02225 DRTS Project description

This project focuses on the modeling, simulation, and analysis of an Advanced Driver-Assistance System (ADAS) implemented on a multicore platform using a hierarchical scheduling approach. The ADAS application has both critical and non-critical functions, and they are modeled using a set of periodic and sporadic tasks, respectively. The goal is to develop tools to simulate the system's behavior and analyze its schedulability, ensuring that all tasks meet their deadlines.

1 System Model

The students are responsible for developing and motivating their own models, and for creating test cases for their project. We will provide a few example test cases and information related to ADAS applications in `DTU Learn/Content/Project`, which can serve as a starting point.

- **Hardware model:** The system runs on a multicore platform, with cores of different performance. The performance will affect the worst-case execution times of tasks[1]. Each core runs its own hierarchical scheduling system, and the analysis techniques must be applied independently for each core.
- **Scheduling policies:** We use a hierarchical scheduling approach on each core. Components in the hierarchical system can use different scheduling algorithms. In this project, components can use either Earliest Deadline First (EDF) or Fixed-Priority Preemptive Scheduling (FPS), where Rate Monotonic (RM) can be used as a priority assignment policy for FPS.
- **Resource supply model:** We use the Bounded Delay Resource (BDR) model to represent resource allocation between hierarchical levels[2]. A BDR model is characterized by parameters (α, Δ) where α is the *resource availability factor* (representing the fraction of processing time guaranteed on average) and Δ is the *maximum delay in resource allocation* (bounding the worst-case delay in getting the allocated resource). Each component on each core will have its own *BDR interface*, which abstractly defines its resource demand using (α, Δ).
- **Application model:** The system consists of hard periodic tasks and soft sporadic tasks. Hard tasks have fixed periods and deadlines and are scheduled directly by the top-level scheduler. The soft sporadic tasks have minimum inter-arrival times (MIT) and deadlines. They are handled by the second-level scheduler. Tasks are statically assigned to cores.

2 Project Requirements

Any programming language or libraries can be used. Guidelines on project structure, scope, and the use of Generative AI are provided in `DTU Learn/Content/Project`, where you will also find advice on which aspects to focus on. Your task is to develop a simulator and an analysis tool and compare their results.

1. **Input Model:** Your simulator and analysis tool should read input files specifying:
   - The application model (hard periodic tasks, soft sporadic tasks with their parameters)
   - The platform model (cores with their performance factors)
   - Task-to-component assignments (the hierarchical structure)
   - Task-to-core assignments
   - For the simulator: initial resource supply parameters for components. For the analysis tool, you will calculate resource supply parameters based on the component interfaces.

[1] A simple solution is to consider a speed factor f for a core and multiply the given *WCET* with f. For example, if a core is 50% slower, then the given *WCET*, corresponding to an $f = 1$, would be *WCET* $\cdot\, 0.5$.

[2] This model was presented in lecture 3 and it is covered in Section 3 of the chapter "Hierarchical Scheduling" from *Handbook of Real-Time Computing*, 2022 (available in `DTU Learn/Content/Project`).

2. **Simulator:**
    - Simulates the execution of tasks on each core according to the hierarchical scheduling approach.
    - Uses the provided initial resource supply parameters to allocate processing time from parent components to child components.
    - Reports observed response times (average and maximum) for each task and resource supply task utilization metrics.
    - Justify your modeling decisions and simulation parameters.
3. **Analysis Tool:**
    - Reads the same input files as the simulator. For the analysis tool, you will **calculate the BDR interface parameters** (α, Δ) for each component. These parameters abstractly represent the resource *demand* of the component. The calculation will be based on its resource demands (workload of tasks within the component) and the component's scheduling algorithm (EDF or FPS with RM priority assignment)). The goal is to find the smallest possible Δ (delay) and corresponding α (utilization) such that a BDR resource with these parameters can satisfy the component's resource needs. This may involve an iterative process to find suitable (α, Δ) values that ensure the Supply Bound Function (SBF) is always greater than or equal to the Demand Bound Function (DBF) of the component's workload. These BDR interface parameters (α, Δ) will then define the resource supply tasks for each component at the parent level. Theorem 3 (Half-Half Algorithm) in Section 3.3.3 of the handbook chapter describes how to transform these BDR interfaces into resource supply tasks, e.g., if needed for your simulation.
    - Implements a schedulability analysis to determine if the tasks are schedulable on each core.
    - Reports the schedulability results. Note that the *core analysis techniques* in this project focus on schedulability tests *based on demand and supply bound functions*. While these techniques can be adapted to estimate Worst-Case Response Times (WCRTs), *explicitly computing WCRTs* is suggested as an optional extension.

3 Analysis Techniques

You are expected to use the compositional analysis techniques from the chapter "Hierarchical Scheduling" from *Handbook of Real-Time Computing*, 2022 (available in `DTU Learn/Content/Project`). Specifically, you should refer to **Section 3: Compositional Framework for HSS** with focus on **Section 3.3: Compositional Framework Based on Bounded Delay Resource Model**.

The following equations and theorems from the chapter are particularly relevant for this project. These analysis techniques provide a compositional approach to analyze the schedulability of your hierarchical system by considering the resource demand at each level and the resource supply from the upper level. The BDR model and Half-Half Theorem allow for compositional analysis by abstracting resource demands and providing a way to implement resource supply in hierarchical scheduling.

- **Demand Bound Functions (DBFs): Section 3.1**. You will need to calculate the Demand Bound Functions for both EDF and FPS schedulers.
    - For workloads scheduled by EDF, use **Equation (2) or (3)** to calculate $\mathrm{dbf}_{\mathrm{EDF}}(W, t)$. These equations calculate the maximum resource demand of a task set W under EDF within a time interval t, considering implicit or explicit deadlines respectively.
    - For workloads scheduled by FPS, use **Equation (4)** to calculate $\mathrm{dbf}_{\mathrm{FPS}}(W, t, i)$. This equation calculates the worst-case demand of task τ_i under FPS, including the demand from higher priority tasks.
- **Supply Bound Function for BDR: Section 3.3.2**. Since you are modeling the resource supply using the Bounded Delay Resource (BDR) model, you will need to use the Supply Bound Function for BDR.
    - Use **Equation (6)** to calculate $\mathrm{sbf}_{\mathrm{BDR}}(R, t)$. This equation computes the minimum resource supply provided by a BDR resource $R = (\alpha, \Delta)$ within a time interval t.

- **Hierarchical Scheduling with BDR: Section 3.3.3**. To analyze the schedulability of components within the hierarchical system:
  - Use **Theorem 1** to check if a set of BDR interfaces can be scheduled by a parent BDR interface.
  - Use **Theorem 3** (half-half algorithm) to transform BDR interfaces into resource supply tasks for scheduling at the parent level. For each component on each core in the hierarchical system, you must compute a BDR interface (α, Δ) that abstracts its resource demands. Note that the Half-Half Theorem is used to derive the parameters (budget C_{supply} and period T_{supply}) of the resource supply tasks (which you need to simulate) from the BDR interface (α, Δ), not to calculate the BDR interface itself.

4 Optional Extensions

- **Worst-Case Response Time Analysis:** The analysis techniques in the project focus on schedulability checks but do not explicitly compute the WCRTs. An advanced extension would be to develop WCRT calculation methods for tasks in hierarchical scheduling with BDR. For a task τ_i in a component with BDR interface (α, Δ), the WCRT must account for both the task's execution demand and the limited resource availability[3].
- **Different resource models**: Explore using the Periodic Resource Model (PRM) or Explicit Deadline Periodic (EDP) resource model instead of BDR and compare their effectiveness in terms of schedulability, resource utilization, and complexity of analysis.
- **Inter-task Communication:** Model and analyze communication delays between tasks, and discuss how these delays could be incorporated into the compositional schedulability analysis framework.
- **System Optimization:** Consider:
  - Core assignment optimization
  - Resource model parameter tuning for better responsiveness

[3] WCRT for FPS: Shin, Insik, and Insup Lee. "Periodic resource model for compositional real-time guarantees." *RTSS: Real-Time Systems Symposium*, 2003.
WCRT for EDF: Guan, Nan, and Wang Yi. "General and efficient response time analysis for EDF scheduling." *DATE: Design, Automation & Test in Europe Conference & Exhibition*, 2014. The PDFs are available at `DTU Learn/Content/Project`