Applying a Real-Time Scheduling Analyzer: “Cheddar”

Jeremy Kuzub, Carleton University
Systems and Computer Engineering
November 2008
Presentation

“Cheddar” Simulator

Scheduling Basics

Mars Pathfinder ‘97 case study
Overview

- Real Time Embedded Systems deal with the real world

- Did your engineering team design the system to work perfectly...

...or work perfectly *on time*?
Brief overview of scheduling

- i.e. This presentation!

- Will the system get all tasks completed by their deadlines using limited resources?

- More complex systems yield answers of potentially limited scope
Many industrial projects do not perform performance analysis with real time scheduling theory even if the demand for the use of this theory is large.

- Can We Increase the Usability of Real Time Scheduling Theory? The Cheddar Project.
  Lecture Notes in Computer Science, Volume 5026/2008
What is Cheddar?

- packages which includes most current feasibility tests and most of the classical real time scheduling algorithms

- A Paradigm for flexible analysis tool architecture:
  - full auto: decides which test cases to apply
  - manual: interactive test case specification using built in toolset
  - extensible framework (i.e Scheduler design).

- Results export as XML for tool chain integration
Cheddar – Use cases

- Do a Scheduling Simulation
- Do a Feasibility Test
- Design a Scheduler
- Design a simple system
- Use Task Editor
- Use Resource Editor
- Use Processor Editor
- Import AADL File
- Interpret .sc script
- Import an system architecture
Cheddar – in your workflow

Eclipse IDE

TOPCASED development tool suite

Cheddar scheduling and feasibility tools

Open-Source workflow: Mission critical (aerospace), allows academic collaboration
Cheddar – in your workflow

Ellidiss STOOD IDE (not free)

AADL System Description

Cheddar scheduling and feasibility tools

List of other tool chains using AADL at
http://la.sei.cmu.edu/aadlinfo/AADLTools.html
Cheddar – in your workflow

IBM Ration Rose Visual Development

AADL Description

Cheddar scheduling and feasibility tools

List of other tool chains using AADL at
http://la.sei.cmu.edu/aadlinfosite/AADLTools.html
Our example: Pathfinder ‘97

- NASA “faster better cheaper” paradigm for robotic exploration
- Instrumented rover and lander
- Rover was semi-autonomous – it decided some of the experiment and mobility sequences
- VxWorks embedded RTOS on 68000-series processor, single core, multithread.
- Highest priority first scheduler with equal time round robin, tie-break
Potential Issues

- COTS operating system, reuse of existing hardware – not all behavioral aspects known.

- A tight budget reused system bus architecture from a much more complex mission with different requirements – some processes were ‘grandfathered’ in
However...

- The partial autonomy of the rover allowed it to collect levels of data that exceed the tested data rates in medium priority tasks. And...

- Random system reboots on Mars

- All commands lost in a reset
Simulation Setup: Structure

(higher priority denoted by lower priority number)
Simulation Setup: Behavior

Mars Soujourner Task Interaction

:bc_sched:Task

:start:

:bc_dist:Task

Request semaphore

semaphore granted

:datbus::sharedResource

Release semaphore

tasks completed
From the Pathfinder Team:
Nominal Schedule

The *** are periods when tasks other than the ones listed are executing.

- **t1** - bus hardware starts via hardware control on the 8 Hz boundary.
  The transactions for the this cycle had been set up by the previous execution of the bc_sched task.
- **t2** - 1553 traffic is complete and the bc_dist task is awakened.
- **t3** - bc_dist task has completed all of the data distribution
- **t4** - bc_sched task is awakened to setup transactions for the next cycle
- **t5** - bc_sched activity is complete
Simple schedulers:
Mars Exploration Analogy

Arrange list of science tasks by increasing time to complete

Do first task on this list

Arrange experiments to run from least-to-most energy consuming

Start next experiment on list

Stop all science activity if new message arrives from Earth

All science tasks resume on full receipt of message
Cheddar scheduler language
Rate Monotonic Algorithm

1 election_section:
2 return min_to_index(dynamic_priority);
3 end section;
Cheddar scheduler language:
Earliest Deadline First

1 start_section:
2    dynamic_priority : array (tasks_range) of integer;
3 end section;
4
5 priority_section:
6    dynamic_priority := tasks.start_time +
7     tasks.deadline + ((tasks.activation_number-1)*tasks.period);
8 end section;
9
10 election_section:
11    return min_to_index(dynamic_priority);
12 end section;
Dependencies Editor

Tasks, Messages, Buffers
Processor Editor

Multiple Processors, each with a different, customizable scheduler
Task Editor

Multiple Tasks, each with a Processor and characterization
Shared Resource Editor

Multiple Tasks, blocking time begin & end, Resource sharing protocols
Cheddar Scheduling Simulation:
Nominal instrument data load

Data management period: 125msec (3 shown)

Fortunately, preemption does impact bc_sched deadline

Scheduling simulation, Processor procl:
- Number of context switches : 10
- Number of preemptions : 3
- Task response time computed from simulation:
  bc_dist => 7/worst
  bc_sched => 1/worst
  mid_level_tasks => 5/worst
- No deadline missed in the computed scheduling: the task set seems to be schedulable.

Scheduling simulation, Processor procl:
- No priority inversion found in the last simulation.
Cheddar Scheduling Simulation:
Heavy instrument data load

**preemption does impact**
**bc_sched deadline!**

Data management period: 125msec (3 shown)

**Scheduling simulation, Processor proc1 :**
- Number of context switches : 7
- Number of preemptions : 1
- Task response time computed from simulation :
  - bc_dist => 8/worst
  - bc_sched => 2/worst, missed its deadline (deadline = 8; completion time = 9)
  - mid_level_tasks => 6/worst
- Some task deadlines will be missed : the task set is not schedulable.

**Scheduling simulation, Processor proc1 :**
- Priority inversion from simulation :
  - bc_sched has a priority inversion on the resource dataBus from the time 7 to the time 8.
Scheduling simulation, Processor proc1:
- Number of context switches: 7
- Number of preemptions: 2
- Task response time computed from simulation:
  - bc_dist => 2/worst
  - bc_sched => 1/worst
  - mid_level_tasks => 11/worst, missed its deadline (deadline = 9; completion time = 11), missed its deadline (deadline = 17; completion time = 20)
- Some task deadlines will be missed: the task set is not schedulable.

Scheduling simulation, Processor proc1:
- No priority inversion found in the last simulation.
This was the problem

- Priority inversion was the issue

- In practice the cause was lack of man hours for testing – human preemption – landing the rover was more important task.

- The solution to switch on priority inheritance ("PIP") in the shared mutex semaphore.
Characteristics of useful modeling tools: Does Cheddar meet these criteria?

- Abstract?
  - Emphasize important aspects while removing irrelevant ones?
    - Yes: Modular, extensible system components

- Understandable?
  - Expressed in a form that is readily understood by observers?
    - Yes: Visual representation, verbose text, XML output, GUI input

- Accurate?
  - Faithfully represents the modeled system?
    - AADL input or plug-in, known and referenced algorithms and schedulers

- Predictive?
  - Can be used to answer questions about the modeled system?
    - Yes, prediction on schedulability and feasibility can be answered

- Inexpensive?
  - Cheddar is free, the learning curve is relatively shallow
Conclusion

- Knowing what to test is not the same as having time to test it

- Even the best in the world miss something, so provide contingency shells

- Great simulation tools may not be used if they are not integrated into existing workflow.
References

1. Can We Increase the Usability of Real Time Scheduling Theory? The Cheddar Project. Lecture Notes in Computer Science, Volume 5026/2008
APPENDIX & REFERENCE
Having a real model – the on earth duplicate

- Random system reboots on mars – all commands lost in a reset
- Replicating the fault was successful when command set and science data load were recreated on the ground
Having a real model – the on earth duplicate

- Random system reboots on mars – all commands lost in a reset
- Replicating the fault was successful when command set and science data load were recreated on the ground
A clue: process priorities

- The main supervisory scheduling process – all work must be completed in 0.125 second cycles – fixed scheduler that overlayes preemptive scheduling system!
- Legacy from previous spacecraft from which databus was derived.
A clue: process priorities

- Under every conceivable worst case test on the ground, 0.125 sec was not a problem – this clock preempted all other tasks
- Since this was the highest priority task – no problem….but…
A low priority task...

- AS/MT task did not work within the standard paradigm of data transmission on the bus
- Instead it used a legacy queued communications channel....a shared resource with other medium-priority tasks
Cheddar – in your workflow

- Intergration with other open source case tools – topcased and STOOD – as a plugin
- Can read AADL files
- Conceptual compatibility – standard,ized algorithms implemented and referenced to papers. Contains
- Standalone capability
- Shedualibility
- Feasobility
- Custmizable
- Integrated with industry tool as plugins and with files types. provides the most known real time scheduling feasibility
- tests and scheduling algorithms, it was primilary used in order to check that the
- first AADL standard can be actually analyzed with real time scheduling theory
- tools. [Cheddar: investigating the usability of real time scheduling theory]