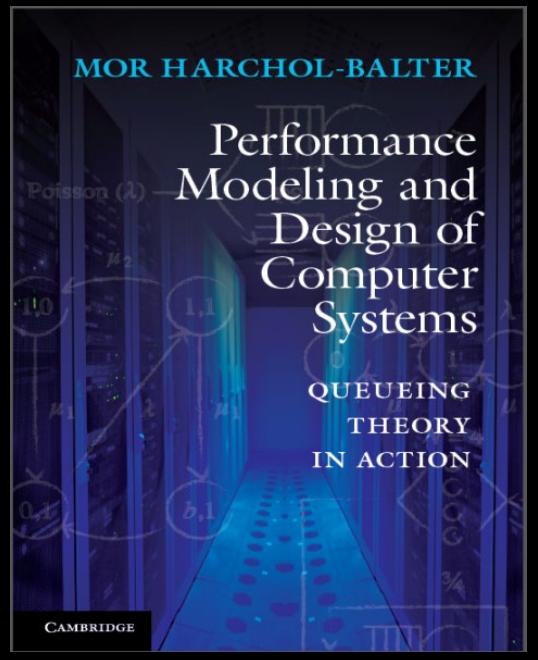
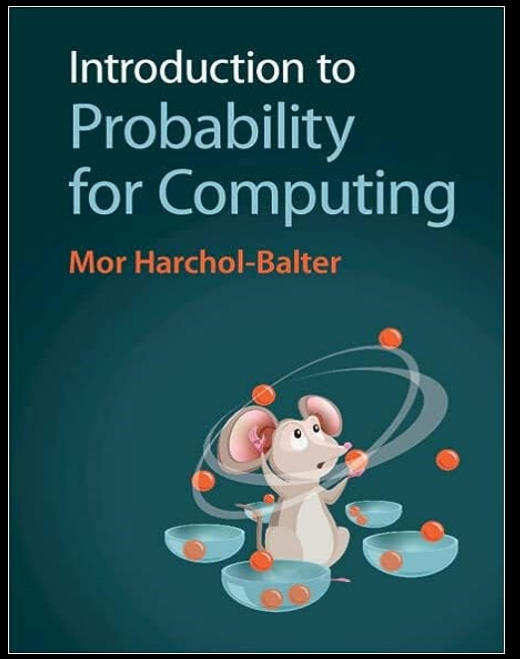
Optimal Core Allocation for Parallel Speedup Jobs

www.performancemodeling.org



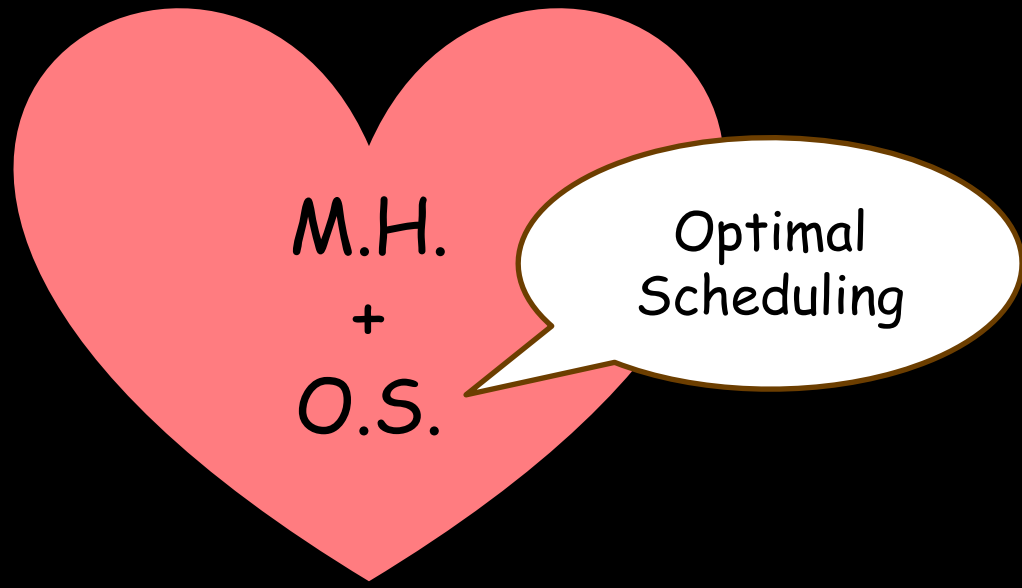
Mor Harchol-Balter
CMU
harchol@cs.cmu.edu

www.probabilitybook.org



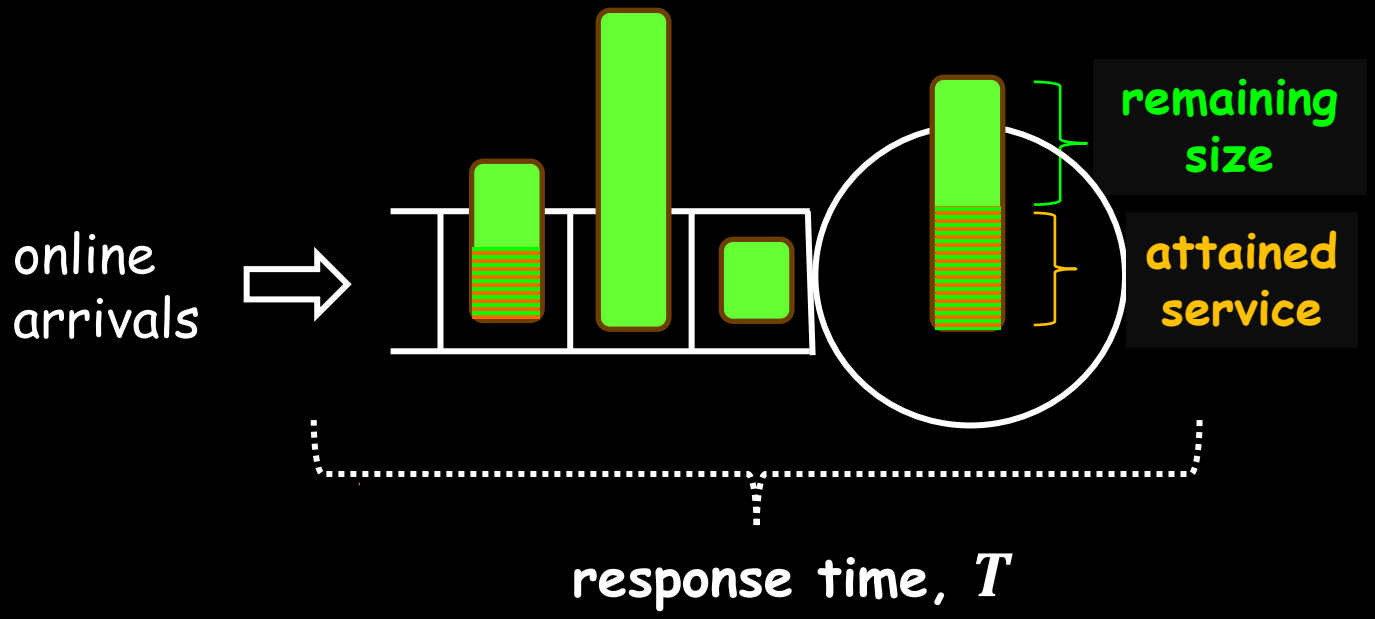
Outline

- Brief primer on scheduling



- Scheduling today's parallel speedup jobs

Scheduling 101



Job sizes are known

Fully preemptible

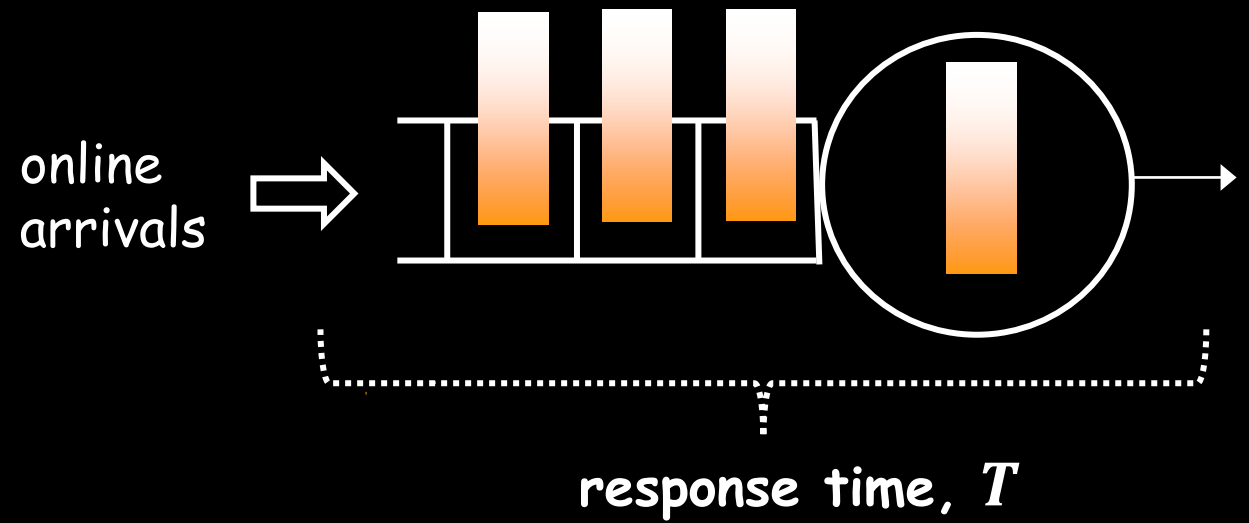
Q: What scheduling policy minimizes $E[T]$?

Shortest Remaining Processing Time (SRPT)

optimal in worst-case!

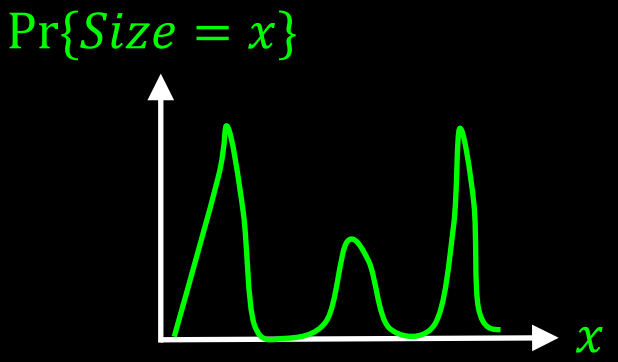
M/G/1 response time analysis – [Schrage 1966]

Scheduling 102 – Unknown job sizes



Job sizes unknown

But know job size distribution



Q: What scheduling policy minimizes $\boldsymbol{E}[T]$?

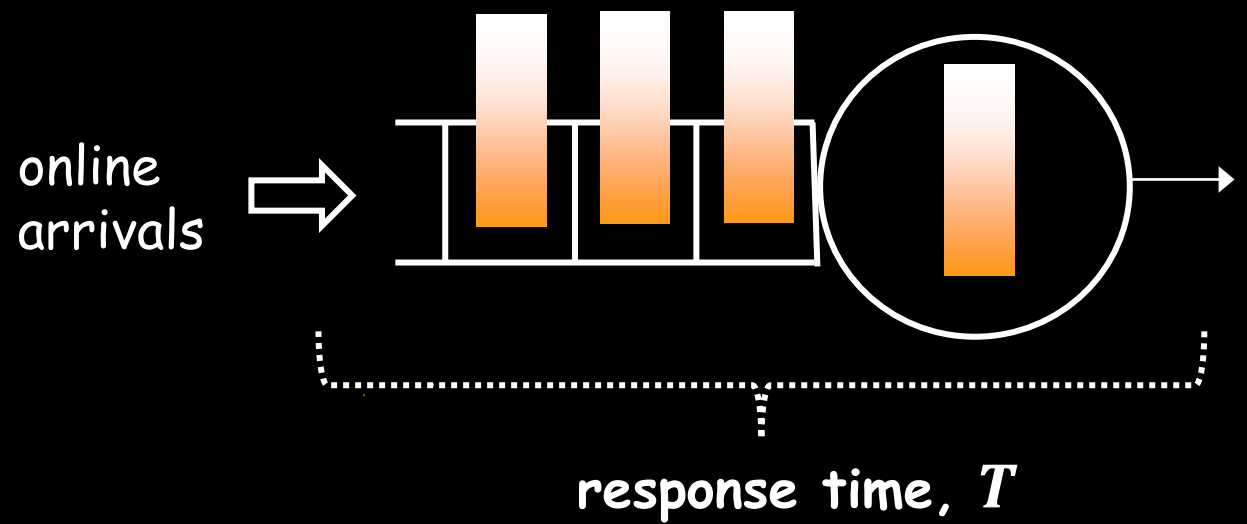
Shortest Expected Remaining Processing Time (SERPT) ?

Run job with smallest:

$$\boldsymbol{E}[Size - a \mid Size > a]$$

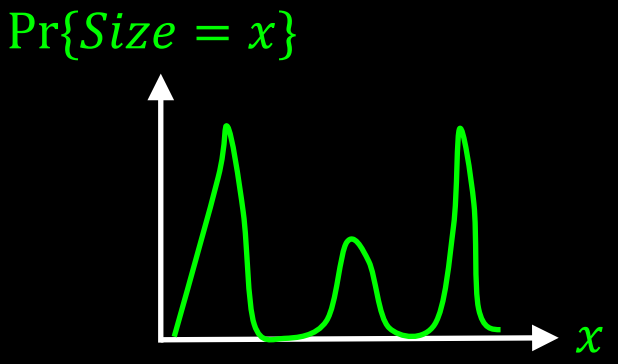
attained service

Scheduling 102 – Unknown job sizes



Job sizes unknown

But know job size distribution

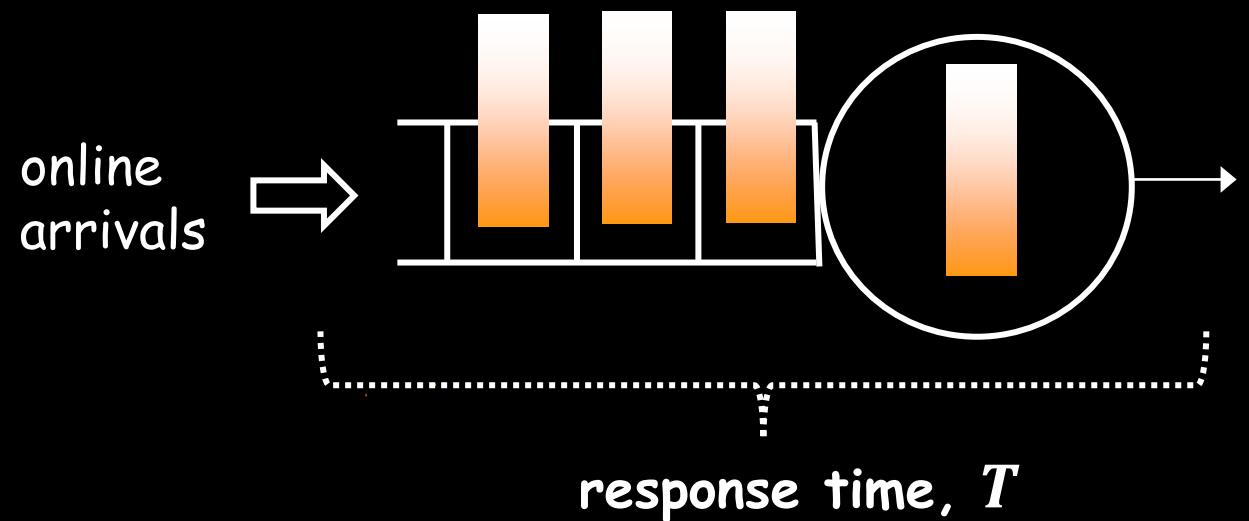


Gittins Index Policy: Run job with highest index

attained service

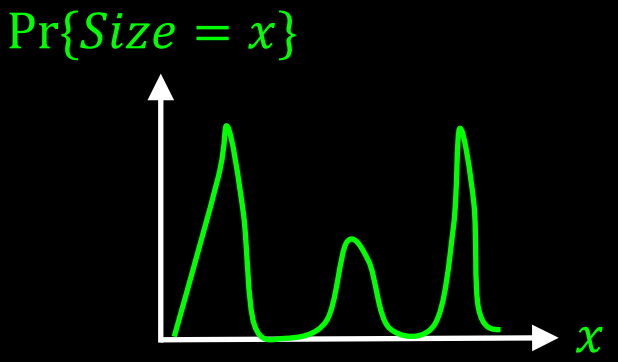
$$Index(a) = \sup_{y>a} \frac{\boldsymbol{P}\{Size \le y \mid Size \ge a\}}{\boldsymbol{E}[\min(Size, y) - a \mid Size \ge a\}} \approx \sup_{y>a} \frac{\boldsymbol{P}\{\text{Job finishes in next } y - a\}}{y - a}$$

Scheduling 102 – Unknown job sizes



Job sizes unknown

But know job size distribution



Gittins Policy:
[Gittins '89], [Gittins, Glazebrook, Weber 2011]

M/G/1 response time analysis:

"SOAP" paper

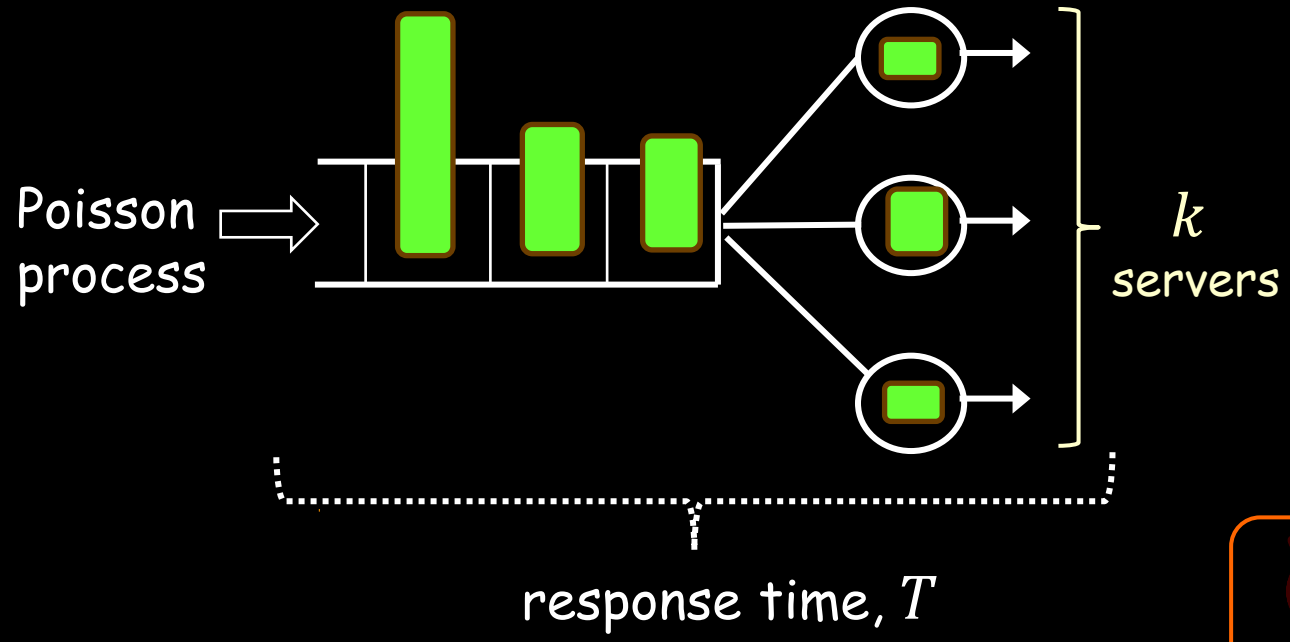
[Scully, Harchol-Balter, Scheller-Wolf 2018]

Gittins Index Policy: highest index first

attained service

$$Index(a) = \sup_{y>a} \frac{\boldsymbol{P}\{Size \le y \mid Size \ge a\}}{\boldsymbol{E}[\min(Size, y) - a \mid Size \ge a\}}$$

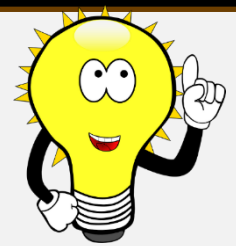
Scheduling 103 – M/G/k



Job sizes are known

Not worst-case optimal! [LR 1997]

Q: What scheduling policy minimizes $\boldsymbol{E}[T]$?

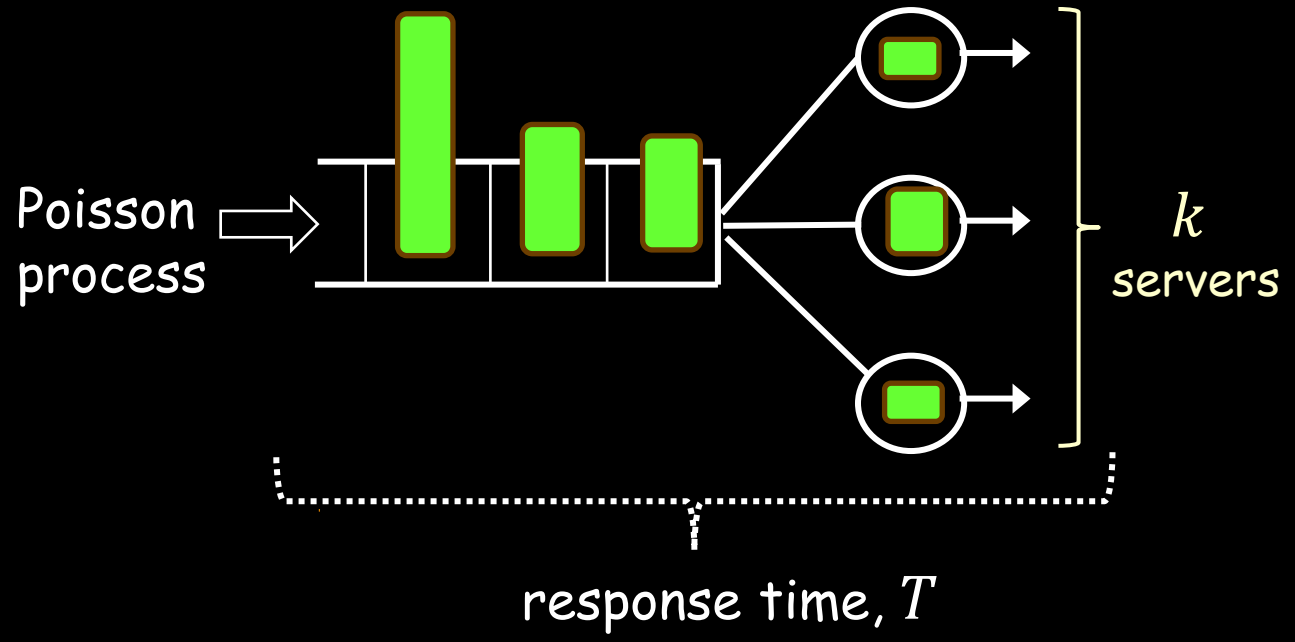


SRPT-k : At all times run the k jobs with shortest remaining processing time

THM: $$\lim_{\rho \to 1} \frac{\boldsymbol{E}[T]^{SRPT-k}}{\boldsymbol{E}[T]^{SRPT-1}} = 1$$

Similar result when job sizes unknown!

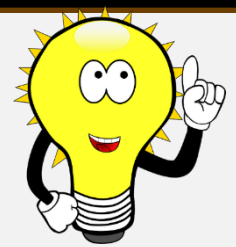
Scheduling 103 – M/G/k



Job sizes are known

SRPT-k result:
[Grosof, Scully, Harchol-Balter 2018]

Gittins-k result:
[Grosof, Scully, Harchol-Balter 2020]



SRPT-k : At all times run the k jobs with shortest remaining processing time

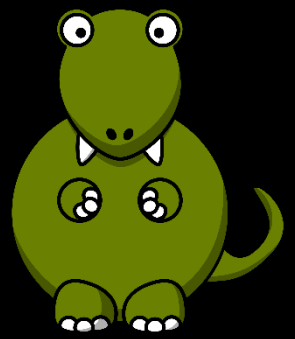
THM: $$\lim_{\rho \to 1} \frac{\boldsymbol{E}[T]^{SRPT-k}}{\boldsymbol{E}[T]^{SRPT-1}} = 1$$

Similar result when job sizes unknown!

Outline

- ☑ Brief primer on scheduling

Each job runs on a SINGLE server/core



- ☐ Scheduling today's parallel speedup jobs

Typical job runs on MULTIPLE cores simultaneously

A job runs on MANY cores simultaneously

Numbers from Google Borg 2020 trace
[Tirmazi et al., EUROSYS 2020]

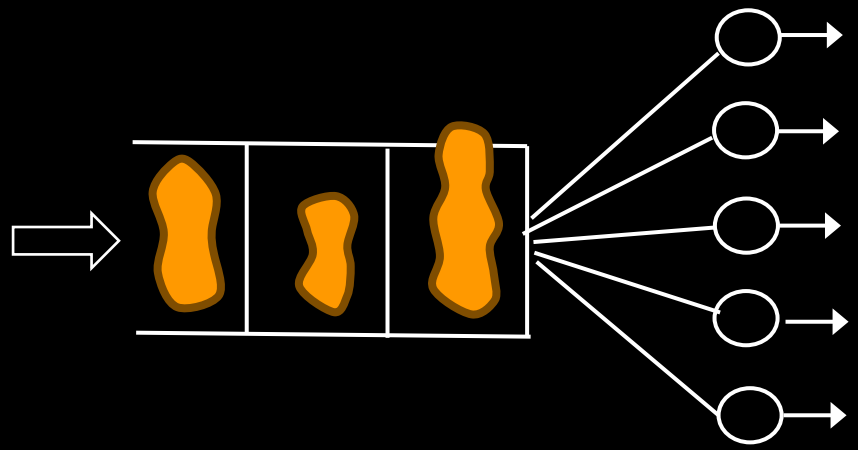
*Numbers are normalized

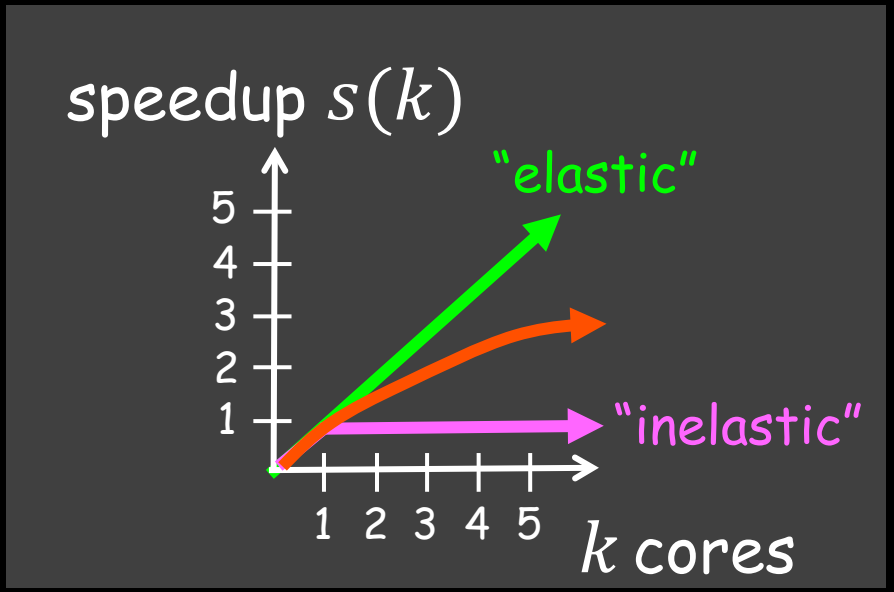
Parallel Speedup Model

Malleable jobs

- Can run job on ANY number cores
  Can change job's allocation as it runs

- Execution time depends on
  number of cores allocated





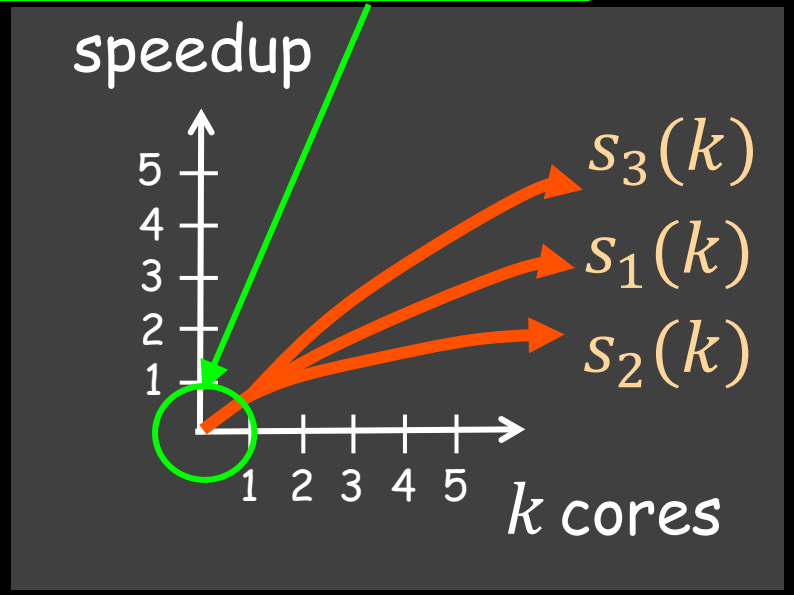
- database queries
- machine learning jobs
- some scientific computing jobs

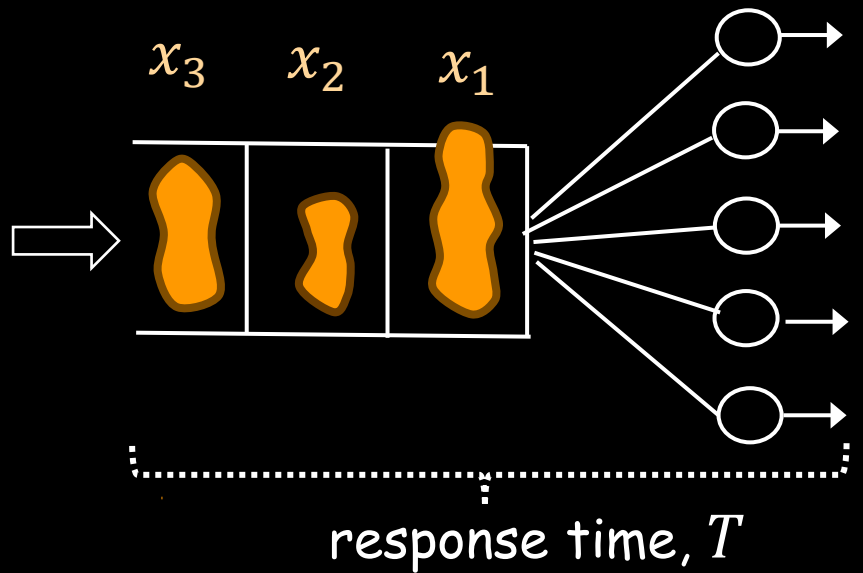
Parallel Speedup Model

Each job i has:

- inherent duration, x_i
- speedup function $s_i(k)$

Assume throughout: $\forall i$, $s_i(k) = k$ if $0 \le k \le 1$





	Execution time
When run on 1 core	x_i
When run on k cores	$\frac{x_i}{s_i(k)}$

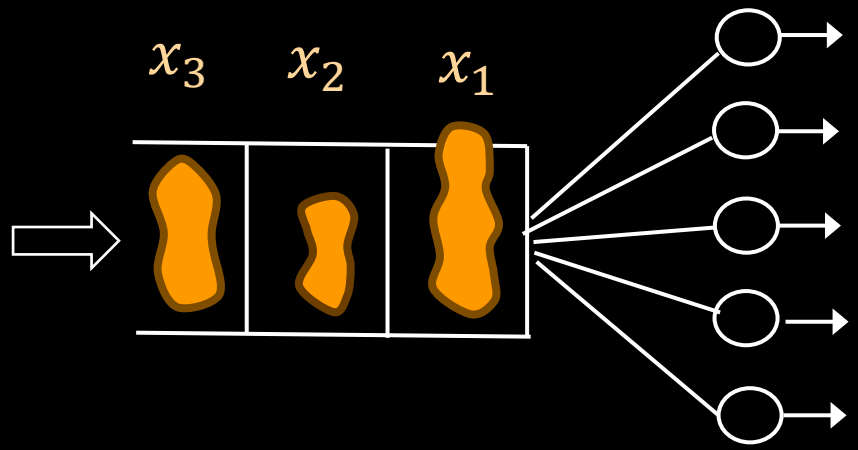
GOAL:

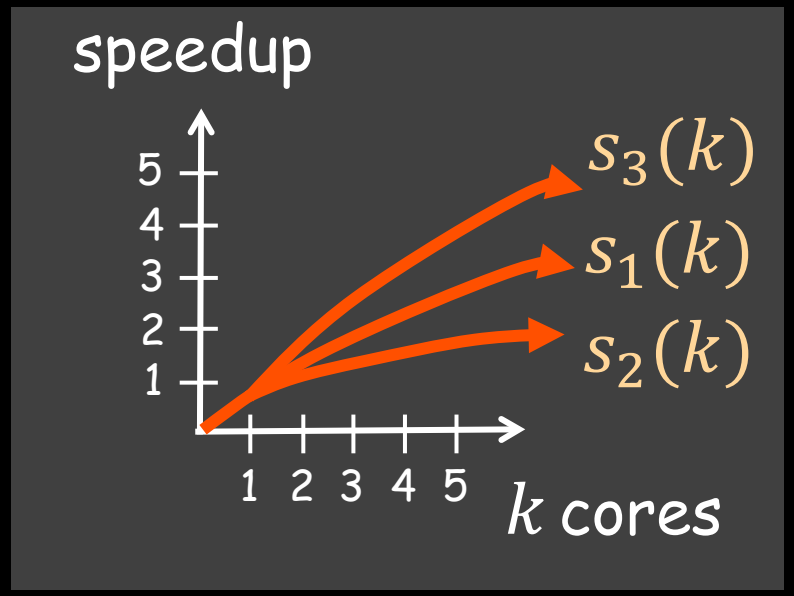
Minimize mean response time

Parallel Speedup Model

Each job i has:

- inherent duration, x_i
- speedup function $s_i(k)$





	Execution time
When run on 1 core	x_i
When run on k cores	$\frac{x_i}{s_i(k)}$

OUTLINE:

Minimize $\boldsymbol{E}[T]$

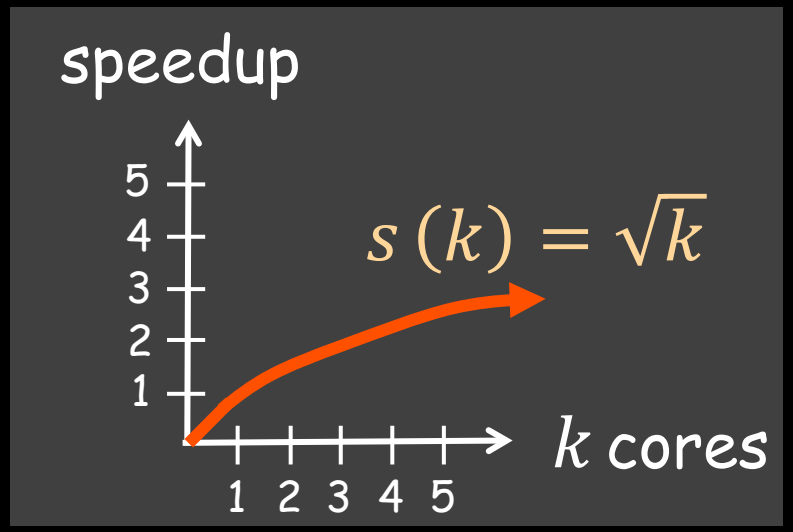
1) Single Speedup Function

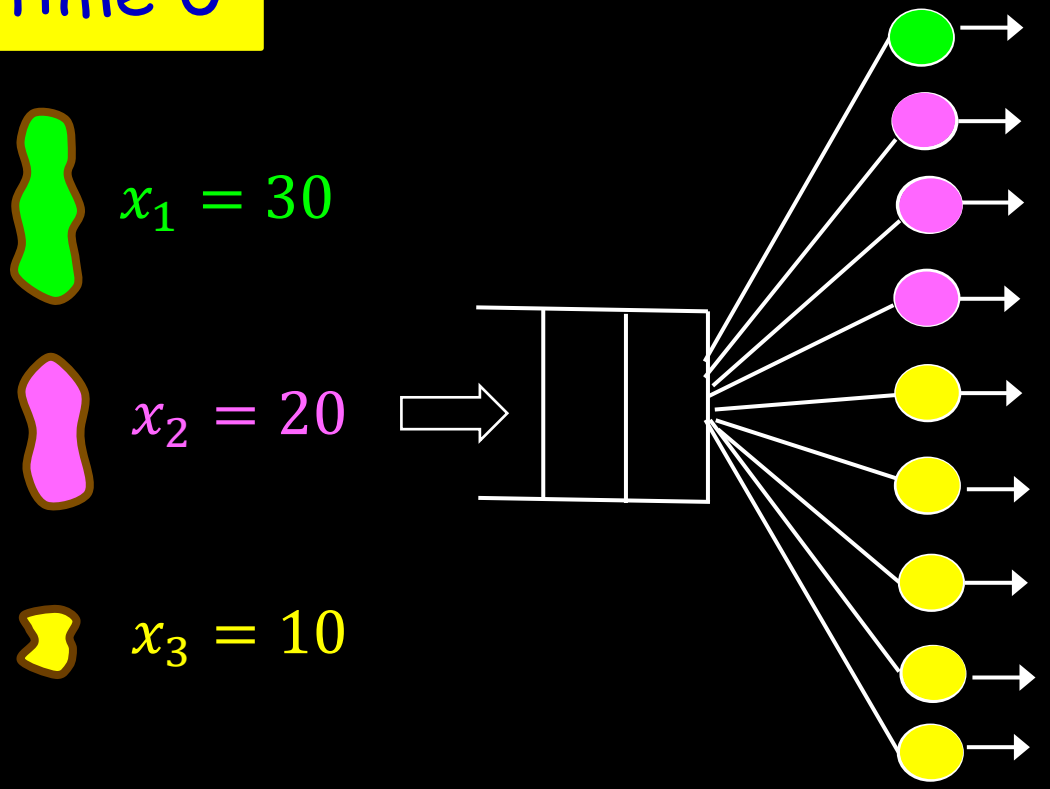
2) Multiple Speedup Functions

All jobs have SAME speedup function

All present at time 0

Speedup of all jobs:





favor shortest

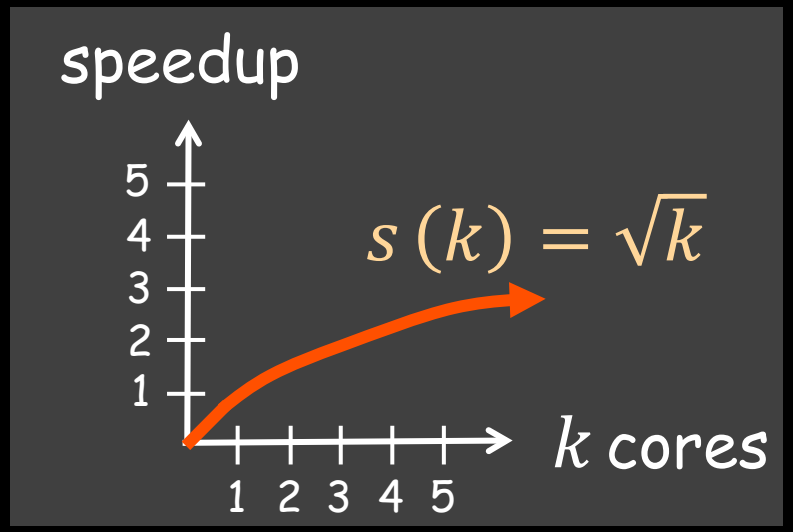
maximize efficiency

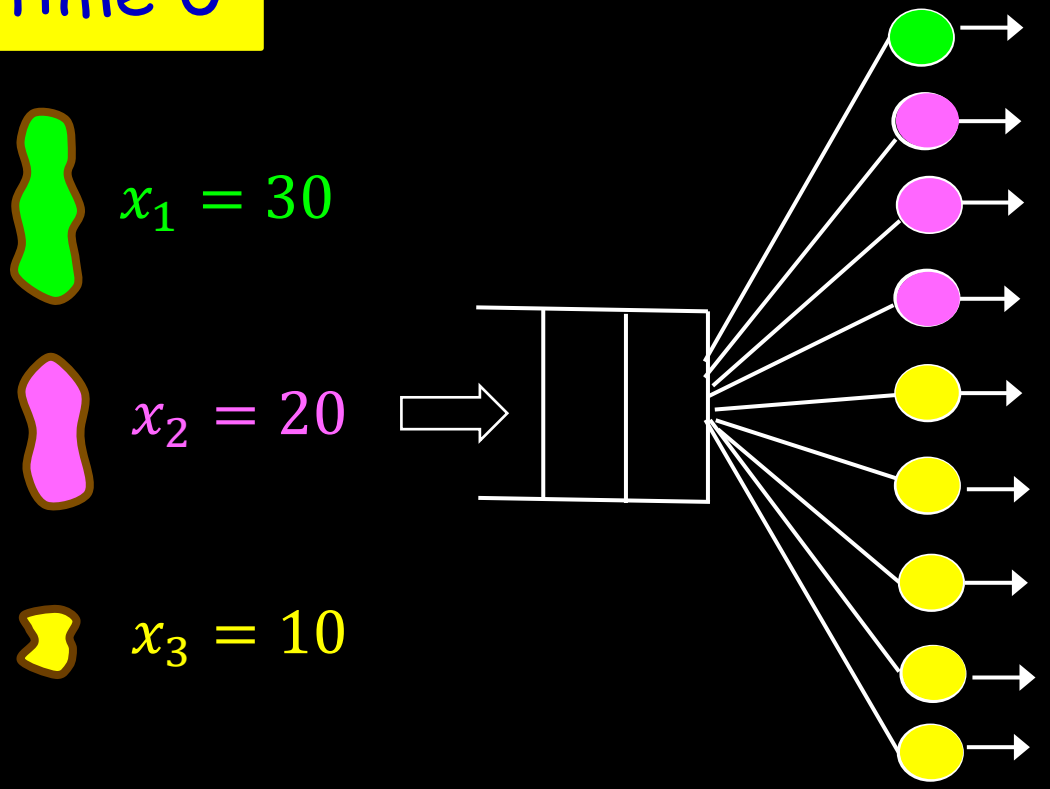
heSRPT

All jobs have SAME speedup function

All present at time 0

Speedup of all jobs:





favor shortest

maximize efficiency

Theorem (heSRPT):

[Berg, Vesilo, Harchol-Balter, Perf. Eval. 2020]

$m(t) = \#\text{jobs at time } t$

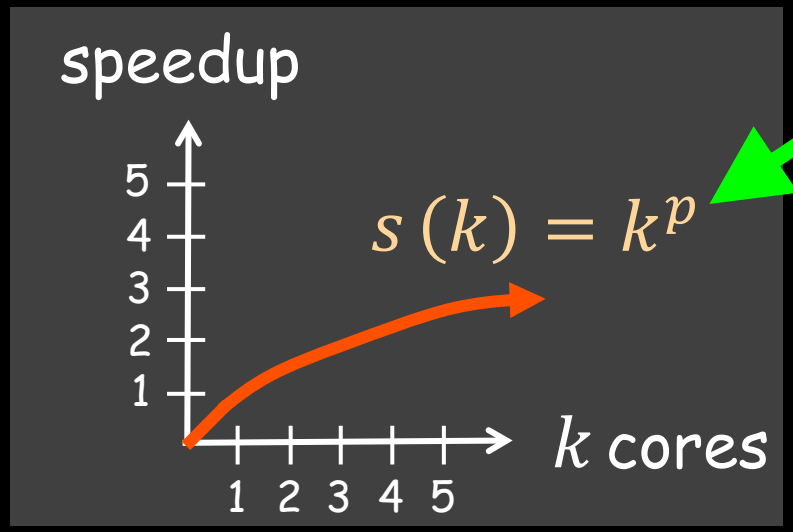
$x_1 > x_2 > x_3 > \cdots > x_{m(t)}$

$$\text{Opt share to job } i = \left(\frac{i}{m(t)}\right)^2 - \left(\frac{i-1}{m(t)}\right)^2$$

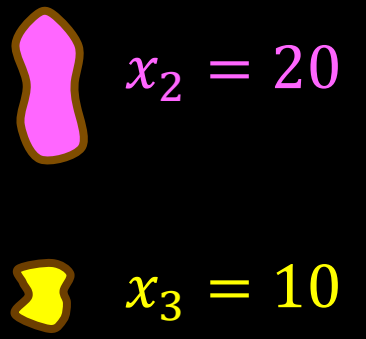
All jobs have SAME speedup function

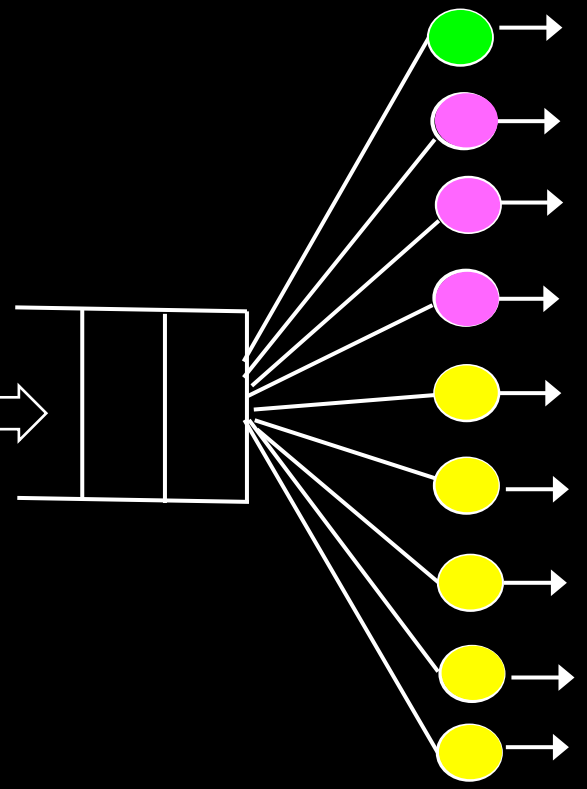
All present at time 0

Speedup of all jobs:



New work coming soon allowing any concave speedup function
[Li, Ju, Eryilmaz, Shroff '26]





favor shortest

maximize efficiency

Theorem (heSRPT):
[Berg, Vesilo, Harchol-Balter, Perf. Eval. 2020]

$m(t) = \#\text{jobs at time } t$

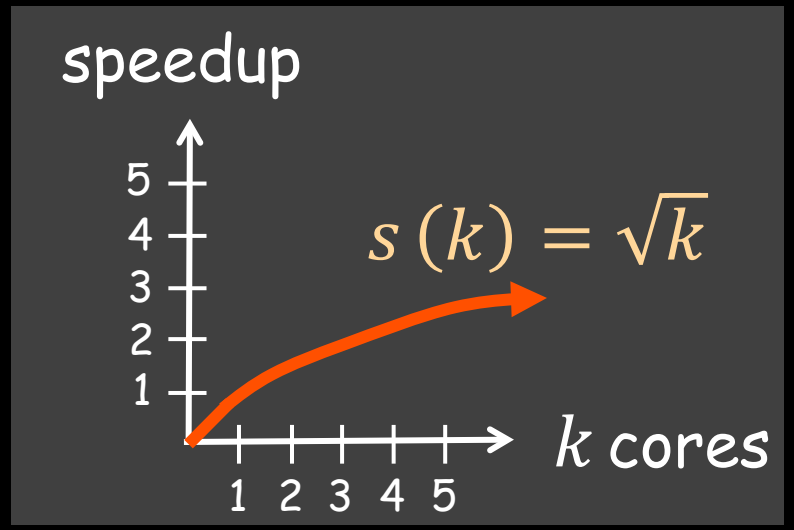
$x_1 > x_2 > x_3 > \cdots > x_{m(t)}$

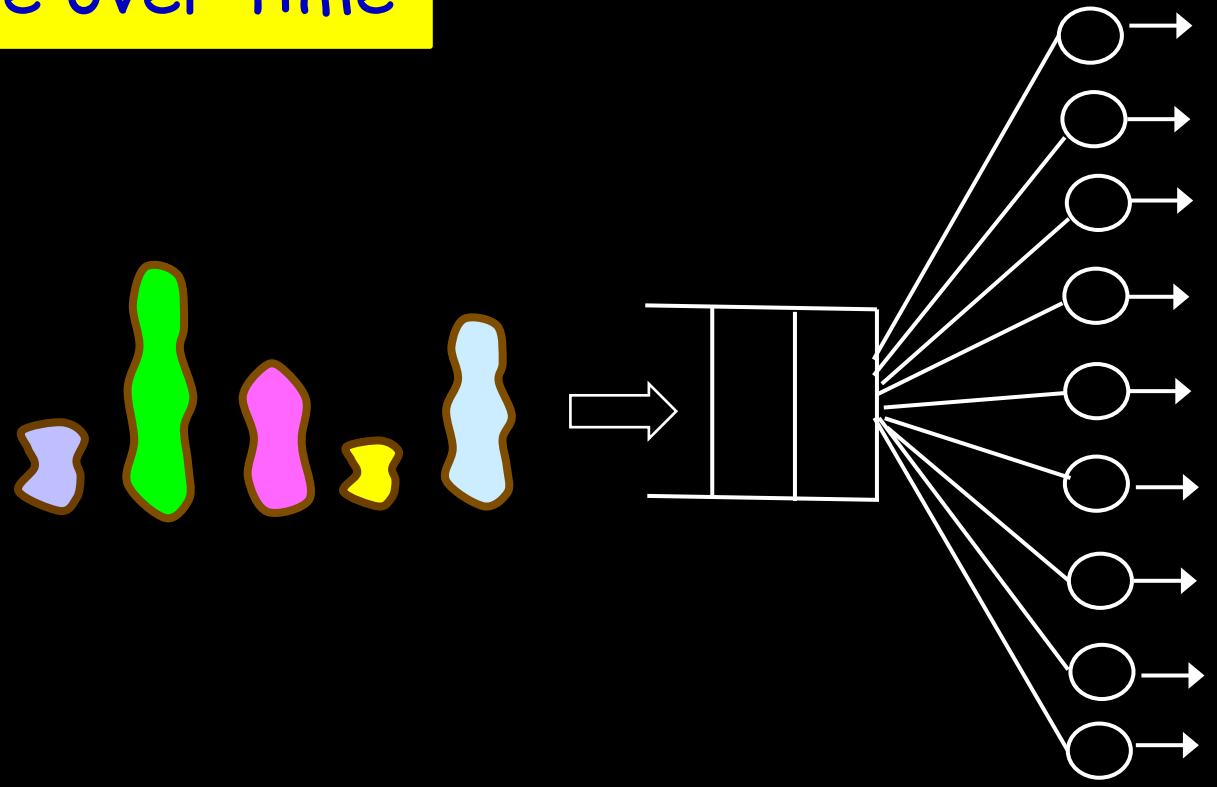
$$\text{Opt share to job } i = \left(\frac{i}{m(t)}\right)^{\frac{1}{1-p}} - \left(\frac{i-1}{m(t)}\right)^{\frac{1}{1-p}}$$

All jobs have SAME speedup function

jobs arrive over time

Speedup of all jobs:





favor shortest

maximize efficiency

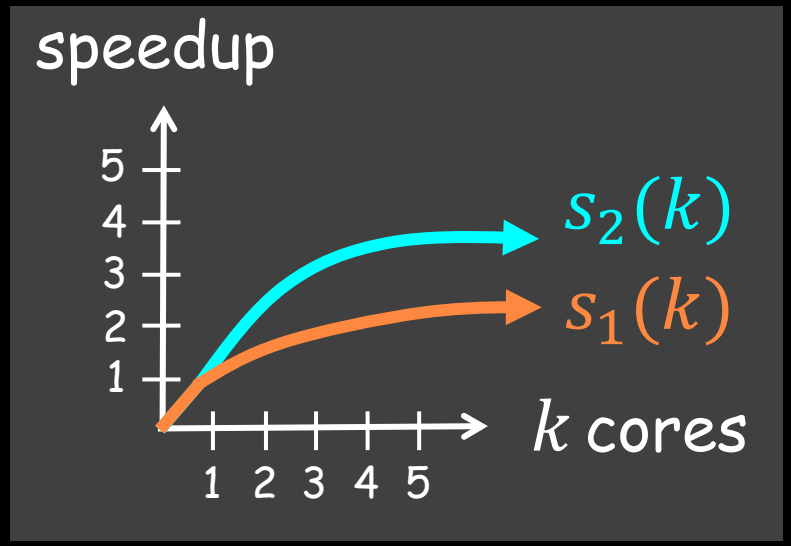
More relevant in light load

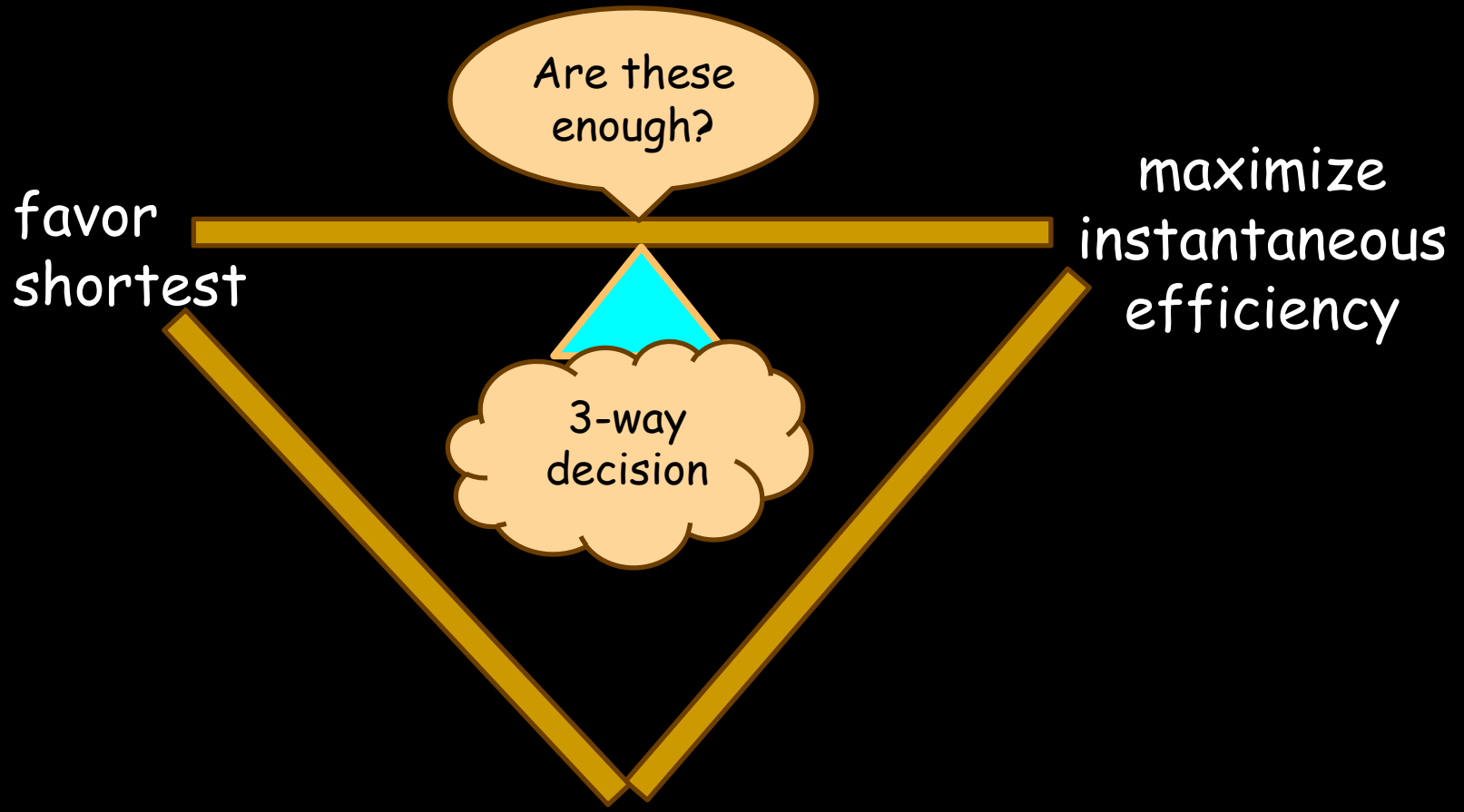
More relevant in heavy load

Q: Tradeoff?

MULTIPLE speedup functions

jobs arrive over time

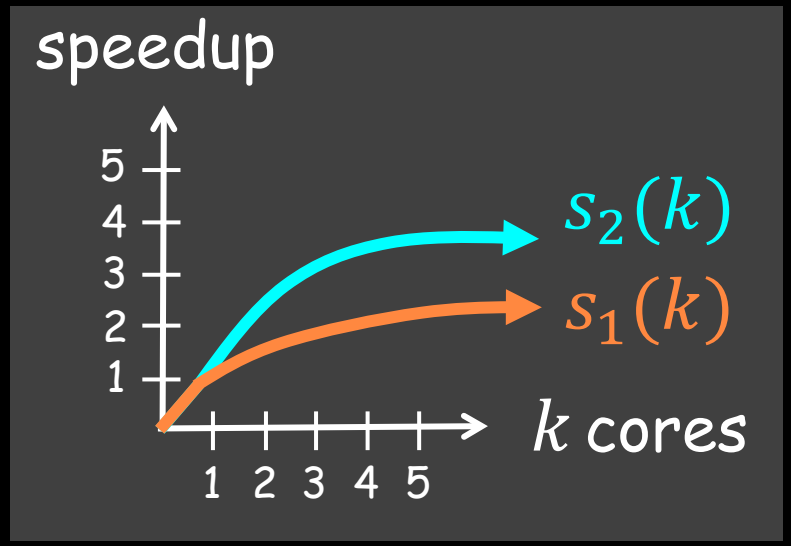




Defer Parallelizable Work
(maximize future efficiency)

MULTIPLE speedup functions: Simpler Job Sizes

jobs arrive over time



$Sizes \sim Exp(\mu)$

~~favor shortest~~

GREEDY class

maximize instantaneous efficiency

3-way decision

?

Pick the GREEDY policy that defers the most parallelizable

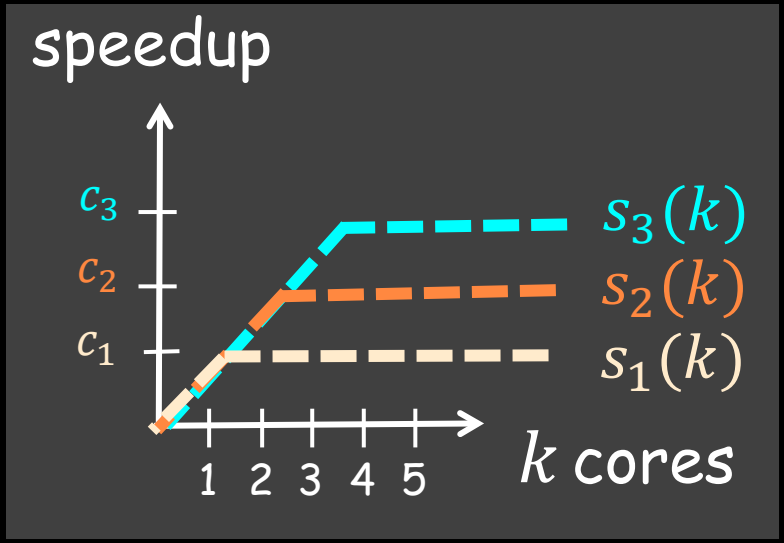
Still not OPT, but close.

Defer Parallelizable Work
(maximize future efficiency)

[Berg, Dorsman, Harchol-Balter Sigmetrics 2018]

MULTIPLE speedup functions: Simpler speedup fcns

jobs arrive over time



favor shortest

~~maximize instantaneous efficiency~~

?

THM: In super-Non-Deg-Slowdown,
- SERPT is opt.
- Least Parallelizable First is sub-opt

THM: In sub-Halfin-Whitt,
- Least Parallelizable First is opt.
- SERPT is sub-opt

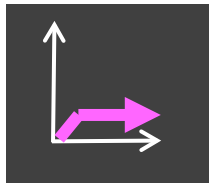
Defer Parallelizable Work (maximize future efficiency)

[Berg, Wang, Moseley, Harchol-Balter, 2025 in submission]

POP QUIZ

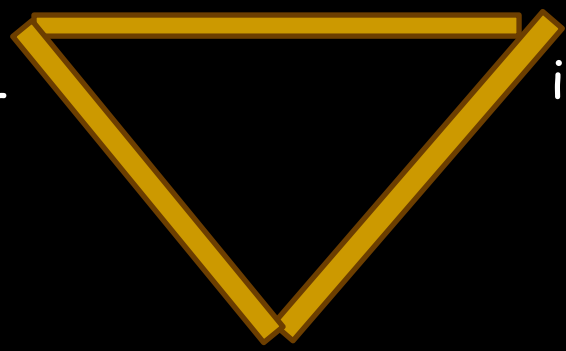
- ML Training jobs are Elastic & Large

- ML Inference jobs are Inelastic & Small



Q: Given a stream of training and inference jobs, how should we schedule them to minimize mean response time?

favor shortest



~~maximize instantaneous efficiency~~

Defer Parallelizable Work
(maximize future efficiency)

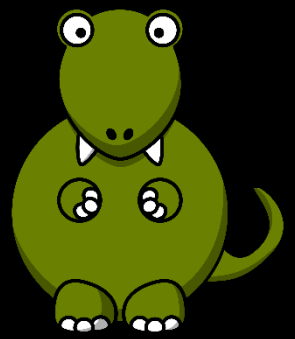
A: Favor Inference Jobs!

[Berg, Harchol, Moseley, Wang, Whitehouse SPAA 2020]

Outline

- ☑ Brief primer on scheduling

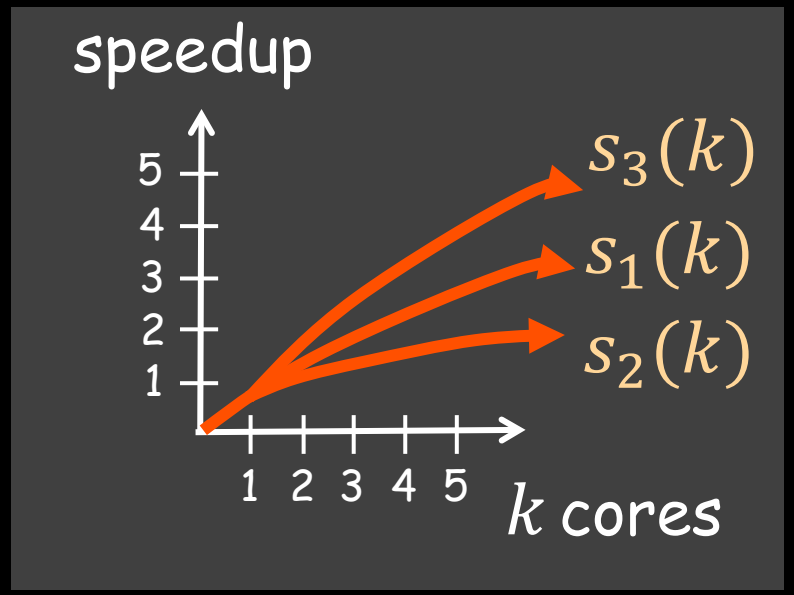
Each job runs on a SINGLE server/core



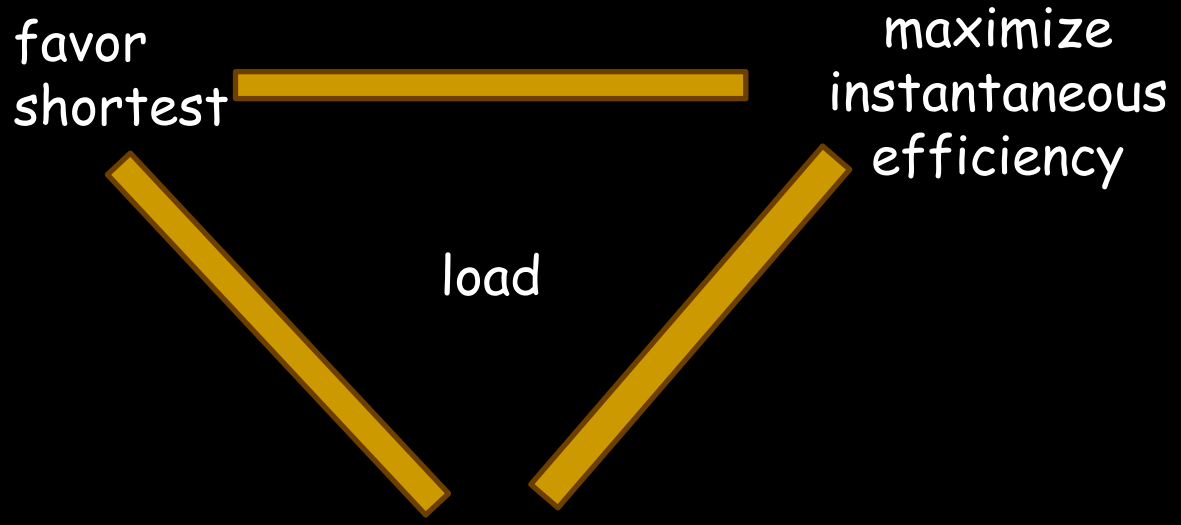
- ☑ Scheduling today's parallel speedup jobs

Typical job runs on MULTIPLE cores simultaneously

Conclusion: Parallel Speedup Jobs



	Execution time
When run on 1 core	x_i
When run on k cores	$\frac{x_i}{s_i(k)}$



Mor Harchol-Balter harchol@cs.cmu.edu

Our papers on Parallel Speedup Jobs

Benjamin Berg, Rein Vesilo and Mor Harchol-Balter. ``heSRPT: Parallel Scheduling to Minimize Mean Slowdown." In *Performance Evaluation*, vol. 144, 2020, pp. 1-20. Previously appeared in *38th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation (IFIP Performance 2020)* , Milan, Italy, November 2020. DOI 10.1016/j.peva.2020.102147 pdf .

Ben Berg, Jan-Pieter Dorsman, Mor Harchol-Balter. ``Towards Optimality in Parallel Job Scheduling." *Proceedings of the ACM on Measurement and Analysis of Computer Systems (ACM SIGMETRICS 2018)* vol. 1, no. 2, pp. 40:1 -- 40:30, December 2017, Los Angeles, CA. pdf .

Asymptotically optimal scheduling of multiple parallelizable job classes. B Berg, B Moseley, W Wang, M Harchol-Balter Under submission to Stochastic Systems 2025. arXiv:2404.00346 https://arxiv.org/pdf/2404.00346

Ben Berg, Mor Harchol-Balter. ``Optimal Scheduling of Parallel Jobs with Unknown Service Requirements." *Handbook of Research on Methodologies and Applications of Supercomputing*, 2021. pp. 18-40. DOI 10.4018/978-1-7998-7156-9.ch003. pdf .

Ben Berg, Justin Whitehouse, Ben Moseley, Weina Wang, Mor Harchol-Balter. ``The Case for Phase-Aware Scheduling of Parallelizable Jobs." *Performance Evaluation, vol. 153, 2022, pp. 1-20. Previously appeared in 38th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation, Milan, Italy, November 2021.* pdf . *DOI 10.1016/j.peva.2021.102246.*

Benjamin Berg, Mor Harchol-Balter, Ben Moseley, Weina Wang, Justin Whitehouse. ``Optimal Resource Allocation for Elastic and Inelastic Jobs.'' *32nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '20)* , Philadelphia, PA, July 2020, pp. 75-87. DOI 10.1145/3350755.3400265. pdf .

Our papers on Scheduling in 1-server-per-job

Ziv Scully, Mor Harchol-Balter, Alan Scheller-Wolf. ``SOAP: One Clean Analysis of All Age-Based Scheduling Policies." *Proceedings of the ACM on Measurement and Analysis of Computer Systems (ACM SIGMETRICS 2018)* vol. 2, no. 1, pp. 16:1 -- 16:30, March 2018, Los Angeles, CA. APS Best Student Paper Award Finalist pdf .

Ziv Scully, Mor Harchol-Balter. ``SOAP Bubbles: Robust Scheduling under Adversarial Noise.'' *56th Annual Allerton Conference on Communication, Control, and Computing (Allerton 2018)* , Urbana-Champaign, IL, October 2018. pdf , pp. 144--154.

Ziv Scully, Isaac Grosof and Mor Harchol-Balter. ``Optimal Multiserver Scheduling with Unknown Job Sizes in Heavy Traffic." In *Performance Evaluation*, vol. 145, 2021, pp. 1-31. Previously appeared in *38th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation (IFIP Performance 2020)* , Milan, Italy, November 2020. DOI 10.1016/j.peva.2020.102150 pdf

Isaac Grosof, Ziv Scully, Mor Harchol-Balter, Alan Scheller-Wolf. ``Optimal Scheduling in the Multiserver-job Model under Heavy Traffic.'' *Proceedings of the ACM Measurement and Analysis of Computer Systems -- SIGMETRICS:* Volume 6, Number 3, Article 51 (Sigmetrics 2023), pp. 51:1 -- 51:32, December 2022. DOI 10.1145/3570612. pdf .

Ziv Scully, Isaac Grosof and Mor Harchol-Balter. ``The Gittins Policy is Nearly Optimal in the M/G/k under Extremely General Conditions." *Proceedings of the ACM Measurement and Analysis of Computer Systems -- SIGMETRICS:* Volume 4, Number 3, Article 43 (Dec 2020), pp. 43:1 -- 43:29, June 2021, Beijing, China. DOI 10.1145/3428328. INFORMS 2022 George Nicholson Prize Winner. pdf

Ziv Scully, Mor Harchol-Balter, Alan Scheller-Wolf. ``Simple Near-Optimal Scheduling for the M/G/1." *Proceedings of the ACM Measurement and Analysis of Computer Systems -- SIGMETRICS:* Volume 4, Issue 1, Article 11 (March 2020), pp. 11:1 -- 11:29, June 2020, Boston, MA. pdf .