NPM-BUNDLE: Non-Preemptive Multitask Scheduling for Jobs with BUNDLE-Based Thread-Level Scheduling (Artifact)

Corey Tessler
Wayne State University, Detroit, Michigan, USA
corey.tessler@wayne.edu

Nathan Fisher
Wayne State Univeristy, Detroit, Michigan, USA
fishern@wayne.edu

Abstract

The BUNDLE and BUNDLEP scheduling algorithms are cache-cognizant thread-level scheduling algorithms and associated worst case execution time and cache overhead (WCETO) techniques for hard real-time multi-threaded tasks. The BUNDLE-based approaches utilize the inter-thread cache benefit to reduce WCETO values for jobs. Currently, the BUNDLE-based approaches are limited to scheduling a single task. This work aims to expand the applicability of BUNDLE-based scheduling to multiple task multi-threaded task sets.

BUNDLE-based scheduling leverages knowledge of potential cache conflicts to selectively preempt one thread in favor of another from the same job. This thread-level preemption is a requirement for the run-time behavior and WCETO calculation to receive the benefit of BUNDLE-based approaches. This work proposes scheduling BUNDLE-based jobs non-preemptively according to the earliest deadline first (EDF) policy. Jobs are forbidden from preempting one another, while threads within a job are allowed to preempt other threads.

An accompanying schedulability test is provided, named Threads Per Job (TPJ). TPJ is a novel schedulability test, input is a task set specification which may be transformed (under certain restrictions); dividing threads among tasks in an effort to find a feasible task set. Enhanced by the flexibility to transform task sets and taking advantage of the inter-thread cache benefit, the evaluation shows TPJ scheduling task sets fully preemptive EDF cannot.

2012 ACM Subject Classification Computer systems organization → Real-time systems; Software and its engineering → Real-time schedulability

Keywords and phrases Scheduling algorithms, Cache Memory, Multi-threading, Static Analysis

Digital Object Identifier 10.4230/DARTS.5.1.2

Related Article Corey Tessler and Nathan Fisher, "NPM-BUNDLE: Non-Preemptive Multitask Scheduling for Jobs with BUNDLE-Based Thread-Level Scheduling", in 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), LIPIcs, Vol. 133, pp. 15:1–15:23, 2019.
https://dx.doi.org/10.4230/LIPIcs.ECRTS.2019.15

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

1 Scope

The artifacts for non-preemptive multi-task BUNDLE (NPM-BUNDLE) allow an independent party to recreate and expand upon the results presented in the research. The primary focus is reproduction of the ten graphs that summarize the schedulability ratios of preemptive EDF, non-preemptive EDF, and the proposed Threads Per Job (TPJ) algorithms.

Data for the graphs is supplied by the creation and analysis of synthetic task sets. Synthetic task sets and their analysis depend on two components 1.) the libsched library 2.) a framework utilizing libsched. Both of these components are pre-built and configured in the supplied virtual machine of the artifact.

The virtual machine image and accompanying instructions provide direction for reproducing the results presented in the research, as well as additional results and methods for tailoring the data set size and parameters to suit subsequent research.

2 Content

When the artifact package is expanded, it includes:

- `NPM-BUNDLE-01.zip`: A virtual machine image compatible with Virtual Box.
- `NPM-BUNDLE-artifact-eval.pdf`: A document describing the use of the environment present on the virtual machine to generate synthetic tasks and analyze their results. Additionally, this document provides instructions on 1.) how to acquire and install the libsched library 2.) modify parameters of data generation used by the framework.

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 as `artifact.tgz` at:

- Long URL: `https://waynestateprod-my.sharepoint.com/:f:/g/personal/fh3227_wayne_edu/EprJwQXLTvBFo6n5XGh5-PUBBIL87kMSCfwKUTH8vlyx4w?e=Fcp1yY`
- Short URL: `https://bit.ly/2VwM3eF`

4 Tested platforms

The virtual machine is known to operate correctly on a host system with an Intel(R) Core i5-4690K at 3.50 GHz with 16 gigabytes of memory. Two cores and one gigabyte of memory was dedicated to the virtual machine guest. With these resources the results are generated in the range of $[2, 72]$ hours. The variance is due to the nature of synthetic task parameters being generated by pseudo-random algorithms.

5 License

6 MD5 sum of the artifact

9b89f75e677bd37728faac2b920ae240

7 Size of the artifact

5.4 GiB