Increasing Fixed-Priority Schedulability using Non-Periodic Load Shapers

Mitra Nasri*  Geoffrey Nelissen
Load Shaping

- It is a technique that **shapes the workload** of each job of a task
- Can be implemented as a **reservation server** (for each task)

In this work: the budget is replenished with a **given pattern** that might **not be periodic**

**Resemblance**

Traffic shaping: a computer network traffic management technique which delays some or all datagrams to bring them into compliance with a desired traffic profile [wiki].

Traffic shaping acts like a simple periodic reservation server.
Period transformation is based on dividing a task into “equal” sub-tasks which have the same period and budget.

**Normal workload**

\( \tau_1: (C_1 = 10, T_1 = 30, D_1 = 30) \)

**Simple periodic reservation**

\( \tau^R_1: (2, 6, 6) \)

**Load shaping**

\( \tau^S_1: (\langle 3, 0, 10 \rangle, \langle 7, 10, 30 \rangle) \)

Each “sub-task” has a WCET, relative release time, and deadline.
What can We Do with Load Shaping?

Improving schedulability of fixed-priority scheduling algorithms (FPS)
We use a shaper for $\tau_1$ that has two sub-tasks.
Example

If period transformation was used, 3 sub-tasks were needed per job

We use a shaper for $\tau_1$ that has two sub-tasks

$\tau_2: (24, 40)$

$\tau_1: (12, 30)$
Open Problems

- **Given:**
  - Uniprocessor
  - Periodic tasks
  - Release offsets
  - Constrained (or arbitrary) deadline
  - Dependent (or independent)

- **Scheduled by FPS**

- **Problem 1:** Find shaper parameters such that the task set becomes schedulable and the number of sub-tasks is minimized.
What can We Do with Load Shaping?

Improving schedulability of fixed-priority scheduling algorithms (FPS)

Optimizing/reducing the number of preemptions
Optimizing/Reducing the Number of Preemptions

- Use FPS to schedule the tasks
- Run each sub-task non-preemptively

\[ \tau_i: (C_i = 10, T_i = 30, D_i = 30) \]

Normal workload

Load Shaping

\[ \tau_i^S: ((3, 0, 10), (7, 10, 30)) \]

If a task that too long is scheduled, other tasks with smaller period may miss their deadline.

We can break it such that it does not cause long blocking to other tasks.
Load Shaping v.s. Fixed Preemption Point Placement

- Assume that each sub-task in load shaping is non-preemptive

Normal workload
\[ \tau_i: (C_i = 10, T_i = 30, D_i = 30) \]

Fixed Preemption Point
\[ \tau_i: (C_i = \{3, 7\}, T_i = 30, D_i = 30) \]
\[ \tau_1: (C_1 = 5, T_1 = 10, D_1 = 7) \]

Load Shaping
\[ \tau_i^S: (\{3, 0,10\}, \{7, 10,30\}) \]
\[ \tau_1: (C_1 = 5, T_1 = 10, D_1 = 7) \]

FPS is work-conserving

Deadline miss
Assume that each sub-task in load shaping is non-preemptive.

With load shaping, we make FPS behave like a non-work-conserving algorithm.

Load Shaping

\( \tau_i^S: (\langle 3, 0, 10 \rangle, \langle 7, 10, 30 \rangle) \)

\( \tau_1: (C_1 = 5, T_1 = 10, D_1 = 7) \)
Open Problems, cont.

- **Given:**
  - Uniprocessor
  - Periodic tasks
  - Release offsets
  - Constrained (or arbitrary) deadline
  - Dependent (or independent)

- **Scheduled by FPS**

- **Problem 2:** Find shaper parameters such that the task set remains/becomes schedulable and each sub-task is potentially executed non-preemptively.
Load shaping is a technique to shape the workload using a reservation server that is replenished with a given pattern.

It is a generalization of periodic reservation servers.

- Improving FPS schedulability
- A new approach for limited-preemptive scheduling
- Open problems are finding shapers’ parameters