Evaluation Framework for Self-Suspending Task Systems

Georg von der Brüggen, Milad Nayebi, Junjie Shi, Kuan-Hsun Chen, and Jian-Jia Chen
TU Dortmund University, Dortmund, Germany

***Abstract*—In the real-time systems community, the performance of newly designed scheduling algorithms or schedulability tests is typically examined either by theoretical methods, like dominance relations or speedup factors, or using empirical evaluations. Such empirical methods often evaluate the acceptance ratio of the new algorithm/test compare to other algorithms/tests based on synthesized task sets. However, it is often difficult to perform such a comparison since not all implementations are publicly available, use a different programming language, a different task generator, etc. Hence, for self-suspending tasks, we provide an easy to use evaluation framework with a pre-implemented task generator, some pre-implemented schedulability tests, and an integrated plotting tool. The framework is written in Python and can be extended by including additional schedulability tests.**

I. Introduction

Self-suspension behaviour has analysed in the real-time systems community as it results from multiple application scenarios like 1) cloud offloading in the *Internet of Things* era, 2) multiprocessor resource synchronization, 3) interactions with external devices, e.g., GPUs, accelerators, and I/O devices, etc. In general, self-suspension behaviour occurs when a task leaves the processor without being either finished or preempted by a higher-priority task. Three models for such self-suspension behaviour have been examined:

1) The **segmented** model, where the self-suspension behaviour is described by a precise pattern of interleaving execution segments and suspension intervals.
2) The **dynamic** model, where the self-suspension behaviour is described by two upper bounds on the total worst-case execution time (WCET) and the total suspension time. It assumes that a task can suspend itself an infinity amount of times as long as the upper bound on the total suspension time is respected.
3) The **hybrid** models [4], that provide different tradeoffs between the overly flexible dynamic model and the overly restrictive segmented model, assuming different levels of information in addition to the bounds on WCET and suspension time, i.e., at least the number of suspensions.

A review for the state-of-the-art of self-suspension can be found in [1]. Multiple scheduling algorithms and related schedulability tests have been proposed. Our goal is to provide an easy to use framework that allows to compare these schedulability tests based on synthesized task sets.

II. The Framework

The provided Python framework evaluates scheduling algorithms and the related schedulability tests based on randomly generated implicit-deadline task sets. The evaluation setup can be configured using multiple parameters like the number of tasks, the utilization range that is considered, the relative length of the suspension interval, the number of sets per configuration, etc. The GUI of our framework is shown in Figure 1. Furthermore, it is also possible to configure the evaluation setup of the framework using a config file, running it based on a command line. The results of the tests are stored in numpy files and can also directly be printed into pdfs. The framework also supports to plot previously collected data.

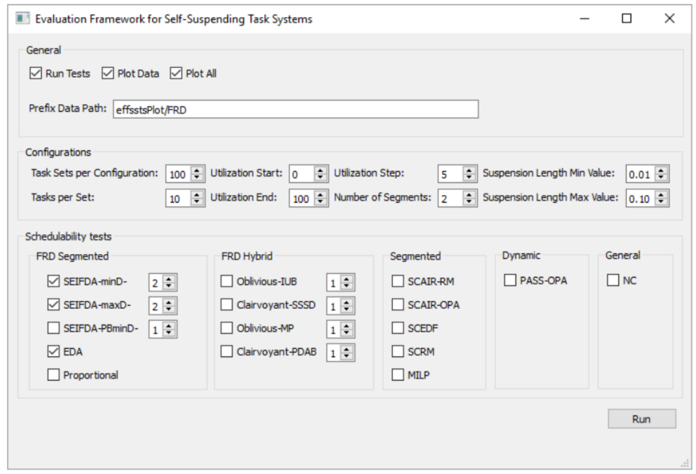

Fig. 1: The GUI of our framework.

The framework already includes schedulability tests for multiple algorithms, e.g., SEIFDA [5], SCAIR [3], PASS [2]. We plan to include additional schedulability tests from the literature, implement additional functionalities, and keep the framework up to date. This is supported by the underlying Python architecture, since Python allows to include, for instance, C or C++ code as well as MILP solvers.

References

[1] Jian-Jia Chen, Georg von der Brüggen, Wen-Hung Huang, and Cong Liu. State of the art for scheduling and analyzing self-suspending sporadic real-time tasks. In *RTCSA 2017*, pages 1–10.
[2] Wen-Hung Huang, Jian-Jia Chen, Husheng Zhou, and Cong Liu. PASS: Priority assignment of real-time tasks with dynamic suspending behavior under fixed-priority scheduling. In *DAC 2015*.
[3] Lea Schonberger, Wen-Hung Huang, Georg von der Brüggen, Kuan-Hsun Chen, and Jian-Jia Chen. Schedulability analysis and priority assignment for segmented self-suspending tasks. In *RTCSA 2018*.
[4] Georg von der Brüggen, Wen-Hung Huang, and Jian-Jia Chen. Hybrid self-suspension models in real-time embedded systems. In *RTCSA 2017*.
[5] Georg von der Brüggen, Wen-Hung Huang, Jian-Jia Chen, and Cong Liu. Uniprocessor scheduling strategies for self-suspending task systems. In *RTNS 2018*.