Panel Discussions

Towards ***Bespoke Graceful Degradation*** in Mixed-Criticality Systems (Panel Presentation)

6th International Workshop on Mixed-Criticality Systems (WMC)
at the 39th IEEE Real-Time Systems Symposium, Nashville, TN, USA
Tuesday, December 11th, 2018

Chris Gill
Department of Computer Science and Engineering
Washington University in St. Louis, MO, USA
cdgill@wustl.edu

Can We Manage *Utilization* Gain/Loss Gracefully?

Task	E_i	Z_i	$m_i[0]$	$m_i[1]$	$m_i[2]$
τ_1	5	0	8	*6*	*4*
τ_2	4	0	6	*4*	*2*
τ_3	3	1	4	**6**	*2*
τ_4	2	1	4	**6**	*2*
τ_5	1	2	2	2	**6**

$$U_i[j] = \frac{C_i[j]}{D_i[j]}$$

$$m_i[j] = \left\lceil \frac{C_i[j] - L_i[j]}{D_i[j] - L_i[j]} \right\rceil$$

degraded

overload

nominal

- Nominal, overload, degraded ranges of utilization
  - » ***(increasing?)*** nominal utilization below criticality level Z_i
  - » ***(maximum?)*** overload utilization at task's designated level Z_i
  - » ***(decreasing?)*** degraded utilizations at even higher levels

Cut (Utilization) to Fit: One Size Doesn’t Fit All

- Can run same amount of work less frequently
  - » Orr et al. RTNS ’18 (linear FEM component of RTHS)
  - » Su et al., DATE ‘13, RTCSA ’14 (alternative periods)
- Can run less work in same amount of time
  - » Liu et al. RTSS 2016; Huang et al. RTNS ’18 (mprcs cmptng)
  - » Anytime algorithms more generally (declare victory and retreat)
- Exploiting both forms of tailoring at once?
  - » Need to define carefully ***how much*** to modify work vs. rate
  - » E.g., run an ε-less-precise calculation λ-slower (tune ε vs. λ)

Say What You Mean, Mean What You Say

- Specify utilization at every criticality level for each task
  - » Highest-criticality tasks already will do this under Vestal model
  - » ***All*** lower-criticality tasks ***must*** do this if they can't be dropped

- Co-design parameters, constraints, objectives carefully
  - » If (and only if) platform allows, remove unnecessary pessimism
  - » Design to minimize each task's footprint at each criticality level
    - Minimum should still ***meet the task's constraints***
    - Higher should ***improve optimization objectives*** monotonically

!! !? ?! ??

Feedback is Welcome

Mixed-Criticality Scheduling with Varying Processor Supply in Compositional Real-Time Systems

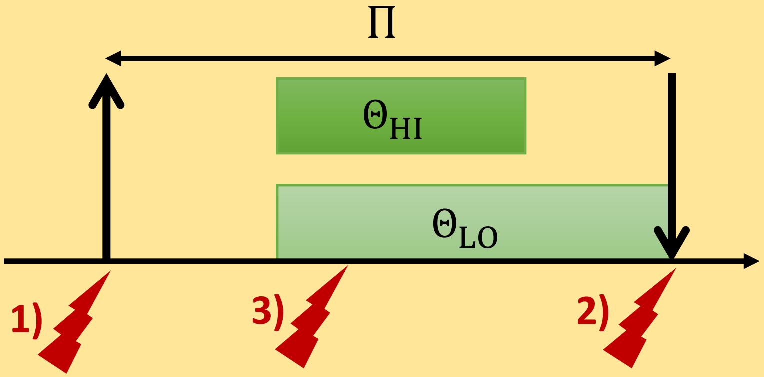
Kecheng Yang, Department of Computer Science, Texas State University

Uncertainties may trigger a criticality **mode switch** (e.g., LO to HI) in MCS

WCETs [*e.g.*, Vestal *RTSS* '07]
Periods, Deadlines [*e.g.*, Baruah, *RTSS* '16]
Processor Speed [*e.g.*, Baruah and Guo, *RTSS* '13]
Processor Supply? from partially available processor(s)

When?
How are the uncertainties **monitored**?
Signal the reduction of supply
1) **before** the replenishment period
2) **after** the replenishment period
3) **during** the replenishment period

E.g., periodic resource model [Shin and Lee, *RTSS* '03]
Θ units budgets every Π time units
uncertainties on the budgets (Θ_{HI} v.s. Θ_{LO})?



Potential avenues:
In additional to **bounding and shaping** the **demand** from the tasks [Ekberg and Yi, *ECRTS* '12], it might need to **bound and shape** the **supply** from the processor(s) as well.

Other resource model?
Multiple Π in addition to Θ?
Multiprocessor?

Challenges of MCS

Iain Bate

University of York

Future Embedded Systems

- **Common themes based on discussions with avionics, automotive and other manufacturers**
- **System consists of platform plus other hardware, e.g. sensors and actuators**
- **Platform can mean**
  - Processing platform - Processor + software
  - Whole system platform, e.g. the car or aircraft
- **Environment is the context the system operates and includes the users**

Key Challenges

- **Realities throws up lots of research and implementation challenges!!**
- **For example, shock and horror!!**
  - Real systems don't have independent tasks
  - Dependencies:
    - Explicit, e.g. transactions
    - Implicit, e.g. caches etc
  - Real systems have a RTOS which analysis needs to allow for
  - The magic C_i figure is not as simple as it seems
- **Systems will feature more un-certainties**
  - These should be welcomed and embraced rather than avoided

Key Challenges

- **Need to build confidence → digital twinning**
  - Start off with simulation
  - Move to rig testing
  - Progress to pre-deployment testing
  - Continue into service
  - Validate and refine at each stage
  - As confidence grows, then trust and capability can be extended
- **(Whole system) platform will have greater connectivity**
- **Maintenance cycles need to be shorter**
  - Need more data to support maintenance
  - Cloud-based analytics

Processing Platform

- **Multi-core – task allocation needs understanding of shared resource usage**
- **Mixed-criticality versus Resilient Scheduling**
  - Need an effective balance between efficient use of resources and achieving certification
  - Mixed-criticality doesn’t deliver this both in “name” and the model
  - ***Functional hazard-related criticality and ability to skip some jobs not directly linked***
  - Systems should meet their requirements
  - It is rarely acceptable to say 5% service is lost
  - Loss of service (duration and arrival rate) needs to be understood and specified

Processing Platform

- **Reduce RTOS overheads**
  - Reduce the number of tasks
  - Reduce the number of context switches
- **Where does C_{Lo} and C_{Hi} come from?**
  - We have lots of data but decisions are needed
  - Need to give right balance between flexibility and how often mode changes happen
  - Being able to use the distribution might be better
- **Task allocation**
  - Needs to support the previous points
  - For example, try to segregate a task sensitive to shared resource usage X from a task using varying amount of X

System

- An appropriate model is needed from which code can be generated
- Models have to allow for the real behaviours of the platform
- Code from a number of models need to be integrated
- Resultant code needs to be efficient on targeted platform
- Review and change of the models is an issue
- Models have to be shown to be valid

Environment

- System and platform models should result in appropriate response
- Balance between efficient and effective system, and the need for dependability including safety

Resource-efficient timing isolation for adaptive mixed-criticality systems with multiple types of shared resources.

- Vestal’s adaptive mode-based model promotes resource efficiency.
    - At mode change, processor resources are taken away from lower-criticality (or lower-importance) tasks and given to the higher-criticality (importance) tasks.
    - The same approach could be applied to other resources
        - Memory bandwidth, caches etc.
- Single-Core Equivalence framework (SCE)
    - Partitions access to shared resources or regulates access to them (Memguard, cache-partitioning, Palloc etc).
    - Meant to facilitate the portability of applications to multicores.
    - It can also make mixed-criticality applications more timing-predictable.
- Challenge: readjust tasks’ resource access budgets at mode-change AND do this for multiple resource types.
    - Memory access budgets, cache partitions, amount of memory pages locked etc…