EVA-rt-Engine: Evaluation, Verification and Analysis Engine for Real-Time Applications Schedulability

Yuri Andriaccio
Scuola Superiore Sant'Anna
Pisa, Italy
yuri.andriaccio@santannapisa.it

Luca Abeni
Scuola Superiore Sant'Anna
Pisa, Italy
luca.abeni@santannapisa.it

Abstract

This paper introduces **EVA-rt-Engine**, a new schedulability analysis and design library released under the GPLv3 open-source license. The tool features the implementation of standard *state-of-the-art* analysis techniques for uniprocessor and global multiprocessor scheduling platforms, as well as more advanced hierarchical scheduling models, analyses, and design tools to aid in the development of component-based real-time software systems. The tool, which is compared with the state of the art in this paper, is currently used to aid the development of a Linux kernel patch that introduces hierarchical scheduling policies. **Demo video link:** https://drive.google.com/file/d/1iACiAZuMBi8Uq0taTkl7pbVjyNcp6jZm/view

CCS Concepts

• **Software and its engineering** → **Real-time schedulability**.

Keywords

Real-Time; Scheduling; Operating Systems

ACM Reference Format:
Yuri Andriaccio and Luca Abeni. 2026. EVA-rt-Engine: Evaluation, Verification and Analysis Engine for Real-Time Applications Schedulability. In *Companion of the 17th ACM/SPEC International Conference on Performance Engineering (ICPE Companion '26), May 04–08, 2026, Florence, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3777911.3799727

1 Introduction

The rise of Cloud-Edge computing and IoT technologies has led to the development of complex systems that require timely and accurate results. To meet real-time constraints, these systems are often built using a component-based architecture, allowing for independent and maintainable units instead of monolithic designs. To meet real-time demands, systems are combined with hierarchical scheduling frameworks, ensuring that components are executed independently and with guaranteed temporal isolation.

To support the development of such systems, a new schedulability analysis tool **EVA-rt-Engine** (*EVA-rt* for short) has been implemented. This library provides schedulability analyses and design space exploration for real-time systems using several *state-of-the-art* algorithms. The tool, distributed under the GPLv3 open-source license, was originally conceived to aid in the development of the Hierarchical Constant Bandwidth Server (HCBS) Linux kernel patchset [1], as a replacement for a custom version of the CARTS tool [5]. The library [1] has since been extended to support schedulability analyses for other platforms and scheduling algorithms, including those unrelated to hierarchical scheduling, with the aim of providing a generic, open-source library to aid in the development of real-time system design and tools.

In the accompanying demo video *EVA-rt* is shown in action with two simple designers, to demonstrate the ease of extendibility of the tool. The first designer finds the minimum number of CPUs necessary to schedule a taskset on global Earliest Deadline First (EDF). The second designer finds the best resource reservation for hierarchical scheduling of the input taskset.

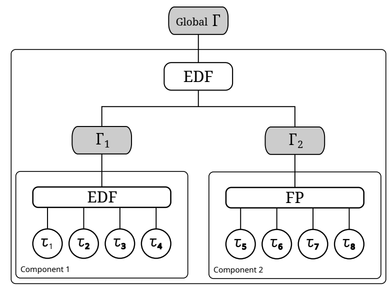

Figure 1: Hierarchical Scheduling example: Component 1 is scheduled using the resource Γ_1 using EDF, while Component 2 is scheduled using the resource Γ_2 using FP.

2 Theoretical Foundation

The schedulability tool presented in this paper is currently built upon the classic *periodic-task* model [4]: tasks τ_i of a taskset τ are tuples $\langle C_i, D_i, T_i \rangle$ which represent a task's Worst-Case Execution Time (WCET), relative Deadline and Period (or minimum interarrival time). Currently *EVA-rt* implements a number of *state-of-the-art* algorithms for *fully-preemptive* scheduling platforms, both for *uni-processors* and *multi-processor* architectures.

The library also focuses on providing schedulability analyses and designers for hierarchical scheduling frameworks, in particular supporting the Periodic Resource (PR) [6] and Multi-Processor Periodic Resource (MPR) [3] models. A simple example, illustrated in figure 1, shows a computing platform which executes two different tasksets/components on the same computing resources. The first taskset is scheduled, on its portion of computing time, using Earliest Deadline First (EDF), while the second taskset is scheduled using Fixed Priority (FP) scheduling. Briefly, the Periodic Resource model $\Gamma(\Pi, \Theta)$ characterizes a partitioned resource that guarantees

[1] Available at https://github.com/Yurand2000/EVA-rt-Engine

ACM ISBN 979-8-4007-2326-1/2026/05
https://doi.org/10.1145/3777911.3799727

allocations of Θ (with $\Theta \in [0, \Pi]$) time units every Π time units on a uni-processor system. The MPR model, an extension of the former, characterizes a resource $\Gamma(\Pi, \Theta, m')$ that guarantees allocations of Θ (with $\Theta \in [0, m'\Pi]$) time units every Π time units with at most concurrency m' on multi-processor systems. Both these models are employed in *EVA-rt* to represent hierarchical resource sharing for tasksets scheduled under EDF and FP, performing schedulability analyses derived from [2, 3, 6].

3 Design of EVA-rt-Engine

EVA-rt-Engine is an open-source library developed using the Rust programming language and released under the GPLv3 license. The goal of this tool is to provide easy-to-access (and extend) implementations of *state-of-the-art* schedulability analyzers and designers, to be integrated in existing tools and applications which aid the development of real-time systems. The Rust programming language was chosen for its strong type system, extensive and easy-to-use ecosystem of libraries, security features, and performance comparable to that of the C language.

The library is mainly a collection of algorithms, but it has been designed to use common interfaces and types to maintain consistency between the different analyses. While interfaces are agnostic of the specific taskset and model types, all algorithms which are based on the classic *periodic resource* model use the type `RTTask` to model tasks and the array reference type `&[RTTask]` to represent tasksets. The custom numeric type `Time` is also extensively used to represent time values.

Schedulability analyses of *EVA-rt* must implement the generic interface `SchedAnalysis` which provides a fixed structure for the analyzer: the `check_preconditions` method must only perform a precondition check on the input taskset, `run_test` must instead perform the schedulability test assuming that all the preconditions are satisfied. A simple example is the classic Liu & Layland [4] Earliest Deadline First (EDF) schedulability test, which requires that all the tasks have *implicit* relative deadlines.

In addition to analyzers, *EVA-rt* also supports designers, i.e., algorithms which search for the best scheduling parameters of the given platform. Examples of scheduling parameters are the minimum number of CPUs for a global scheduling policy, the taskset partition on the platform cores for partitioned scheduling policies, and the server's parameters for hierarchical scheduling.

Designers in *EVA-rt* must implement the `SchedDesign` interface, which provides a fixed structure similar to the analyzer counterpart: it requires the methods `check_preconditions` and `run_designer`. While the first method is self explanatory, the second must produce in output the best scheduling parameters for the given scheduling algorithm and platform assuming that all the preconditions hold.

4 Use Case: Hierarchical Scheduling

EVA-rt-Engine 's initial goal was to provide an open-source analysis framework for hierarchical scheduling, with particular focus on the MPR model [3]. Its first use case was to substitute and improve upon a custom version of the Compositional Analysis of Real-Time Systems (CARTS) tool [5], to be provided alongside the ongoing Linux kernel patch-set [2] on Hierarchical Constant Bandwidth Server (HCBS) [1] for the generation of synthetic tests for the patch-set. Experiments were conducted as part of this study to compare *EVA-rt* with the CARTS tool, specifically on the use case of Fixed Priority hierarchical scheduling provided by the HCBS patchset, using the MPR model.

A first experiment generated 274 tasksets, using UUnifast Discard, to compare schedulability design results and computation times. The taskset vary in the number of tasks between 2 and 16, total utilization between 0.4 and 8, and task periods of 10 to 100 milliseconds. Both tools aimed to find the best interface for MPR models with periods of 5, 20, and 50 milliseconds, with 10 nanoseconds resolution. The presented tool gives equal or better results than CARTS on 818 tests over 820, and better results in 286 of these tests. Performance-wise, it outperforms CARTS on 739 tests over 820, with an average design time of 140ms, 65% less time on average compared to CARTS, on a *Intel Core Ultra-5 125H*.

A second experiment consists in demonstrating the ease of extensibility of *EVA-rt*: while CARTS allows generating the best MPR model given the schedulability period, the *EVA-rt* code can be easily extended to allow for the state space search over a range of possible model periods. The design space search, for periods between 5 and 20 milliseconds, on the aforementioned 274 tasksets takes 1.136 seconds on average for *EVA-rt*, with an average reduction of more than 1700% in computation time to CARTS.

5 Conclusions and Future Works

This paper presents **EVA-rt-Engine**, an open-source schedulability analysis tool for real-time systems. An overview of the theoretical foundations has been given, with particular focus on the hierarchical scheduling frameworks and models currently implemented. Technical design decisions are highlighted. A small comparison study was performed against the CARTS tool [5] to assess the validity of *EVA-rt*'s results and the performance improvement over the CARTS software. A notable use case, aiding the development of the HCBS patchset [1], was additionally presented.

Future plans for this tool involve the implementation of many more *state-of-the-art* analyses and designers, supporting different preemption or task models, with particular focus on other hierarchical scheduling models.

[2]https://lore.kernel.org/all/20251201124205.11169-1-yurand2000@gmail.com/

References

[1] Yuri Andriaccio, Luca Abeni, and Massimo Torquati. 2025. Scheduling IoT Applications in Real-Time Control Groups. In *2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. 01–08. doi:10.1109/DCOSS-IoT65416.2025.00070 ISSN: 2325-2944.

[2] Marko Bertogna, Michele Cirinei, and Giuseppe Lipari. 2009. Schedulability Analysis of Global Scheduling Algorithms on Multiprocessor Platforms. *IEEE Transactions on Parallel and Distributed Systems* 20, 4 (April 2009), 553–566. doi:10.1109/TPDS.2008.129

[3] Arvind Easwaran, Insik Shin, and Insup Lee. 2009. Optimal virtual cluster-based multiprocessor scheduling. *Real-Time Systems* 43, 1 (Sept. 2009), 25–59. doi:10.1007/s11241-009-9073-x

[4] C. L. Liu and James W. Layland. 1973. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *J. ACM* 20, 1 (Jan. 1973), 46–61. doi:10.1145/321738.321743

[5] Linh T. X. Phan, Jaewoo Lee, Arvind Easwaran, Vinay Ramaswamy, Sanjian Chen, Insup Lee, and Oleg Sokolsky. 2011. CARTS: a tool for compositional analysis of real-time systems. *ACM SIGBED Review* 8, 1 (March 2011), 62–63. doi:10.1145/1967021.1967029

[6] Insik Shin and Insup Lee. 2003. Periodic resource model for compositional real-time guarantees. In *RTSS 2003. 24th IEEE Real-Time Systems Symposium, 2003.* 2–13. doi:10.1109/REAL.2003.1253249