technische universität dortmund

computer science 12

Master/Bachelor Thesis

Exploiting concrete present redundancy in schedule encodings

Lars Willemsen
Prof. Dr. Jian-Jia Chen

Otto-Hahn Str. 16
Technische Universität Dortmund
Email: lars.willemsen@tu-dortmund.de
December 4, 2025

In Real Time Systems timing guarantees are of critical importance. Historically real time workload were encoded as something called *cyclic directives*; static tables that expect to periodically repeat. Such static encoding methods have phased out in favor of *dynamic scheduling methods* in most fields, but static schedule encodings are still prevalent in aviation and aerospace hardware[1]. Furthermore, static scheduling is used in new emergent technologies such as TSN (Time Sensitive Networking)[2].

When encoding a schedule $\mathcal{S}$ with an *encoding strategy* ϕ we obtain an **encoding**. However, many of such encodings are larger than they need to be and we aim to compress them based on repeating patterns. There are three distinct forms of redundancy in an encoding classified as:

- **Concrete, present redundancy:** we observe patterns in the encoding that are immediately redundant; jobs that repeat periodically across the entire encoding
- **Abstract, present redundancy:** we observe patterns repeat in the encoding, but not necessarily for the right entries. Some small modifications could lead to concrete redundancy
- **Non-present redundancy:** we observe patterns that aren't fully supported in the encoding and need modifications to become present redundancy. This is a very hard problem to solve

In Figure 1 a schedule encoding $\mathcal{E}_{\mathcal{S}}(\phi)$ is depicted that has a concrete present redundancy for τ_1 due to its repeating pattern. The encoding of this schedule could be compressed by dividing out this pattern. Specifically, the entries for τ_1 from $[4, 6)$ and $[8, 10)$ are *redundant*.

In this thesis,[1] you will familiarize yourself with schedule compression strategies and various common scheduling algorithms. You will attempt to figure out what properties lead to concrete redundancy through

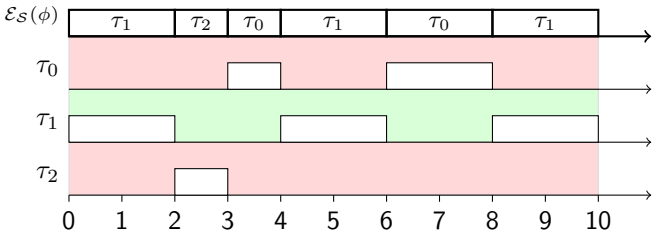

Figure 1: A schedule with a repeating pattern for τ_1

theorems and observations. Furthermore, you will provide experimental results to corroborate on your findings and demonstrate quantitatively that you can isolate this form of redundancy through your analysis.

For master students we can make this problem significantly more demanding by *mechanically proving* your properties and potentially looking into other forms of redundancy than merely concrete.

Required Skills:

- Recommended intermediate/skilled level experience with Python3
- Analytical skills and potentially some knowledge of formal methods
- Proactivity and creativity, capable of reasoning about sensible solutions

Acquired Skills after the thesis:

- Learned how to generate and present relevant data in a convincing way
- Learned how to present analytical results in a formally sound way
- Potentially contribute to active research!

References:

[1] TTTech hardware
[2] TSN Taskgroup

[1] Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.