System Mode Changes - General and Criticality-Based

Alan Burns

University of York, UK

Introduction

- Many protocols aimed at MCS involve a criticality mode change
- System starts in LO-crit mode
  - all tasks meet their deadlines
- If any HI-crit task executes for more than C(LO) then switch to HI-crit mode
  - all HI-crit tasks meet their deadlines
- If appropriate switch back to LO-crit mode

Criticality-Based Mode Change

- Is this the same as a general mode change?
- This paper aims to review general mode changes and classify a criticality-based mode change
- It also addresses the issues arising from a system with general and criticality-based mode changes

General Mode Changes

- Type – what are the different classes
- Trigger – what causes a mode change
- Definition – the software, and its operational parameters, that constitute a mode
- Protocol – how is the mode change managed
- Attributes – properties of modes
- Analysis – verify the timing properties of the system during a mode change

Type

- Normal Functional Modes
- Exceptional Functional Modes (sometimes called *Operational mode*)
- Degraded Functional Modes (sometimes called simply *Graceful Degradation*)

Trigger, or request

- timed-triggered
- event-triggered

Definition of Task Behaviour

- run unaffected in both modes
- run only in the old mode
- run only in new mode
- run in both modes but have their defining parameters changed
  - could be the criticality classification that changes
  - could be WCET

Protocol

- Immediate
  - abort or abandon old mode tasks
- Bounded
  - wait for old mode ro finish
- Phased
  - both modes active for a bounded time

Attributes

- re-entrant (best not to abort)
- one-shot (can abort)
- initial
- sink
- connected or strongly connected

Analysis

- easy for Immediate
- quite easy for Bounded
- quite difficult for Phased
- very difficult for Phased during Phased

What is a Criticality Mode Change?

- For this talk just two levels, HI and LO
- System starts in LO, all tasks execute within C(LO)s
  - all tasks meet their deadlines
- Any task executed for more than C(LO) then switch to HI
  - all HI-crit tasks meet their deadlines
- If appropriate switch back to LO

LO to HI

- LO is initial
- HI is a sink (but may never be entered)
- Both are one-shot
- Event-triggered

LO to HI

- Immediate
- Some tasks only run in old mode
- Some tasks run in both but defining parameters change
- Graceful Degradation
  - if there are L criticality levels, L-1 are degraded modes

HI to LO

- LO is initial
- LO and HI are re-entrant
- Event-triggered
- Bounded

HI to LO

- Some tasks only run in new mode
- Some tasks run in both but defining parameters change
- Exceptional/Operational

Fault Tolerance

- Following partial failure, degraded service
- Followed by active recovery, or
- System (channel) re-start
  - cold or warm re-start
  - move from sink to initial

Both General and Criticality Modes

- Gets complicated
- Is a real problem
- Some modes will be degraded modes for both
- See paper for some initial thoughts

Conclusion

- Mode changes are a reality in many systems
- They can be characterised in many ways
- I believe the closest match for a criticality mode change is graceful degradation followed by ‘restart’ or an operational mode change