DAC YF

Multi-core Safety-Critical Memory Manager

Alexy Torres Aurora Dugo, Gabriela Nicolescu, Polytechnique Montreal

DESIGN AUTOMATION CONFERENCE
FROM CHIPS TO SYSTEMS – LEARN TODAY, CREATE TOMORROW

Context

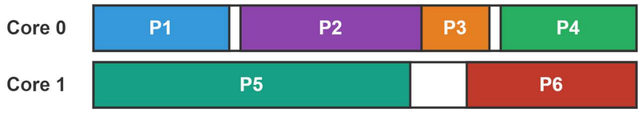
Avionic software -> **real-time, deterministic and safety-critical**

Hard partitioned CPU time / resources

Interferences appear in Multi-Core architectures
- Shared resources mechanisms
- Contention

Processors evict other cores' data from shared **caches**

Shared **bus** get contended: arbitration generates **unpredictable delays**



Major Time Frame scheduling table

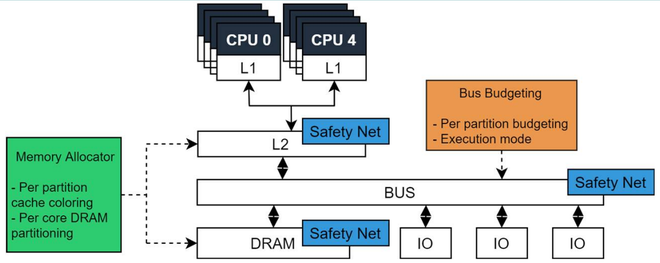
Objectives

Bound bus contention to ensure QoS
- Memory bandwidth limitation

Remove shared cache interferences
- Application space coloring / partitioning

Ease certification process
- Set of formalized constraints
- Run-time monitoring

Approach



Distributed **monitoring** mechanism
Rely on **Commercial Off-The-Shelf** hardware
Private mechanism – Each core managed **independently**

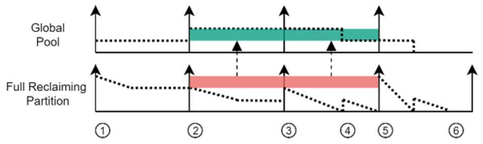
Memory Coloring

Physical address multiplexing

Physical address **multiplexing**
- DRAM Bank bit selection
- Cache Set bit selection

Virtual / Physical translation used to choose where to place the application's data

Bus Access Throttling



Throttling mechanism timeline

Per application **throttling**
- SOTA: per CPU
- Allows **finer** configuration

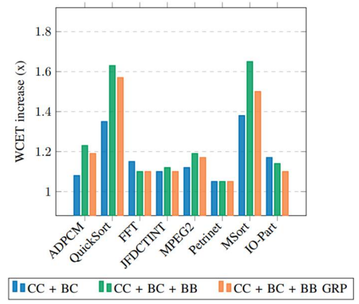
4 **reclaiming** modes

Run-Time Monitoring (Safety Net)

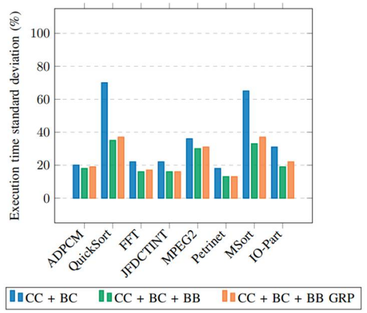
Formal constraints – Used in formal system's verification
Run-time monitoring – Recovery in case of unhandled interference

$$\forall t \in [0; Q], G(t) + \sum_{\forall A_i} A_i^B \times R(A_i, t) \leq B_{max}$$

Results



Execution time increase



Predictability analysis

Increased predictability by 68% on average

Increased execution time (+22.3%) on 2 cores

Still **advantageous** compared to single core (less than 100% slowdown)

Small overhead introduced by the isolation mechanism (less than 2%)

AMP architecture ensures **scalability**

Conclusion

Our work ensures correct **isolation** of shared resources and removes state-of-the-art limitations

The presented methodology is applied in a commercial RTOS, **extended to 4 cores**

The **low overhead** makes our proposition applicable without hard constraints on the system.

NSERC CRSNG CRIAQ MANNARINO HES Lab POLY MTL