Evaluating Mixed Criticality Scheduling Algorithms with Realistic Workloads

David Griffin, Iain Bate, Benjamin Lesage, Frank Soboczenski

Structure of the Presentation

- **Case for Scenario-Based Assessment (SBA)**
- **Review of ECRTS work**
  - ECRTS 2015 paper that included a SBA
- **A Case for a better fault model**
- **Creating an improved Task Set Generator (TSG)**
- **Does having a different TSG make a difference to our results from ECRTS?**

Case for SBA

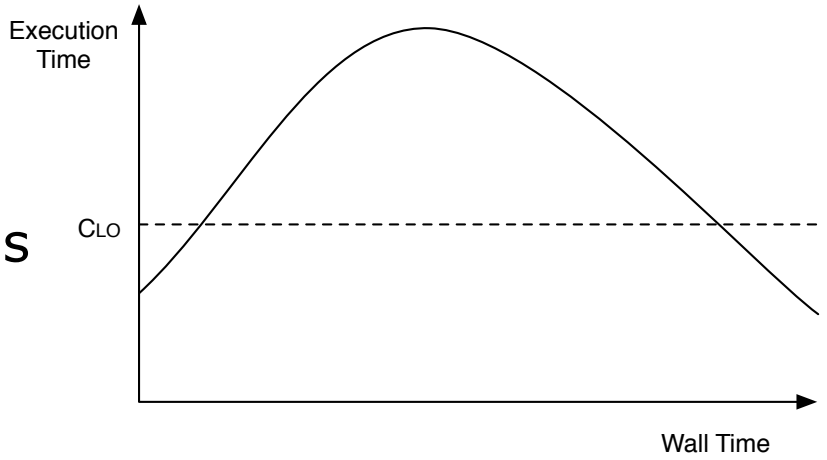
- **Static analysis is perfect for definitive answer**
  - Assuming our proofs are valid ☺
- **Its of less use if we want to know**
  - How much service Lower-Criticality Tasks (LCT) get?
  - That is their availability, i.e. how often we enter a Higher-Criticality Mode (HCM) and for how long?
  - For a safety case this is important information as lower-criticality doesn't always mean no criticality
  - E.g. an aircraft's navigation system is unlikely to be the highest level of system
- **In ECRTS 2015, Alan, Rob and I showed how SBA could provide useful evidence**
  - Showed how often LCTs had service
- **Note the concept of probabilistic guarantees is not yet accepted in many domains but MCS needs it**
  - The concepts of modes, including for fault tolerance, is accepted

Review of ECRTS Work

- **Paper showed that**
  - Bailout Protocol (BP) gave LCTs better service than AMC+
  - Showed having slack time helped both BP and AMC+
- **TSG based on Uunifast**
  - Independent identically distributed (i.i.d.) timing failures
  - Failures are exceedances of a WCET value, e.g. C_{LO}
  - Initial failure rate chosen as 10^{-4}
  - Others have suggested an initial failure rate of 10^{-16}
  - Uniform random used to generate execution times

A Case for a Better Fault Model

- **For individual tasks i.i.d. failures unrealistic**
  - Evidence from industry timing failures normally caused by fault accommodation code
  - E.g. sensor and comms errors due to interference, or state unexpected and untested
  - Failures very unlikely to be for a single cycle
- **Suggest more realistic fault model features:**
  - Initial Failure Rate (IFR)
  - Duration of failure
  - Size and shape of failures



A Case for a Better Fault Model

- **The pWCET normally fits different distributions to data, e.g. Gumbel**
- **Therefore should ideally select samples from something other than uniform random**

A Case for a Better Fault Model

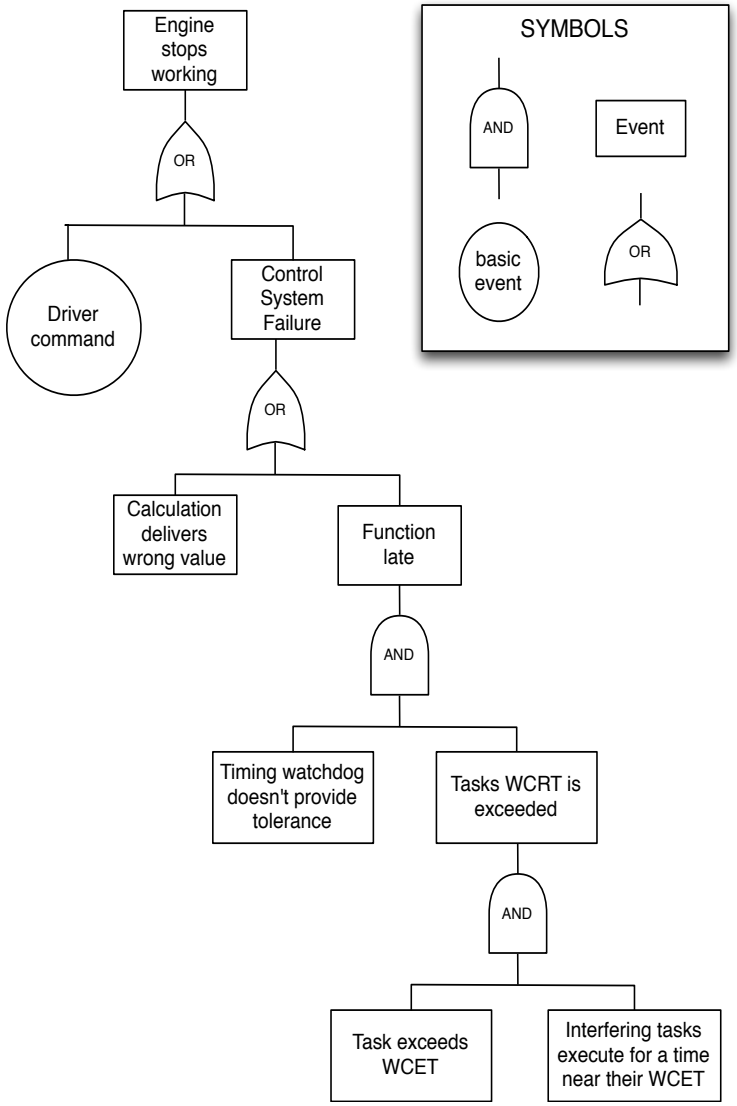
- **Initial failure rate should come from system safety analysis**
  - Logic behind previous 10^{-16} figure is as follows
  - Some standards mention one hazardous event in 10^{9} operational hours for activities with highest criticality
  - A software task may execute every few milli-seconds which means over a million times an hour
  - Therefore IFR should approaches once in every 10^{16} releases of the task
- **Observations typically stop before 10^{5}**
  - High WaterMark (HWM) normally tight even if not sound
  - Due to gap between 10^{5} and 10^{16} means, are we effectively guessing?

A Case for a Better Fault Model

- **If a single point of failure can lead to a hazard then certification standards demand extra level of rigour**
- **Software can't itself cause a single point of failure**
  - There has to be physical devices involved
  - Nobody would trust software that much
  - Systems typically have a timing watchdog
  - Reasonable to reset computer-based system especially if there are replicas without a common-mode failure

A Case for a Better Fault Model

- **Simplified example**
- **Not shown but single missing value wouldn't stop engine**
  - Algorithms designed to be tolerant
  - Previous value could be used, e.g. use the same fuel valve setting
  - Engine has inertia
- **Timing watchdog (TW) provides tolerance**
  - Accepted MTBF is 10^6 hours
  - Both TW and task(s) have to fail
- **Exceeding a task's WCRT may involve a number of tasks**
  - Some faults may affect multiple tasks, however may be better to reset quickly
  - We have analysis to help understand dependencies



A Case for a Better Fault Model

- **Discussions with industry suggest for MCS:**
  - C_{LO} could be HWM based on comprehensive testing
  - IFR for C_{LO} would therefore be somewhere between 10^4 and 10^5 based on standard testing literature
  - IFR for C_{HI} could be around 10^6 as
    - Software not expected to be more reliable than TW
    - Plenty of fault tolerance
    - No point having unusable (due to pessimism) WCET
  - With "controlled experiment" could show these values relate well to actual WCET
    - Controlled experiment gives actual WCETs (RTNS 2015)
  - Note - Not every system continuous control

A Case for a Better Fault Model

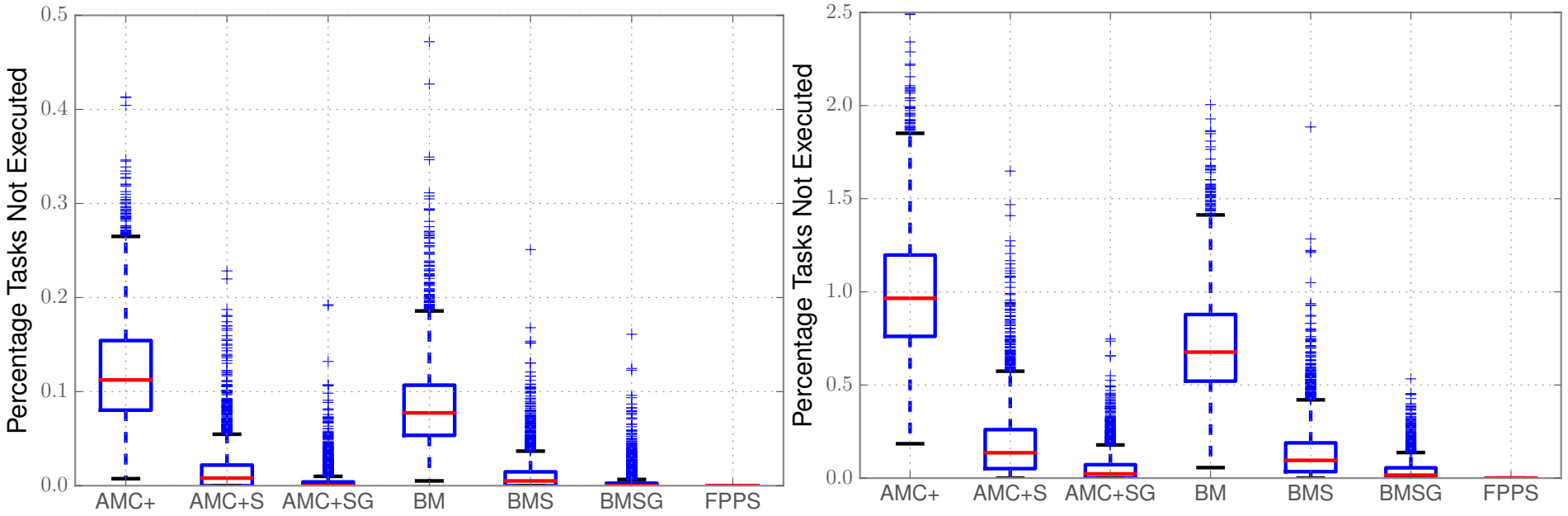
- **QUESTION 1: How do you generate a target reliability (IFR and duration) for WCRT?**
- **QUESTION 2: Given a target reliability for WCRT, can we derive appropriate target reliability for each task's WCET?**
- **QUESTION 3: Do probabilistic approaches change the way we do timing tolerance?**
- **QUESTION 4: Can the regulatory authorities change their policy about probabilistic guarantees?**

Creating an Improved TSG

- **Some previous work that has used MBPTA to generate execution time profiles**
  - Often based on Cumulative Distribution Functions (CDF)
- **Previous work (RTNS 2015) generated a fault model**
  - Used lossy comprehensive and Markov chains to understand the duration and magnitude of failures
  - Failure threshold (in terms of exceedance threshold) could be chosen
- **Combined to form a TSG called DepET**
  - Basis was to set failure threshold at different levels
  - Use fault model in bands from one threshold to the next
  - Source can be found at http://rtslab.wikispaces.com/Experiment+Source+Code

Does a Different TSG Change Results?

- **Repeated some of the trials from ECRTS 2015 with DEPET**
  - LHS: Independent failures, RHS: Dependent failures, Both: IFR = 0.1%
- **Trends were similar**
  - i.e. Scheduling policy X gave Y% better service to LCTs than policy Z
- **Absolute values of service were different**
  - In partly due to IFR meaning dependent case had many more failures



Summary

- **I think we have to go beyond static analysis**
- **"Real" industrial needs raises some cool academic challenges**
  - Only raised a few here
- **As soon as we do, we either**
  - Make simple assumptions leading to answers with questionable worth
  - We hit some very complex (interesting) problems
- **Biggest issue is possibly changing industrial and regulatory practice**
  - Best to ignore until we have solid solutions
  - Note - its not what standards say that matters but what is expected in meeting them

Acknowledgements

- **Patrick Graydon of NASA for comments**
- **Mälardalen University SYNOPSIS project**
- **EPSRC funded MCC**
- **EU funded PROXIMA**