Asymptotic Competitive Analysis of Task Scheduling Algorithms on a Fault-Prone Machine

Elli Zavou[1,2] (Part of a collaborative work with A. Fernández Anta[1], C. Georgiou[3] and D.R. Kowalski[4])
[1]IMDEA Networks Institute, [2]Universidad Carlos III de Madrid, [3]University of Cyprus, [4]University of Liverpool

Developing the Science of Networks

institute IMDEA networks

Introduction

Problem definition and Model.

Efficient and reliable task executions on a fault-prone machine, under worst-case occurrence of crashes, and dynamic task arrivals of different computation times (costs)

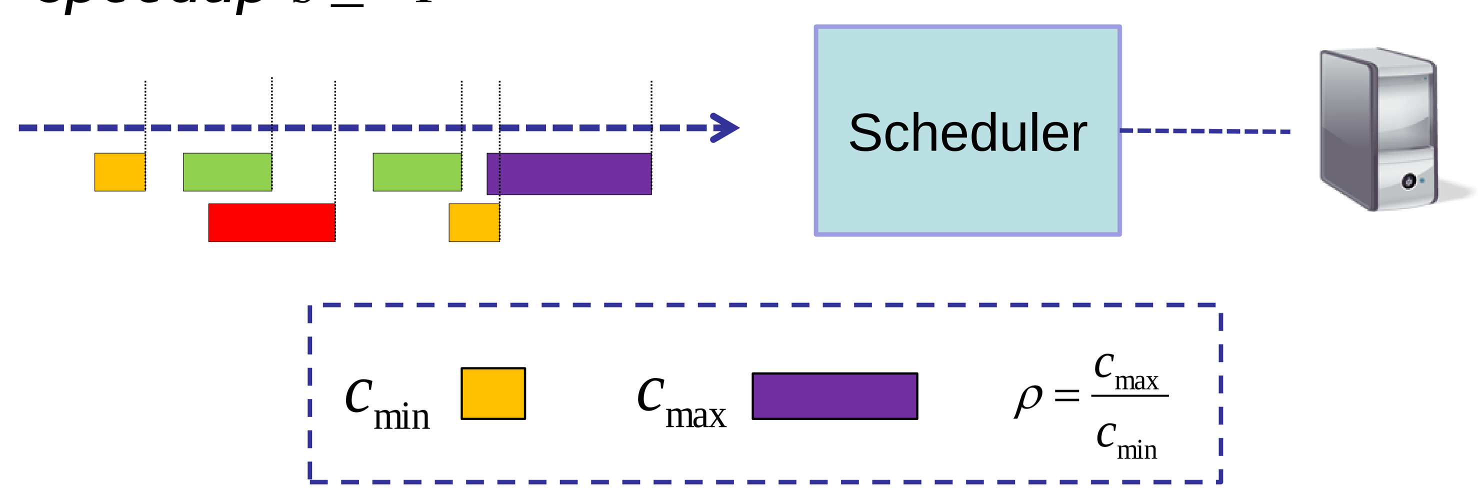
We perform *competitive analysis* of online scheduling algorithms under adversarial crash and arrival patterns **[1,3]**

Evaluation of 4 popular scheduling algorithms:

- Longest In System ***(LIS)***
- Shortest In System ***(SIS)***
- Largest Processing Time ***(LPT)***
- Shortest Processing Time ***(SPT)***

Introduce ***speed augmentation*** **[1,2]**

- *speedup* $s \geq 1$



Efficiency measures. *Asymptotic competitiveness*

Machine Utilization

Completed Cost $\mathcal{C}^s(ALG,A,E) = \inf_{A\in A, E\in E, X\in X} \lim_{t\to\infty} \frac{C_t^s(ALG,A,E)}{C_t^1(X,A,E)}$

Buffering

Pending Cost $\mathcal{P}^s(ALG,A,E) = \sup_{A\in A, E\in E, X\in X} \lim_{t\to\infty} \frac{P_t^s(ALG,A,E)}{P_t^1(X,A,E)}$

Fairness

Latency $\mathcal{L}^s(ALG,A,E) = \inf_{A\in A, E\in E, X\in X} \lim_{t\to\infty} \frac{L_t^s(ALG,A,E)}{L_t^1(X,A,E)}$

A : set of arrival patterns E : set of error patterns

$ALG \in \{LIS, SIS, LPT, SPT\}$ X : offline optimal algorithm

Speedup $s < \rho$

Algorithm	Completed Cost $\mathcal{C}^s(ALG,A,E)$	Pending Cost $\mathcal{P}^s(ALG,A,E)$	Latency $\mathcal{L}^s(ALG,A,E)$
LIS	0	∞	∞
SIS	0	∞	∞
LPT	0	∞	∞
SPT	$\left[\frac{1}{2+\rho}, \frac{\lceil\rho\rceil-1}{\lceil\rho\rceil-1+\rho}\right]$ *	∞	∞

* Lower bound holds for two task costs

If tasks can have *ANY* cost $c \in [c_{min}, c_{max}]$, and there is *no speedup* ($s = 1$), ***SPT*** is *NOT* completed cost *competitive*

Intuition for SPT lower bound:

1) Associate each task τ completed by X, with the following tasks completed by SPT up to current time t:
   - The same task τ
   - Task w begin executed at t_τ (time at which τ was scheduled), if it was not interrupted by a crash.
2) Prove that all tasks completed by SPT are associated with some task completed by X
3) Prove that there are tasks of at most c_{max} aggregate cost completed by X, not associated with tasks completed by SPT.

Speedup $s \geq \rho$

For all work conserving algorithms.

Speedup s	Completed Cost $\mathcal{C}^s(ALG,A,E)$	Pending Cost $\mathcal{P}^s(ALG,A,E)$
$s \geq \rho$	$[1/\rho, s]$	$[1, \rho]$
$s \geq 1+\rho$	$[1, s]$	1

Red bounds hold ***when*** the ***queue*** of pending tasks ***never*** becomes ***empty*** after a point in time

Intuition for $s \geq \rho$ general bounds:

We first show that for every task executed by X, ALG is also able to complete at least one task.

The results follow from the ***observation*** that the cost that can be completed by ALG is at least c_{min}/c_{max} times the cost of any task completed by X.

More specific results for each algorithm.

1-cost competitiveness achieved with speedup $s < 1+\rho$ by all algorithms

$c < \frac{c_{min}+c_{max}}{s}$

ALG	Speedup *s*	Comp. Cost	Pend. Cost	Latency
LIS	$s \in [\rho, 1+1/\rho]$	$\left[\frac{1}{\rho}, \frac{1}{2}+\frac{1}{2\rho}\right]$	$\left[\frac{1+\rho}{2}, \rho\right]$	(0,1]
	$s \in [\max\{\rho, 1+1/\rho\}, 2]$	$\left[\frac{1}{\rho}, \frac{1}{2}+\frac{c_{min}}{2\rho}\right]$	$\left[\frac{1}{2}+\frac{c}{2c_{min}}, \rho\right]$	(0,1]
	$s \geq \max\{\rho, 2\}$	[1,s]	1	(0,1]
SIS	$s \in [\rho, 1+1/\rho]$	$\left[\frac{1}{\rho}, \frac{c_{min}}{c}\right]$	$\left[\frac{c/c_{min}+\rho}{1+\rho}, \rho\right]$	∞
	$s \geq 1+1/\rho$	[1,s]	1	∞
LPT	$s \geq \rho$	[1,s]	1	∞
SPT	$s \geq \rho$	[1,s]	1	∞

Proving that, at every time instant that ***LIS*** completes a task, say **w**, there is some task **τ** pending in ***X***, for at least as long as **w**.

References

[1] Antonio Fernández Anta, Chryssis Georgiou, Dariusz R. Kowalski, and Elli Zavou. *Online parallel scheduling of non-uniform tasks: Trading failures for energy*. In FCT, pages 145-158, 2013.
[2] Leah Epstein and Rob van Stee. *Online bin packing with resource augmentation*. Discrete Optimization, 4(34):322 – 333, 2007.
[3] Eric Sanlaville and Gunter Schmidt. *Machine scheduling with availability constraints*. Acta Informatica, 35(9):795-811, 1998.

© IMDEA Networks Institute