Optimization Metrics for Batch-Scheduler Evaluation

(how to use them slightly better)

Guillaume Pallez

Inria

Scheduling Variable Capacity Resources for Sustainability, March’23

Work done with Robin Boezennc and Fanny Dufossé
How to evaluate the Performance of a Scheduling solution?

1. Presentation and discussion of several objectives;
2. Analysis of a mainstream Use-Case
1. Presentation and discussion of several objectives;
2. Analysis of a mainstream Use-Case

Some notations, given job $J_i$:

- $r_i$: Release time (aka submission time)
- $C_i$: Completion time
- $t_i^{\text{real}}$: Length (aka execution time)
- $t_i^{\text{wait}} = C_i - r_i - t_i^{\text{real}}$: Waiting time
**Mean Bounded Slowdown (I)**

\[ S^b_i = \max \left( \frac{C_i - r_i}{\max \left( t_i^{\text{real}}, \tau \right)}, 1 \right) \rightarrow \text{minimize} \sum_i S^b_i \]

Used in *many* recent work; Also called *flow, stretch*..
Mean Bounded Slowdown (I)

\[ S_i^b = \max \left( \frac{C_i - r_i}{\max \left( t_i^{\text{real}}, \tau \right)}, 1 \right) \rightarrow \text{minimize} \sum_i S_i^b \]

Used in many recent work; Also called flow, stretch..

Why do people like it?

- Includes all applications
- Some sort of fairness
Mean Bounded Slowdown (I)

\[ S_i^b = \max \left( \frac{C_i - r_i}{\max\left( t_i^{\text{real}}, \tau \right)}, 1 \right) \rightarrow \text{minimize} \sum_{i} S_i^b \]

Used in many recent work; Also called flow, stretch..

Why do people like it?

- Includes all applications
- Some sort of fairness

“my students can provide theoretical guarantees”
Mean Bounded Slowdown (II): Limits

Which schedule do you like best?

Top MBSD: 2.8
Bottom MBSD: 1.8
Bottom is "better"!
Why do people like it?
It is easy to significantly improve...
Mean Bounded Slowdown (II): Limits

Which schedule do you like best?

- Top MBSD: 2.8
- Bottom MBSD: 1.8

Bottom is "better"!
Which schedule do you like best?

- Top MBSD: 2.8
- Bottom MBSD: 1.8

Bottom is "better"!

Why do people like it?
It is easy to significantly improve...
Mean/Max Response Time (I)

\[ RT_i = t_i^{\text{wait}} + t_i^{\text{real}} \rightarrow \text{minimize } \max_i RT_i \text{ (or } \sum_i RT_i \text{)} \]

Used in many recent work. Also known as wait time (equivalent).
Mean/Max Response Time ($R_i$)

$$RT_i = t_{i}^{\text{wait}} + t_{i}^{\text{real}} \quad \rightarrow \text{minimize} \quad \max_{i} RT_i \quad (\text{or} \quad \sum_{i} RT_i)$$

Used in many recent work. Also known as wait time (equivalent).

Why do people like it?

- From a user-perspective: gives information (how long before I get a result)

Why don’t people like it?

- No difference between:
  - a 1min long application that waits for 10h
  - a 10h long application that waits for 1min
Which schedule do you like best?
Which schedule do you like best?

- **Top Response Time**: 3.6
- **Bottom Response Time**: 3

Bottom is "better"!
Utilization (I)

On interval $[t_1, t_2]$ ($N$= number of nodes, $W$=amount of work done):

$$U(t_1, t_2) = \frac{W(t_1, t_2)}{N \cdot (t_2 - t_1)}.$$ 

Also called goodput (apparently)
Utilization (I)

On interval $[t_1, t_2]$ ($N =$ number of nodes, $W =$ amount of work done):

$$U(t_1, t_2) = \frac{W(t_1, t_2)}{N \cdot (t_2 - t_1)}.$$ 

Also called goodput (apparently)

Why do people like it?
- Natural perspective of sysadmin

Why don’t people like it?
- Negligible gains in general
Utilization (I)

On interval $[t_1, t_2]$ ($N =$ number of nodes, $W =$ amount of work done):

$$U(t_1, t_2) = \frac{W(t_1, t_2)}{N \cdot (t_2 - t_1)}.$$ 

Also called *goodput* (apparently)

**Why do people like it?**
- Natural perspective of sysadmin

**Why don’t people like it?**
- Negligible gains in general

“It’s a bad objective, no-approximation guarantees or too hard to analyze.”
Utilization highly dependent on job arrival rate.

1. When too few jobs ($U < 93\%$): no utilization difference
2. When too many: system is saturated, analysis is not valid.
Utilization Standard Deviation (I)

- Utilization: 0.625
- Standard Deviation: 0.5625

- Utilization: 0.625
- Standard Deviation: 0.125
**Use-case: Runtime estimates (I)**

**Runtime under-estimated:**
- Job killed, need to resubmit; additional cost to user.
- Waste of system resources.

**Runtime overestimated:**
- May waste system resources (if no backfilling possible).
Use-case: Runtime estimates (II)

Analysis:
- 70 scenarios from Mira and Theta logs
- Experiments on Simgrid
- Exact vs User-walltime for walltime prediction
**Use-case: Runtime estimates (II)**

**Analysis:**
- 70 scenarios from Mira and Theta logs
- Experiments on Simgrid
- **Exact vs User-walltime** for walltime prediction

**Improvement of Exact over User-walltime**
1. Mean Bounded Slowdown;
2. Mean Response Time;
3. Utilization.

Confirms that evaluating an objective is not informative.
More on utilization

(a) Data from Mira

(b) Data from Theta

Figure: Relative improvement of the Utilization of EXACT over USER-WALLTIME as a function of the Utilization
Relative improvement of the standard deviation of the utilization as a function of the relative improvement of the utilization (Mira and Theta data)
Stop using the mean bounded slowdown. Seriously.
Stop using the mean bounded slowdown. Seriously. Understand and analyze critically your performance.

https://hal.inria.fr/hal-04042591