Simultaneous Multithreading Applied to Real Time (Artifact)

Sims Hill Osborne
University of North Carolina, Chapel Hill, North Carolina, USA
shosborn@cs.unc.edu

Joshua J. Bakita
University of North Carolina, Chapel Hill, North Carolina, USA
jbakita@cs.unc.edu

James H. Anderson
University of North Carolina, Chapel Hill, North Carolina, USA
anderson@cs.unc.edu

Abstract

Existing models used in real-time scheduling are inadequate to take advantage of simultaneous multithreading (SMT), which has been shown to improve performance in many areas of computing, but has seen little application to real-time systems. The SMART task model, which allows for combining SMT and real time by accounting for the variable task execution costs caused by SMT, is introduced, along with methods and conditions for scheduling SMT tasks under global earliest-deadline-first scheduling. The benefits of using SMT are demonstrated through a large-scale schedulability study in which we show that task systems with utilizations 30% larger than what would be schedulable without SMT can be correctly scheduled. This artifact includes benchmark experiments used to compare execution times with and without SMT and code to 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 → Multithreading; Software and its engineering → Scheduling

Keywords and phrases real-time systems, simultaneous multithreading, soft real-time, scheduling algorithms

Digital Object Identifier 10.4230/DARTS.5.1.8

Funding Work supported by NSF grants CNS 1409175, CNS 1563845, CNS 1717589, and CPS 1837337, ARO grant W911NF-17-1-0294, and funding from General Motors.

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

Related Article Sims Hill Osborne, Joshua J. Bakita, and James H. Anderson, "Simultaneous Multithreading Applied to Real Time", in 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), LIPIcs, Vol. 133, pp. 3:1–3:22, 2019.
`https://dx.doi.org/10.4230/LIPIcs.ECRTS.2019.3`

Related Conference 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), July 9–12, 2019, Stuttgart, Germany

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 Sec. 4 ("SMT and Execution Times") in the related paper. The schedulability tests generate the results shown in Sec. 5 ("Schedulability Experiments") of the related paper.

2 Content

The artifact package includes:

- An instruction file, `ArtifactEvaluationInstructions.pdf`.
- Source code for benchmarks, helper scripts to compile, run, and summarize results, a Microsoft Excel file to format results as seen in the paper, and copies of the raw output for the experiments reported in the paper. All this is contained in the folder `interference-benchmark`.
- Code to run the schedulability tests and copies of the raw results as reported in paper can be found in the folders `uniform-normal` and `gaussian-average`. These paths correspond to the two experimental methods described in the paper.
- Script `gen_graphs.py` to generate graphs for all the schedulability study results.

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/`.

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 in Python 3, with some results formatted using Microsoft Excel. They do not have specific hardware requirements, but users should be aware that running all tests as written required over a week on a recent Intel-based server with 24 physical cores and 48 threads. However, the instructions included with the artifact also state how to run alternative, faster, reduced-fidelity tests.

5 License

6 MD5 sum of the artifact

e83e1ed15eea76f1e323d757b42d8ab8

7 Size of the artifact

23.3 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, J. Bakita, and J. H. Anderson. Simultaneous Multithreading Applied to Real Time. In *ECRTS '19*, 2019.