UNIVERSITY OF CAMBRIDGE

Motivating preemptive GPU scheduling for real-time systems (...and the need for better models)

Roy Spliet (Roy.Spliet@cl.cam.co.uk)

Computer Laboratory
University of Cambridge

Contents

Real-time systems - Use-case

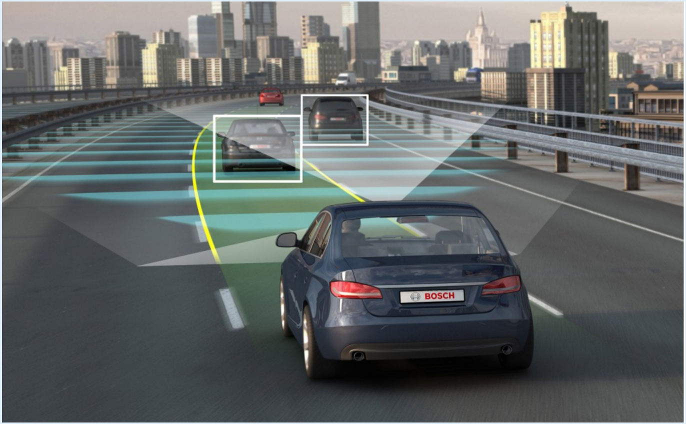


Figure: ©Bosch

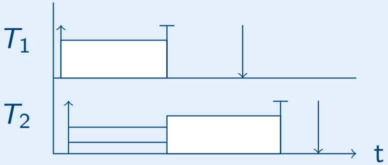
Real-time systems

Primary concern is bound latency.

Then we also care about performance, power consumption...

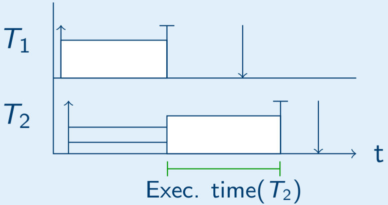
Real-time systems

- WCET: Worst-case *execution* time.
- WCRT: Worst-case *response* time.
  $\rightarrow$ WCET + maximum blocking time



Real-time systems

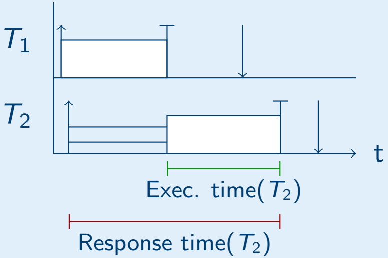
- WCET: Worst-case *execution* time.
- WCRT: Worst-case *response* time.
  - $\rightarrow$ WCET + maximum blocking time



UNIVERSITY OF CAMBRIDGE

Real-time systems

- WCET: Worst-case *execution* time.
- WCRT: Worst-case *response* time.
  → WCET + maximum blocking time



Real-time systems - sporadic task model

System: set of tasks $T_i = \{C_i, D_i, P_i\}$

- C_i: Cost, WCET.
- D_i: Deadline.
- P_i: Period, minimum interval between successive *job* releases.

Utilisation of a “task set”:

$$U = \sum_{i=0}^{n} \frac{C_i}{P_i}$$

Real-time systems - schedulability

Given a set of tasks $\{T_1..T_n\}$, a scheduling method (fixed-priority, earliest-deadline first) and its parameters, can all tasks be guaranteed to *always* make their deadline?

Or: for every task in a task set, is the WCRT smaller than its deadline?

Preemptive GPU scheduling

Preemption for real-time systems is a trade-off between:

- Lower WCRT for high-priority tasks, and
- Higher context switching overhead.

Can the higher overhead be justified by lower WCRT?

Preemptive GPU scheduling - experiment

Experiment: determine schedulability of random tasksets under preemptive vs. non-preemptive scheduling.

Steps:

1. Determine WCRT of non-preemptive context switch.
2. Estimate conservative WCRT of preemptive context switch.
3. Compare schedulability.

Preemptive GPU scheduling - experiment

Step 1: Determine WCRT of non-preemptive context switch.

- NVIDIA (2009): *"The Fermi pipeline is optimized to reduce the cost of an application context switch to* ***below 25 microseconds****."*

Preemptive GPU scheduling - experiment

Step 1: Determine WCRT of non-preemptive context switch.

- NVIDIA (2009): *"The Fermi pipeline is optimized to reduce the cost of an application context switch to* ***below 25 microseconds****."*

- But Nouveau cannot change Fermi memory clock.

Preemptive GPU scheduling - experiment

Step 1: Determine WCRT of non-preemptive context switch.

- Variety of Kepler GPUs (2012-2014) with different:
  - Context size,
  - Maximum memory bandwidth.
- Same conditions:
  - Nouveau: max clockspeed,
  - Samples: 20,000,000/run,
  - Resolution: 1600x1200,
  - Workload: 1024x768 OpenArena windowed timedemo.
- (Intrusive measurement, max. observed overhead 224ns.)

Preemptive GPU scheduling - results

NVIDIA GPU (SMs)	Max bw	State	Time (μs)			Avg. utilisation	
	GiB/s	KiB	min	avg	max	GiB/s	%
GeForce GT 710 (1)	14.4	$\sim$63.9	9.2	21.5	80.1	2.83	(19.6%)
GeForce GT 640 (2)	28.5	$\sim$68.2	13.6	26.5	43.7	2.45	(8.6%)
GeForce GTX 650 (2)	80.0	$\sim$68.2	12.7	23.2	36.0	2.71	(3.4%)
GeForce GTX 780 (12)	288.4	$\sim$268.6	9.7	20.0	28.6	13.76	(4.8%)

Preemptive GPU scheduling - results

NVIDIA GPU (SMs)	Max bw GiB/s	State KiB	Time (μs) min	avg	max	Avg. utilisation GiB/s	%
GeForce GT 710 (1)	14.4	$\sim$63.9	9.2	21.5	80.1	2.83	(19.6%)
GeForce GT 640 (2)	28.5	$\sim$68.2	13.6	26.5	43.7	2.45	(8.6%)
GeForce GTX 650 (2)	80.0	$\sim$68.2	12.7	23.2	36.0	2.71	(3.4%)
GeForce GTX 780 (12)	288.4	$\sim$268.6	9.7	20.0	28.6	13.76	(4.8%)

NVIDIA's 25μs claim roughly holds on Kepler.

Preemptive GPU scheduling - results

NVIDIA GPU (SMs)	Max bw GiB/s	State KiB	Time (μs) min	avg	max	Avg. utilisation GiB/s	%
GeForce GT 710 (1)	14.4	$\sim$63.9	9.2	21.5	80.1	2.83	(19.6%)
GeForce GT 640 (2)	28.5	$\sim$68.2	13.6	26.5	43.7	2.45	(8.6%)
GeForce GTX 650 (2)	80.0	$\sim$68.2	12.7	23.2	36.0	2.71	(3.4%)
GeForce GTX 780 (12)	288.4	$\sim$268.6	9.7	20.0	28.6	13.76	(4.8%)

NVIDIA's 25μs claim roughly holds on Kepler.

Worst case up to $\sim 3.7\times$ average...

Preemptive GPU scheduling

Step 2: Estimate conservative WCRT of preemptive context switch.

Assumption: WCRT grows linear with size of context.
Each SM:

- 256KiB registers
- Max. 48KiB local memory

Preemptive GPU scheduling

Ex: GeForce GT 640 (2×SM) full-preemption context size:

$$68.2 + 2 \times (256 + 48) = 676.2 KiB$$

Results in following (conservative) estimates:

Ctxswitch type	Avg (μs)	Max (μs)
Non-preemptive (68.2KiB)	26.5	43.7
Preemptive (676.2KiB)	262.7	433.3

Preemptive GPU scheduling

Step 3: determine schedulability of random task sets.

Parameters:

- Uniprocessor EDF scheduling policy.
- 8.1M random tasksets (UUniFast).
- Taskset: two tasks, $1000\mu s \leq P_i < 15000\mu s$.

Ctxswitch type	Avg (μs)	Max (μs)
Non-preemptive (68.2KiB)	26.5	43.7
Preemptive (676.2KiB)	262.7	433.3

UNIVERSITY OF CAMBRIDGE

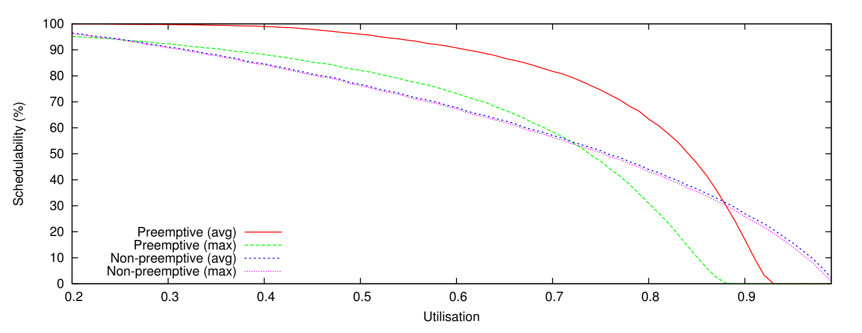
Preemptive GPU scheduling

Step 3: determine schedulability of random task sets.

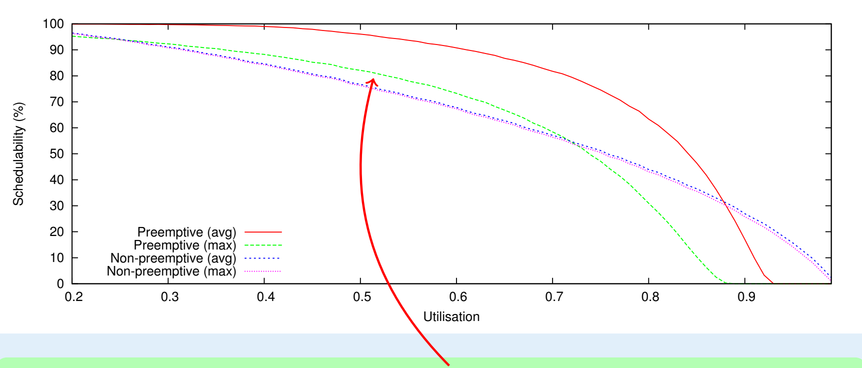
Assumptions:

- NVIDIA Tegra K1-like system (28.5GiB/s, $2\times$SM).
- Non-preemptive context switch: 1 context switch/job.
- Preemptive context switch: 2 context switches/job.

Preemptive GPU scheduling



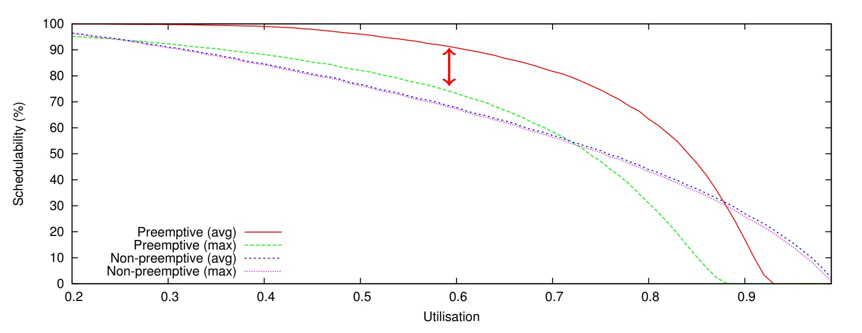
Preemptive GPU scheduling



For $0.25 \leq U \leq 0.72$ full-preempt beneficial

UNIVERSITY OF CAMBRIDGE

Preemptive GPU scheduling



For $0.25 \leq U \leq 0.72$ full-preempt beneficial

Reduce preemptive ctxswitch overhead → higher schedulability.

UNIVERSITY OF CAMBRIDGE

Preemptive GPU scheduling - summary

Preemption for real-time systems is a trade-off between:

- Lower WCRT for high-priority tasks, and
- Higher context switching overhead.

Can the higher overhead be justified by lower WCRT?
$\rightarrow$ Yes

Preemptive GPU scheduling - summary

Preemption for real-time systems is a trade-off between:

- Lower WCRT for high-priority tasks, and
- Higher context switching overhead.

Can the higher overhead be justified by lower WCRT?
→ Yes... under real-time task(/shader/compute kernel) scheduling.

The need for better models

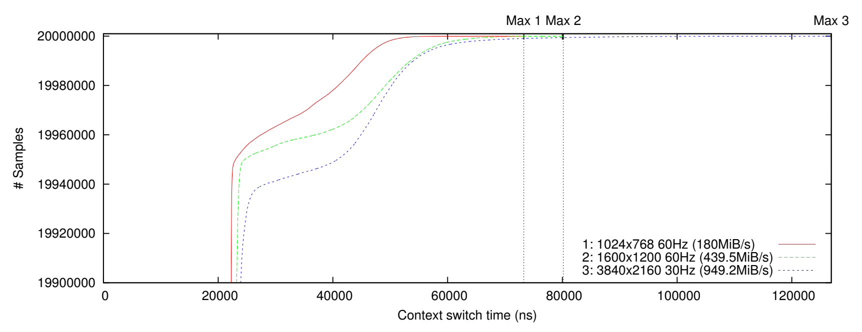
NVIDIA GPU (SMs)	Max bw GiB/s	State KiB	Time (μs) min	avg	max	Avg. utilisation GiB/s	%
GeForce GT 710 (1)	14.4	~63.9	9.2	21.5	80.1	2.83	(19.6%)
GeForce GT 640 (2)	28.5	~68.2	13.6	26.5	43.7	2.45	(8.6%)
GeForce GTX 650 (2)	80.0	~68.2	12.7	23.2	36.0	2.71	(3.4%)
GeForce GTX 780 (12)	288.4	~268.6	9.7	20.0	28.6	13.76	(4.8%)

NVIDIA's 25μs claim roughly holds on Kepler.

Worst case up to ~3.7× average. Why?

The need for better models

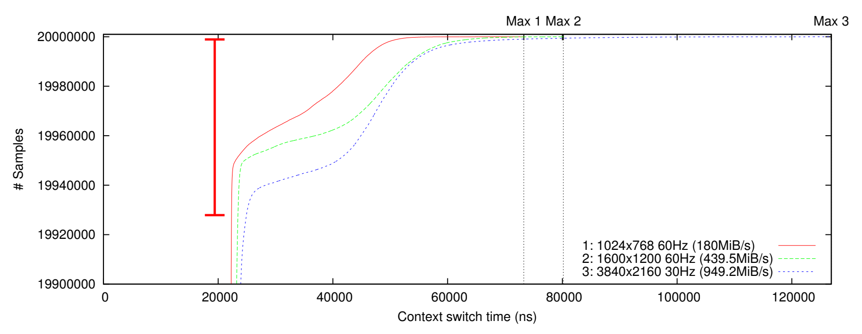
NVIDIA GeForce GT 710:



UNIVERSITY OF CAMBRIDGE

The need for better models

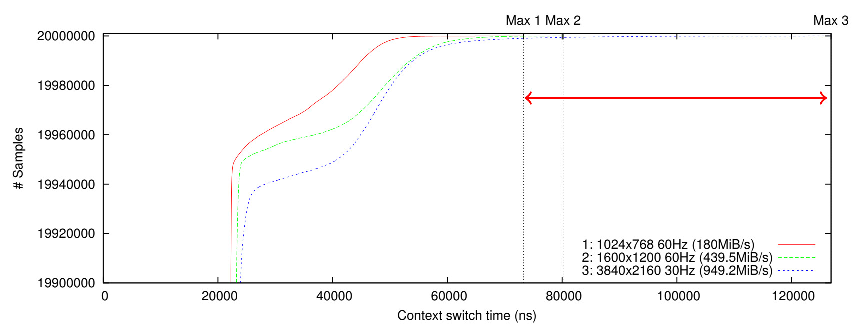
NVIDIA GeForce GT 710:



Tail $\sim 0.3\%$ of all samples.

The need for better models

NVIDIA GeForce GT 710:



Tail $\sim$ 0.3% of all samples.

Maximum correlates with display resolution $\rightarrow$ interference!

The need for better models - summary

- Tasks on GPUs susceptible to performance interference.
- Ex.: display scan-out interferes with context switch.

- Need models to distinguish WCET from WCRT!

Conclusion

Preemptive GPU scheduling for real-time systems

- Use-cases for parallel accelerators in RTS.
  - Autonomous robotics driving force.
- Preemptive scheduling: improved WCRT outweighs overhead.
- We need:
  - More control over task(/shader/compute kernel) scheduling policies.
  - More accurate performance interference models.