Scheduling Strategies for Overloaded Real-Time Systems

Yiqin Gao, Guillaume Pallez, Yves Robert and Frédéric Vivien

April 13, 2022

Motivating article

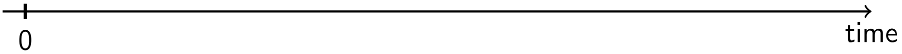
"Stochastic-based robust dynamic resource allocation for independent tasks in a heterogeneous computing system"

by Salehi et al, Journal of Parallel and Distributed Computing Volume 97, November 2016, Pages 96-111

- Heterogeneous oversubscribed system
- Hard deadlines
- Several types of tasks; each type is associated a distribution of task execution times
- A global queue, and one queue per machine
- At each event: recomputation of probabilities of success through convolutions (expensive)

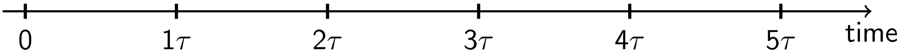
Question: as the distribution of execution times is known, could we precompute things?

Model



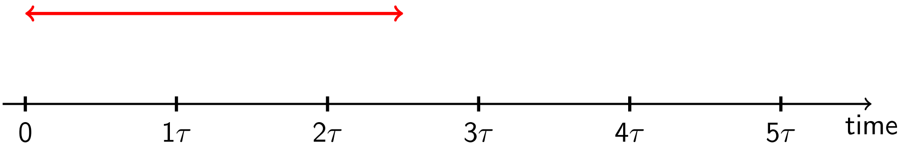
- A single periodic task (and a single processor)
- A new task instance (or job) released with period τ

Model



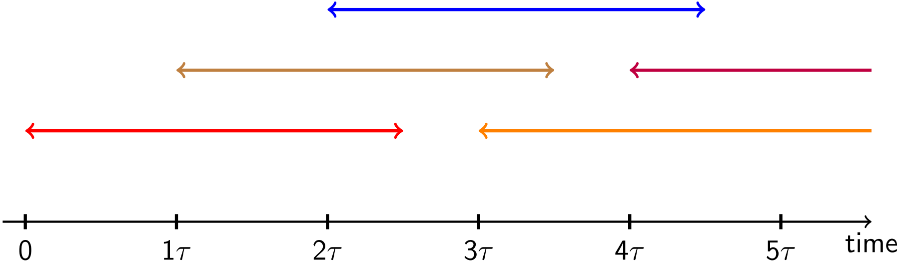
- A single periodic task (and a single processor)
- A new task instance (or job) released with period τ

Model



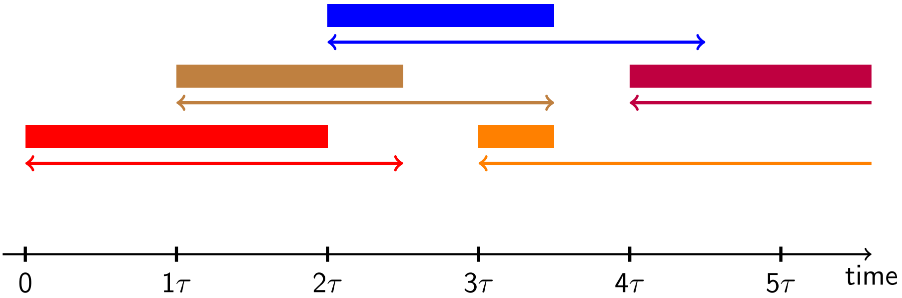
- A single periodic task (and a single processor)
- A new task instance (or job) released with period τ
- Each instance has a relative deadline of D with $D > \tau$
  (in the example $D = 2.5\tau$)

Model



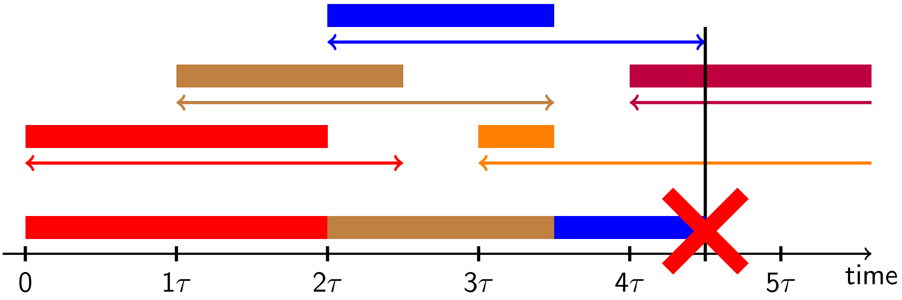
- A single periodic task (and a single processor)
- A new task instance (or job) released with period τ
- Each instance has a relative deadline of D with $D > \tau$ (in the example $D = 2.5\tau$)

Model



- A single periodic task (and a single processor)
- A new task instance (or job) released with period τ
- Each instance has a relative deadline of D with $D > \tau$
  (in the example $D = 2.5\tau$)
- Execution times follow some (known) probability distribution $\mathcal{D}$ (with unbounded support)

Model



- A single periodic task (and a single processor)
- A new task instance (or job) released with period τ
- Each instance has a relative deadline of D with $D > \tau$
  (in the example $D = 2.5\tau$)
- Execution times follow some (known) probability distribution $\mathcal{D}$ (with unbounded support)
- Overloaded system: not all jobs will succeed: a job is killed when it reaches its deadline

Question: How could we optimize such a system?

Objective:

- Deadline Miss Ratio (DMR) or job execution rate

Available mechanisms:

- Do not admit all jobs
- Do not start all admitted jobs
- Kill some running jobs

Overview

1. Scheduling
2. Markov model
3. Evaluation

Overview

1. Scheduling
2. Markov model
3. Evaluation

Scheduling policies

A scheduling policy must decide:

- Which jobs should be admitted in the system
- In which order to execute admitted jobs

- Whether to launch the execution of an admitted job
- When to kill a running job

Scheduling policies

A scheduling policy must decide:

- Which jobs should be admitted in the system
- In which order to execute admitted jobs
  All jobs have the same relative deadline: we always execute jobs in their order of arrival (i.e., *first come first served* and *earliest deadline first*)
- Whether to launch the execution of an admitted job
- When to kill a running job

Admission policies

We consider four admission policies:

1. Admit all jobs
   Tag: *AA* (for *All (jobs) Admitted)*

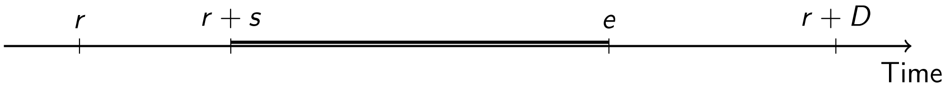
2. Bounded queue of size m
   Tag: $\textsc{Queue}(m)$

3. Random admission policy: a fraction α of jobs are randomly admitted
   Tag: *rand*

4. Periodic admission policy: admission defined by a pattern $\mathcal{A}$ of length $\tau_{\mathcal{A}}$
   Tag: *per*

Control parameters

Example: a successful job released at time r whose execution ends at time e



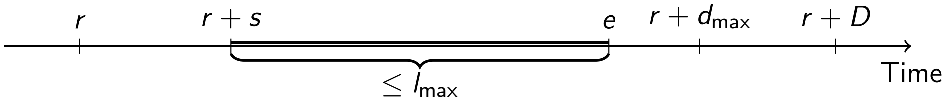
Control parameters

Example: a successful job released at time r whose execution ends at time e

1. Upper bound on completion time: $d_{\max}$
   A job released at time r will have its execution stopped at time $r+d_{\max}$
   (if it has not completed by that time)
   Meaningful values for $d_{\max}$: $\tau \leq d_{\max} \leq D$

Control parameters

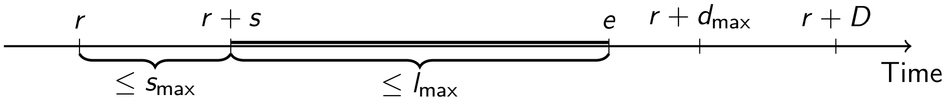
Example: a successful job released at time r whose execution ends at time e



1. Upper bound on completion time: $d_{\max}$
   A job released at time r will have its execution stopped at time $r + d_{\max}$
   (if it has not completed by that time)
   Meaningful values for $d_{\max}$: $\tau \leq d_{\max} \leq D$
2. Upper bound on execution time: $l_{\max}$
   Meaningful values for $l_{\max}$: $\tau \leq l_{\max} \leq d_{\max}$

Control parameters

Example: a successful job released at time r whose execution ends at time e



1. Upper bound on completion time: $d_{\max}$
   A job released at time r will have its execution stopped at time $r + d_{\max}$
   (if it has not completed by that time)
   Meaningful values for $d_{\max}$: $\tau \leq d_{\max} \leq D$
2. Upper bound on execution time: $l_{\max}$
   Meaningful values for $l_{\max}$: $\tau \leq l_{\max} \leq d_{\max}$
3. Upper bound on starting time: $s_{\max}$
   A job released at time r will not be allowed to start its execution later than at time $r + s_{\max}$
   Meaningful values for $s_{\max}$: $0 \leq s_{\max} \leq d_{\max} - \tau$

Scheduling strategies

A scheduling strategy is defined entirely by:

- an admission policy AP
- a triplet $(d_{\max}, l_{\max}, s_{\max})$

1. AP-NeverKill: each admitted job runs until either its completion or it is killed by its deadline
   $(d_{\max}, l_{\max}, s_{\max}) = (D, D, D - \tau)$
2. Buffer(m): bounded queue keeping in the queue the m most recent available jobs
   $(d_{\max}, l_{\max}, s_{\max}) = (D, D, m\tau - 1)$
3. AP-BestDmax, AP-BestLmax, and AP-BestSmax: pick the value for $d_{\max}$ (resp. $l_{\max}$, $s_{\max}$) that minimizes the deadline miss ratio
   We can also optimize for two or three parameters simultaneously: AP-BestDmaxLmaxSmax

Overview

Discrete distributions

If the distribution of job execution times is discrete one can

- Use a Markov chain to describe the system under the NeverKill policy
- Compute its stationary distribution and evaluate its performance

(pre-existing work)

Our aim

- Discretize the continuous distributions
- Model the system with a Markov chain for all admission policies and control parameters

Discretization

- Quantum duration q used to discretize all durations
- Assume τ and D both last an integral number of quanta: $\tau = \left\lfloor \frac{\tau}{q} \right\rfloor q$ and $D = \left\lfloor \frac{D}{q} \right\rfloor q$
- In the following: we set that q lasts one arbitrary unit of time (for the sake of readability) ($q = 1$ with the right choice of time unit)
  The period now lasts τ quanta, and so on
- p_l: probability that a job execution time is equal to l (quanta) in the discretized version of $\mathcal{D}$

$$p_l = \int_{(l-1)q}^{lq} f(x)dx.$$

Notation

- We study the evolution of the system due to the execution of a job S released at a time r
- s: the time job S has to wait after its arrival in the system until the server is available (Server available for S at the date $r + s$)
- Job S is followed by a job T which is released at time $r + \tau$
- We compute the probability $\mathcal{P}_{s,t}$ that the server is available at time $(r + \tau) + t$ for job T if it was available at time $r + s$ for job S

All jobs are admitted $\tau = 4$, $D = 12$, $d_{\max} = D$, $l_{\max} = 6$, and $s_{\max} = 5$

$$\mathcal{P}_{s,t} = \begin{bmatrix}
\sum_{i=1}^{4} p_i & p_5 & 1 - \sum_{i=1}^{5} p_i & 0 & 0 & 0 & 0 & 0 & 0 \\
\sum_{i=1}^{3} p_i & p_4 & p_5 & 1 - \sum_{i=1}^{5} p_i & 0 & 0 & 0 & 0 & 0 \\
\sum_{i=1}^{2} p_i & p_3 & p_4 & p_5 & 1 - \sum_{i=1}^{5} p_i & 0 & 0 & 0 & 0 \\
p_1 & p_2 & p_3 & p_4 & p_5 & 1 - \sum_{i=1}^{5} p_i & 0 & 0 & 0 \\
0 & p_1 & p_2 & p_3 & p_4 & p_5 & 1 - \sum_{i=1}^{5} p_i & 0 & 0 \\
0 & 0 & p_1 & p_2 & p_3 & p_4 & p_5 & 1 - \sum_{i=1}^{5} p_i & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}$$

All jobs are admitted

Case $0 \leq s \leq s_{\max}$ and $t = 0$

$$\mathcal{P}_{s,0} = \sum_{l=1}^{\min\{l_{\max}, d_{\max}-s, \tau-s\}} p_l + \sum_{\substack{l \geq \min\{l_{\max}, d_{\max}-s\}+1 \\ \min\{l_{\max}, d_{\max}-s\} \leq \tau - s}} p_l$$

Case $0 \leq s \leq s_{\max}$ and $t \geq 1$

- *Case* $\max\{1, 1+s-\tau\} \leq t \leq s - \tau + \min\{l_{\max}, d_{\max} - s\} - 1$: $\quad \mathcal{P}_{s,t} = p_{t-s+\tau}$
- *Case* $\max\{1, 1+s-\tau\} \leq t$ *and* $t = s - \tau + \min\{l_{\max}, d_{\max} - s\}$: $\mathcal{P}_{s,t} = 1 - \sum_{l=1}^{\min\{l_{\max}, d_{\max}-s\}-1} p_l$
- *Case* $t > s - \tau + \min\{l_{\max}, d_{\max} - s\}$ *or* $1 \leq t < 1 + s - \tau$: $\quad \mathcal{P}_{s,t} = 0$

Case $s > s_{\max}$

$$\begin{cases} \mathcal{P}_{s,\max\{0,s-\tau\}} = 1 & \\ \mathcal{P}_{s,t} = 0 & \text{if } t \neq \max\{0, s-\tau\} \end{cases}$$

Asymptotic behavior

Theorem

The Markov chains are all both irreducible and aperiodic, and they all admit a unique stationary distribution.

(We show that all possible states are reachable from the initial state $s = 0$, that there is a path from any state to the state 0, and that there is a loop from state 0 to itself.)

Corollary

For any admission policy and any choice of the parameters $d_{\max}$, $l_{\max}$, *and* $s_{\max}$, *the system converges to a unique asymptotic behavior.*

Resolution and Complexity

We solve the linear system

$$\pi_\infty^t \begin{bmatrix} \sum_{i=1}^{4} p_i & p_5 & 1-\sum_{i=1}^{5} p_i & 0 & 0 & 0 & 0 & 0 & 0 \\ \sum_{i=1}^{3} p_i & p_4 & p_5 & 1-\sum_{i=1}^{5} p_i & 0 & 0 & 0 & 0 & 0 \\ \sum_{i=1}^{2} p_i & p_3 & p_4 & p_5 & 1-\sum_{i=1}^{5} p_i & 0 & 0 & 0 & 0 \\ p_1 & p_2 & p_3 & p_4 & p_5 & 1-\sum_{i=1}^{5} p_i & 0 & 0 & 0 \\ 0 & p_1 & p_2 & p_3 & p_4 & p_5 & 1-\sum_{i=1}^{5} p_i & 0 & 0 \\ 0 & 0 & p_1 & p_2 & p_3 & p_4 & p_5 & 1-\sum_{i=1}^{5} p_i & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} = \pi_\infty^t \qquad \text{with} \qquad \textstyle\sum_{s=0}^{\sigma} \pi_\infty^{(s)} = 1$$

Deadline miss ratio: $\mathrm{DMR}^{AA} = 1 - \sum_{s=0}^{\sigma} \pi_\infty(s) \sum_{l=1}^{\min\{l_{\max}, d_{\max}-s\}} p_l$

Theorem

The optimal value of $p \leq 3$ parameters (chosen from $d_{\max}$, $l_{\max}$, and $s_{\max}$) can be computed in time $O(\beta^3(\frac{D}{q})^p)$, where β is the number of states of the Markov chain.

Overview

Distributions: all have a mean of 1

Bimodal exponential	$\mu_1 = 1.005, \mu_2 = 0.995$
	$\mu_1 = 0.1, \mu_2 = 1.9$
Bimodal truncated normal	$\mu_1 = 0.5, \sigma_1 \approx 0.534, \mu_2 = 1, \sigma_2 \approx 1.068$
	$\mu_1 = 0.01, \sigma_1 \approx 0.178, \mu_2 = 1, \sigma_2 \approx 1.782$
Exponential	$\mu = 1$
Gamma	$k = \frac{1}{3} \approx 0.333, \theta = 3$
Half-normal	$\sigma = \sqrt{\frac{\pi}{2}} \approx 1.253$
Inverse Gamma	$\alpha = \frac{7}{3} \approx 2.333, \beta = \frac{4}{3} \approx 1.333$
Log-normal	$\mu = 1, \sigma = 0.5$
	$\mu = 1, \sigma = 3$
Truncated normal	$\mu = 0.8, \sigma \approx 0.754$
Uniform	$a = 0, b = 2$
Weibull	$k \approx 0.411, \lambda = \frac{1}{\Gamma\left(1+\frac{1}{k}\right)} \approx 0.324$
	$k = 1.5, \lambda = \frac{1}{\Gamma\left(1+\frac{1}{k}\right)} \approx 1.108$

Accuracy of simulations

14 distributions $\times$ 19 periods $\times$ 5 relative deadlines = 1330 settings

100 random experiments for each parameter setting

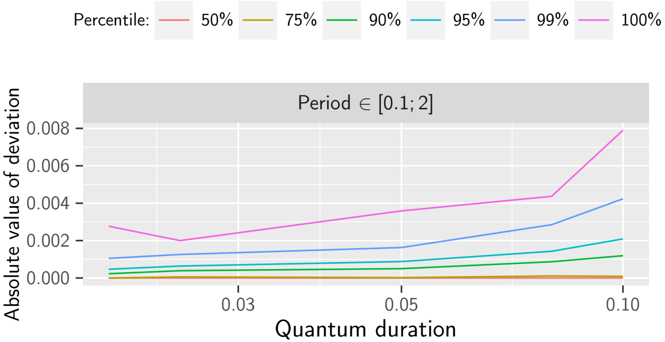
We report the quantiles on the standard deviation of all scenarios

Simulations with 10^6 jobs are reliable (take 0.01 second)

Discretization quantum

We plot the percentiles of the function

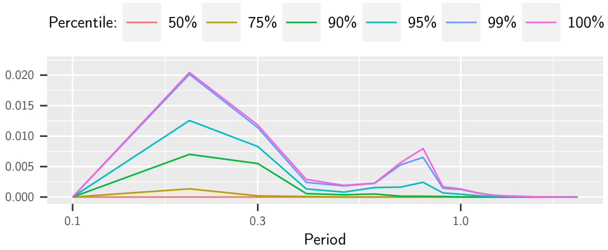
$$q \mapsto |\mathrm{DMR}(\textsc{BestSmax}(q)) - \mathrm{DMR}(\textsc{BestSmax}(0.001))|$$



A quantum $q = 0.1$ is precise enough

All jobs admitted: impact of $d_{\max}$, $l_{\max}$, and $s_{\max}$

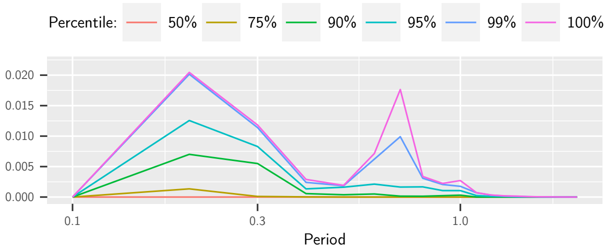
Absolute difference of the achieved Deadline Miss Ratio



BESTSMAX vs. BESTDMAXLMAXSMAX

All jobs admitted: impact of $d_{\max}$, $l_{\max}$, and $s_{\max}$

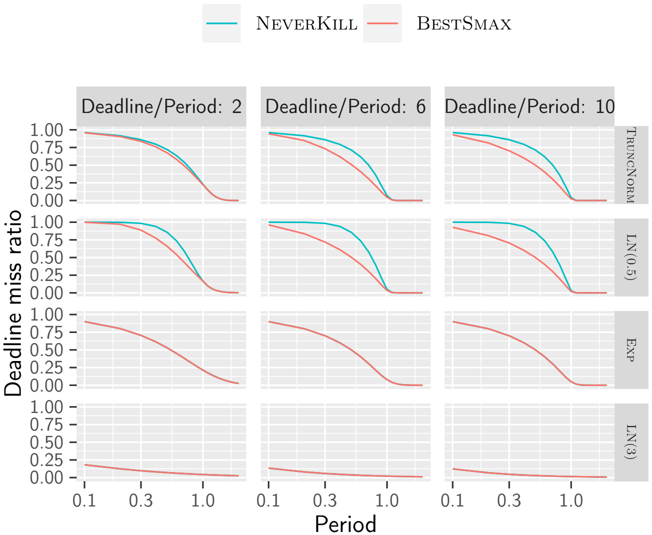
Absolute difference of the achieved Deadline Miss Ratio



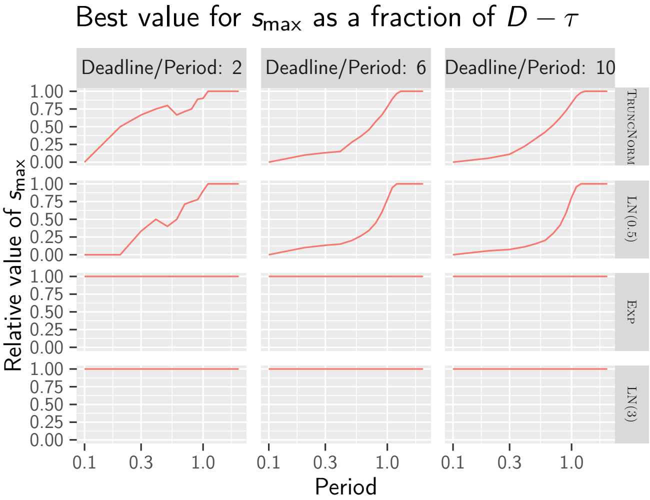
BestDmaxLmax vs. NeverKill

Parameters $d_{\max}$ and $l_{\max}$ play no significant role in optimizing the DMR

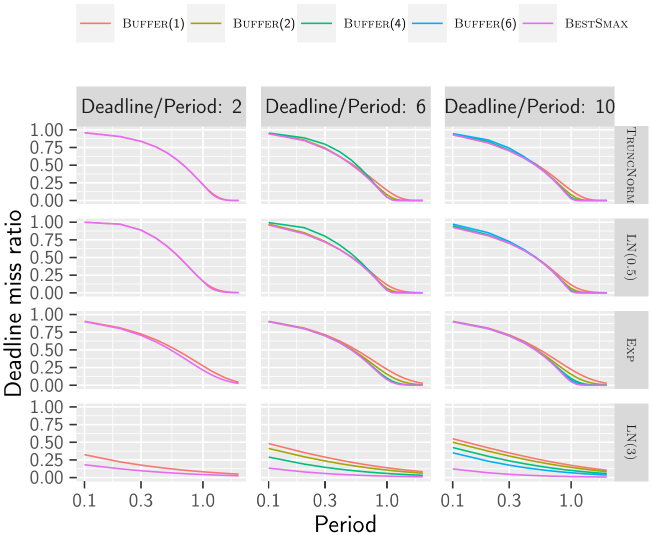
All jobs admitted: impact of s_{max}



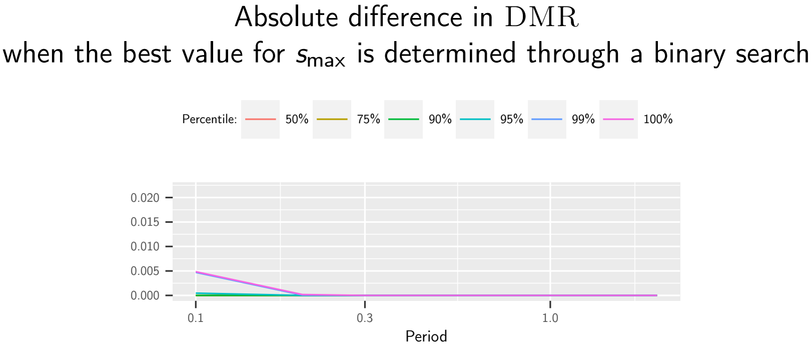
All jobs admitted: best value for s_{max}



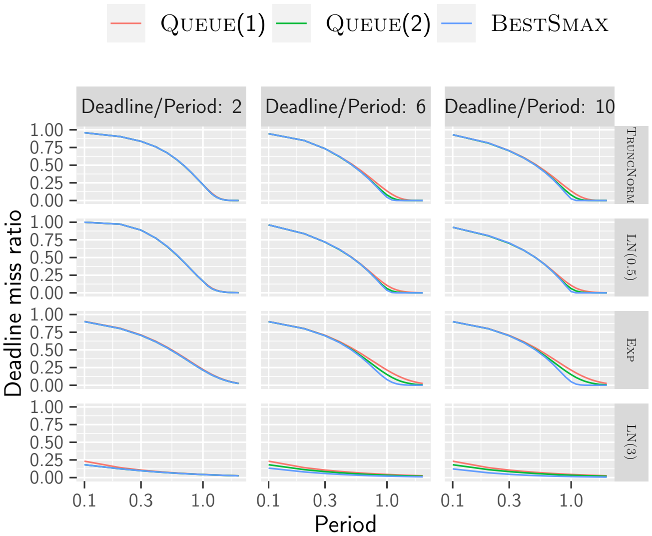
All jobs admitted: Buffer(m)



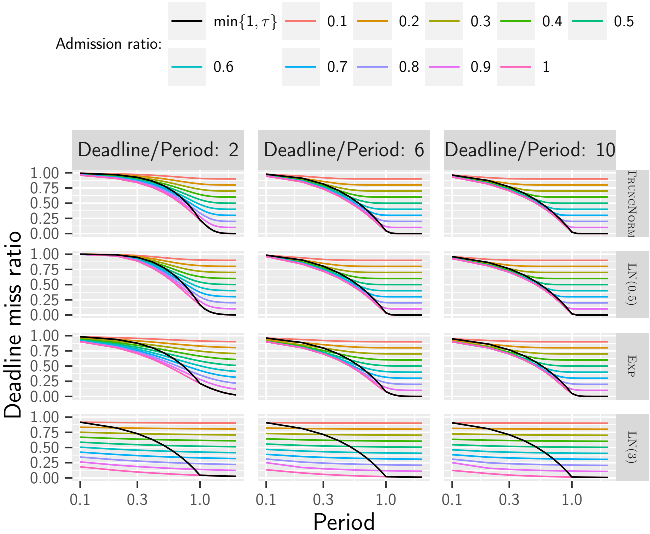
All jobs admitted: binary search for best value for $s_{\max}$



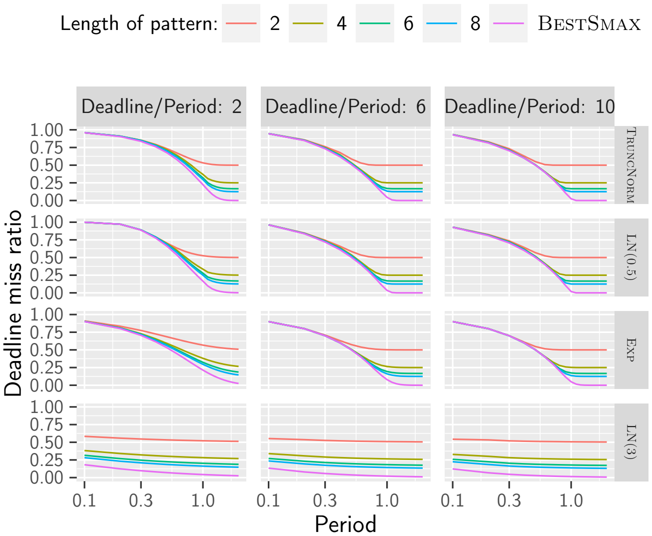
Admission through bounded queues



Random admission



Periodic admission



Conclusions and future work

Conclusions

- To minimize the DMR, it is always better to admit all jobs and then to control their execution, than to reject some jobs at submission time
- The only (studied?) parameter that matters is $s_{\max}$, the maximum allowed waiting time
- Discrete distributions can be modeled, analyzed and optimized using Markov chains
- Continuous settings cannot be investigated analytically, but a rough discretization is sufficient for their analysis

Future work

- What if the task is not exactly periodic?
- What if there are several tasks?
- What if there are several processors?