



SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

Adaptive Scheduling Parameters Manager for SCHED_DEADLINE

Alessio Balsini
a.balsini@sssup.it

Università di Pisa, Scuola Superiore Sant'Anna

Workshop on Real-Time Scheduling in the Linux Kernel
27 June 2014



Introduction

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

- Context: soft real-time periodic tasks scheduling
- Subcontext: multimedia audio/video reproduction
- Problem: tradeoff between overprovisioning and QoS
- Solution: a set of tools that manage SCHED_DEADLINE parameters adaptively



Scheduling Soft Real-Time Periodic Tasks

- *What Happens*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Computational request at each activation may heavily differ.



Figure : Ideal



Figure : Real



Scheduling Soft Real-Time Periodic Tasks

- *Examples*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

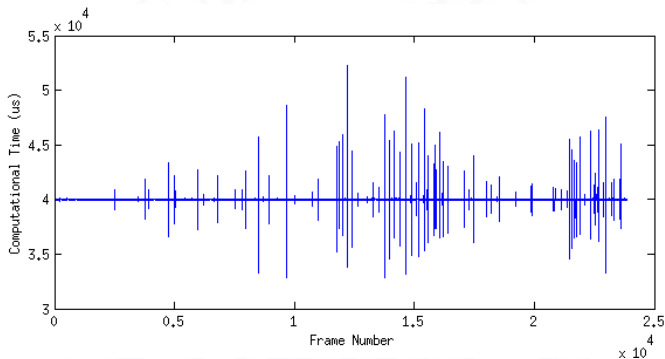


Figure : Back to the Future (MKV)



Scheduling Soft Real-Time Periodic Tasks

- *Examples*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

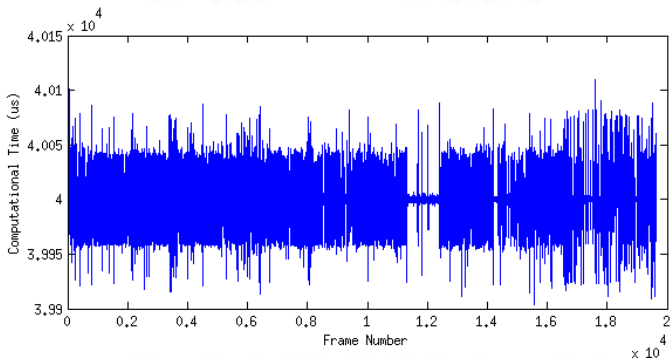


Figure : Blade Runner (AVI)



Scheduling Soft Real-Time Periodic Tasks

- *Examples*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

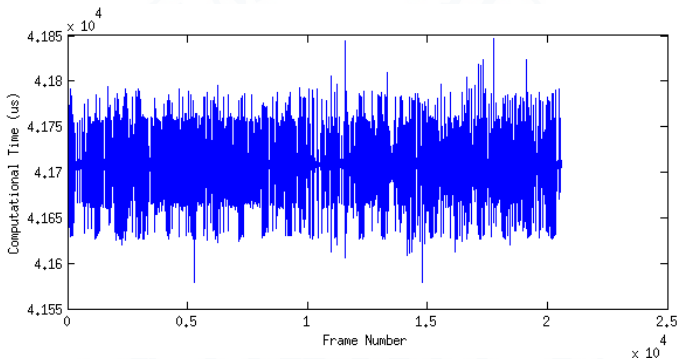


Figure : Superman Returns (MP4)



Scheduling Soft Real-Time Periodic Tasks

- *Relevant Parameters?*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

In SCHED_DEADLINE it is possible to configure (task based)

- Period
- Relative Deadline
- Bandwidth

However, in the considered application context, a single parameter can be enough

Response Time



Scheduling Soft Real-Time Periodic Tasks

- *Relevant Parameters?*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

In SCHED_DEADLINE it is possible to configure (task based)

- Period
- Relative Deadline
- Bandwidth

However, in the considered application context, a single parameter can be enough

Response Time



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

How to generate SCHED_DEADLINE parameters starting from the Response Time?

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

How to generate SCHED_DEADLINE parameters starting from the Response Time?

1 Period

■ Equal to the Response Time

2 Relative Deadline

3 Bandwidth



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

How to generate SCHED_DEADLINE parameters starting from the Response Time?

1 Period

- Equal to the Response Time

2 Relative Deadline

- Equal to the Period

3 Bandwidth



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

How to generate SCHED_DEADLINE parameters starting from the Response Time?

1 Period

- Equal to the Response Time

2 Relative Deadline

- Equal to the Period

3 Bandwidth



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

How to generate SCHED_DEADLINE parameters starting from the Response Time?

- 1 Period
 - Equal to the Response Time
- 2 Relative Deadline
 - Equal to the Period
- 3 Bandwidth



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

How to generate SCHED_DEADLINE parameters starting from the Response Time?

- 1 Period
 - Equal to the Response Time
- 2 Relative Deadline
 - Equal to the Period
- 3 Bandwidth
 - ?



Problem

Response Time to SCHED_DEADLINE: $\mathbb{R} \rightarrow \mathbb{R}^3$

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction
Problem
Optimistic vs
Pessimistic

Tools
Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions
Performance
Hands On

References
Repositories
References

How to generate SCHED_DEADLINE parameters starting from the Response Time?

- 1 Period
 - Equal to the Response Time
- 2 Relative Deadline
 - Equal to the Period
- 3 Bandwidth
 - ?



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Warning: Choosing the bandwidth may cause headaches

Optimistic



Pessimistic

Low GoS



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Warning: Choosing the bandwidth may cause headaches

Optimistic



Pessimistic

Low QoS

Resources

High QoS

Resources



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Warning: Choosing the bandwidth may cause headaches

Optimistic



Pessimistic

1 Low QoS

2 Resources-driven?



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Warning: Choosing the bandwidth may cause headaches

Optimistic



Pessimistic

1 Low QoS

2 Resources-driven?



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Warning: Choosing the bandwidth may cause headaches

Optimistic



Pessimistic

1 Low QoS

2 Resources-driven?



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Warning: Choosing the bandwidth may cause headaches

Optimistic

1 Low QoS

2 Resources-driven?



Pessimistic

1 Best QoS

2 Waste of resources



Computational Requirements

- *Is the glass half empty or half full?*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

Warning: Choosing the bandwidth may cause headaches

Optimistic

1 Low QoS

2 Resources-driven?



Pessimistic

1 Best QoS

2 Waste of resources



Another possible approach? Dynamic!

- *Let's see how much you drank*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

The bandwidth is dynamically (periodically) chosen *for each* *SCHED_DEADLINE* task, depending on the history of the required computational times.

It is a feedback loop controller.

But if this controller modifies the bandwidth,
isn't it just like removing the CBS from
SCHED_DEADLINE?

Yes and no.



Another possible approach? Dynamic!

- *Let's see how much you drank*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

The bandwidth is dynamically (periodically) chosen *for each* `SCHED_DEADLINE` task, depending on the history of the required computational times.

It is a feedback loop controller.

But if this controller modifies the bandwidth,
isn't it just like removing the CBS from
`SCHED_DEADLINE`?

Yes and no.

Adaptation delay for the transitory



Another possible approach? Dynamic!

- *Let's see how much you drank*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

The bandwidth is dynamically (periodically) chosen *for each* `SCHED_DEADLINE` task, depending on the history of the required computational times.

It is a feedback loop controller.

But if this controller modifies the bandwidth,
isn't it just like removing the CBS from
`SCHED_DEADLINE`?

Yes and no.



Another possible approach? Dynamic!

- *Let's see how much you drank*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

The bandwidth is dynamically (periodically) chosen *for each* `SCHED_DEADLINE` task, depending on the history of the required computational times.

It is a feedback loop controller.

But if this controller modifies the bandwidth,
isn't it just like removing the CBS from
`SCHED_DEADLINE`?

Yes and no.

1 Adaptation delay: for the transitory

2 Global controller: for the fairness



Another possible approach? Dynamic!

- *Let's see how much you drank*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

The bandwidth is dynamically (periodically) chosen *for each* `SCHED_DEADLINE` task, depending on the history of the required computational times.

It is a feedback loop controller.

But if this controller modifies the bandwidth,
isn't it just like removing the CBS from
`SCHED_DEADLINE`?

Yes and no.

- 1 Adaptation delay: for the transitory
- 2 Global controller: for the fairness



Tools

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References





Tools

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

- **Kernel Module:** SCHED_DEADLINE Spy
- **Daemon:** SCHED_DEADLINE Dynamic Manager
- **Configuration GUI:** SchedConfigTool



Hi-level Point of View

- Overall Block Scheme

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

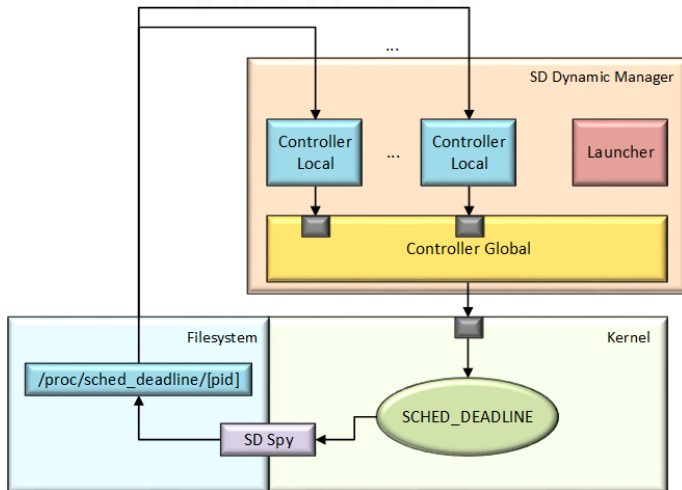
Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References





Kernel Module: SCHED_DEADLINE Spy

- Userspace

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

This module creates a file for each SCHED_DEADLINE task

/proc/sched_deadline/[PID]

Containing four columns:

1401468028	22757242	10149132	N
1401468028	22757242	86353	N
1401468028	62757243	37679835	Y
1401468028	94757243	26311134	N
...			

- first two columns are the kernel time (seconds and nanoseconds) of the measurement
- third column is the job execution time (nanoseconds)
- last column says if execution exceeds the bandwidth (Yes/No)



Kernel Module: SCHED_DEADLINE Spy

- *Implementation Hints*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

Kernel probes: *Kprobes*

Probes are placed around the kernel and the instrumentation codes are executed when the processor encounters those probe point. With *Jprobes* it is also possible to access function arguments.

This module attaches probes to

enqueue_task_dl
and
update_dl_entity

Jprobes are used to create tasks' statistics.

Other callbacks are provided, managing all the statistics sequential files.



Daemon: SCHED_DEADLINE Dynamic Manager

- Interface

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

This tool provides the following DBus interface

core.sched.dl.ProcessManager

with the following methods:

- **xml**: requires a string input, corresponding to the path of the XML file containing the task information
- **fixed_add**: adds a new fixed task to the control list, with the defined SCHED_DEADLINE parameters
- **fixed_launch**: creates a new fixed task and adds it to the control list, with the defined SCHED_DEADLINE parameters
- **control**: adds a new dynamic task to the control list, with the defined response time parameter
- **launch**: creates a new dynamic task and adds it to the control list, with the defined response time parameter



Daemon: SCHED_DEADLINE Dynamic Manager

- Interface, Sample Configuration

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

```
<?xml version="1.0"?>
<SchedulingAlgorithm name="SCHED_DEADLINE">
  <path>/usr/bin/executable</path>
  <args>-p parameter</args>
  <runtime>28000000</runtime>
  <deadline>33333333</deadline>
  <period>33333333</period>
</SchedulingAlgorithm>
```

XML configuration for SCHED_DEADLINE Dynamic Manager



Daemon: SCHED_DEADLINE Dynamic Manager

- *Implementation, Controller Local*

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

- One controller local for each dynamically scheduled task
- It performs the following operations cyclically
 - obtain task statistics
 - run the Control Algorithm to calculate the best utilization factor
 - send the computed utilization factor to the global controller

Note: The current control algorithm implements the worst case within a window of samples



Daemon: SCHED_DEADLINE Dynamic Manager

- Implementation, Controller Global

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

It performs the following operations cyclically

- check the schedulability of all the SCHED_DEADLINE utilization factors

$$\sum_{i=1}^n \frac{B_{D,i}}{T_{D,i}} + \sum_{i=1}^m \frac{B_{F,i}}{T_{F,i}} \leq B_{SD}$$

, D : Dynamic, F : Fixed

- if not verified, use the *Spring With no Length Constraints* algorithm to compress the dynamic tasks' requirements

$$\forall i, U'_{D,i} = \frac{B_{D,i}}{T_{D,i}} - (U_D - B_{residual}) \cdot \frac{T_{D,i}}{\sum_{i=1}^n T_{D,i}}$$

- update SCHED_DEADLINE parameters



Configuration Generator: SchedConfigTool

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon

Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

The screenshot shows the 'Sched Config Tool' window. It has a menu bar with 'File' and a help icon. Below the menu bar, there's a tab labeled 'SCHED_DEADLINE'. Under this tab, there's a section for 'QoS' with a text input field containing '12345678' and a label 'Response Time'. At the bottom, there are two input fields: one for 'Path' containing '/usr/bin/myMoviePlayer' and one for 'Args' containing '-v myVideo.mp4'. On the right side, there are two tabs: 'XML' and 'JSON'. The 'XML' tab is active, showing a preview of the generated XML configuration. Below the XML preview, there's a status box that says 'Validation successful'.

Sched Config Tool

File ?

SCHED_DEADLINE

QoS

12345678 Response Time

/usr/bin/myMoviePlayer Path

-v myVideo.mp4 Args

XML JSON

```
<?xml version="1.0"?>
<SchedulingAlgorithm
name="QoS_Feedback">
  <path>/usr/bin/myMoviePlayer</path>
  <args>-v myVideo.mp4</args>
  <responsetime>12345678</
responsetime>
</SchedulingAlgorithm>
```

Validation successful



Configuration Generator: SchedConfigTool

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem

Optimistic vs
Pessimistic

Tools

Overview

Kernel Module

Daemon

Configuration
Generator GUI

Conclusions

Performance

Hands On

References

Repositories

References

The screenshot shows the 'Sched Config Tool' window. The title bar includes a menu icon, a question mark, and standard window controls. The main area is titled 'SCHED_DEADLINE'. It contains three input fields with labels: 'Period' (value: 123456789), 'Deadline' (value: 123456789), and 'Run Time' (value: 12345678). Below these is a 'QoS' section with two input fields: 'Path' (value: /usr/bin/myMoviePlayer) and 'Args' (value: -v myVideo.mp4). On the right, there are tabs for 'XML' and 'JSON'. The 'XML' tab is active, displaying the following XML code:

```
<?xml version="1.0"?>
<SchedulingAlgorithm
name="SCHED_DEADLINE">
  <path>/usr/bin/myMoviePlayer</path>
  <args>-v myVideo.mp4</args>
  <period>123456789</period>
  <deadline>123456789</deadline>
  <runtime>12345678</runtime>
</SchedulingAlgorithm>
```

At the bottom right, a status box indicates 'Validation successful'.



Performance

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References





Response Times

- MPlayer Without SCHED_DEADLINE

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

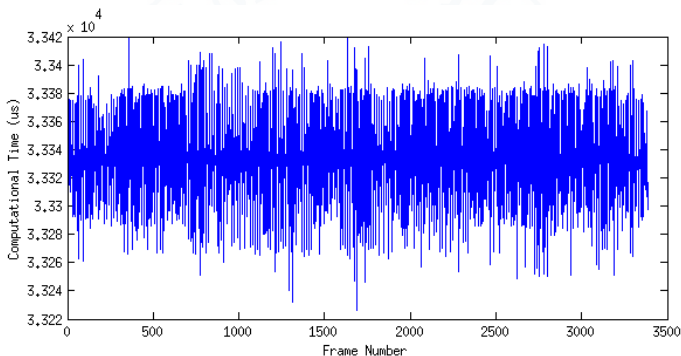


Figure : Eve Online Rubicon (MP4) without SCHED_DEADLINE



Response Times

- MPlayer With Dynamic Manager, Alone

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

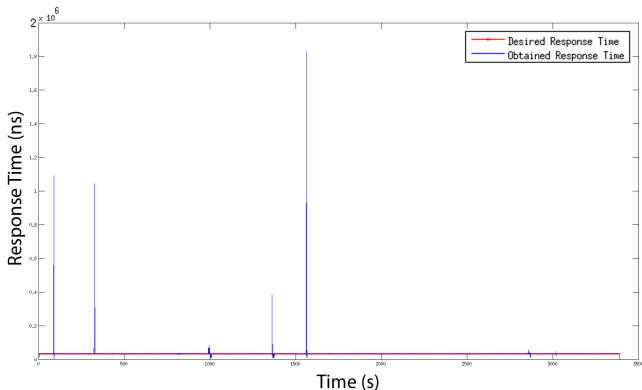


Figure : Eve Online Rubicon (MP4) Dynamic Manager, Alone

Controller Global period: 1s, Controller Local window size: 50



Response Times

- *MPlayer With Dynamic Manager, With Fixed*

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

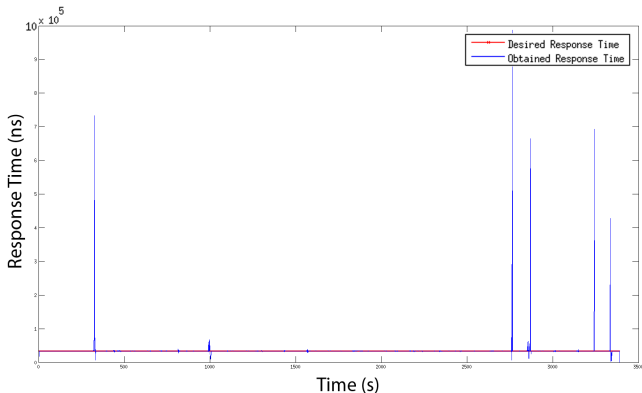


Figure : Eve Online Rubicon (MP4) Dynamic Manager, Running Together With Several Fixed Parameters SCHED_DEADLINE Tasks

Controller Global period: 1s, Controller Local window size: 50



Response Times

- MPlayer With Dynamic Manager, With Other

SCHED_DL:
Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction
Problem
Optimistic vs
Pessimistic

Tools
Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions
Performance
Hands On

References
Repositories
References

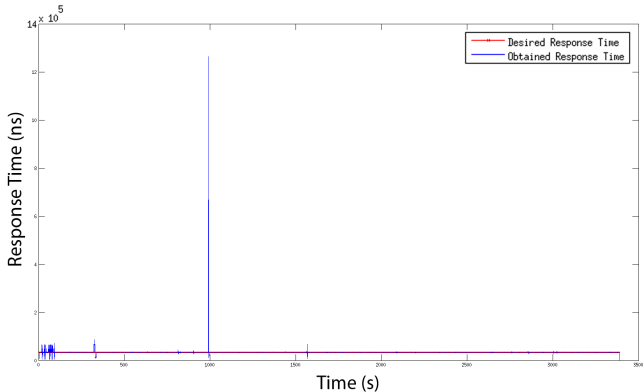


Figure : Eve Online Rubicon (MP4) Dynamic Manager, Running Together With Several Other Linux Tasks

Controller Global period: 1s, Controller Local window size: 50



Practical Session

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References





Repositories

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

■ SCHED_DEADLINE Spy

► github.com/balsini/sched-deadline-spy

■ SCHED_DEADLINE Dynamic Manager

► github.com/balsini/sched-deadline-dynamic-manager

■ SchedConfigTool

► github.com/balsini/SchedConfigTool



References

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References



L. Palopoli, T. Cucinotta, L. Marzario, G. Lipari, *AQoSA - Adaptive Quality of Service Architecture*. Wiley InterScience, 2008.



G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications 3rd*. Springer Publishing Company, 2011.



Thank You

SCHED_DL:

Adaptive
Scheduling
Parameters
Manager

A.Balsini

Introduction

Problem
Optimistic vs
Pessimistic

Tools

Overview
Kernel Module
Daemon
Configuration
Generator GUI

Conclusions

Performance
Hands On

References

Repositories
References

Alessio Balsini



a.balsini@sssup.it