.edu/
anderson@cs.unc.edu

Abstract

The applicability of Simultaneous Multithreading (SMT) to real-time systems has been hampered by the difficulty of obtaining reliable execution costs in an SMT-enabled system. This problem is addressed from two directions. A scheduler is introduced, CERT-MT, that minimizes SMT-related timing variations, and two new timing analysis methods – one based on the binomial distribution and one based on Cantelli's Inequality – are given. Both methods estimate probabilistic WCETs and attach statistical confidence levels to those estimates. The timing analyses are applied to tasks executing with and without SMT, and it is found that in some cases, two tasks utilizing SMT can be safely executed in less time than would be needed for either task by itself. A large-scale schedulability study is conducted, showing that CERT-MT can schedule systems with total utilizations twice what could otherwise be achieved. This artifact includes benchmark experiments used to compare execution times with and without SMT and code to analyze the benchmark experiments and duplicate the reported schedulability experiments.

2012 ACM Subject Classification Computer systems organization → Real-time systems; Computer systems organization → Real-time system specification; Software and its engineering → Scheduling; Software and its engineering → Multithreading

Keywords and phrases real-time systems, simultaneous multithreading, real-time, scheduling algorithms, timing analysis, probability, statistics

Digital Object Identifier 10.4230/DARTS.6.1.1

Funding Work was supported by NSF grants CNS 1563845, CNS 1717589, and CPS 1837337, ARO grant W911NF-17-1-0294, ONR grant N00014-20-1-2698, and funding from General Motors.

Acknowledgements We wish to thank the anonymous reviewers for their time and suggested improvements.

Related Article Sims Hill Osborne and James H. Anderson, "Simultaneous Multithreading and Hard Real Time: Can It Be Safe?", in 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020), LIPIcs, Vol. 165, pp. 14:1–14:25, 2020. https://doi.org/10.4230/LIPIcs.ECRTS.2020.14

Related Conference 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020), July 7–10, 2020, Virtual Conference

1 Scope

This artifact was used to generate the benchmark and schedulability results in the related conference paper [2]. More specifically, the artifact can be divided into two sections: benchmark tests and schedulability tests. The benchmark tests, based on the TACLeBench sequential benchmarks [1], are the basis of Secs. 5.1 and 5.2 ("Timing Analysis Assessment" and "Execution Times with and Without SMT", respectively) in the related paper. The schedulability tests generate the results shown in Sec. 5.3 ("Schedulability Study") the related paper. The benchmark tests are closely related to those used in our previous paper, [3], and detailed in the artifact [4].

2 Content

The artifact package includes:

- An instruction file, `ArtifactInstructions.pdf`.
- Source code for benchmarks and helper scripts to compile and run the benchmarks. All this is contained in the folder *benchmarkExecution*.
- Code to analyze the output of the benchmark execution tests, along with the original output of our own tests, in the folder *benchmarkAnalysis*.
- The optimization program that forms our scheduler in the folder *scheduler*.
- Our schedulability results, graphs and the program to produce graphs from our raw results are also in the folder *scheduler*.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `http://jamesanderson.web.unc.edu/` and `https://github.com/shosborn/SMT-HRT-ECRTS20`. Due to file size restrictions, some of the raw data is excluded from the github version.

4 Tested platforms

The benchmark tests were originally conducted on an Intel Xeon Silver 4110 2.1 GHz (Skylake) CPU running Ubuntu 16.04.6. They are expected to run on any Hyperthreading-enabled Intel platform, which includes most Xeon and Core processors running Linux. Exact results will vary based on platform, but general trends are expected to hold. Running benchmarks on a virtual machine is not expected to reproduce the observed trends.

Schedulability tests were performed using Gurobi (`www.gurobi.com`), a commercial mathematical optimization solver. A free, full-featured academic license is available. The front-end for the Gurobi solver was programmed in Python 3, using the gurobipy module. Instructions for installing gurobipy are available as part of the overall Gurobi installation instructions provided on the Gurobi website.

There are no specific hardware requirements for replicating our schedulability results, but users should be aware that running all tests as written required close to a week on a large-scale research machine using a high degree of parallelism. It is helpful, but not necessary, to have access to such a system that uses the SLURM job submission system. The instructions included with the artifact also state how to run alternative, faster, reduced-fidelity tests.

5 License

6 MD5 sum of the artifact

03dd6255f2cee5f39b0977e0cacddd5b

7 Size of the artifact

521 MB (compressed)

References

1 H. Falk, S. Altmeyer, P. Hellinckx, B. Lisper, W. Puffitsch, C. Rochange, M. Schoeberl, R. B. Sorensen, P. Wagemann, and S. Wegener. TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research. In *WCET 2016*, pages 2:1–2:10, 2016.

2 S. Osborne and J. H. Anderson. Simultaneous multithreading and hard real time: Can it be safe? In *ECRTS 2020 (conditionally accepted)*, 2020.

3 S. Osborne, J. Bakita, and J. H. Anderson. Simultaneous multithreading applied to real time. In *ECRTS '19*, 2019.

4 S. Osborne, J. Bakita, and J. H. Anderson. Simultaneous Multithreading Applied to Real Time (Artifact). *Dagstuhl Artifacts Series*, 5(1):8:1–8:2, 2019. `doi:10.4230/DARTS.5.1.8`.