Interference analysis algorithm in $O(n^2)$ for many-core response time computation, able to quickly schedule thousands of tasks

Scaling Up the Memory Interference Analysis for Hard Real-Time Many-Core Systems

Maximilien Dupont de Dinechin¹, Matheus Schuh², Matthieu Moy⁴, Claire Maiza²
¹ Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP
² Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG
³ Kalray S.A.

1 Interference Calculation Problem
- Task graph (DAG)
  - Obtained from Data-Flow languages
  - Dependencies, order, mapping and WCET in isolation are given

- Access to shared resources
  - Controlled by hardware arbiters
  - Huge impact on execution time

- Static Time-Triggered Schedule
  - Release dates for each task
  - WCRT of the program

2 Original Algorithm
- Global fixed-point iterations
- $O(n^4)$ complexity
- Interference and execution time are interdependent problems

3 Our Improved Algorithm
- Schedule computed incrementally
- $O(n^2)$ complexity
- Scales for thousands of tasks

Dead: $n_3$, $n_5$, $n_8$, $n_6$  \[ t \geq \text{end date} \]
Alive: $n_1$, $n_4$, $n_7$, $n_9$  \[ \text{rel date} \leq t < \text{end date} \]
Future: $n_1$, $n_2$, $n_{10}$  \[ t < \text{rel date} \]