Mixed Criticality Scheduling with Memory Regulation

Motivation
- A trend towards running different mixed-criticality applications on a single core.
- Mixed-criticality systems require certification.

**Issue**
- COTS multicore platforms have contention over shared resources
- Hard to analyze WCET of a task due to low predictability. This leads towards pessimistic bounds for WCET, resulting in low resource utilization.

**Proposed Approach**
- Memory regulation technique (Single Core Equivalence - SCE) is adapted. This approach did not explicitly consider mixed criticalities until now.

**Colored Lockdown**
**PALLOC**
**SCE**
**MemGuard**

Contribution
- Vestal’s based L-WCET and H-WCET for 2 level criticality tasks in 2 modes (L and H).
- Ekberg and Yi’s deadline scaling technique extended by incorporating the effects of contention under SCE’s memory regulation scheme.

**What more!**
After the mode change, L-tasks are dropped. Thus the resources allocated to them (like cache partitions) are idle.

Utilize the L-Tasks’ resources for the available H-tasks. Based on Integer Linear Programming (ILP) model taking into account the # pages in the last level cache for both modes, and resource utilization.

**Deadline scaling (No low level effects)**
+ **SCE** (No mixed criticalities)

**Scheduling isolation with mixed criticalities**

**Future Work**
- Design of simulator for the implementation of proposed work.
- Implementation of dbf-based schedulability tests.

**References**