IPDPS
2022 • VIRTUAL •

Bounding the Flow Time in Online Scheduling with Structured Processing Sets

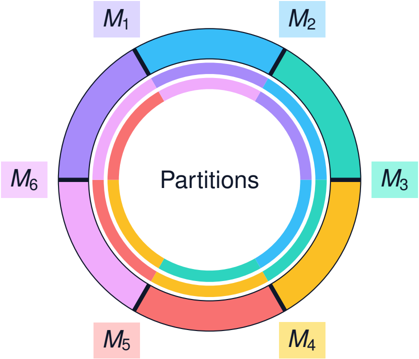
L.-C. Canon[1], A. Dugois[2] and L. Marchal[2]

[1]FEMTO-ST, Univ. Franche-Comté, France
[2]LIP, ENS Lyon, Inria, France

June 1, 2022 – 36th IEEE International Parallel and Distributed Processing Symposium

Introduction

Applicative Context



Key-value store (KVS)

Database whose values are bound to unique keys.

Distributed model

- Each server $M_1, M_2, \ldots$ holds a data partition.
- Partitions are replicated on different servers.
- Several servers may process a read query.

Introduction

General Problem

Scheduling problem

Schedule requests to bound the reponse time F_i of each request i.

Graham	Type	Description
P	Constraint	Homogeneous environment
$\mathcal{M}_i$	Constraint	Restricted assignment
online-r_i	Constraint	Online over time model
$\circ$	Constraint	No preemption
$F_{\max}$	Objective	Maximum response time

Introduction

General Problem

Scheduling problem

Schedule requests to bound the reponse time F_i of each request i.

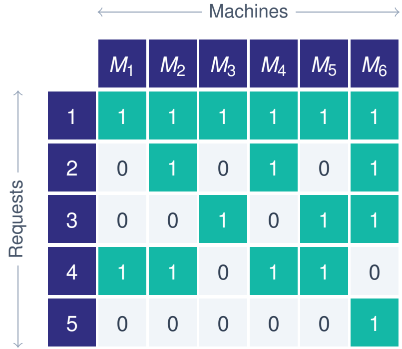
Processing set restriction

$\mathcal{M}_i$ is the subset of machines able to process request i.

Graham	Type	Description
P	Constraint	Homogeneous environment
$\mathcal{M}_i$	Constraint	Restricted assignment
online-r_i	Constraint	Online over time model
$\circ$	Constraint	No preemption
$F_{\max}$	Objective	Maximum response time

Processing Set Structures

General (no structure)



	M_1	M_2	M_3	M_4	M_5	M_6
1	1	1	1	1	1	1
2	0	1	0	1	0	1
3	0	0	1	0	1	1
4	1	1	0	1	1	0
5	0	0	0	0	0	1

General (no structure)

Processing sets exhibit no particular structure.

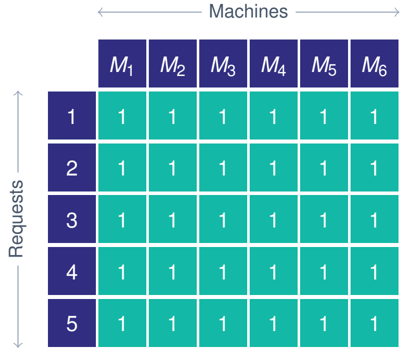
Competitive ratio

Lower bound $= \boldsymbol{\Omega(m)}$ [Anand et al., 2017]

$m =$ number of machines

Processing Set Structures

Filled structure



	M_1	M_2	M_3	M_4	M_5	M_6
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	1	1	1	1	1	1

Filled (full replication)

Any machine can process a given request.

Earliest Finish Time (EFT)

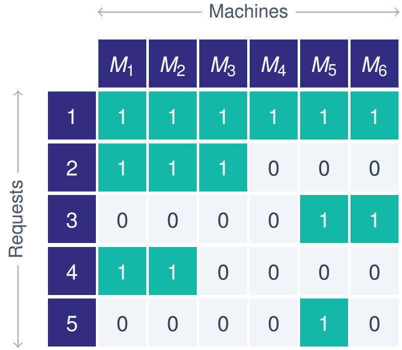
Schedule each arriving request on the eligible machine that completes it first.

Competitive ratio

EFT is ($3 - 2/m$)-competitive [Bender et al., 1998]

Processing Set Structures

Nested structure



	M_1	M_2	M_3	M_4	M_5	M_6
1	1	1	1	1	1	1
2	1	1	1	0	0	0
3	0	0	0	0	1	1
4	1	1	0	0	0	0
5	0	0	0	0	1	0

Nested

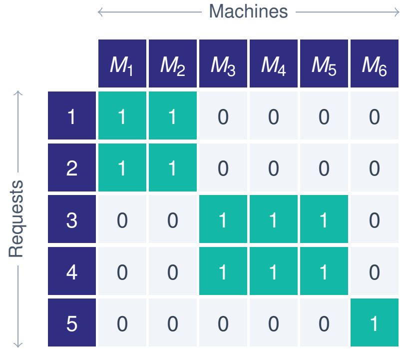
Two processing sets are either nested or disjoint.

Competitive ratio

Lower bound $= \frac{1}{3}\lfloor \log_2(m) + 2 \rfloor$

Processing Set Structures

Disjoint structure



	M_1	M_2	M_3	M_4	M_5	M_6
1	1	1	0	0	0	0
2	1	1	0	0	0	0
3	0	0	1	1	1	0
4	0	0	1	1	1	0
5	0	0	0	0	0	1

Disjoint

Two processing sets are either equal or disjoint.

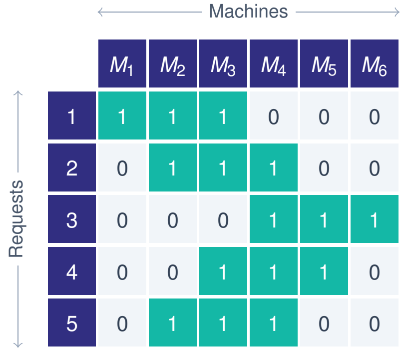
Competitive ratio

EFT is $(3 - 2/\max_i |\mathcal{M}_i|)$-competitive

$|\mathcal{M}_i| =$ number of machines able to process req. i

Processing Set Structures

Fixed-size interval structure



	M_1	M_2	M_3	M_4	M_5	M_6
1	1	1	1	0	0	0
2	0	1	1	1	0	0
3	0	0	0	1	1	1
4	0	0	1	1	1	0
5	0	1	1	1	0	0

Fixed-size interval (common in KVS)

Each processing set is a contiguous interval of size k.

Competitive ratio

Lower bound for EFT $= m - k + 1$

Processing Set Structures

Summary

Processing set structure	Algorithm	Competitive ratio
Filled	EFT	$3 - 2/m$
Disjoint	EFT	$\mathbf{3 - 2/\max_i \|\mathcal{M}_i\|}$
General	Any	$\geq \Omega(m)$
Nested	Any	$\boldsymbol{\geq \frac{1}{3}\lfloor \log_2(m) + 2 \rfloor}$
Interval	EFT	$\boldsymbol{\geq m - k + 1}$

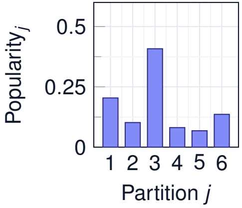
Processing Set Structures

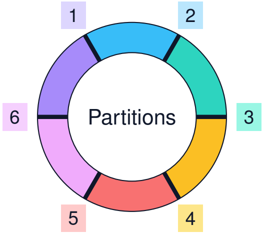
Summary

Processing set structure	Algorithm	Competitive ratio
Filled	EFT	$3 - 2/m$
Disjoint	EFT	$\mathbf{3 - 2/\max_i \lvert\mathcal{M}_i\rvert}$
General	Any	$\geq \Omega(m)$
Nested	Any	$\geq \frac{1}{3}\lfloor\log_2(m) + 2\rfloor$
Interval	EFT	$\geq m - k + 1$

Performance under Biased Popularity

Introducing popularity biases



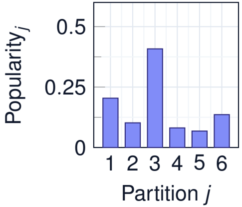


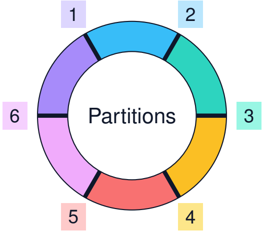
Partition popularity

- Recall each request looks for data.
- Some data may be more popular.
- **Popularity_j = probability to choose j**

Performance under Biased Popularity

Introducing popularity biases



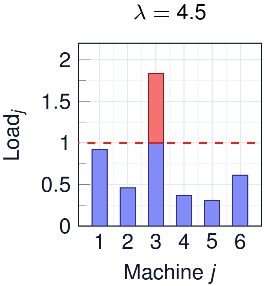


Partition popularity

- Recall each request looks for data.
- Some data may be more popular.
- **Popularity$_j$ = probability to choose j**

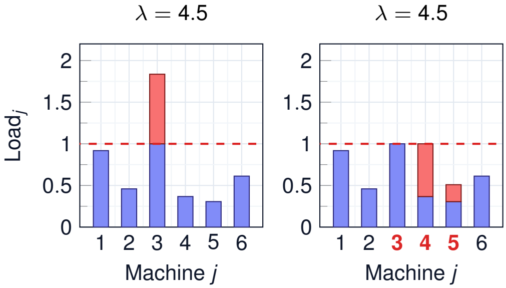
Machine load

- **λ = number of arriving req. per time unit**
- $\text{Load}_j = \lambda \cdot \text{Popularity}_j$
- $\text{Load}_j > 1 \iff j$ is overloaded



Performance under Biased Popularity

Load balancing

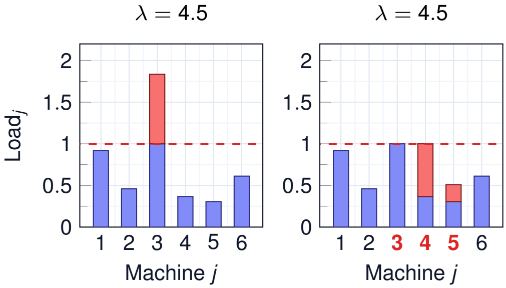


Replicate to load-balance

- Copy partition 3 on machines 4 and 5.
- Hypothesis: perfect load balancer.
- $\lambda = 4.5$ becomes feasible!

Performance under Biased Popularity

Load balancing



$$\begin{aligned} \textbf{maximize} \quad & \lambda \\ \text{subject to} \quad & \forall j, \sum_i a(i,j) = \text{Load}_j, \\ & \forall i, \sum_j a(i,j) \leq 1, \\ & \forall i,j,\ i \notin \text{Rep}_j \implies a(i,j) = 0, \\ & \forall i,j,\ a(i,j) \geq 0 \end{aligned}$$

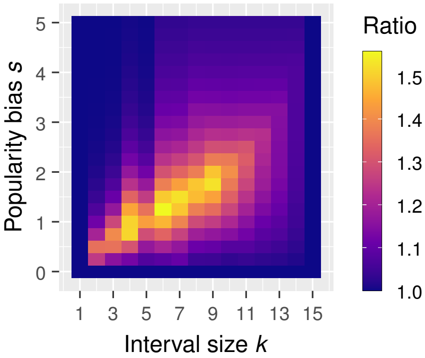
Replicate to load-balance

- Copy partition 3 on machines 4 and 5.
- Hypothesis: perfect load balancer.
- $\lambda = 4.5$ becomes feasible!

Load_j : $\lambda \cdot \text{Popularity}_j$
Rep_j : set of replica of j
$a(i,j)$: work transfer from j to i

Performance under Biased Popularity

Comparing disjoint/interval structures



Max λ for disjoint/interval

- Popularity bias: Zipf's law (bias s).
- Replication strategies:
    - Fixed-size disjoint (size k).
    - Fixed-size interval (size k).
- Solve for each combination of k and s.

$$\mathrm{Ratio}(k, s) = \frac{\lambda_{\mathrm{max}}(\mathrm{interval}, k, s)}{\lambda_{\mathrm{max}}(\mathrm{disjoint}, k, s)}$$

Conclusion

Online problem

Often difficult even with structured processing sets.

Disjoint/interval structures

- **Disjoint**: strong guarantee on max-flow (EFT: $3 - 2/k$)
- **Interval**: higher resilience to load (+50% in some cases)

Perspectives

Is there a structure that offers both max-flow guarantee and good resilience to load?