On Preemption Delay and WCET handling on Reservation frameworks

José Marinho

20th of May, 2010

Instituto Superior de Engenharia do Porto

Research Centre in Real-Time Computing Systems
FCT Research Unit 608

Outline

Introduction

In Real-Time systems:

- Predictability of execution required (Hard RT)
- Requirements less restrictive for Soft RT
- Hardware platforms get increasingly complex
- Strict assurances have the drawback of (sometimes) unnecessary resource wastage
- temporal behaviour of HRT must not be geopardized by other HRT or SRT execution.

Introduction

In Real-Time systems:

- Predictability of execution required (Hard RT)
- Requirements less restrictive for Soft RT
- Hardware platforms get increasingly complex
- Strict assurances have the drawback of (sometimes) unnecessary resource wastage
- temporal behaviour of HRT must not be geopardized by other HRT or SRT execution.

Introduction

In Real-Time systems:

- Predictability of execution required (Hard RT)
- Requirements less restrictive for Soft RT
- Hardware platforms get increasingly complex
- Strict assurances have the drawback of (sometimes) unnecessary resource wastage
- temporal behaviour of HRT must not be geopardized by other HRT or SRT execution.

Introduction

In Real-Time systems:

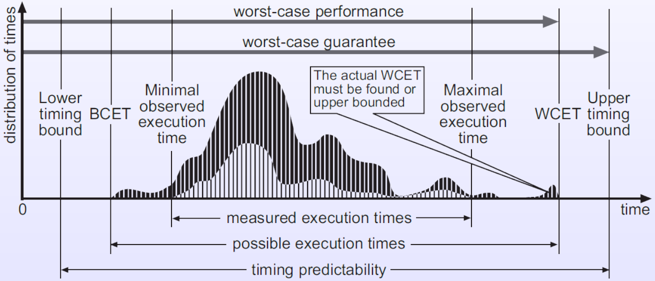
- Predictability of execution required (Hard RT)
- Requirements less restrictive for Soft RT
- Hardware platforms get increasingly complex
- Strict assurances have the drawback of (sometimes) unnecessary resource wastage
- temporal behaviour of HRT must not be geopardized by other HRT or SRT execution.

Introduction

In Real-Time systems:

- Predictability of execution required (Hard RT)
- Requirements less restrictive for Soft RT
- Hardware platforms get increasingly complex
- Strict assurances have the drawback of (sometimes) unnecessary resource wastage
- temporal behaviour of HRT must not be geopardized by other HRT or SRT execution.

WCET Estimation



- difficult to attain in modern architecture, for complex programs.
- impossible to test all possible states in complex programs
- values obtained will always be an estimate for real-world examples

WCET Estimation

- WCET estimations obtained for single task execution or static analysis

WCET Estimation

- WCET estimations obtained for single task execution or static analysis
- systems will generally be preemptive though

WCET Estimation

- WCET estimations obtained for single task execution or static analysis
- systems will generally be preemptive though
- usually preemptions are assumed to have no cost,
- or caches are assumed entirely dirty.

WCET Estimation

- WCET estimations obtained for single task execution or static analysis
- systems will generally be preemptive though
- usually preemptions are assumed to have no cost, vastly simplifying assumption
- or caches are assumed entirely dirty. overestimate

preemption delay:

WCET Estimation

- WCET estimations obtained for single task execution or static analysis
- systems will generally be preemptive though
- usually preemptions are assumed to have no cost, vastly simplifying assumption
- or caches are assumed entirely dirty. overestimate

preemption delay:

- static: code swap, register and control words store, etc..

WCET Estimation

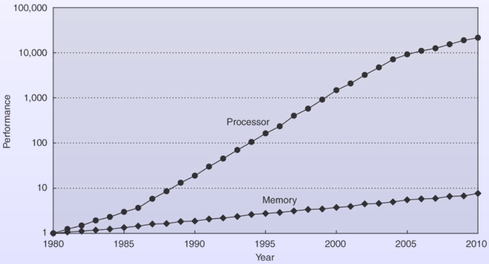
- WCET estimations obtained for single task execution or static analysis
- systems will generally be preemptive though
- usually preemptions are assumed to have no cost, vastly simplifying assumption
- or caches are assumed entirely dirty. overestimate

preemption delay:

- static: code swap, register and control words store, etc..
- dynamic: Caches, TLB, branch prediction

WCET Estimation

- from early times a big difference gap has been observed between processors and memories



- there's the need for a big amount of fast memory and the two requirements clash
- several hierarchy layers are thus introduced

Caches

Fast memory, generally located on-chip
Caches can either be:

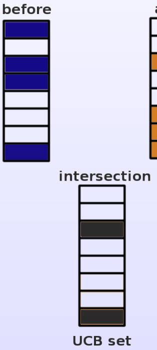
- direct mapped
- set-associative, with possible replacement policies
- fully associative (similar concept to scratchpads)

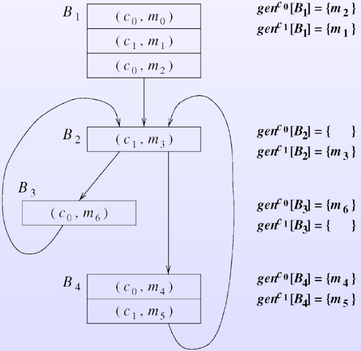
if no partition scheme is used this is a resource shared among tasks.

- interference must be bounded.
- capacity continues to expand

Inst. Cache CRPD Estimation

Lee et. al. 98





- UCB = CRPD bound

Inst. Cache CRPD Estimation

Mitra et. al. 01

- Same method for computation of UCB but more detailed in the information contained.
- cache states represented as sets of tuples instead of a set of memory blocks
- more information is presented in cache allowing for lower overestimation
- maximum set of used blocks by higher priority tasks is computed.
- this information will yield CRPD values for pairs of tasks.

Data Cache CRPD Estimation

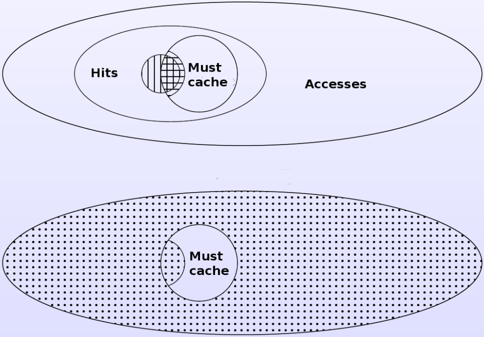
Ramaprasad 06

- assumptions made in previous works render them useless for data caches
- tightly integrated with the WCET analysis
- cache hit/miss pattern is generated
- a chain connects all memory references to the same cache block
- at every program point data CRPD is computed (equal to the amount of lines present)
  - cache blocks not shared with other tasks are not taken into account
  - if previous point in chain was a miss the line is not considered
- pointer arithmetics not supported (static method)

Further Bounding CRPD

Altmeyer 09

- CRPD is meant to be used in conjunction with WCET estimates
- cache analysis is performed in WCET estimation procedure
- two sources of overestimation
- one is actually enough to stay on the safe side of timing analysis



Number of Preemptions

- In the previous referred papers CRPD estimation is just part of the process
- bounding (n) number of preemptions follows
- these bounds will be overly pessimistic in number
- won't take into consideration indirect preemptions.
- resources end up overallocated

Restricting Preemptions

Bertogna et. al. 10

- Restricts preemptions to specific points where CRPD is low
- introduces blocking time
- can affect schedulability
- viability of usage dependent on task-set characteristics
- some task-sets may be unschedulable
- non-determinism is reduced

Bottom Line

Is static analysis what we want?

- as we've seen, there are static analysis tools to estimate the CRPD parameter!

Bottom Line

Is static analysis what we want?

- as we've seen, there are static analysis tools to estimate the CRPD parameter!
- but Open-Systems can't extensively rely on static analysis input.

Bottom Line

Is static analysis what we want?

- there are static analysis tools to over-estimate the CRPD parameter...
- but Open-Systems can't extensively rely on static analysis input.

Bottom Line

Is static analysis what we want?

- there are static analysis tools to over-estimate the CRPD parameter...
- but Open-Systems can't extensively rely on static analysis input.
- bounding preemption number is not really efficient specially in the presence of non-periodic tasks

Bottom Line

Is static analysis what we want?

- there are static analysis tools to over-estimate the CRPD parameter...
- but Open-Systems can't extensively rely on static analysis input.
- bounding preemption number is not really efficient specially in the presence of non-periodic tasks
- there is considerably more knowledge about the system on run-time.

System Model Proposal

- Task set is comprised of sporadic and periodic tasks.
- Caches are assumed to be direct-mapped, though generalisations for set-associative may be devised
  - static analysis enables one to get more info in the case of set-associative caches
- The solution's purpose is to be applied in an open-system configuration
- HRT, SRT and BE tasks co-exist in the same system.

System Model Proposal

- Task set is comprised of sporadic and periodic tasks.
- Caches are assumed to be direct-mapped, though generalisations for set-associative may be devised
  - static analysis enables one to get more info in the case of set-associative caches
- The solution's purpose is to be applied in an open-system configuration
- HRT, SRT and BE tasks co-exist in the same system.

System Model Proposal

- Task set is comprised of sporadic and periodic tasks.
- Caches are assumed to be direct-mapped, though generalisations for set-associative may be devised
  - static analysis enables one to get more info in the case of set-associative caches
- The solution's purpose is to be applied in an open-system configuration
- HRT, SRT and BE tasks co-exist in the same system.

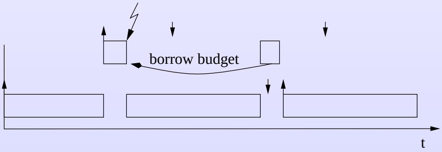
System Model Proposal

- Task set is comprised of sporadic and periodic tasks.
- Caches are assumed to be direct-mapped, though generalisations for set-associative may be devised
  - static analysis enables one to get more info in the case of set-associative caches
- The solution's purpose is to be applied in an open-system configuration
- HRT, SRT and BE tasks co-exist in the same system.

Proposed Solution Statement

Run-time method for CRPD management

- Intend to use RBED framework
  - uses EDF → lower preemption count and it is optimal
  - budget enforcement → jobs can only execute for a specific amount of time
  - Temporal isolation enforced for simple architectures
  - budget overruns → borrow budget from future and advance jobs' deadline to next invocation



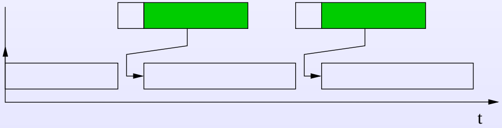
Proposed Solution Statement

- complex architectures with sources of dynamic PD temporal isolation breaks
- reservation $\longrightarrow$ entity responsible for resource allocation (resource manager)
- resource manager must be aware of PD
- Open-systems are increasingly common and must be taken into account

Proposed Solution Statement

- there is still the need to bound and accurately estimate CRPD

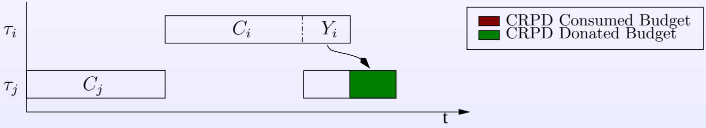
Suppose:



- run-time support has to be on place for tasks with no information (open-system)

Preemption Scenarios

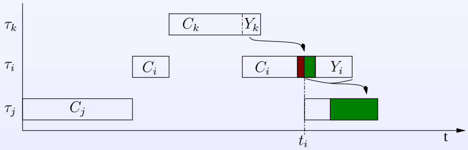
Simple Preemption



- simplest scenario
- direct budget passing according to the level of knowledge on the specific interaction
- in an ideal case the budget would be just enough to account for reloading evicted cache sets

Preemption Scenarios

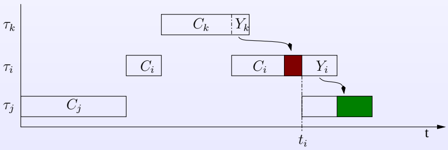
Indirect Preemption



- elaboration of the former example
- more likely to happen in real system execution with high utilizations
- donated CRPD compensation has to be rightfully shared by the queue of preempted applications
- assuming that no task misbehaves, everything will run in the same model than previous scenario

Preemption Scenarios

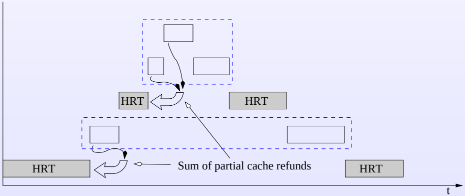
Preemption and Overrun



- when a task misbehaves there is no way to ensure fairness on the budget passing through the chain
- temporal isolation is no longer ensured.
- solution:

Preemption Scenarios

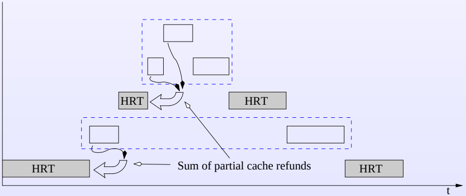
Preemption and Overrun

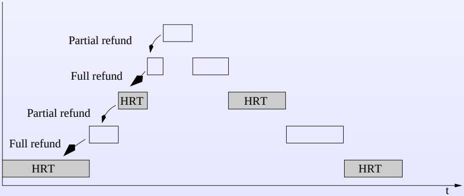


1. set of non-HRT tasks directly pass CRPD refund to the next HRT in preemption queue

Preemption Scenarios

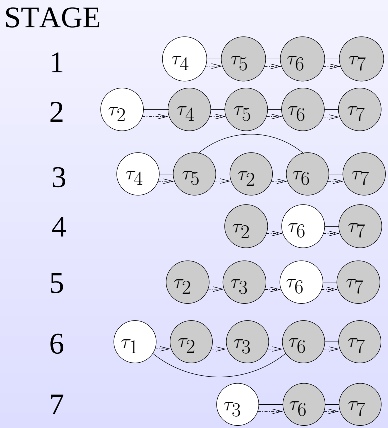
Preemption and Overrun





1. set of non-HRT tasks directly pass CRPD refund to the next HRT in preemption queue
2. ensure that HRT task gets a full refund from preempting task

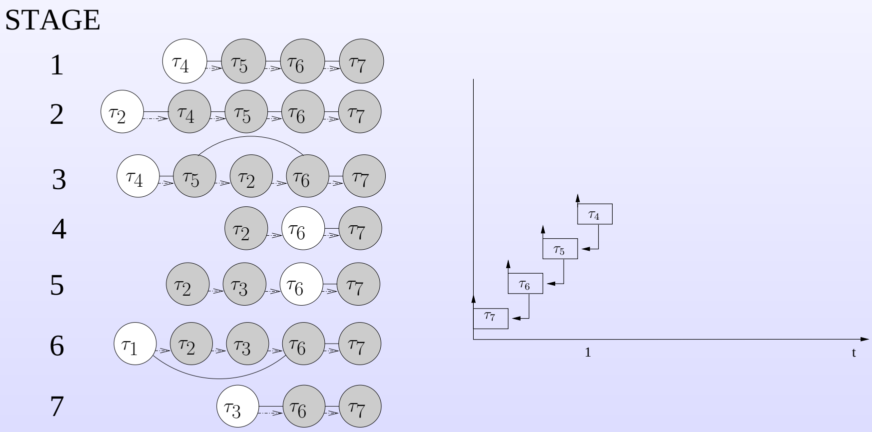
Queue Management



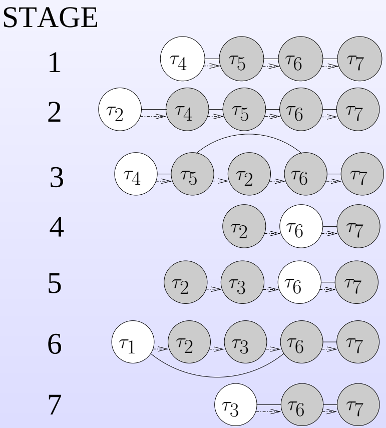
Two queues are maintained:

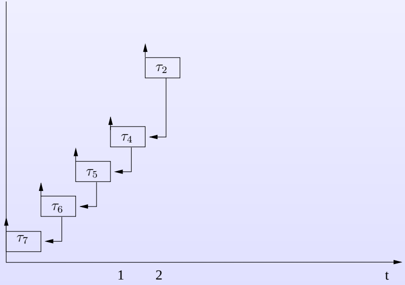
1. ready: already present in the system. Ordered at every task release
2. preemption queue: reordered when preemption occurs

Queue Management

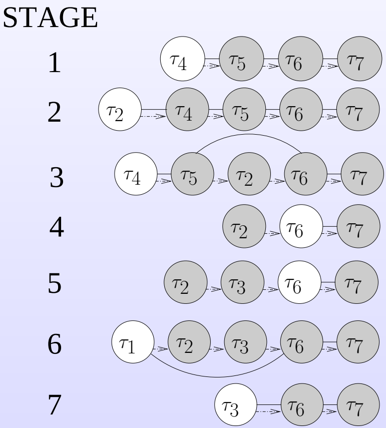


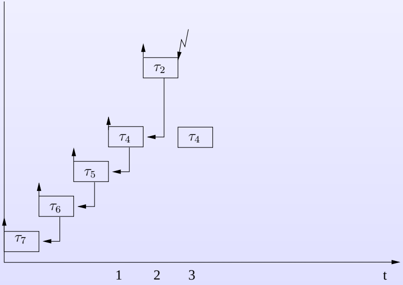
Queue Management



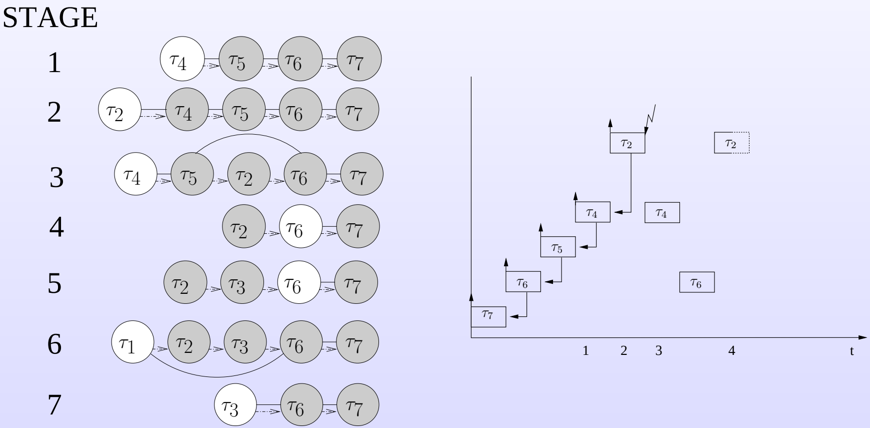


Queue Management

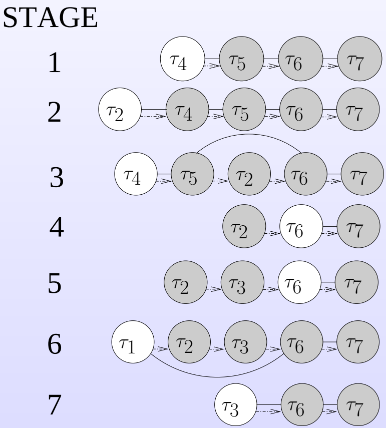


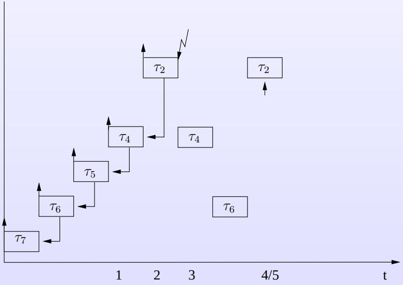


Queue Management

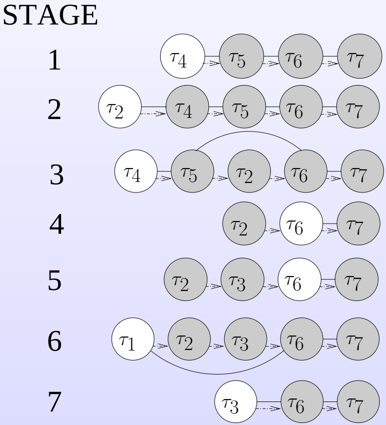


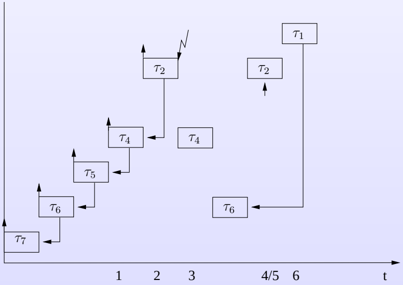
Queue Management



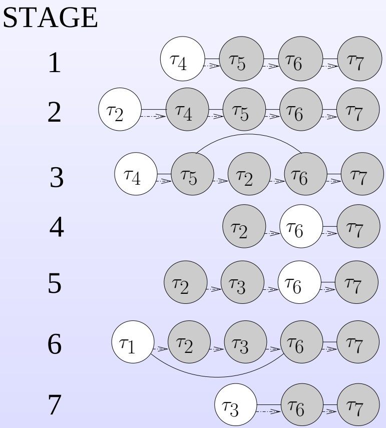


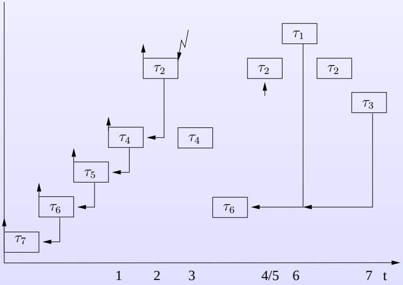
Queue Management





Queue Management





Future Work

- testing of concept in a real-life system is the next step.
- development of on-line measuring mechanisms is the planned way to address the issue.
- 2 solutions provided in Preemption and Overrun will be compared
- Likely outcome:
  - one superior over the other depending on systems configuration

Research Centre in Real-Time Computing Systems

Future Work

- testing of concept in a real-life system is the next step.
- development of on-line measuring mechanisms is the planned way to address the issue.
- 2 solutions provided in Preemption and Overrun will be compared
- Likely outcome:
  - one superior over the other depending on systems configuration

Future Work

- testing of concept in a real-life system is the next step.
- development of on-line measuring mechanisms is the planned way to address the issue.
- 2 solutions provided in Preemption and Overrun will be compared
- Likely outcome:
  - one superior over the other depending on systems configuration