Optimizing Tail Latency in the Light-Tailed M/G/k

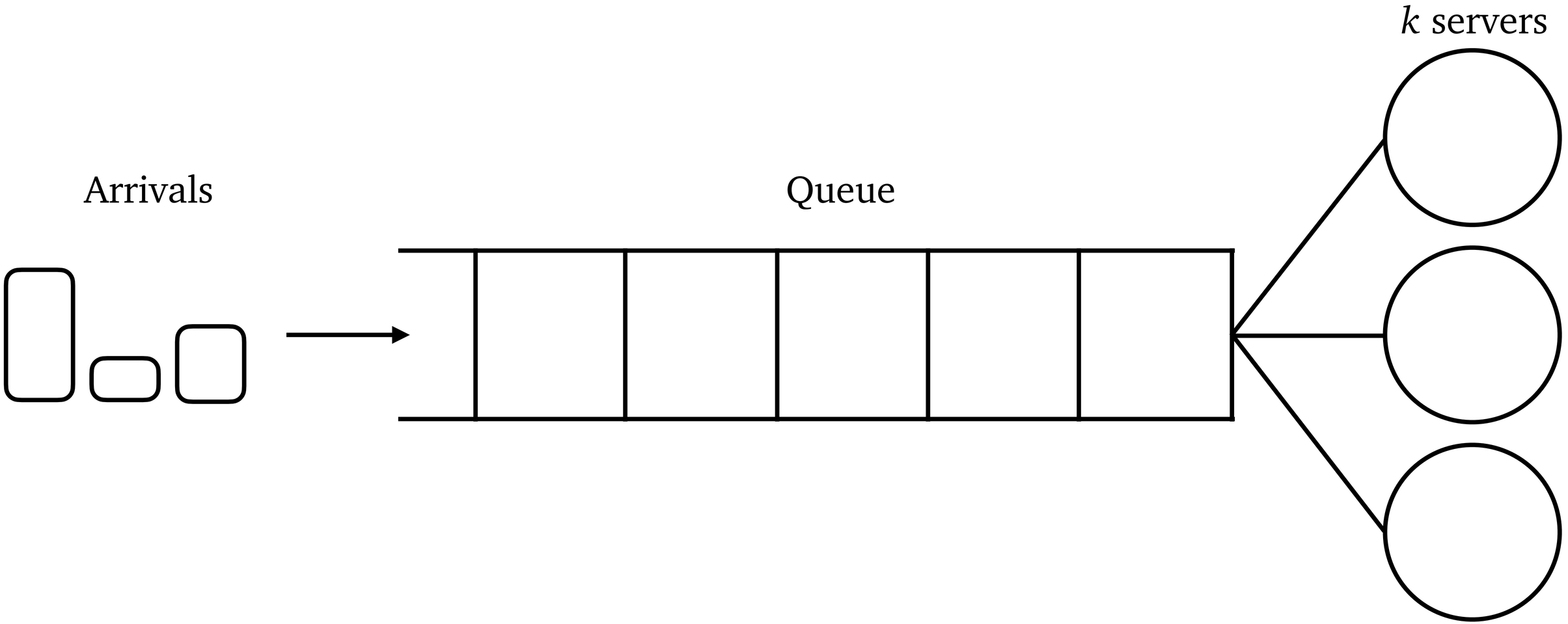
George Yu
Cornell ORIE

Amit Harlev
Cornell CAM

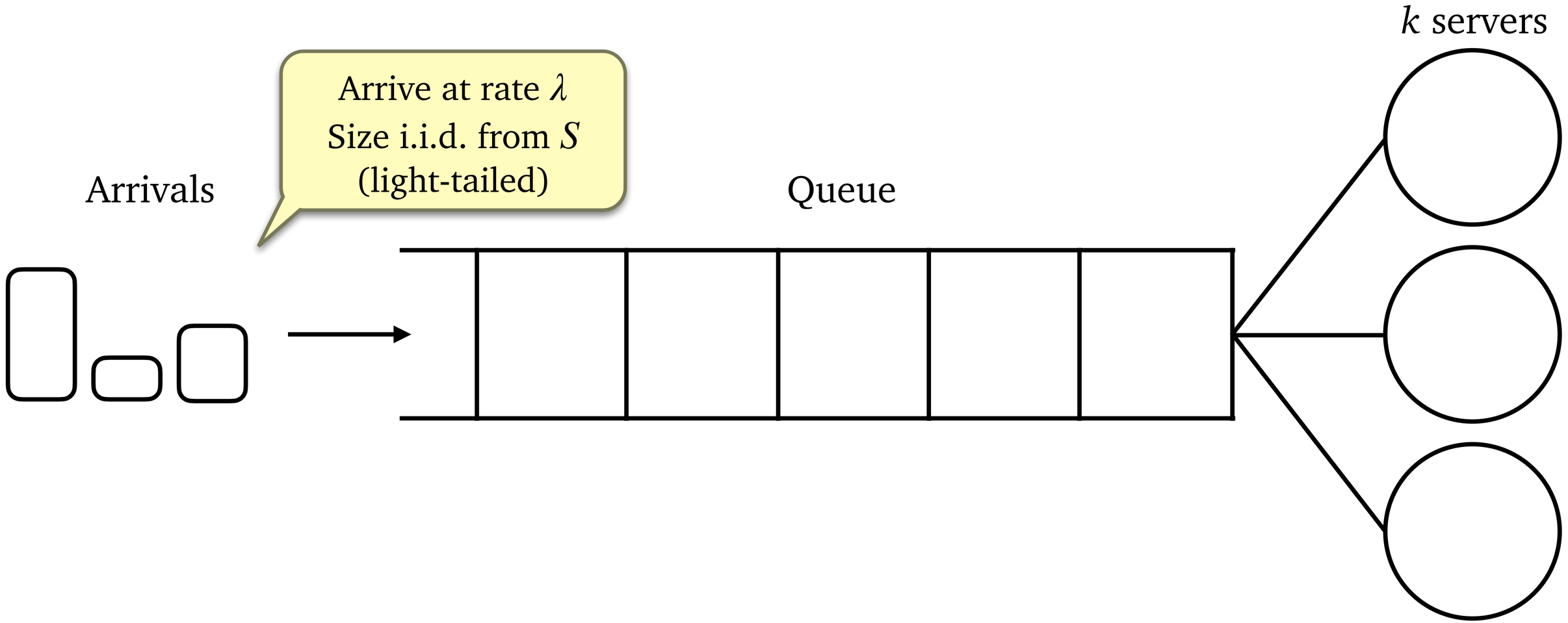
Reevu Adakroy
Cornell University

Ziv Scully
Cornell ORIE

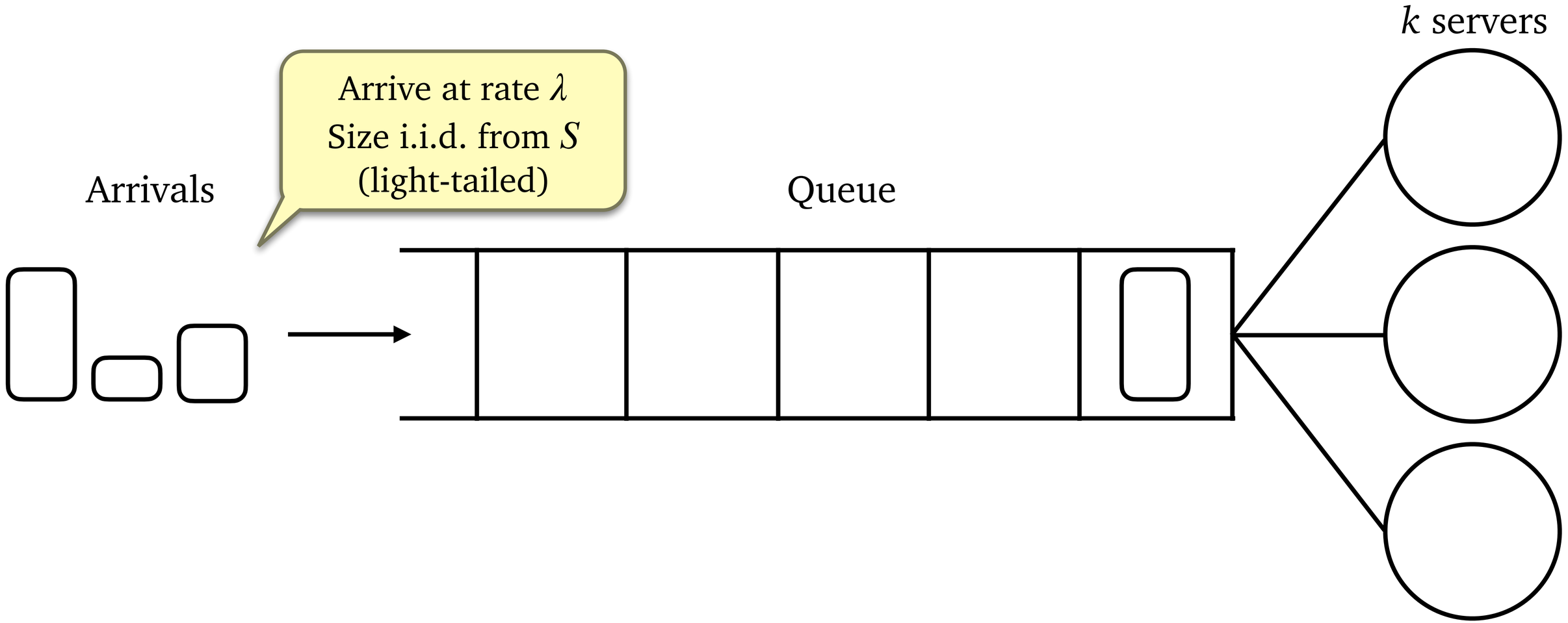
The light-tailed M/G/k



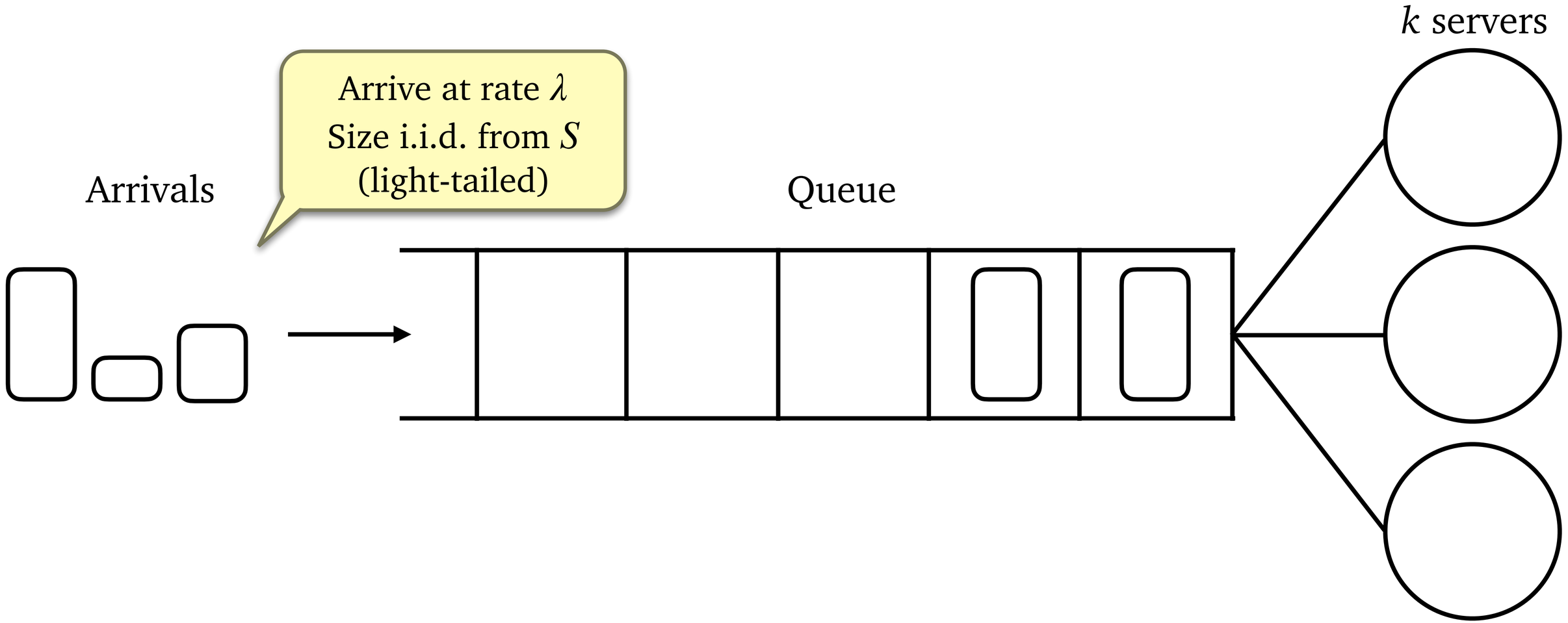
The light-tailed M/G/k



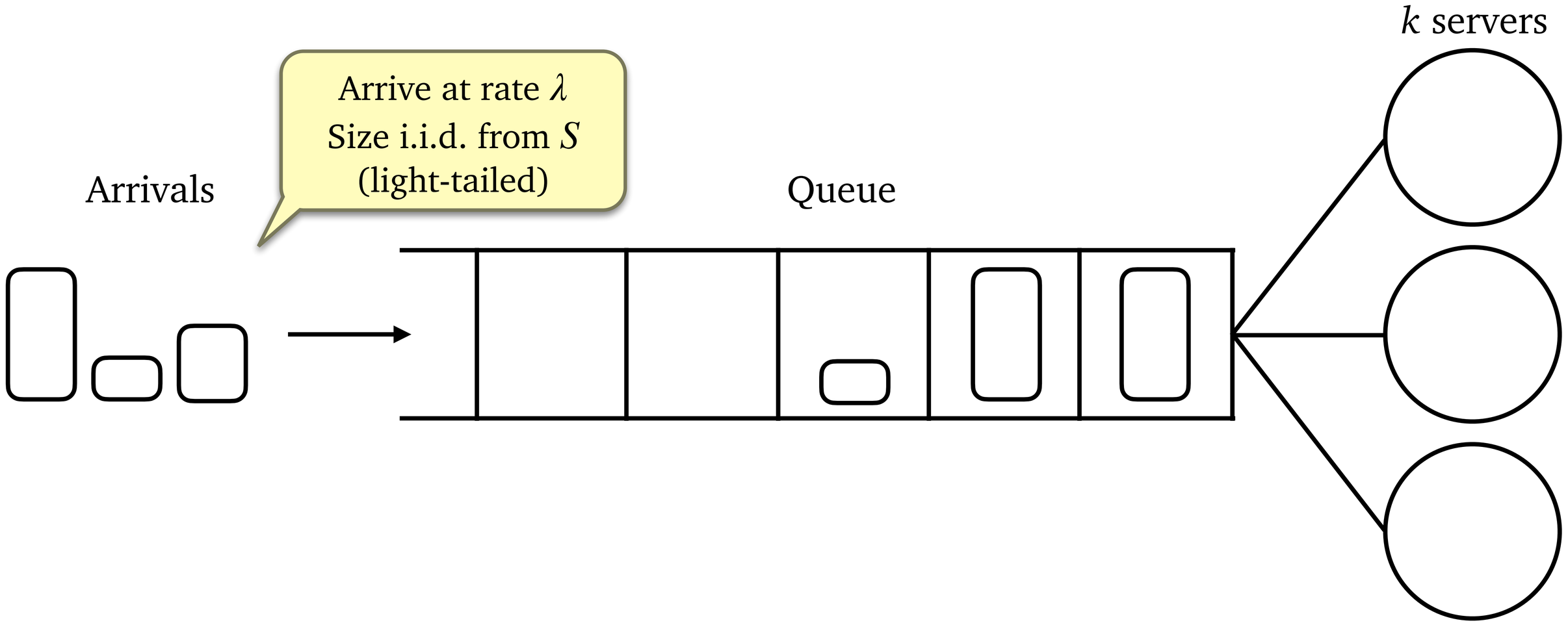
The light-tailed M/G/k



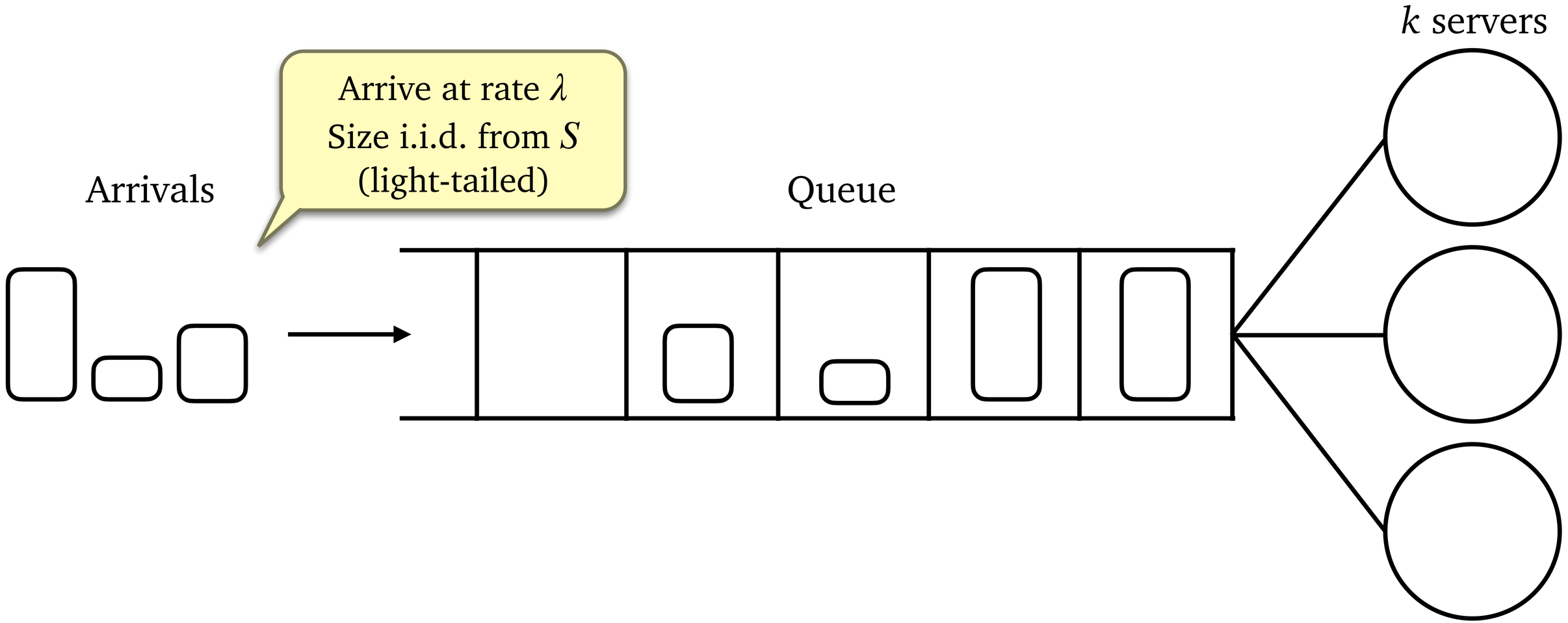
The light-tailed M/G/k



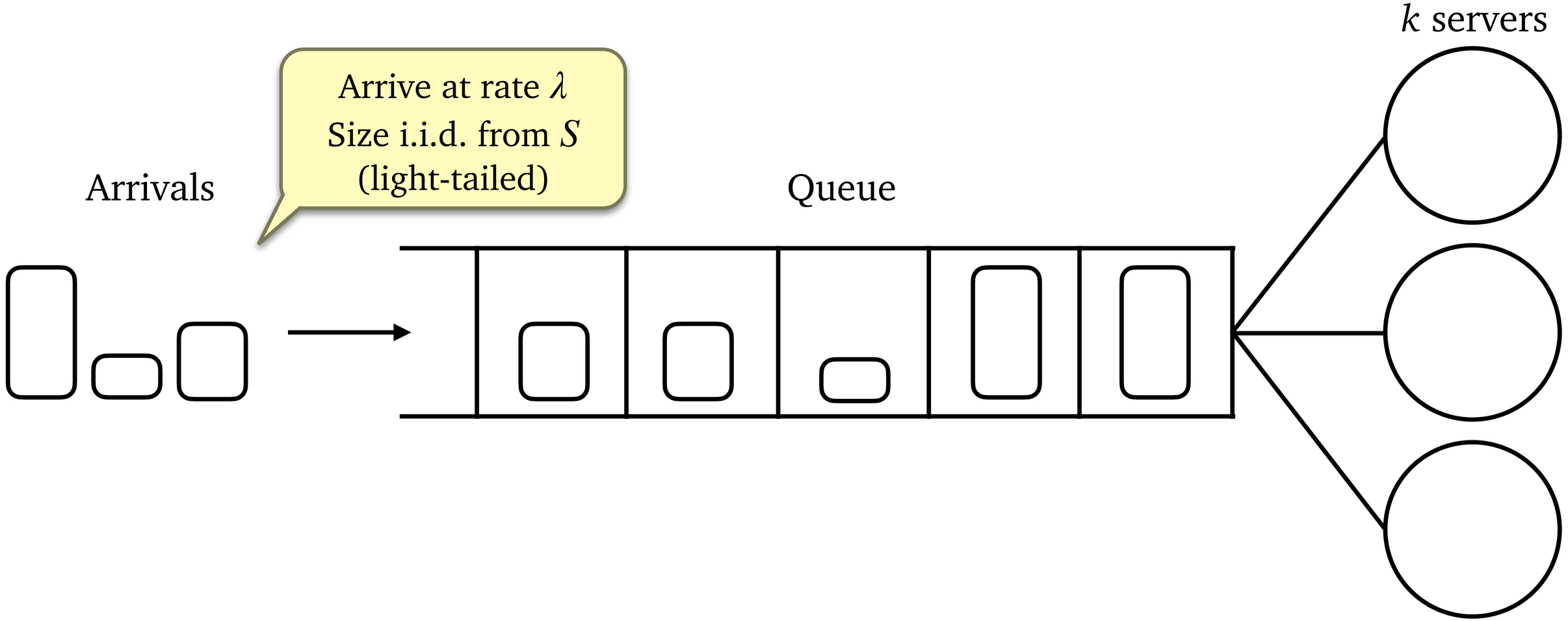
The light-tailed M/G/k



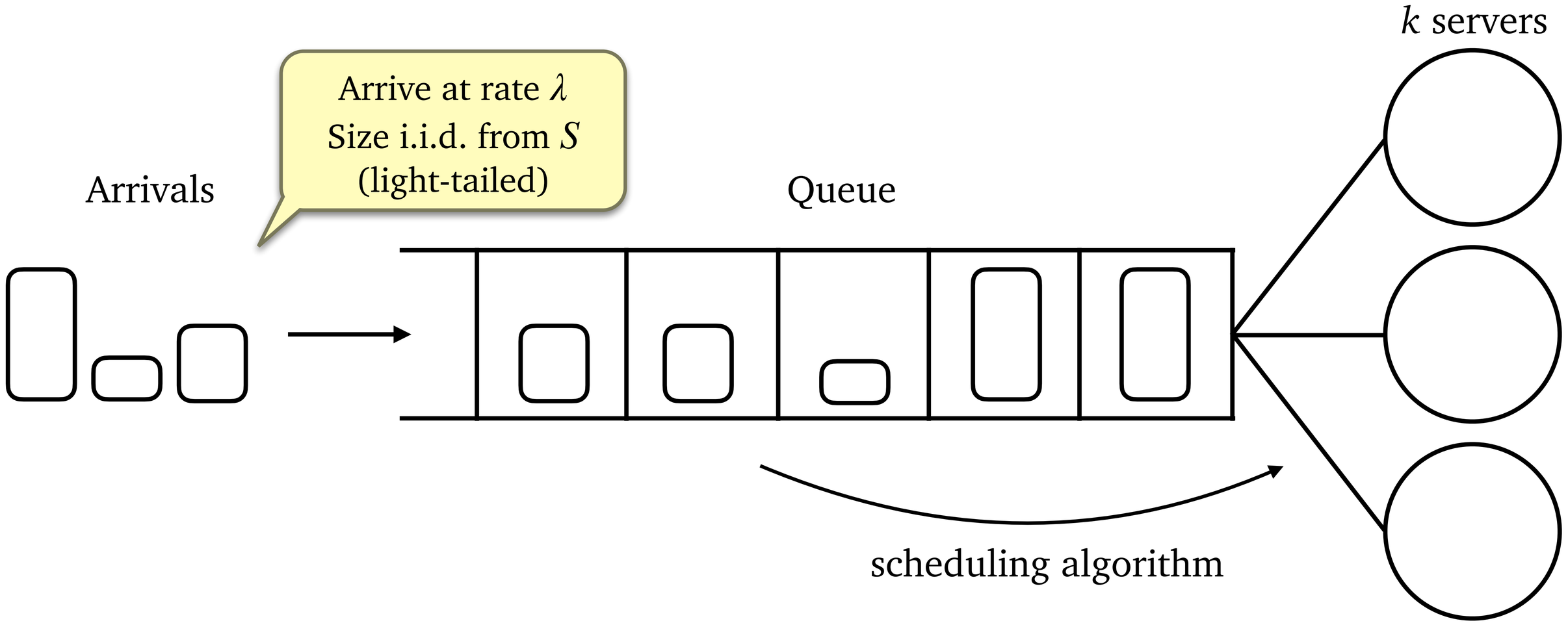
The light-tailed M/G/k



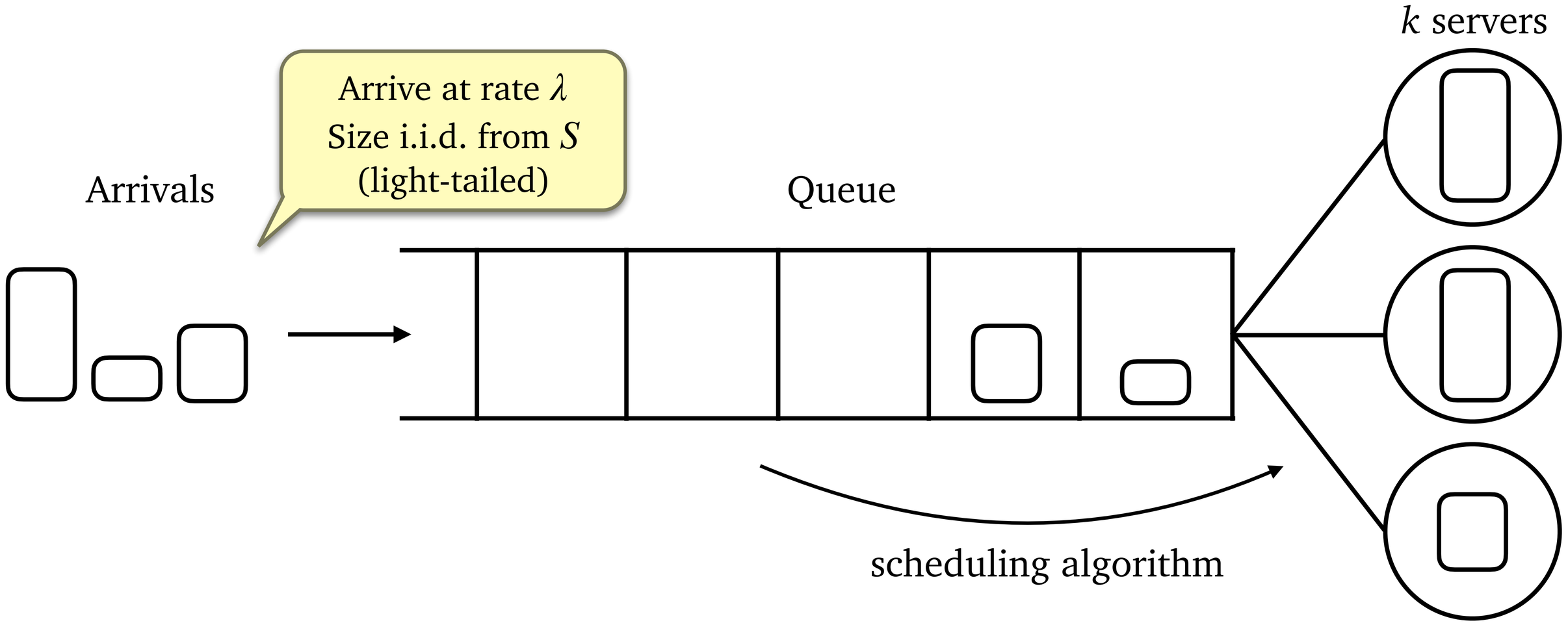
The light-tailed M/G/k



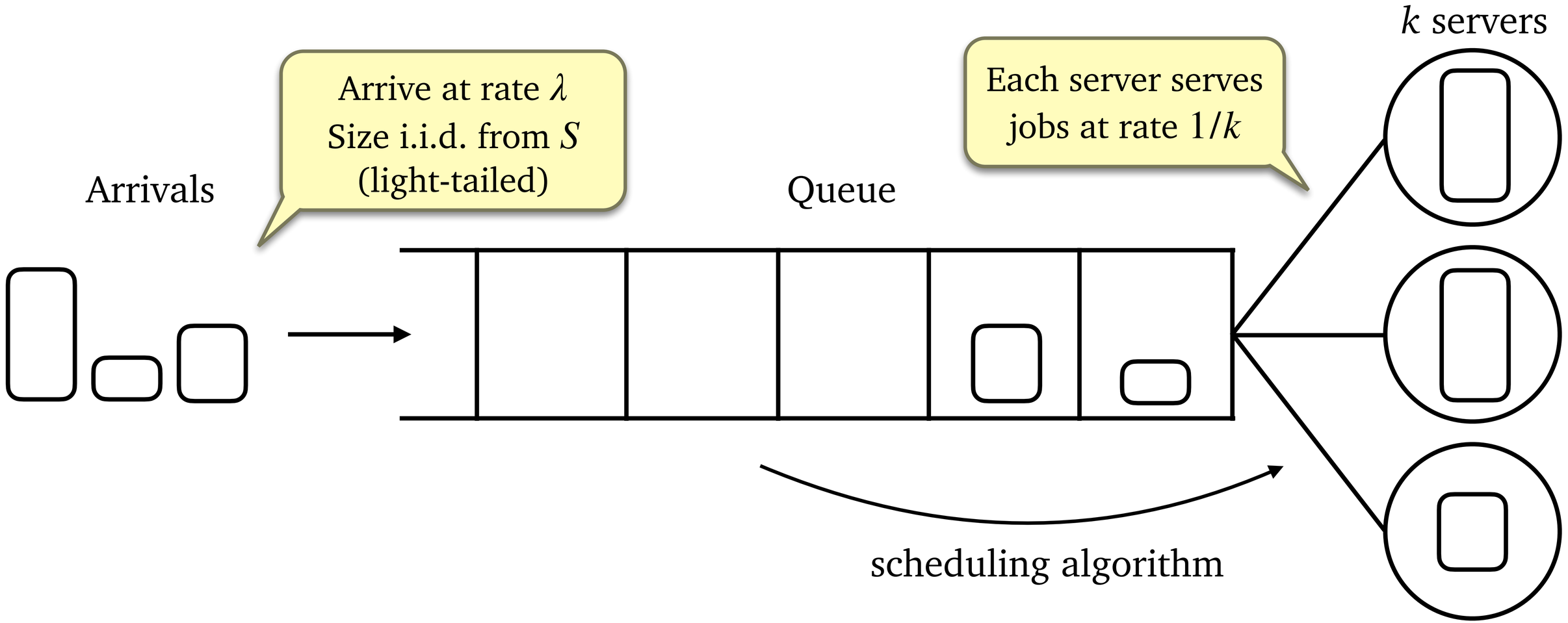
The light-tailed M/G/k



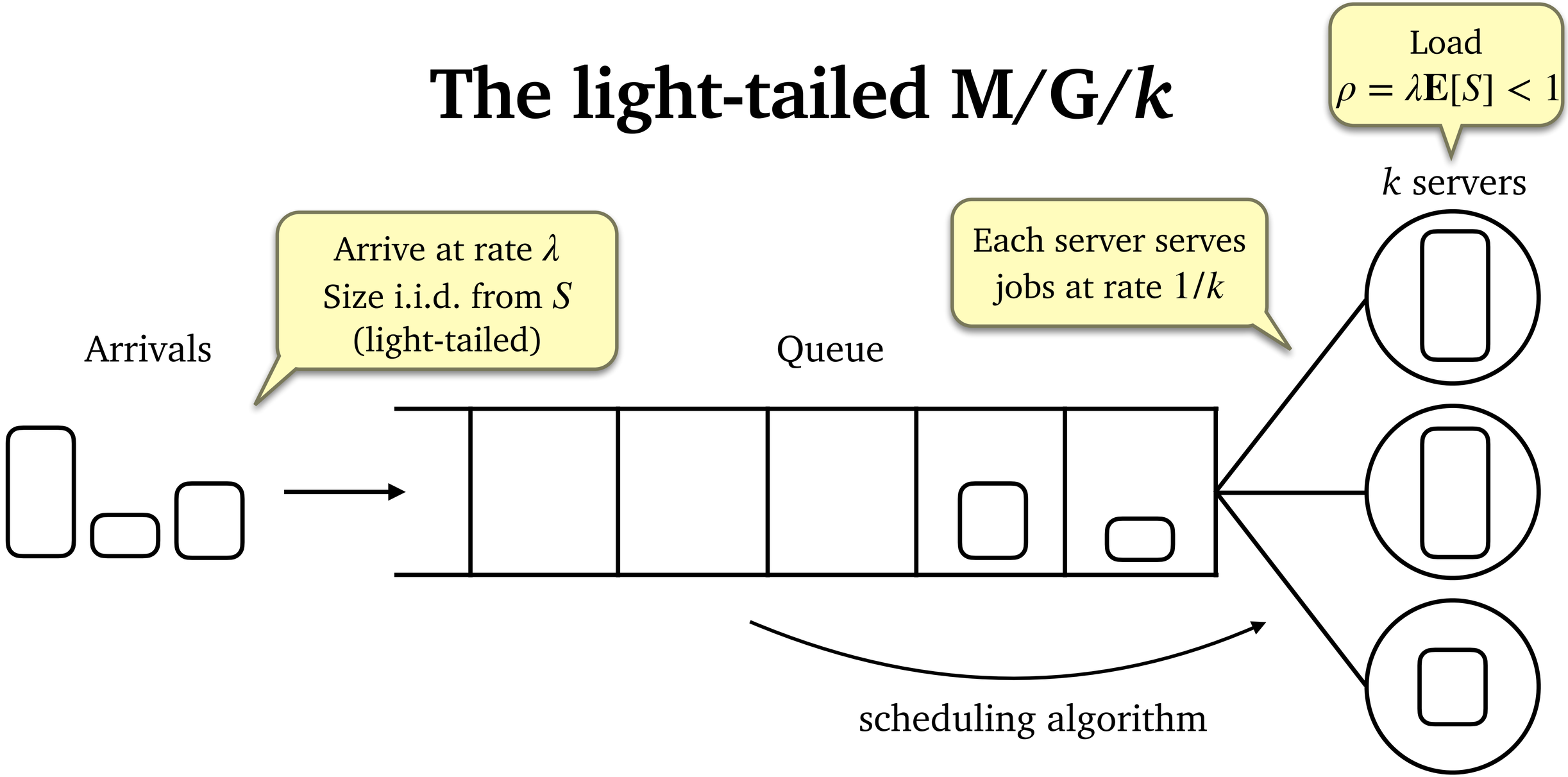
The light-tailed M/G/k



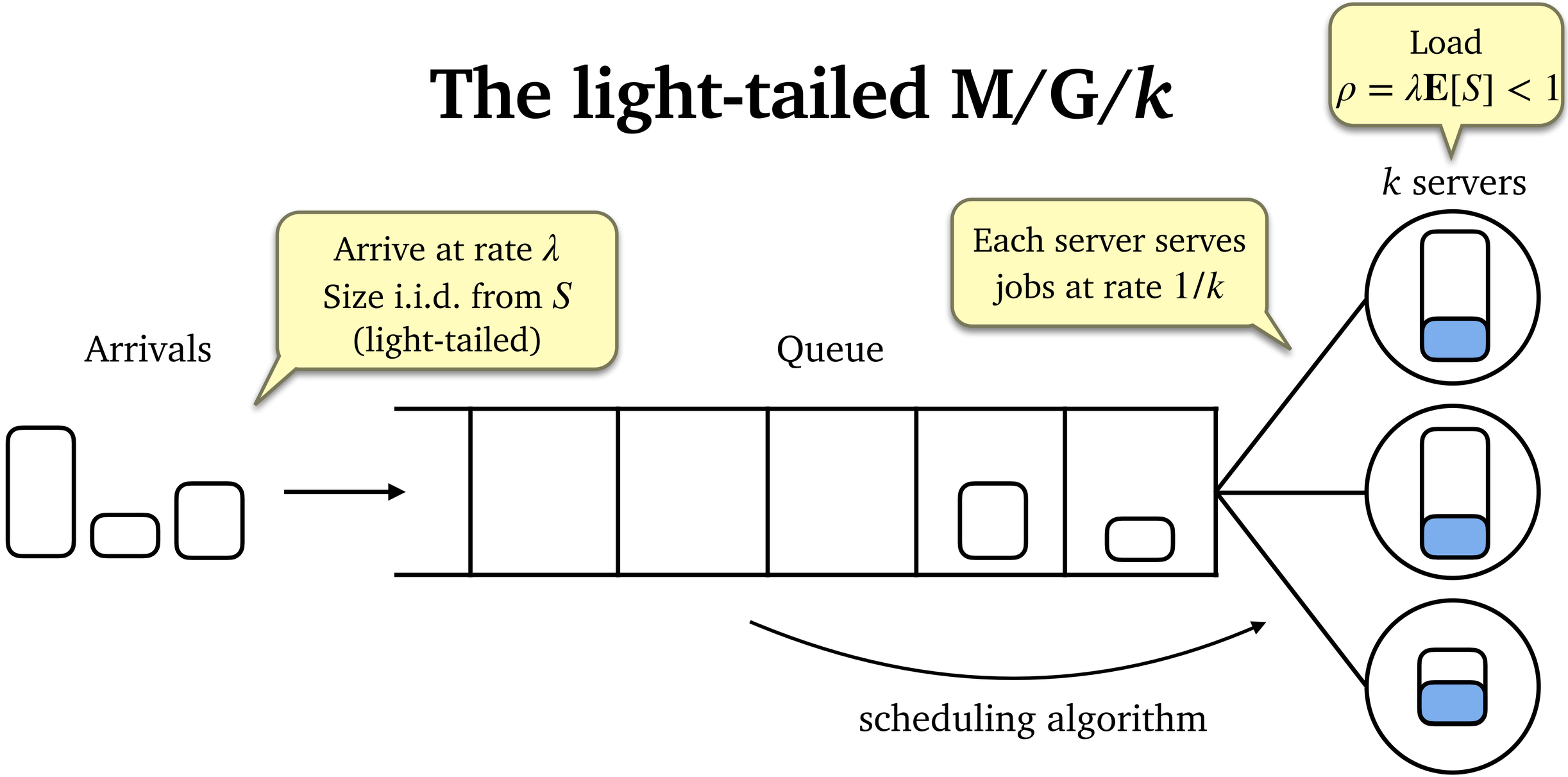
The light-tailed M/G/k



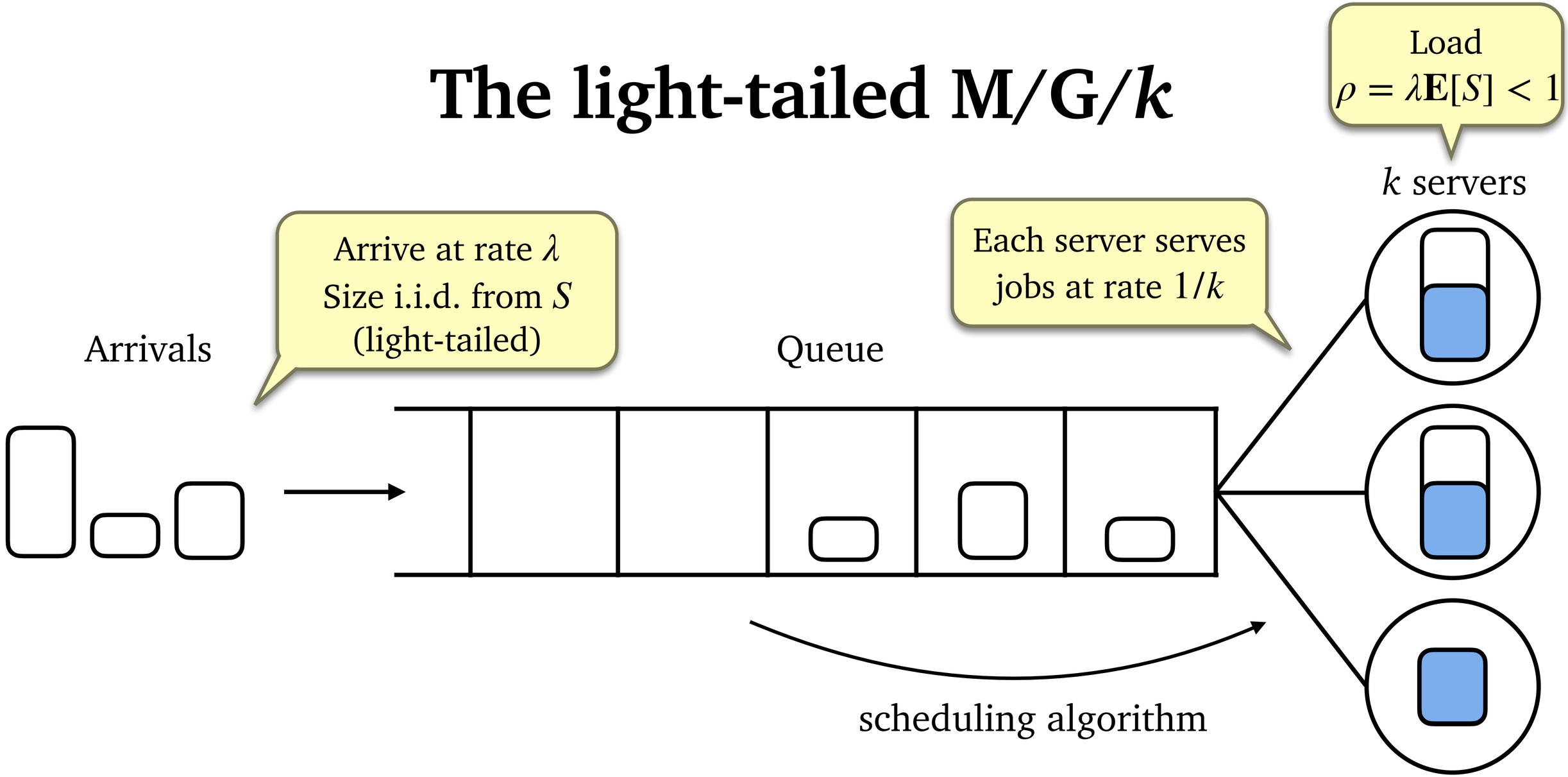
The light-tailed M/G/k



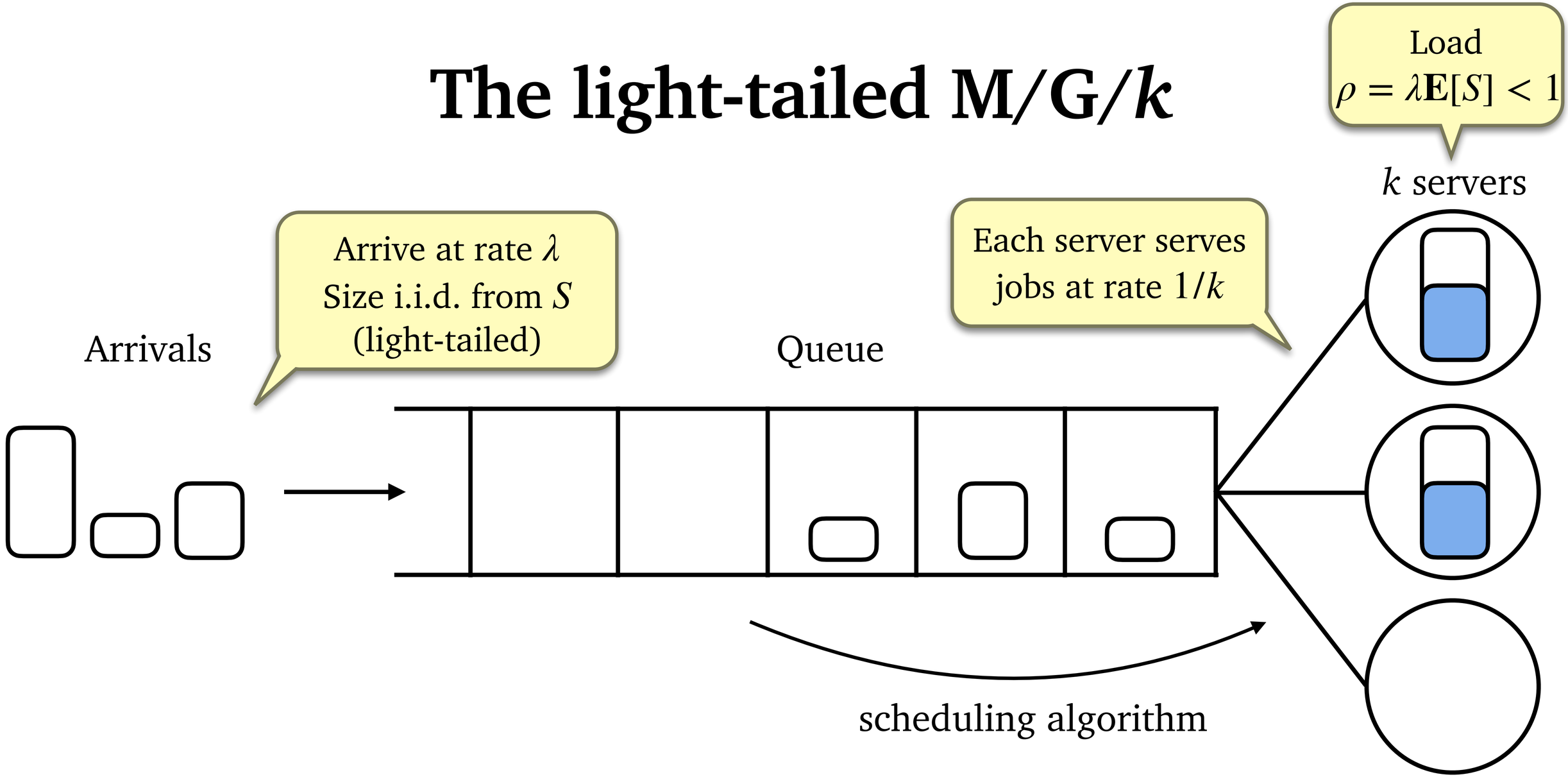
The light-tailed M/G/k



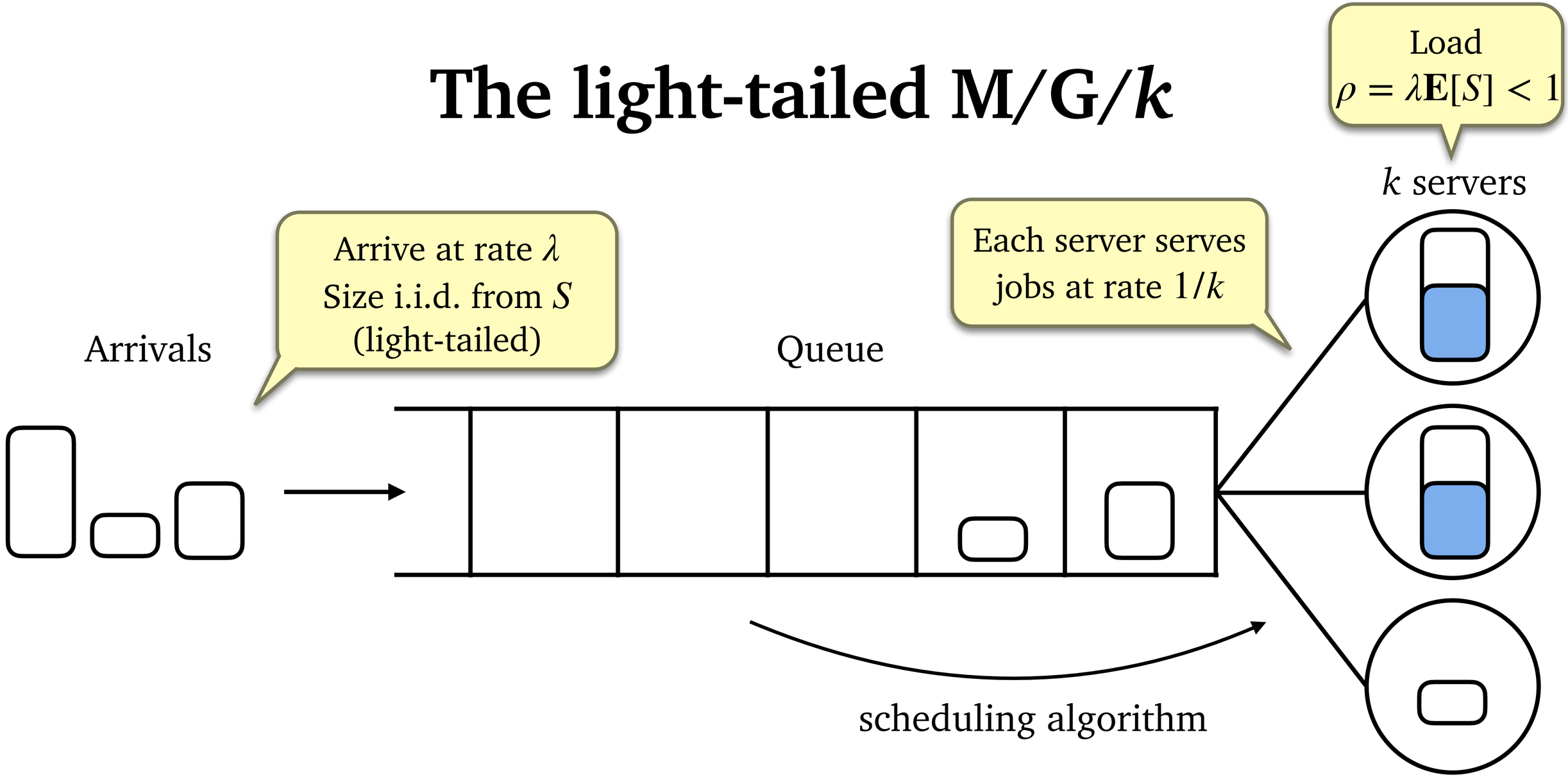
The light-tailed M/G/k



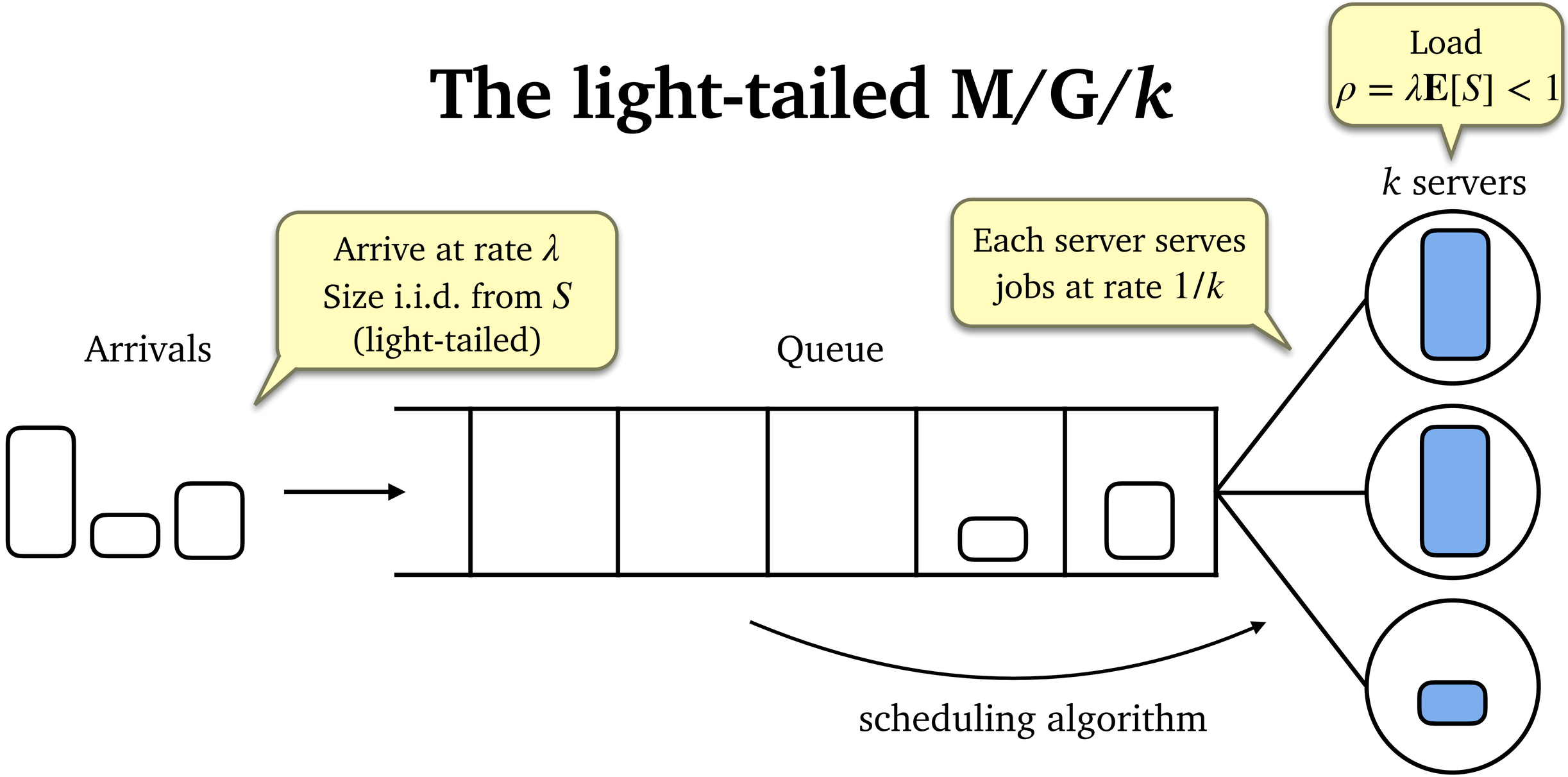
The light-tailed M/G/k



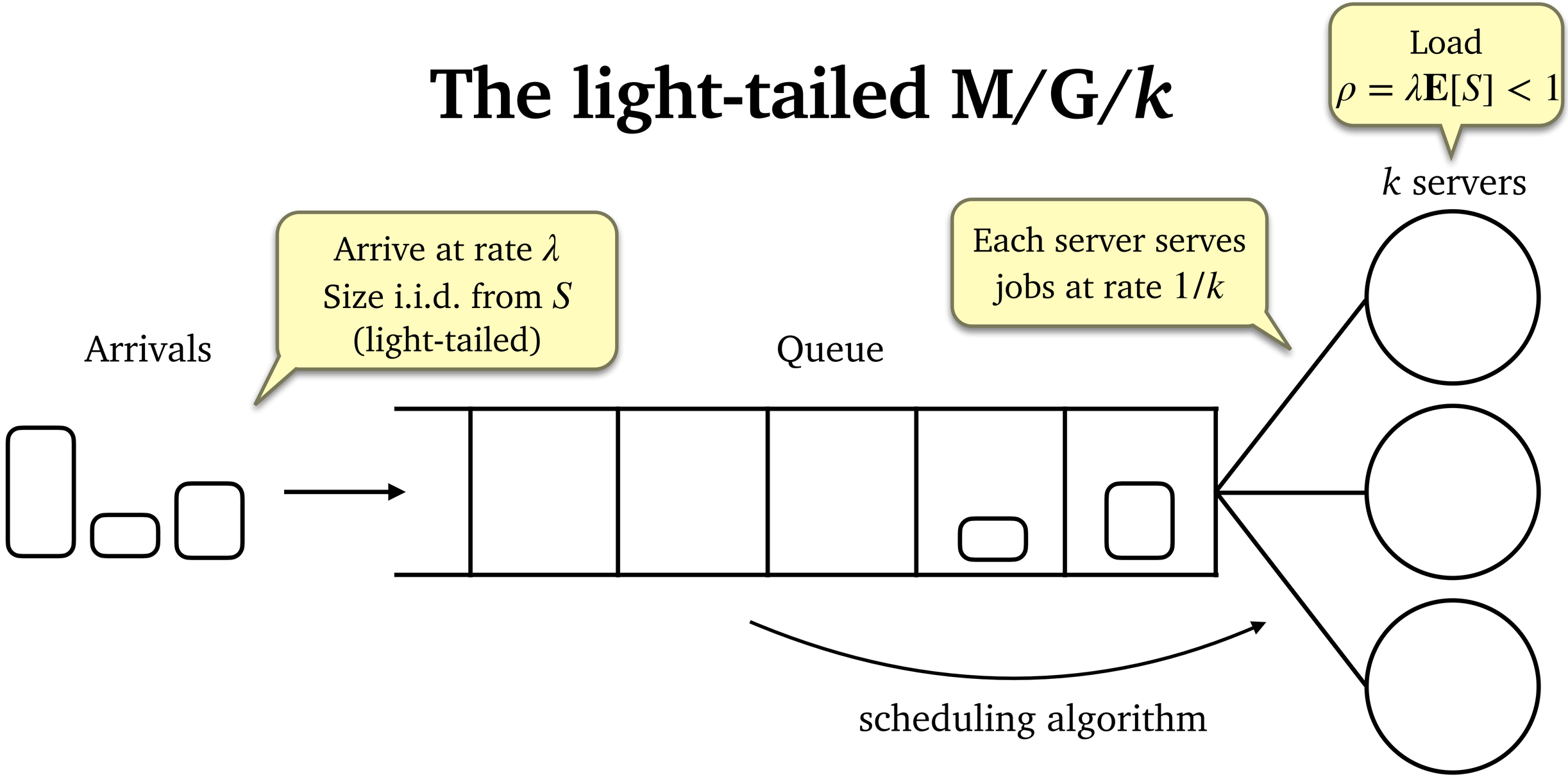
The light-tailed M/G/k



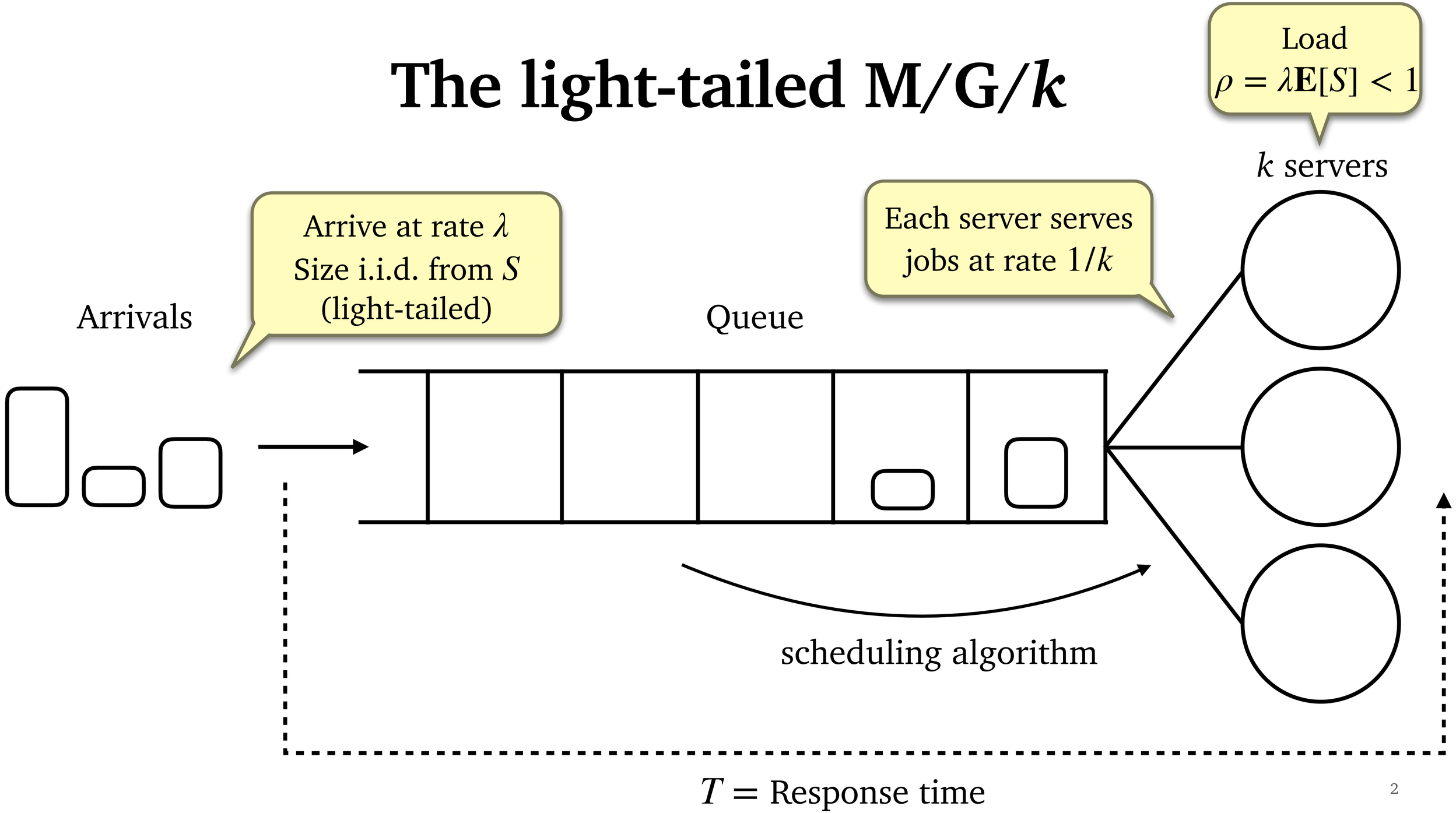
The light-tailed M/G/k



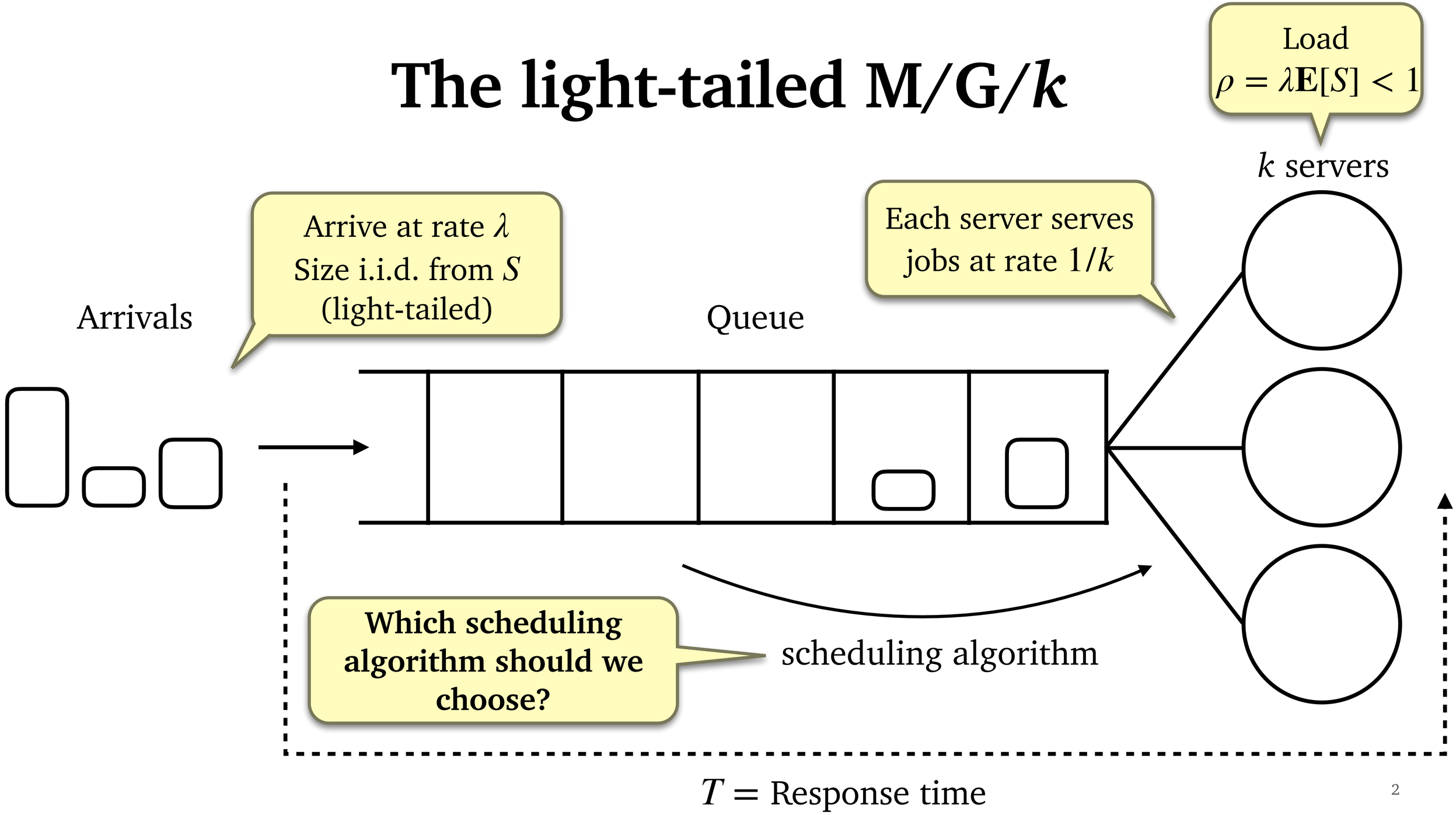
The light-tailed M/G/k



The light-tailed M/G/k



The light-tailed M/G/k



Which scheduling algorithm should we choose?

Which scheduling algorithm should we choose?

Objective \ Setting			
$\min \mathbf{E}[T]$			

Which scheduling algorithm should we choose?

Setting / Objective	single-server ($k = 1$)		
$\min \mathbf{E}[T]$			

Which scheduling algorithm should we choose?

Objective \ Setting	single-server ($k = 1$)		
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)		

Which scheduling algorithm should we choose?

Objective \ Setting	single-server ($k = 1$)		
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)		
$\min \mathbf{P}[T > t]$ “for all large t”			

In practice

Which scheduling algorithm should we choose?

Objective \ Setting	single-server ($k = 1$)		
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)		
$\min \mathbf{P}[T > t]$ "for all large t"	γ-Boost		

In practice

[Yu & Scully 2024]

Which scheduling algorithm should we choose?

Objective \ Setting	single-server ($k = 1$)	multi-server (light traffic, $\rho \ll 1$)		
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)			
$\min \mathbf{P}[T > t]$ "for all large t"	γ-Boost			

In practice

[Yu & Scully 2024]

Which scheduling algorithm should we choose?

Hard!

Objective \ Setting	single-server ($k = 1$)	multi-server (light traffic, $\rho \ll 1$)		
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)			
$\min \mathbf{P}[T > t]$ "for all large t"	γ-Boost [Yu & Scully 2024]			

In practice

Which scheduling algorithm should we choose?

Hard!

Objective \ Setting	single-server ($k = 1$)	multi-server (light traffic, $\rho \ll 1$)	multi-server (heavy traffic, $\rho \to 1$)
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)		
$\min \mathbf{P}[T > t]$ "for all large t"	γ-Boost [Yu & Scully 2024]		

In practice

Which scheduling algorithm should we choose?

Hope ideas carry over

Hard!

Setting / Objective	single-server ($k = 1$)	multi-server (light traffic, $\rho \ll 1$)	multi-server (heavy traffic, $\rho \to 1$)
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)		
$\min \mathbf{P}[T > t]$ "for all large t"	γ-Boost [Yu & Scully 2024]		

In practice

Which scheduling algorithm should we choose?

Hard!

Hope ideas carry over

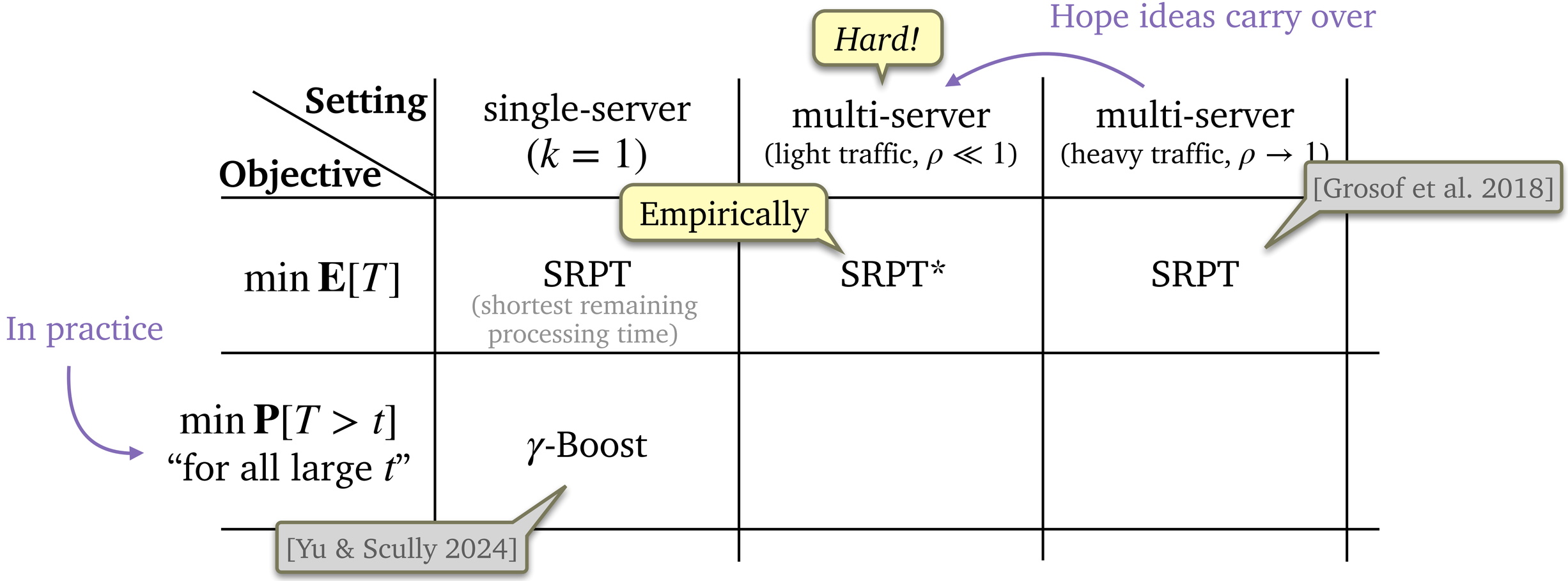
Objective \ Setting	single-server ($k = 1$)	multi-server (light traffic, $\rho \ll 1$)	multi-server (heavy traffic, $\rho \to 1$)
$\min \mathbf{E}[T]$	SRPT (shortest remaining processing time)		SRPT
$\min \mathbf{P}[T > t]$ "for all large t"	γ-Boost		

[Grosof et al. 2018]

In practice

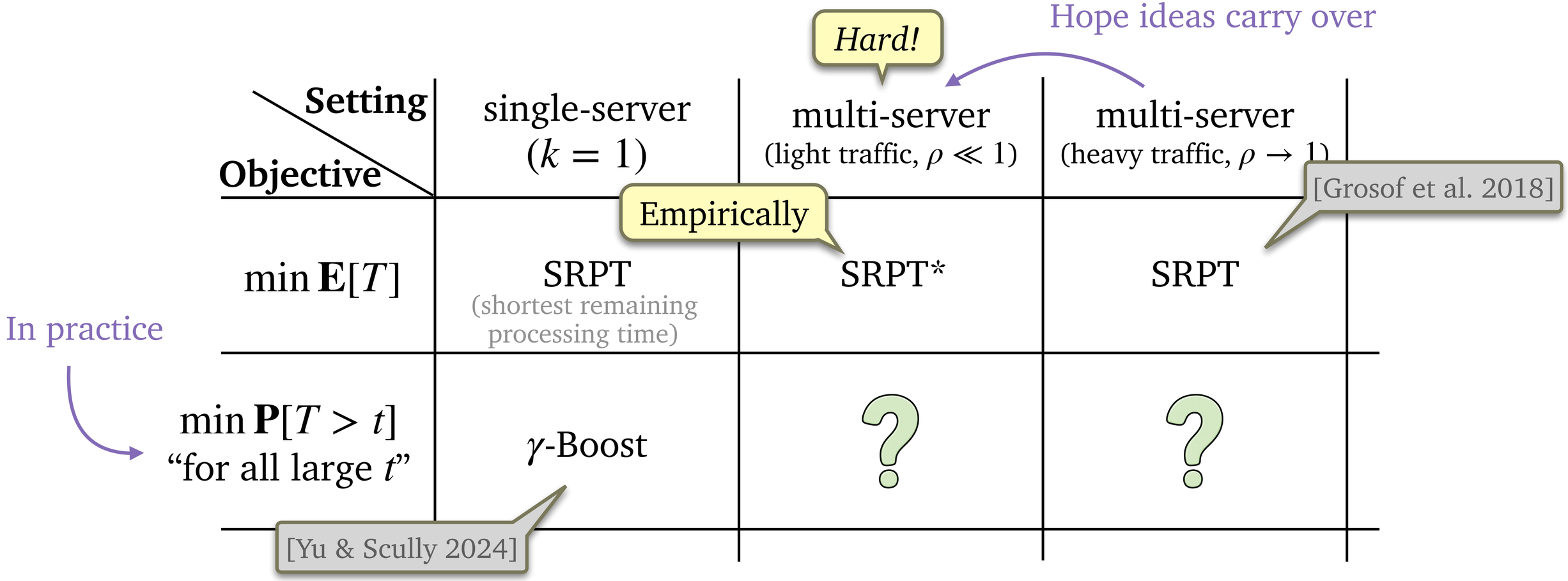
[Yu & Scully 2024]

Which scheduling algorithm should we choose?



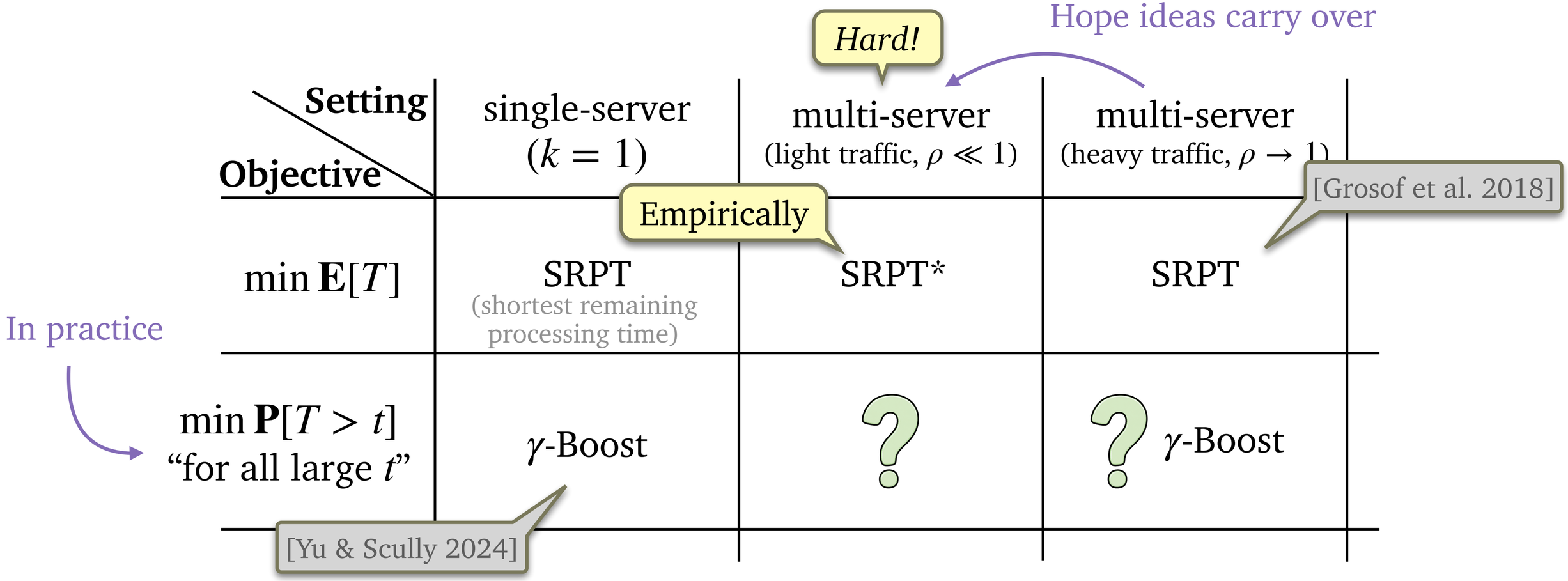
*: You can do a bit better, stay tuned for the next talk!

Which scheduling algorithm should we choose?



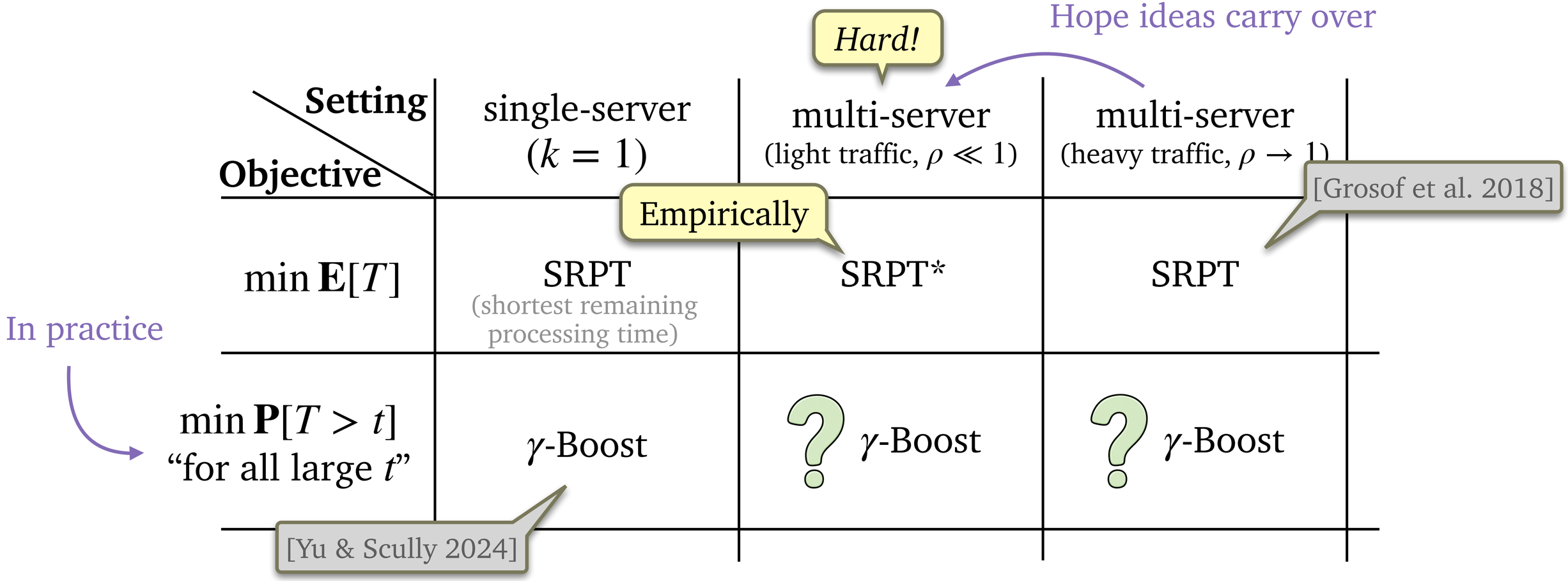
*: You can do a bit better, stay tuned for the next talk!

Which scheduling algorithm should we choose?



*: You can do a bit better, stay tuned for the next talk!

Which scheduling algorithm should we choose?



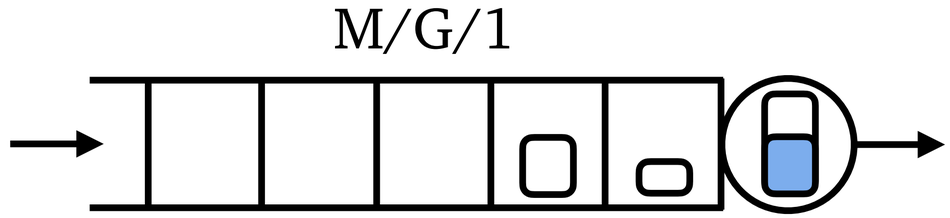
*: You can do a bit better, stay tuned for the next talk!

This talk

This talk

What do we mean by $\min \mathbf{P}[T > t]$ “for all large t”?

What is the γ-Boost policy?

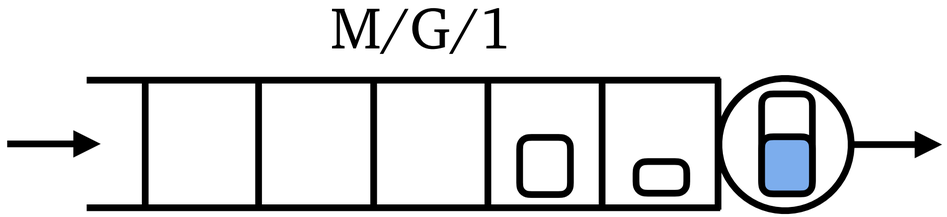


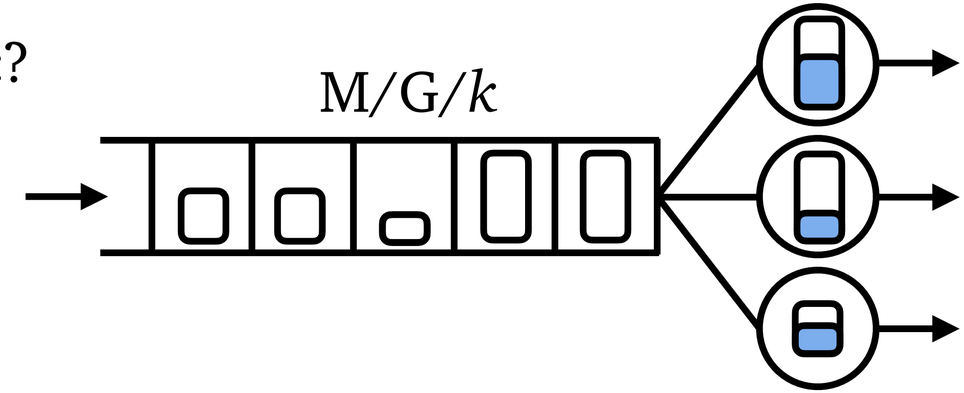
This talk

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?

What is the γ-Boost policy?

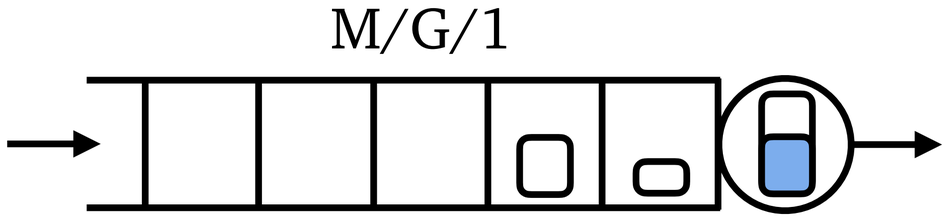
Is γ-Boost tail optimal in the M/G/k in heavy traffic?





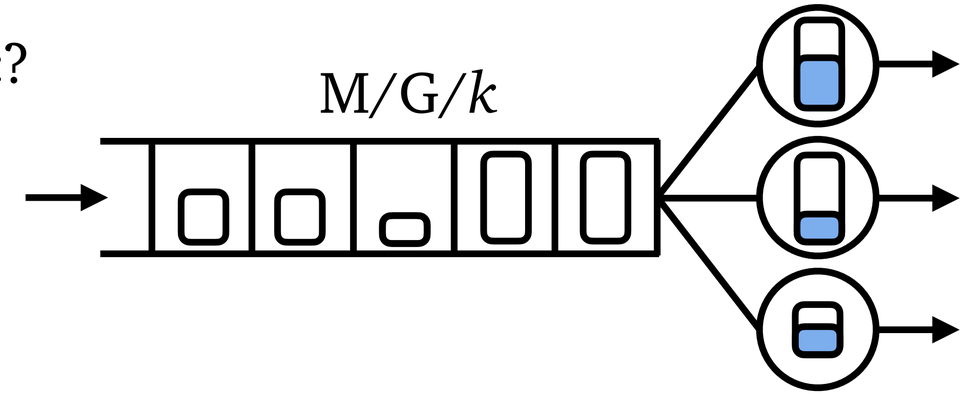
This talk

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?



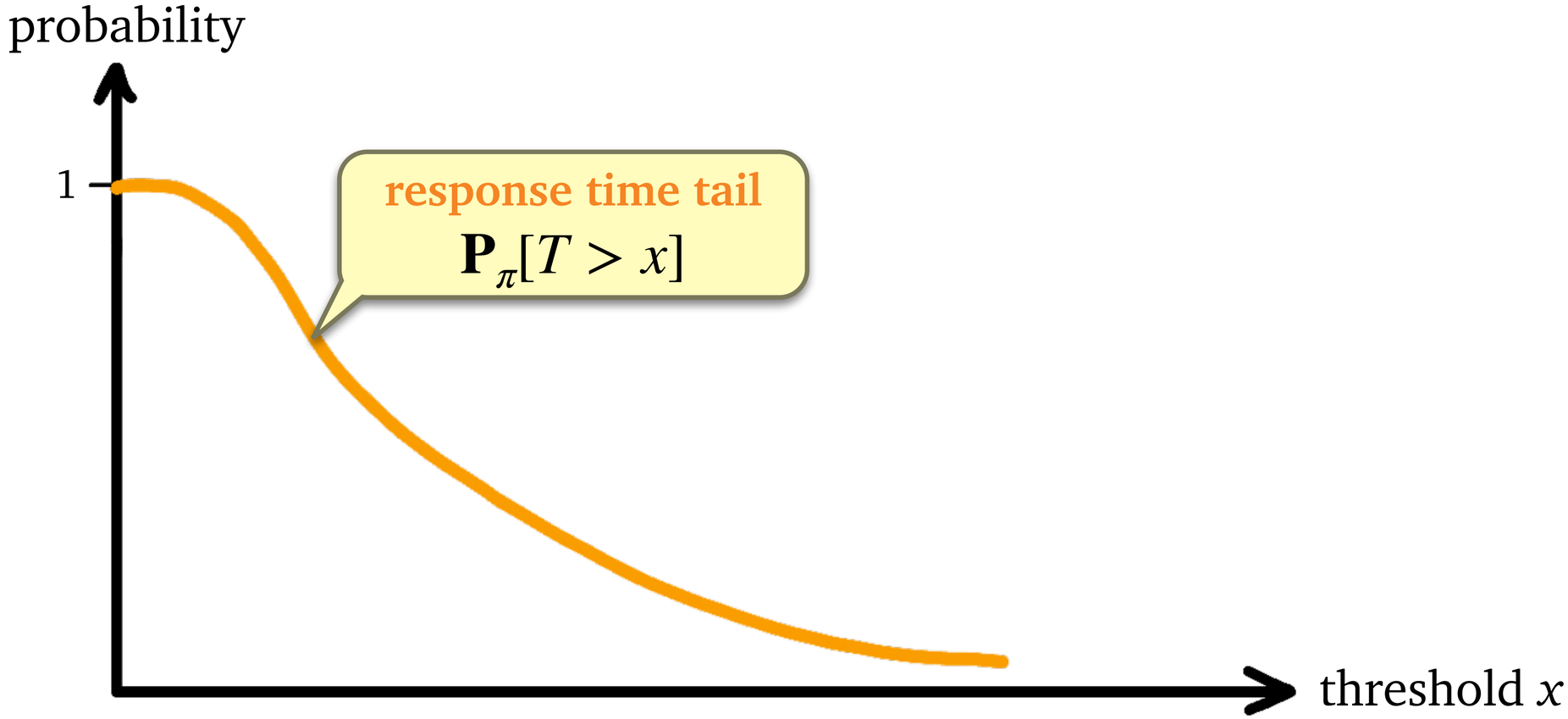
What is the γ-Boost policy?

Is γ-Boost tail optimal in the M/G/k in heavy traffic?

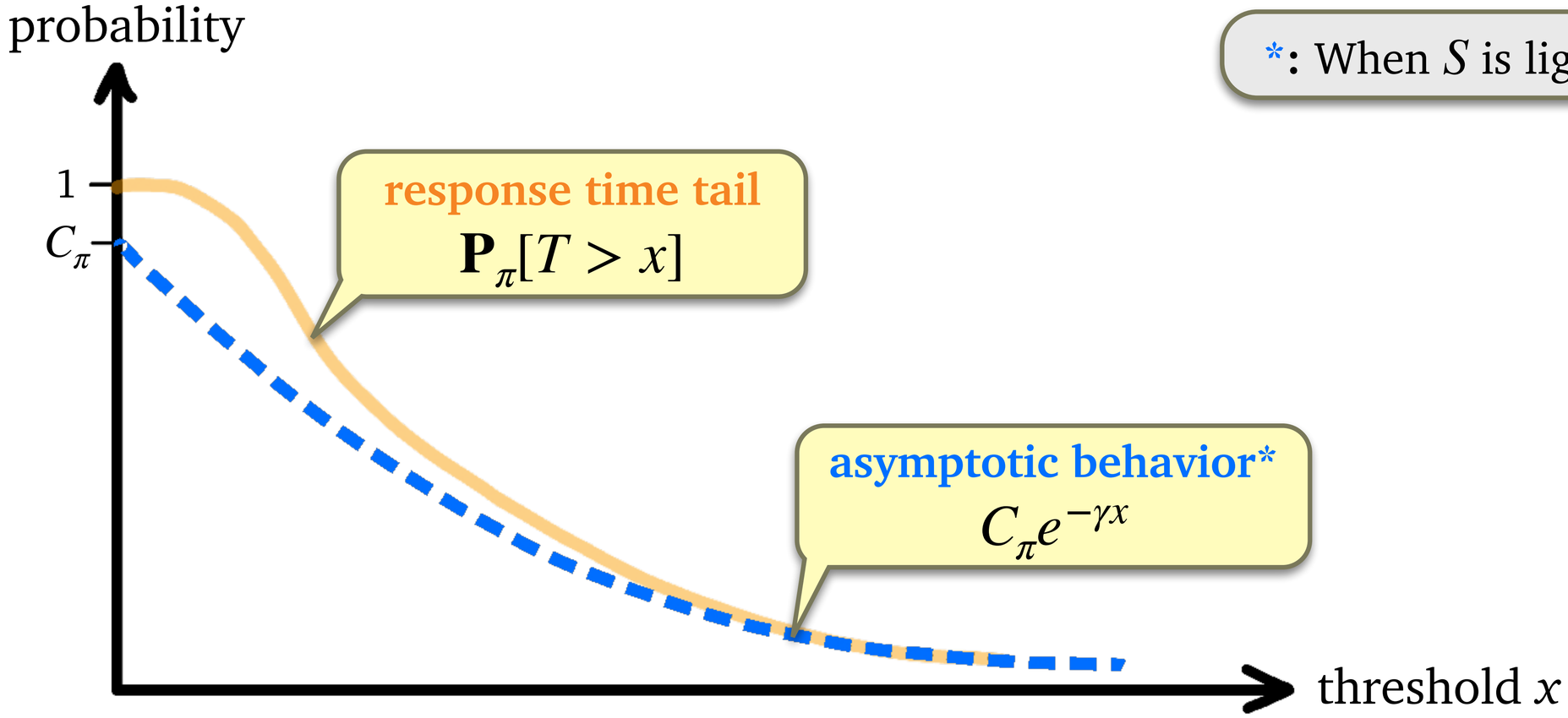


Does γ-Boost have good empirical tail performance in the M/G/k in lighter traffic?

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?



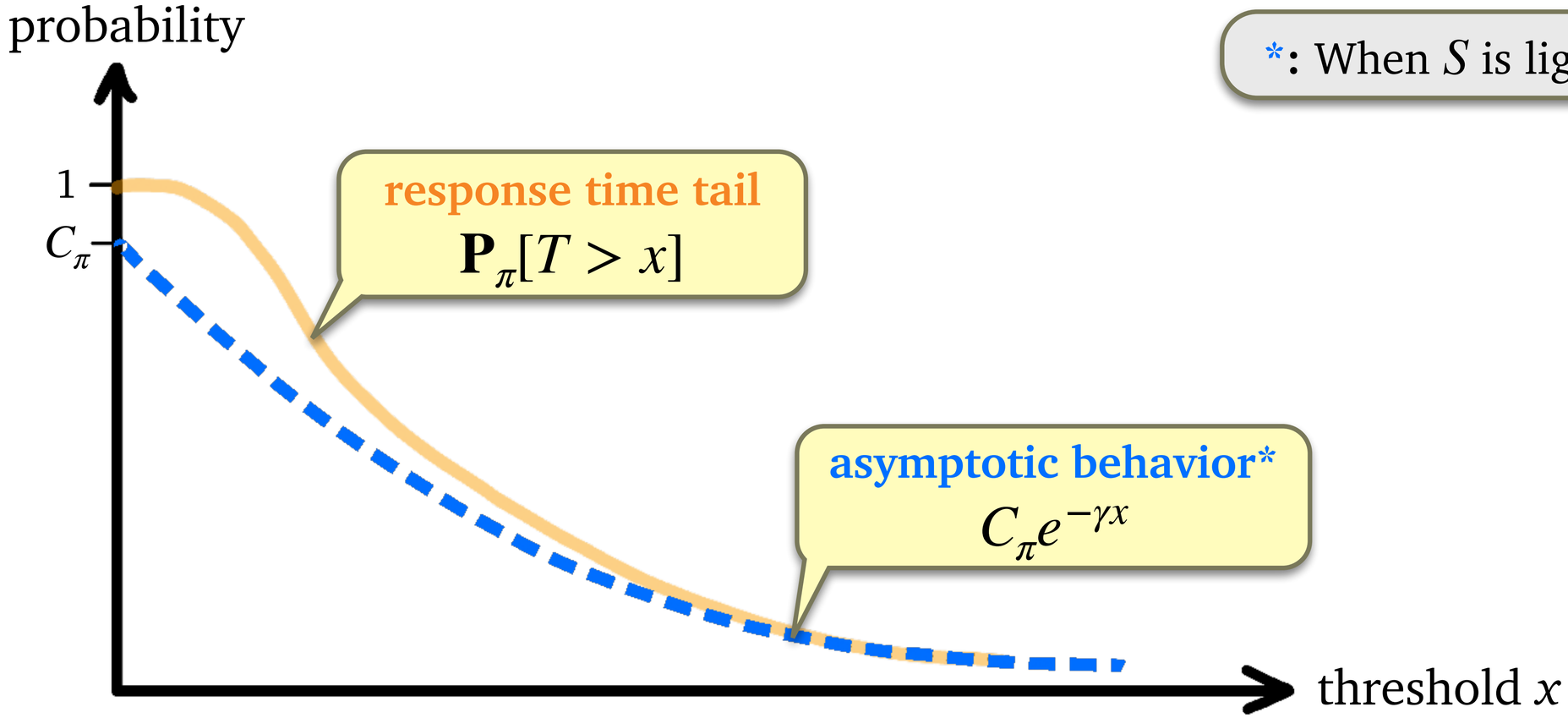
What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?



*: When S is light-tailed.

C_π = *tail constant* of π

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?



*: When S is light-tailed.

C_π = *tail constant* of π

Goal: find π that minimizes C_π

What is the γ-Boost policy?

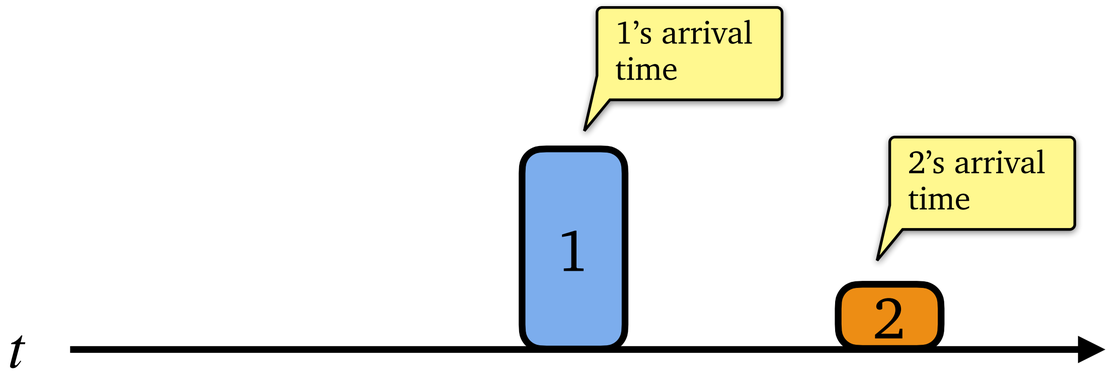
What is the γ-Boost policy?

γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.

$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

What is the γ-Boost policy?

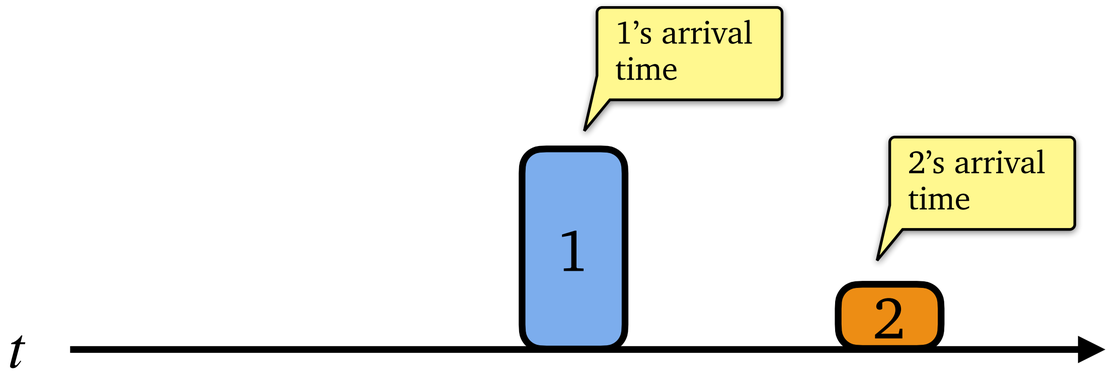
γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.



$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

What is the γ-Boost policy?

γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.

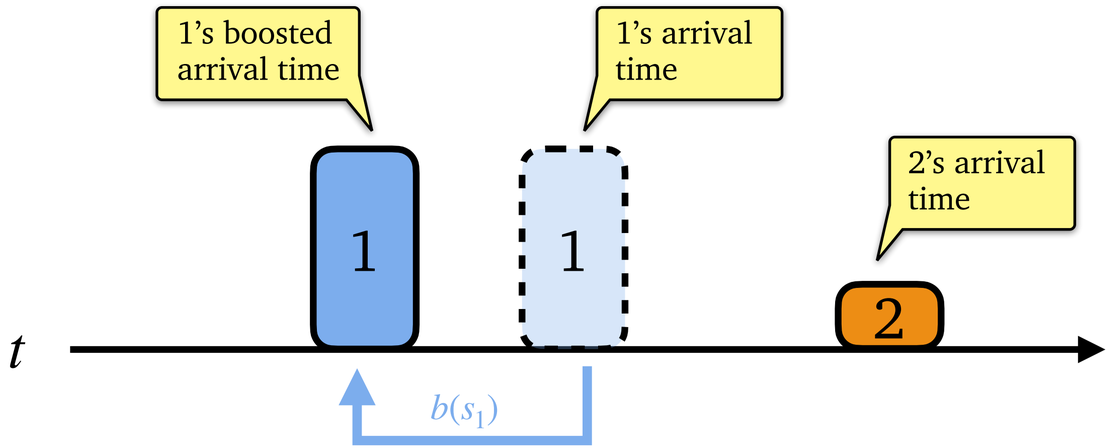


$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

FCFS: serves job 1 then job 2.

What is the γ-Boost policy?

γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.

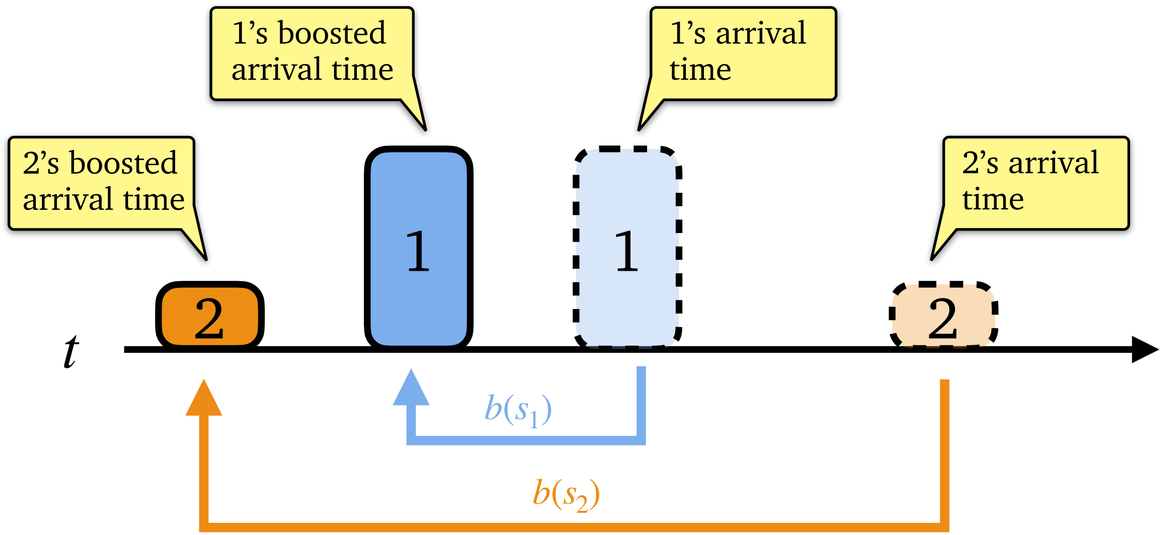


$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

FCFS: serves job 1 then job 2.

What is the γ-Boost policy?

γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.

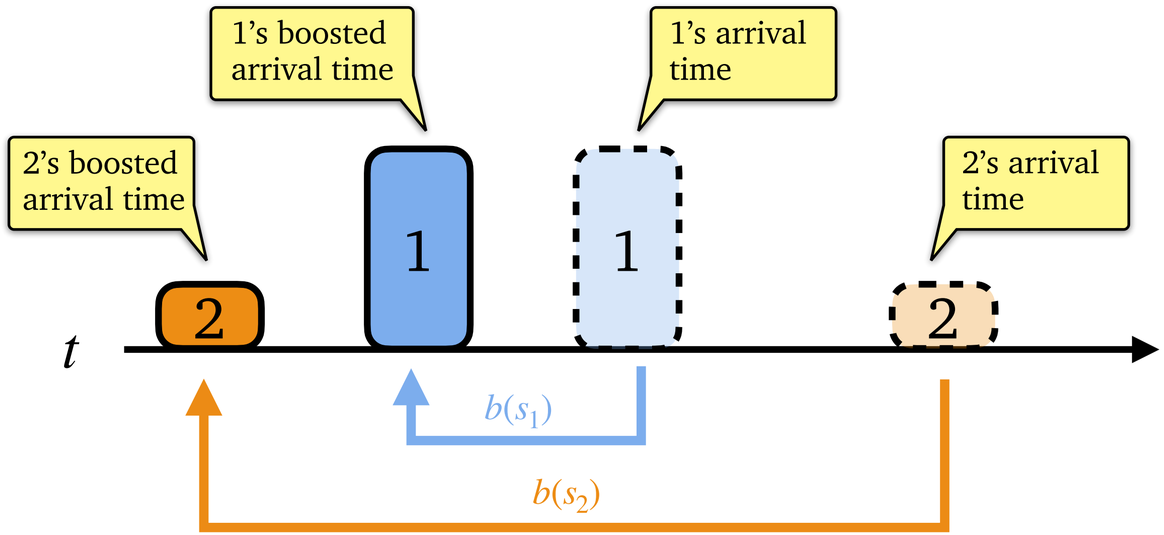


$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

FCFS: serves job 1 then job 2.

What is the γ-Boost policy?

γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.



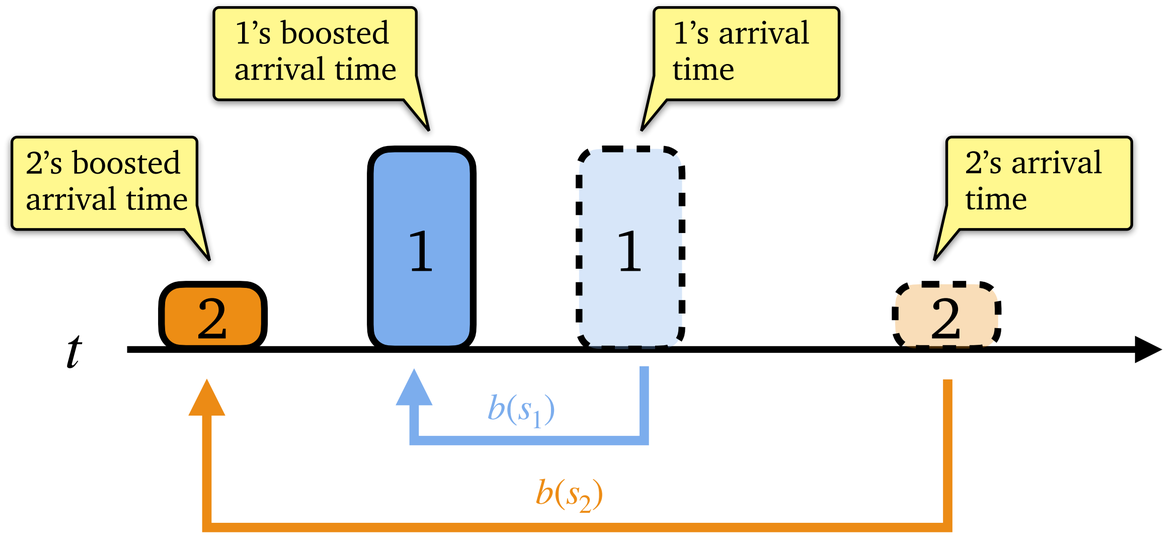
$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

FCFS: serves job 1 then job 2.

γ-Boost: serves job 2 then job 1.

What is the γ-Boost policy?

γ-Boost is *like FCFS*, but gives each job of size s a *boost of* $b(s)$ *to its arrival time*.



$$b(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right)$$

FCFS: serves job 1 then job 2.

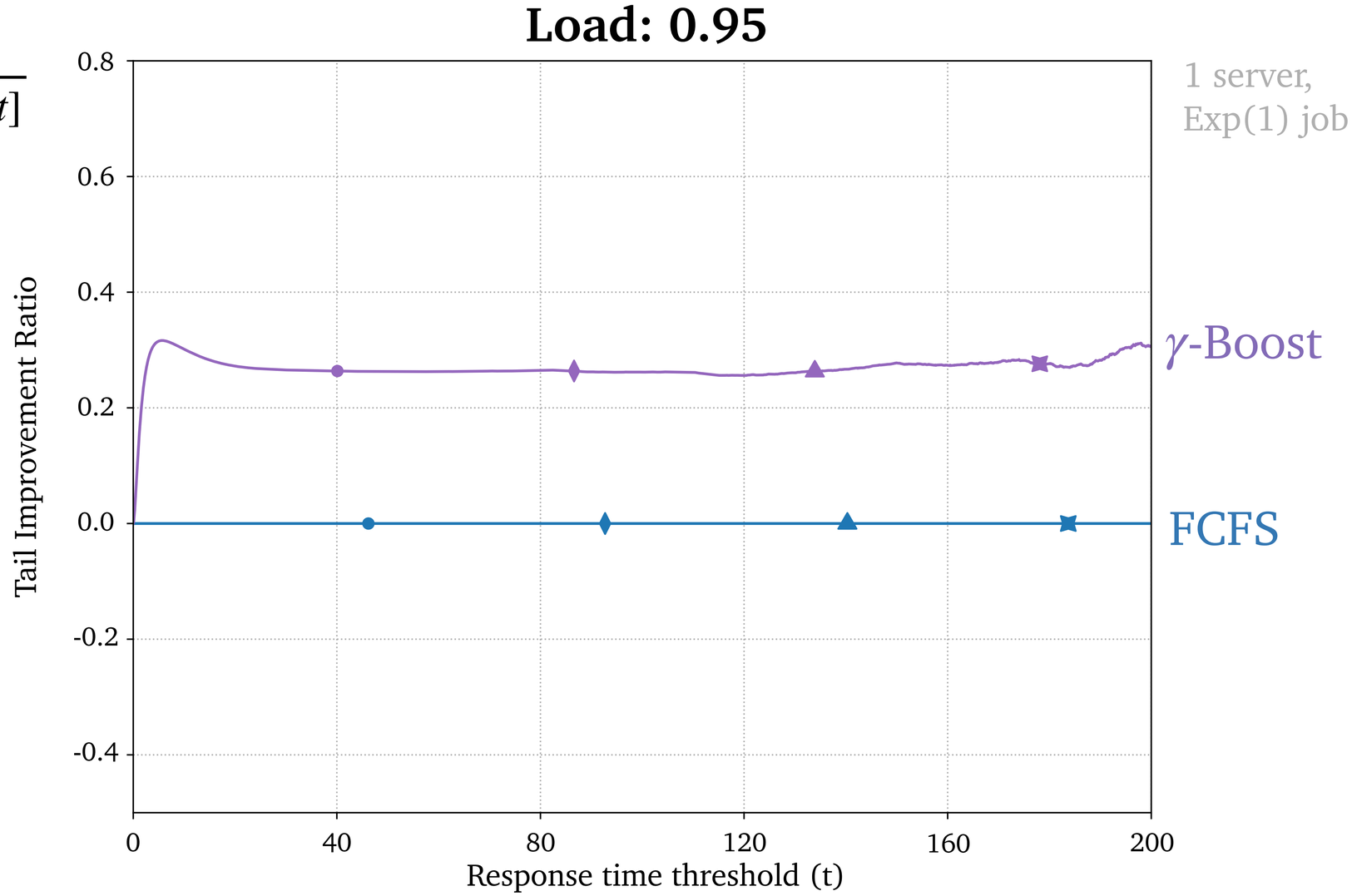
γ-Boost: serves job 2 then job 1.

Theorem: in the M/G/1, $C_{\gamma\text{–Boost}} = \inf_{\pi} C_{\pi}$ [Yu & Scully 2024]

γ-Boost's performance in the M/G/1

$$1 - \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\text{FCFS}} > t]}$$

(Higher is better)

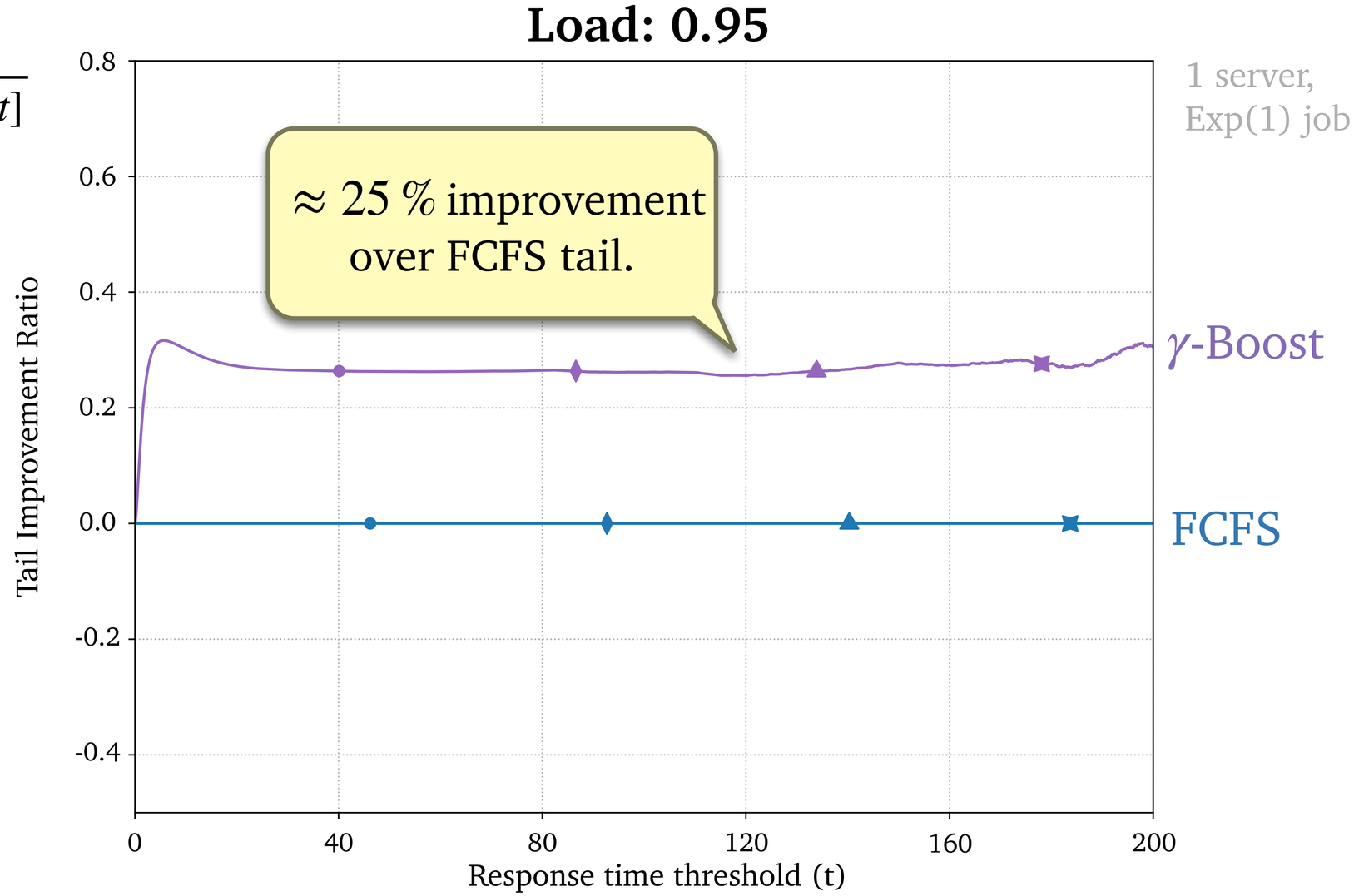


90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

γ-Boost's performance in the M/G/1

$$1 - \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\text{FCFS}} > t]}$$

(Higher is better)



• 90th percentile ♦ 99th percentile ▲ 99.9th percentile ✖ 99.99th percentile ★ 99.999th percentile

This talk

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?

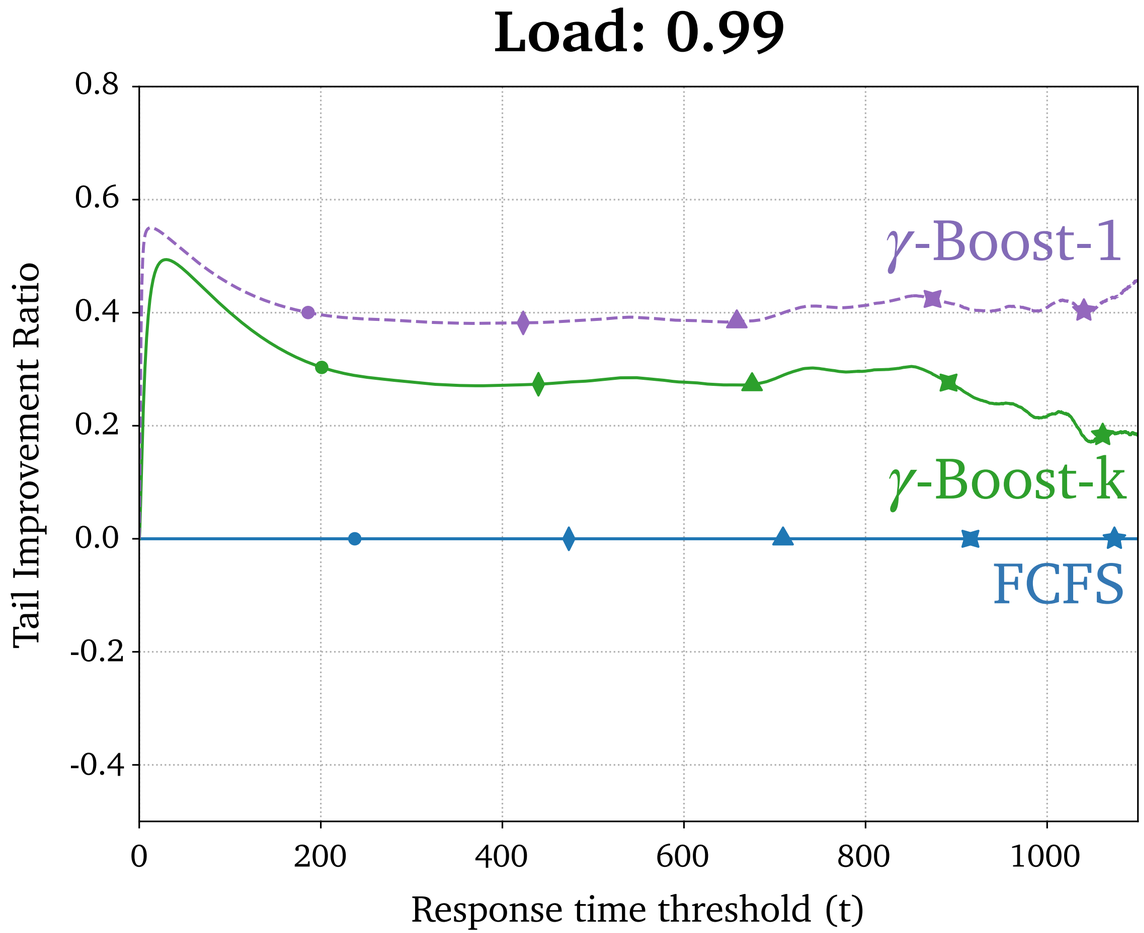
What is the γ-Boost policy?

Is γ-Boost tail optimal in the M/G/k in heavy traffic?

Does γ-Boost have good empirical tail performance in the M/G/k in lighter traffic?

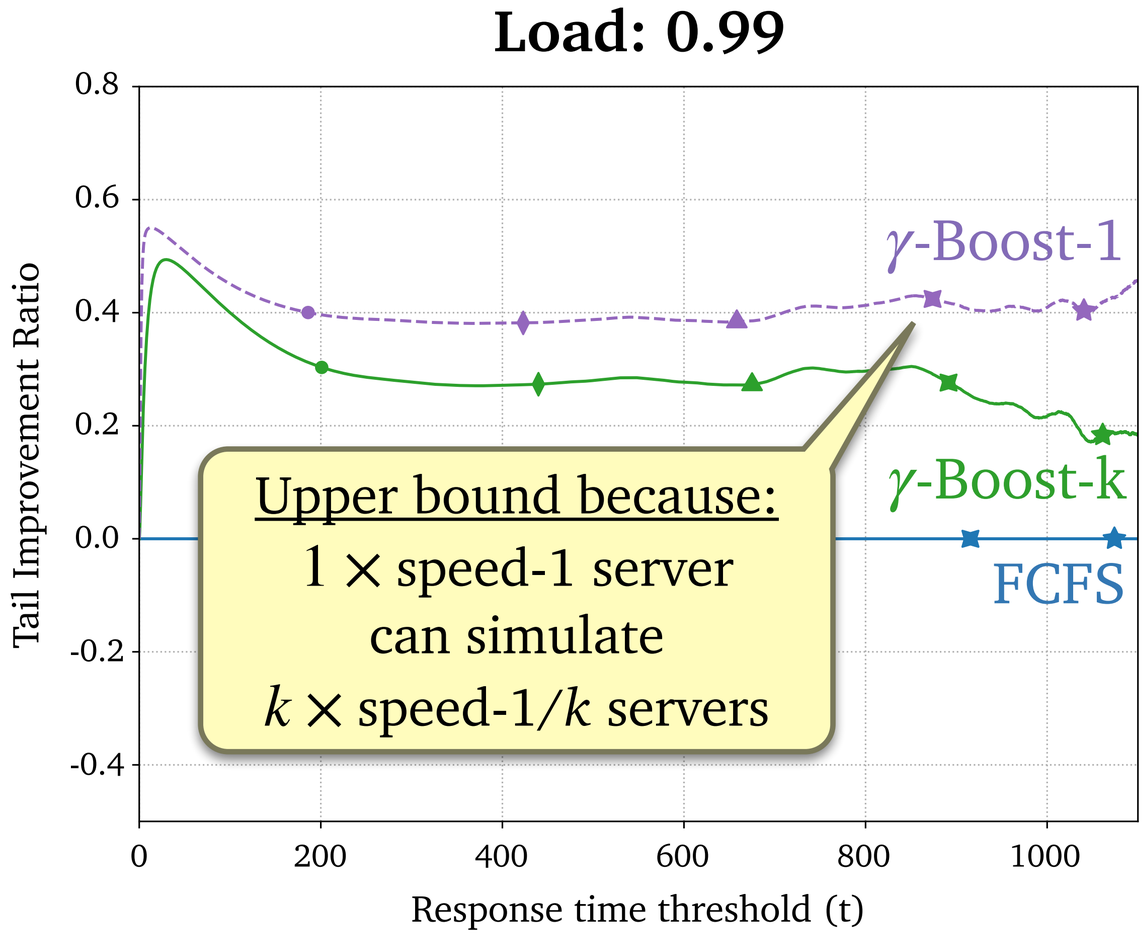
γ-Boost-k performance in heavy traffic

γ-Boost-k performance in heavy traffic



10 servers,
Exp(1) job sizes

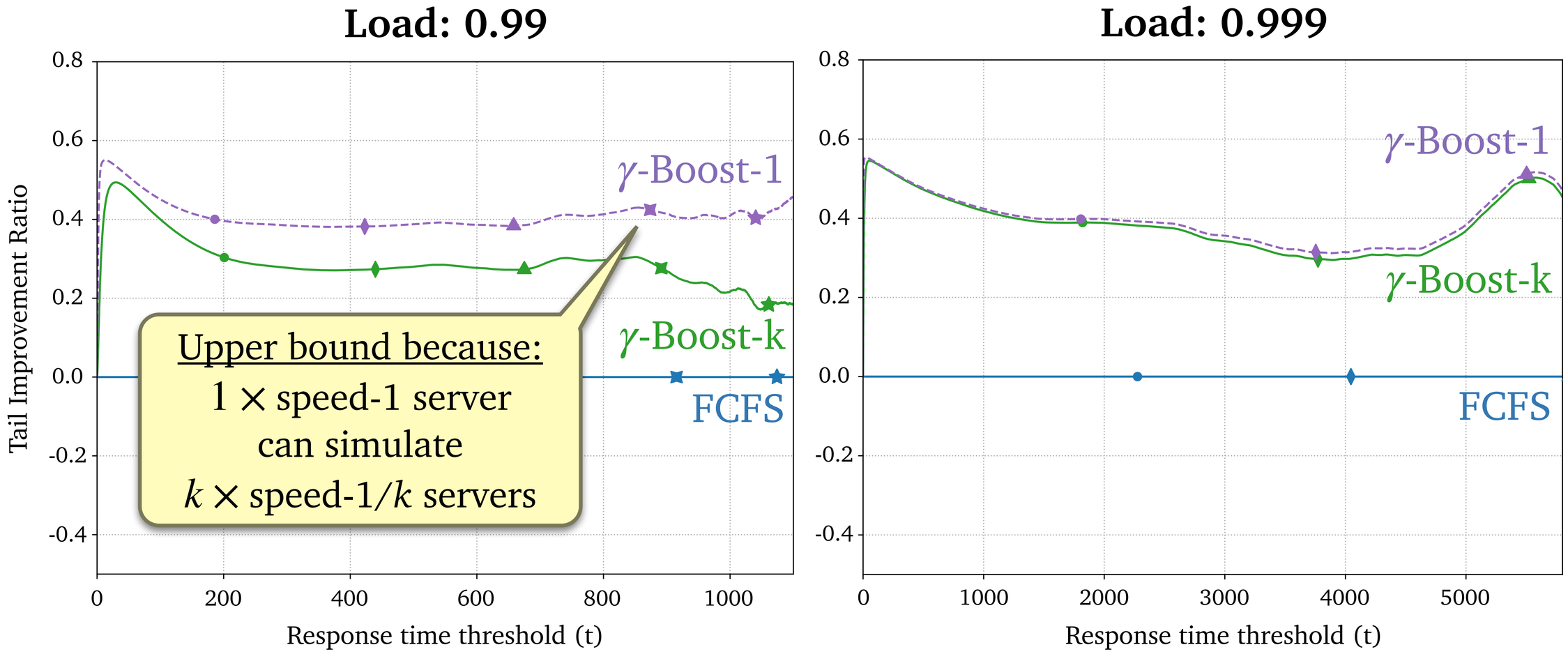
γ-Boost-k performance in heavy traffic



• 90th percentile ♦ 99th percentile ▲ 99.9th percentile ✖ 99.99th percentile ★ 99.999th percentile

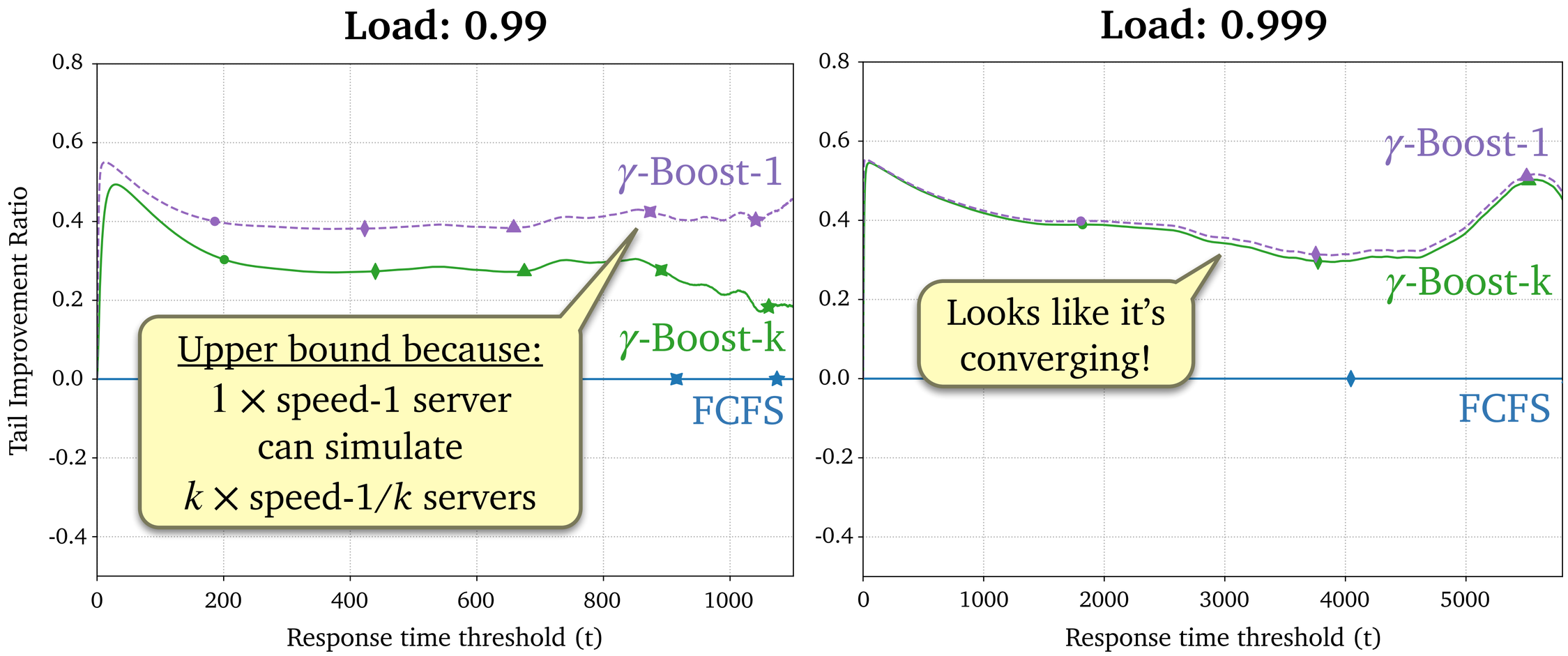
10 servers,
Exp(1) job sizes

γ-Boost-k performance in heavy traffic



10 servers,
Exp(1) job sizes

γ-Boost-k performance in heavy traffic



10 servers,
Exp(1) job sizes

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Main tool: Work Decomposition Law

$$W_{\text{M/G/k}} = W_{\text{M/G/1}} + W_{\text{extra}}$$

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Main tool: Work Decomposition Law

$$W_{\text{M/G/k}} = W_{\text{M/G/1}} + W_{\text{extra}}$$

Independent

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Main tool: Work Decomposition Law

$$W_{\text{M/G/k}} = W_{\text{M/G/1}} + W_{\text{extra}}$$

Depends on π

Independent

Depends on π

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Main tool: Work Decomposition Law

$$W_{\text{M/G/k}} = W_{\text{M/G/1}} + W_{\text{extra}}$$

Depends on π

Independent

Depends on π

Lemma: $\mathbf{P}[W_{\text{extra}} > t] \leq \min\left(1, \frac{k\rho}{1-\rho}\mathbf{P}[kS_{\text{excess}} > t]\right)$

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Main tool: Work Decomposition Law

$$W_{\text{M/G/k}} = W_{\text{M/G/1}} + W_{\text{extra}}$$

Depends on π

Independent

Depends on π

Lemma: $\mathbf{P}[W_{\text{extra}} > t] \leq \min\left(1, \frac{k\rho}{1-\rho}\mathbf{P}[kS_{\text{excess}} > t]\right)$

For all work-conserving policies!

Theorem: $\lim_{\rho \to 1} C_{\gamma\text{-Boost-k}} = C_{\gamma\text{-Boost-1}}$

Main tool: Work Decomposition Law

$$W_{\text{M/G/k}} = W_{\text{M/G/1}} + W_{\text{extra}}$$

Depends on π

Independent

Depends on π

Lemma: $\mathbf{P}[W_{\text{extra}} > t] \leq \min\left(1, \frac{k\rho}{1-\rho}\mathbf{P}[kS_{\text{excess}} > t]\right)$

George's Lemma

For all work-conserving policies!

This talk

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?

What is the γ-Boost policy?

Is γ-Boost tail optimal in the M/G/k in heavy traffic?

Does γ-Boost have good empirical tail performance in the M/G/k in lighter traffic?

This talk

What do we mean by min $\mathbf{P}[T > t]$ "for all large t"?

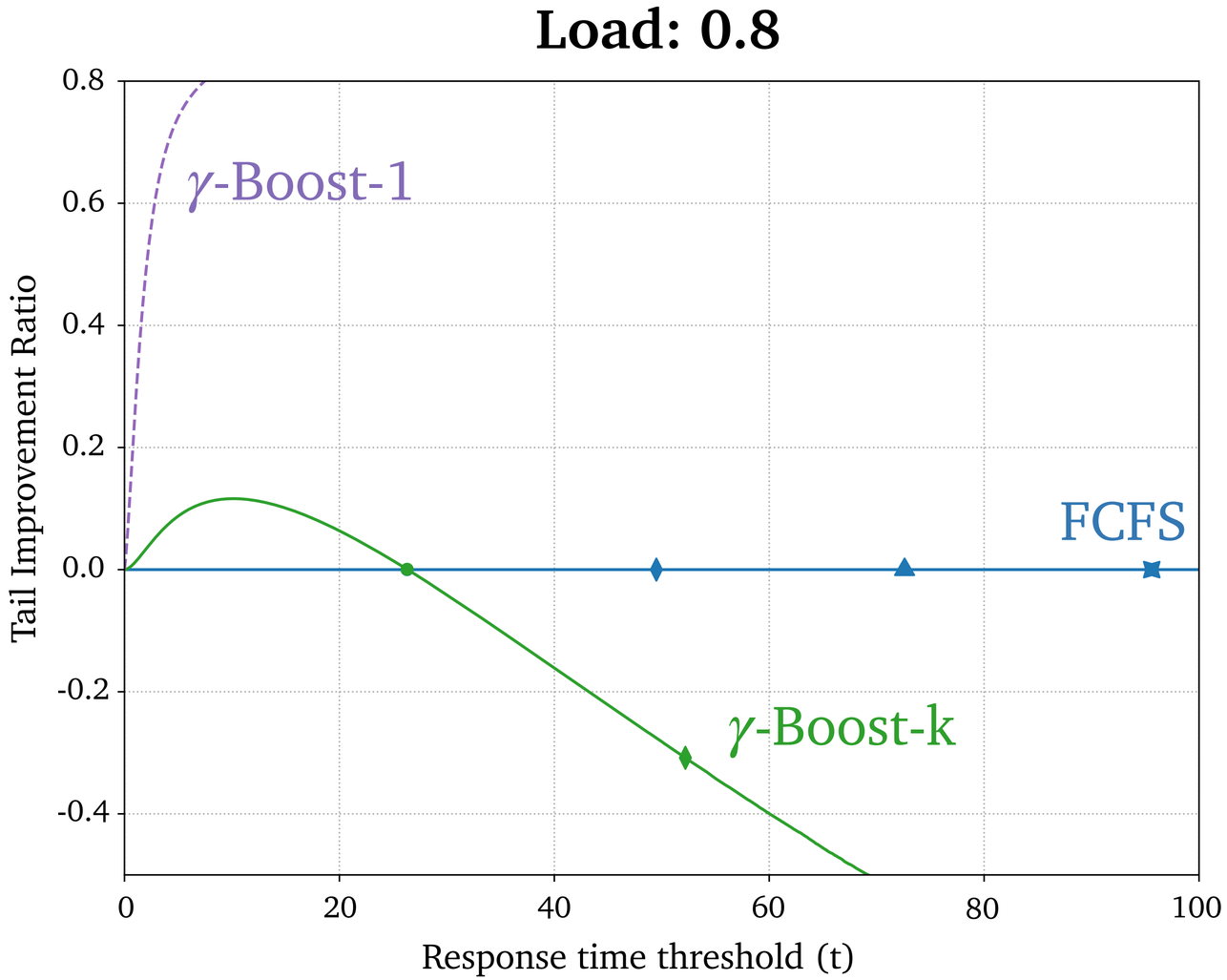
What is the γ-Boost policy?

Is γ-Boost tail optimal in the M/G/k in heavy traffic?

Yes!

Does γ-Boost have good empirical tail performance in the M/G/k in lighter traffic?

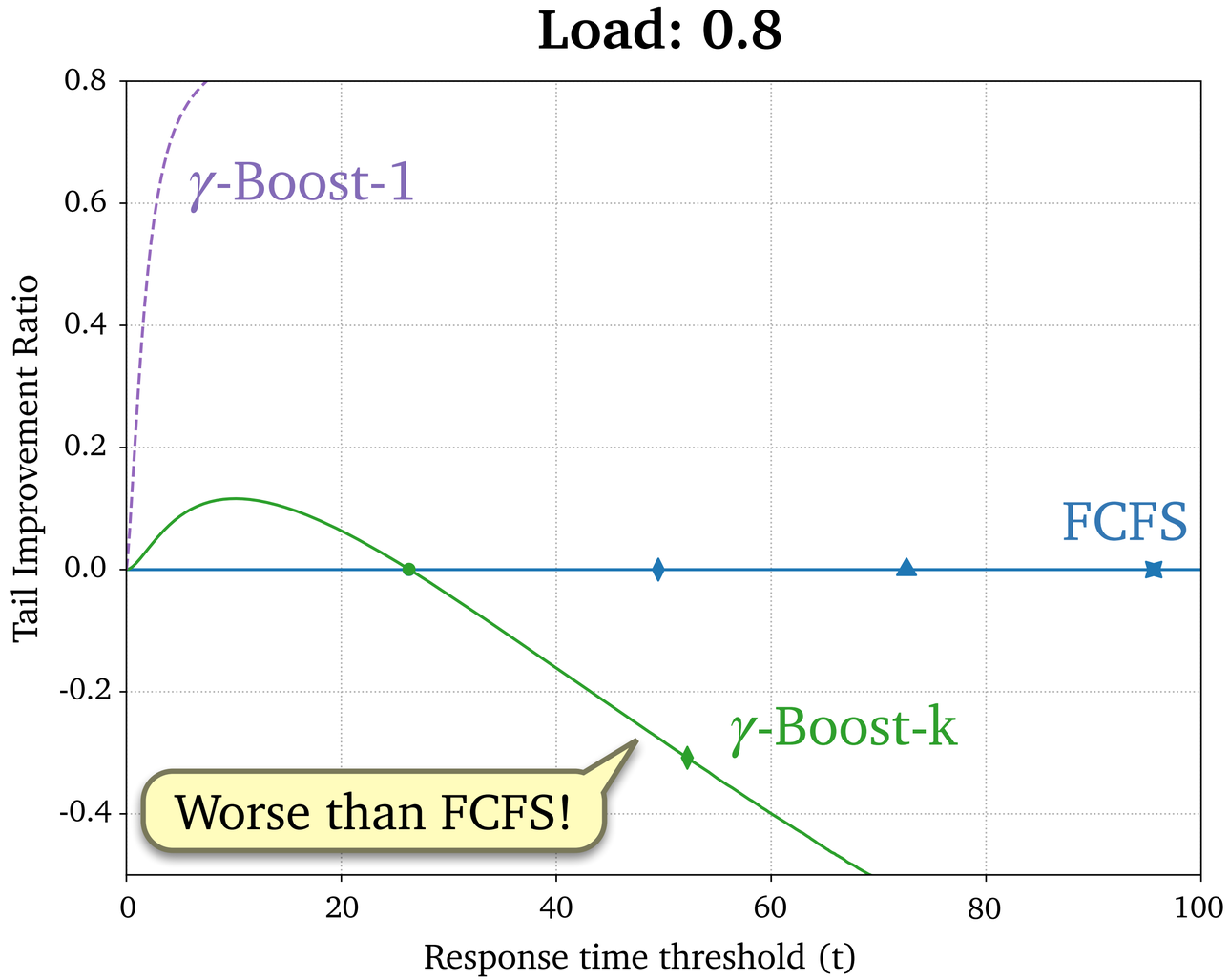
γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

• 90th percentile ♦ 99th percentile ▲ 99.9th percentile ✖ 99.99th percentile ★ 99.999th percentile

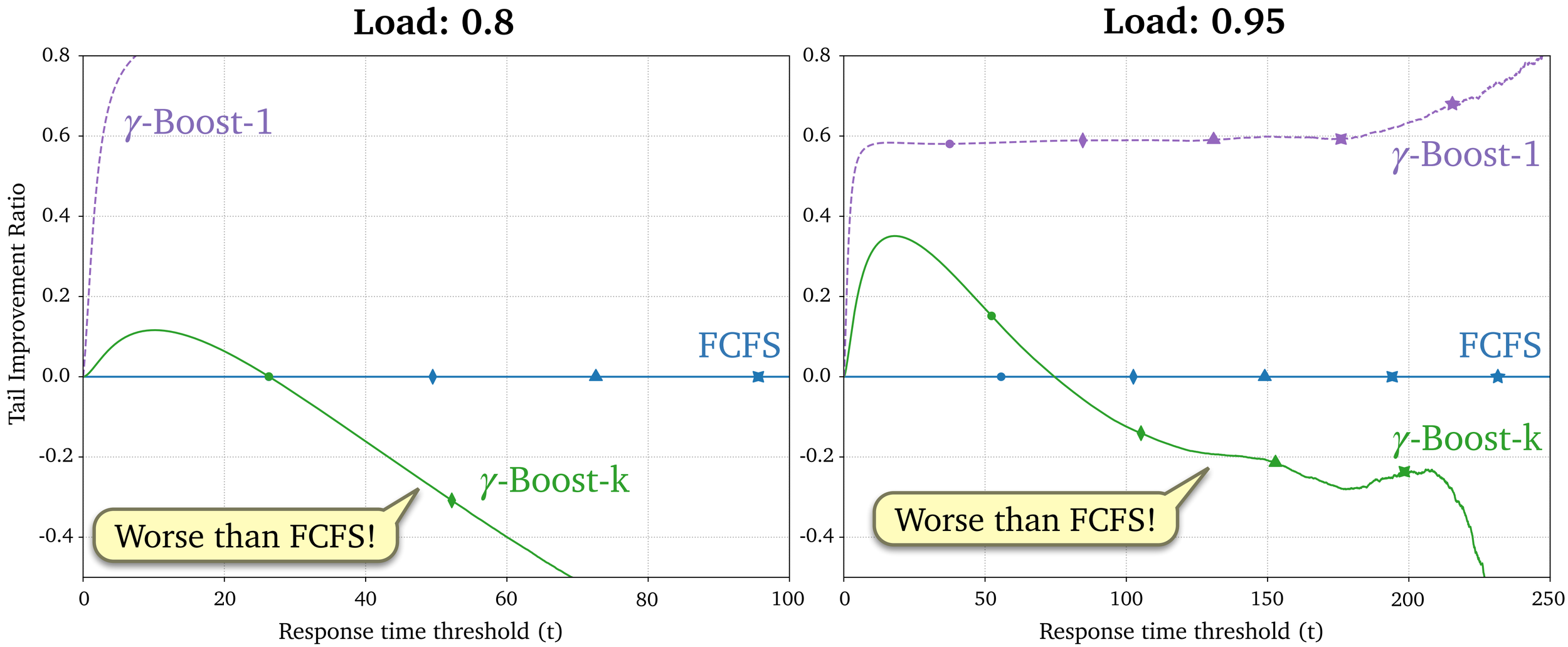
γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

This talk

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?

What is the γ-Boost policy?

Is γ-Boost tail optimal in the M/G/k in heavy traffic?

Yes!

Does γ-Boost have good empirical tail performance in the M/G/k in lighter traffic?

No..

This talk

What do we mean by $\min \mathbf{P}[T > t]$ "for all large t"?

What is the γ-Boost policy?

Is γ-Boost tail optimal in the M/G/k in heavy traffic?

Yes!

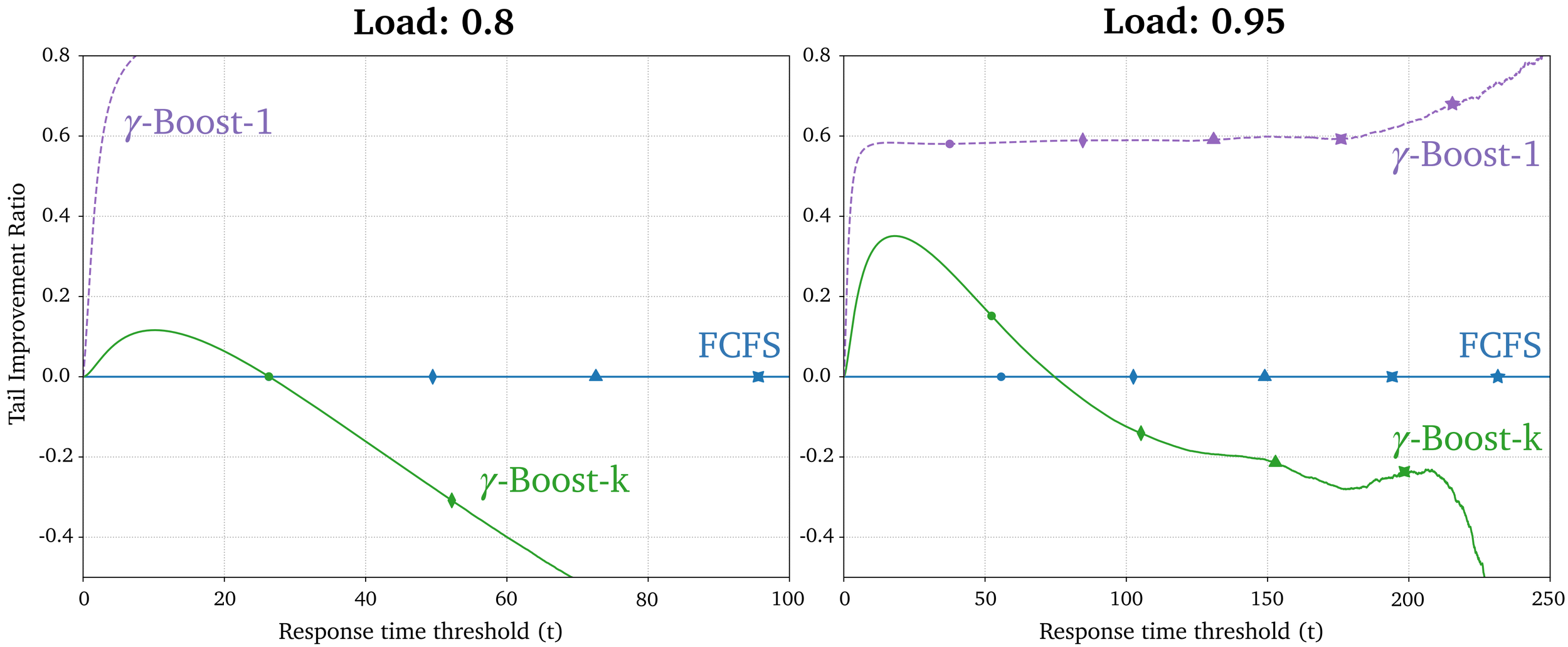
Does γ-Boost have [illegible] ance
in the M/G/k in lighter traffic?

What should we do instead?

No..

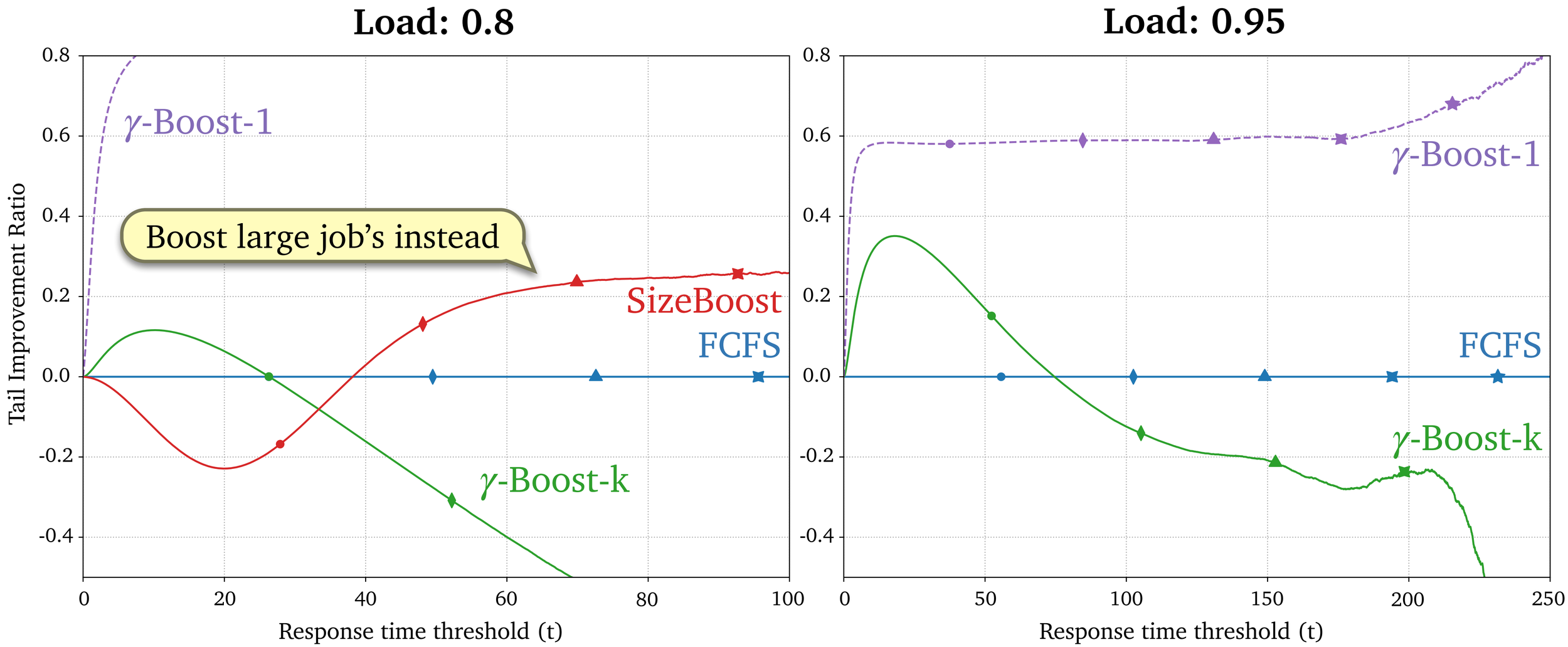
womp womp

γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

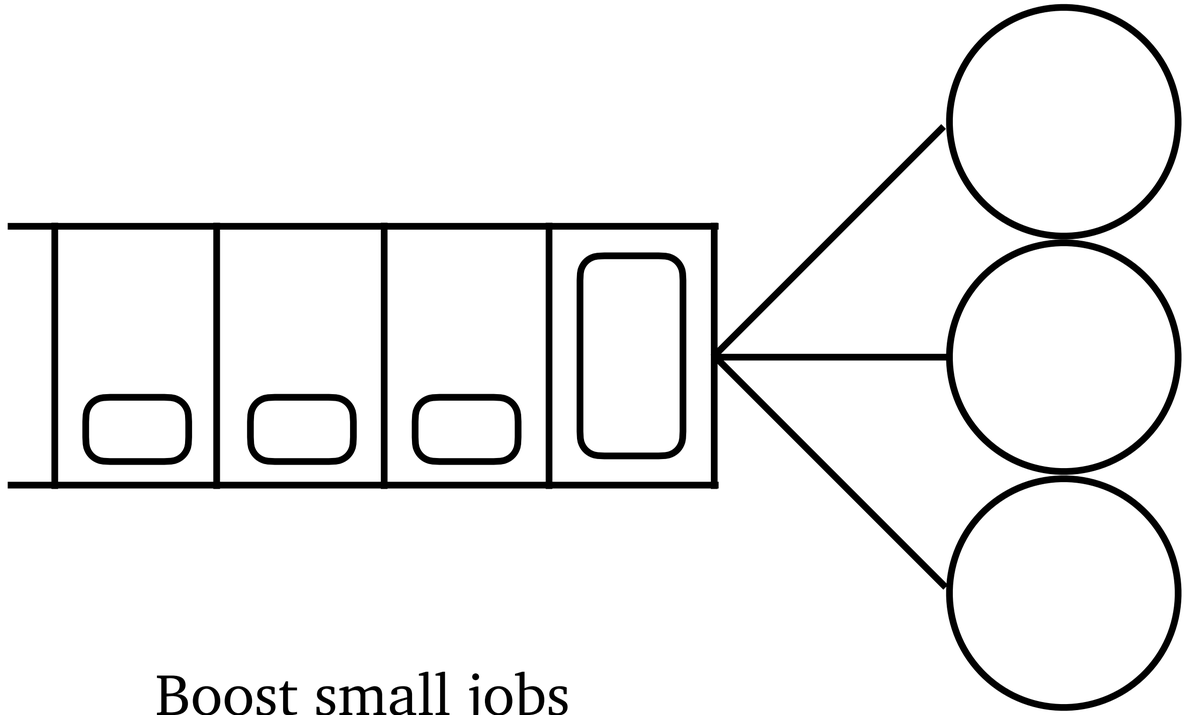
γ-Boost performance in lighter traffic



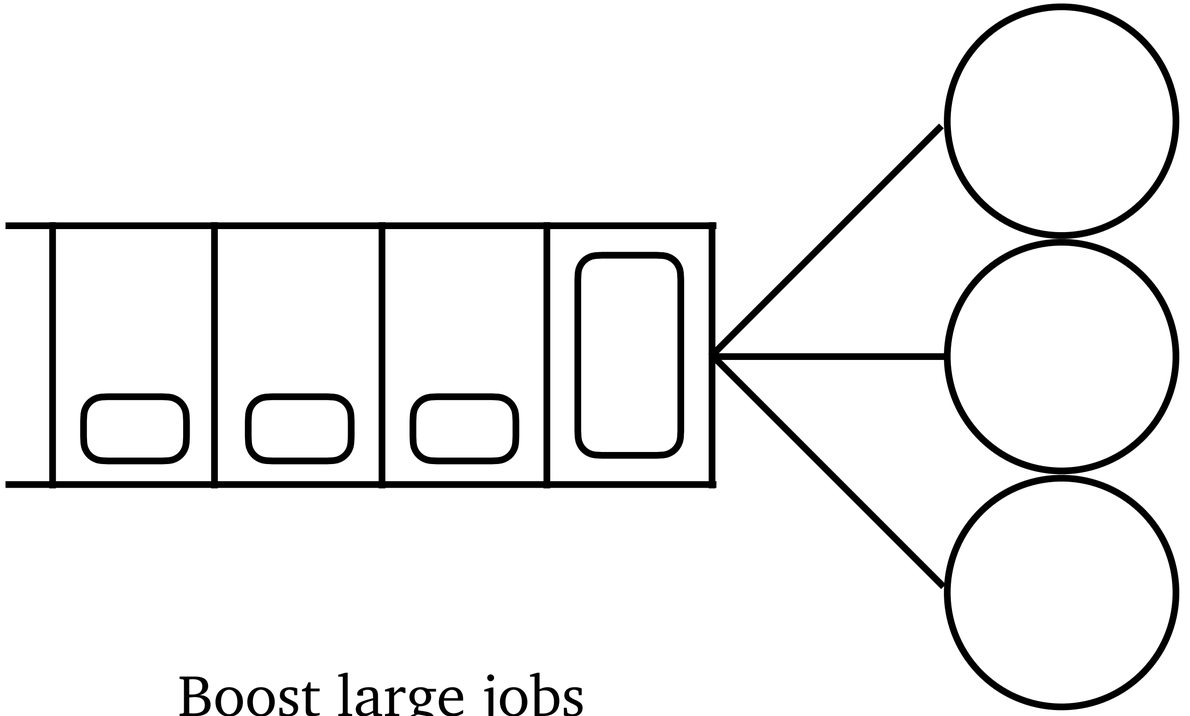
10 servers,
Exp(1) job sizes

• 90th percentile ♦ 99th percentile ▲ 99.9th percentile ✖ 99.99th percentile ★ 99.999th percentile

Why might SizeBoost make sense at low load?

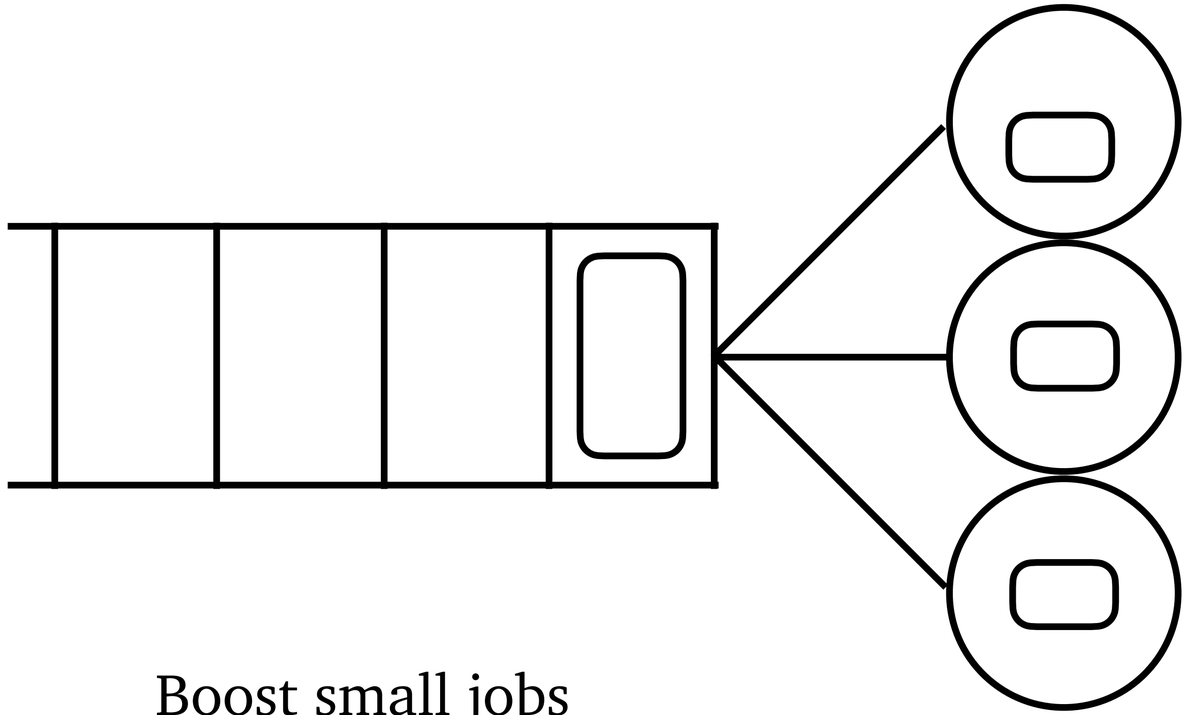


Boost small jobs

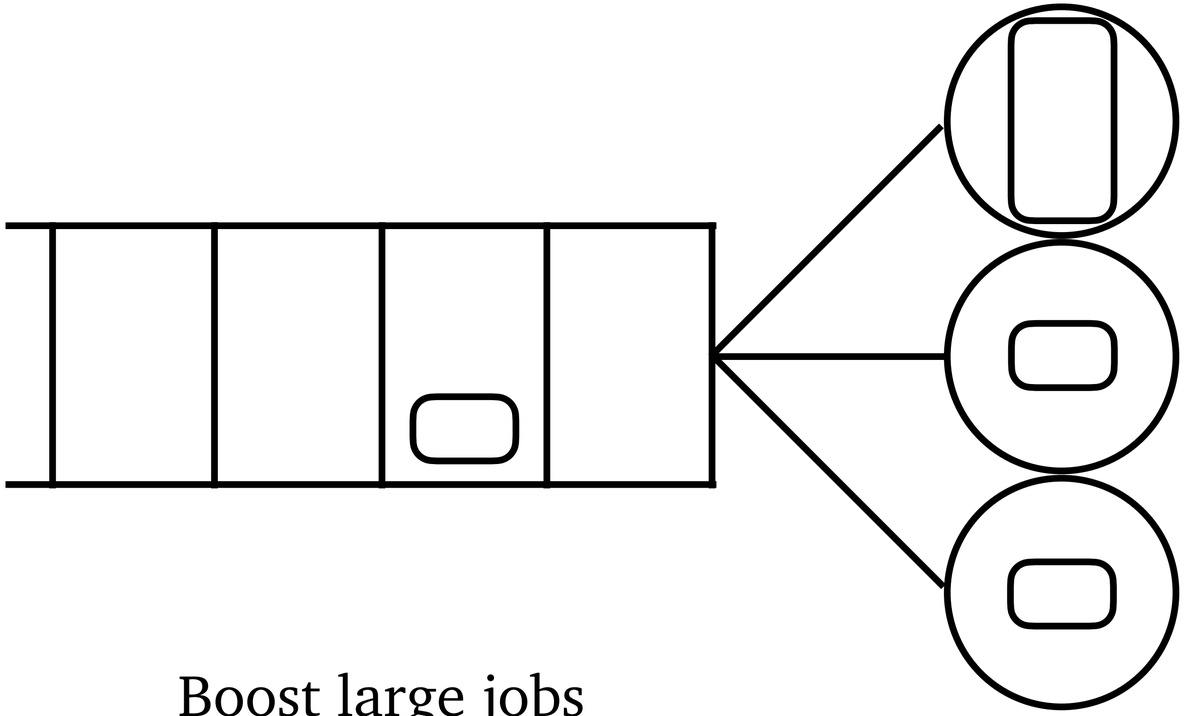


Boost large jobs

Why might SizeBoost make sense at low load?

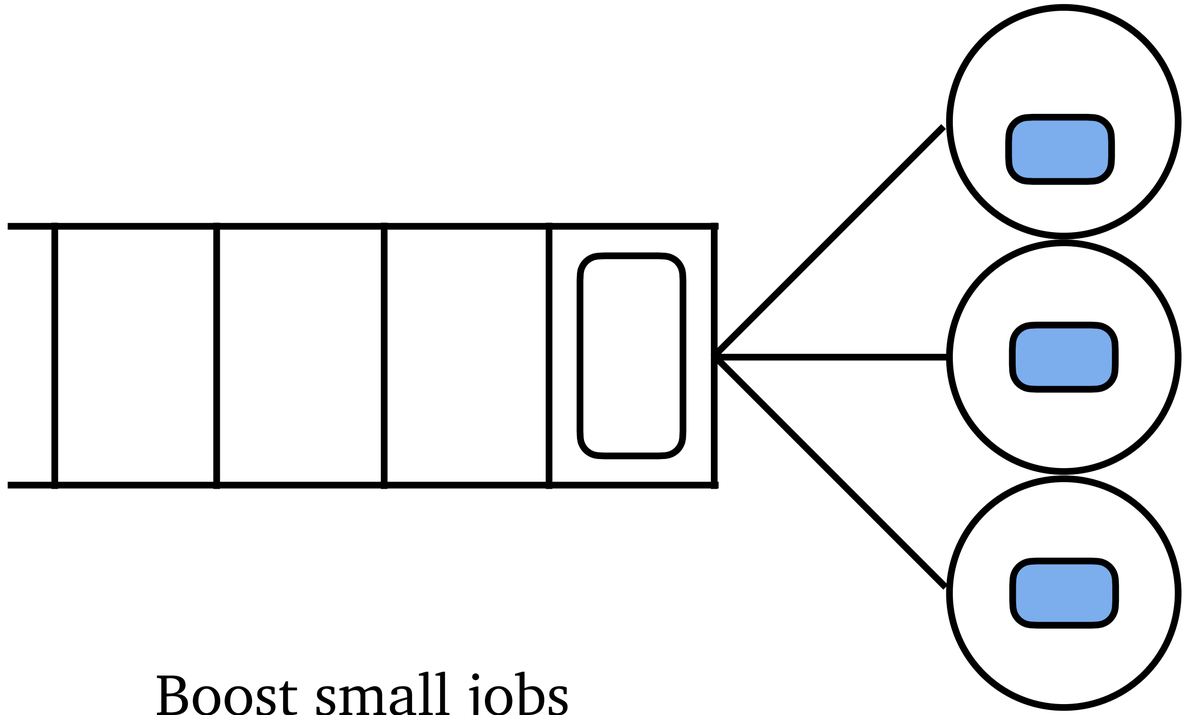


Boost small jobs

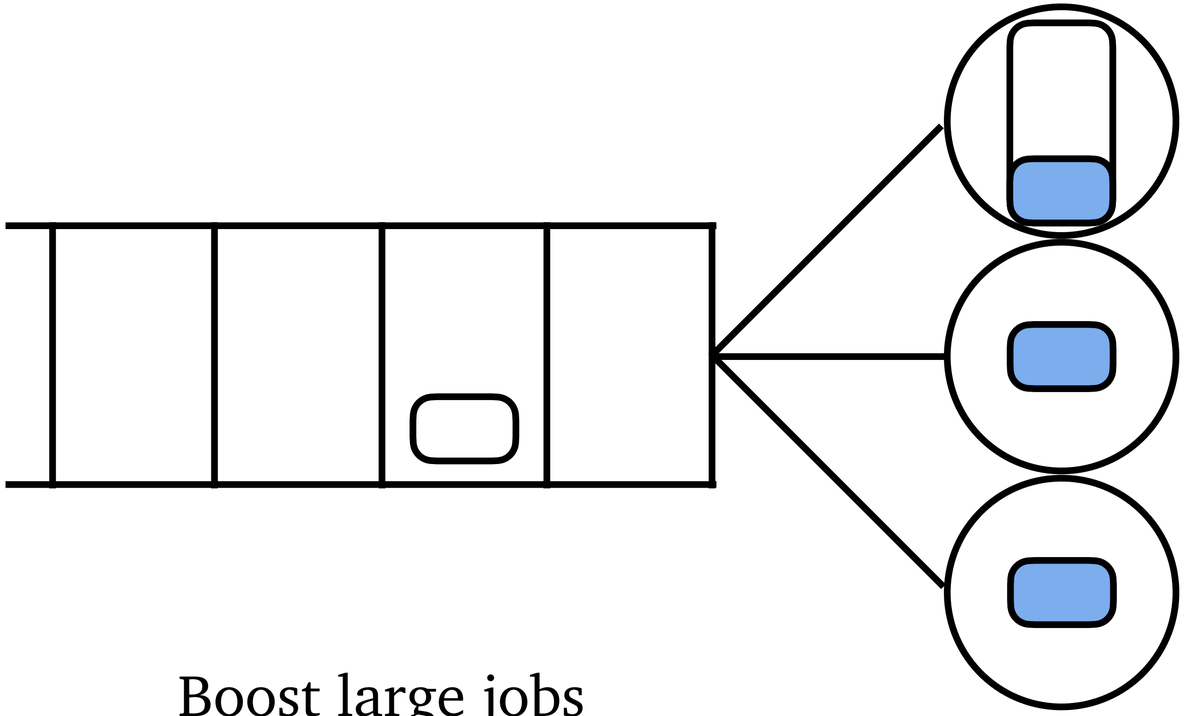


Boost large jobs

Why might SizeBoost make sense at low load?

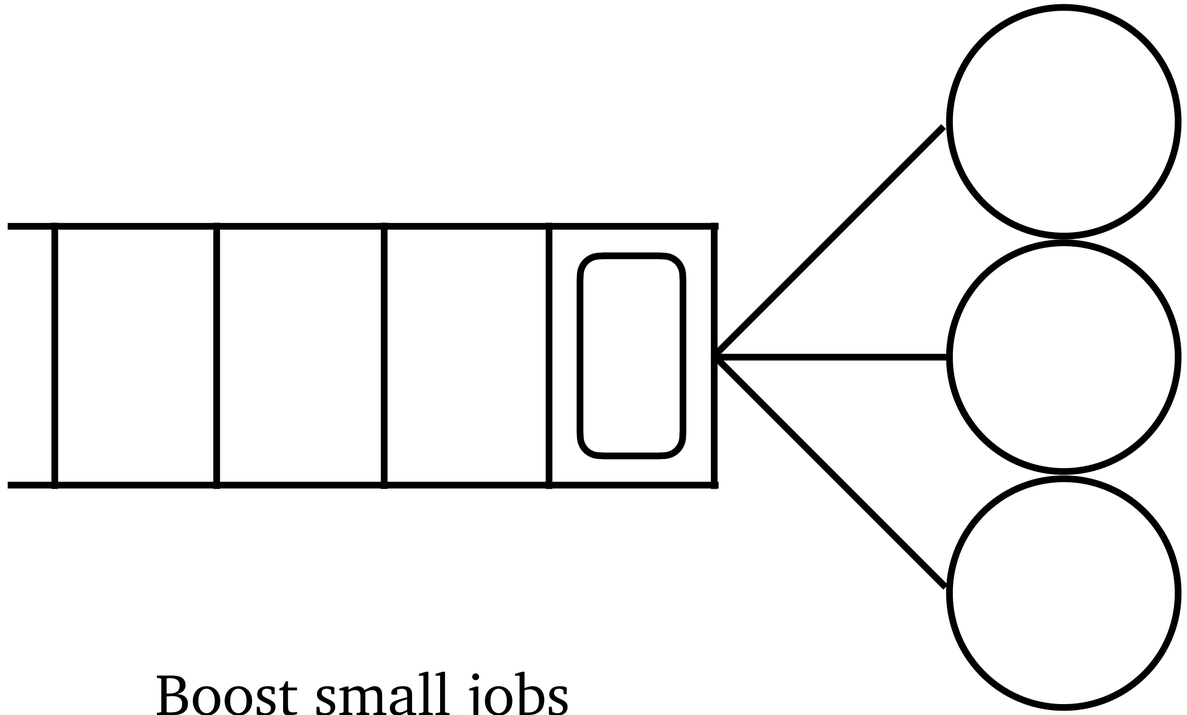


Boost small jobs

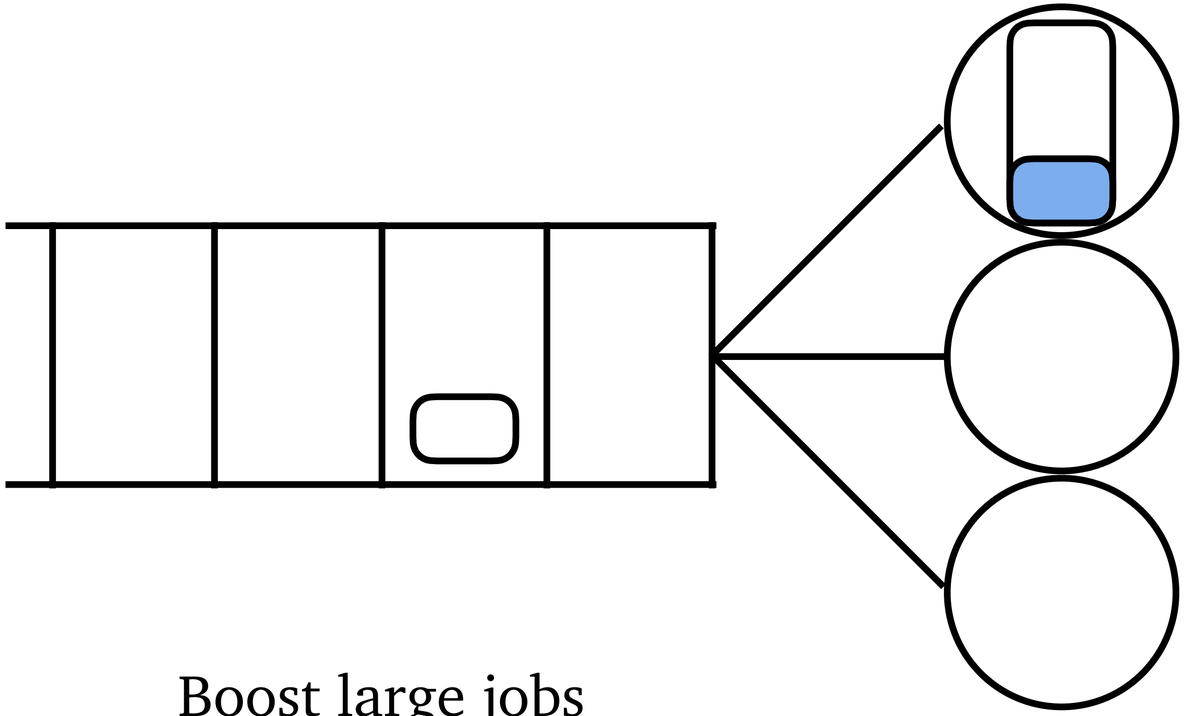


Boost large jobs

Why might SizeBoost make sense at low load?

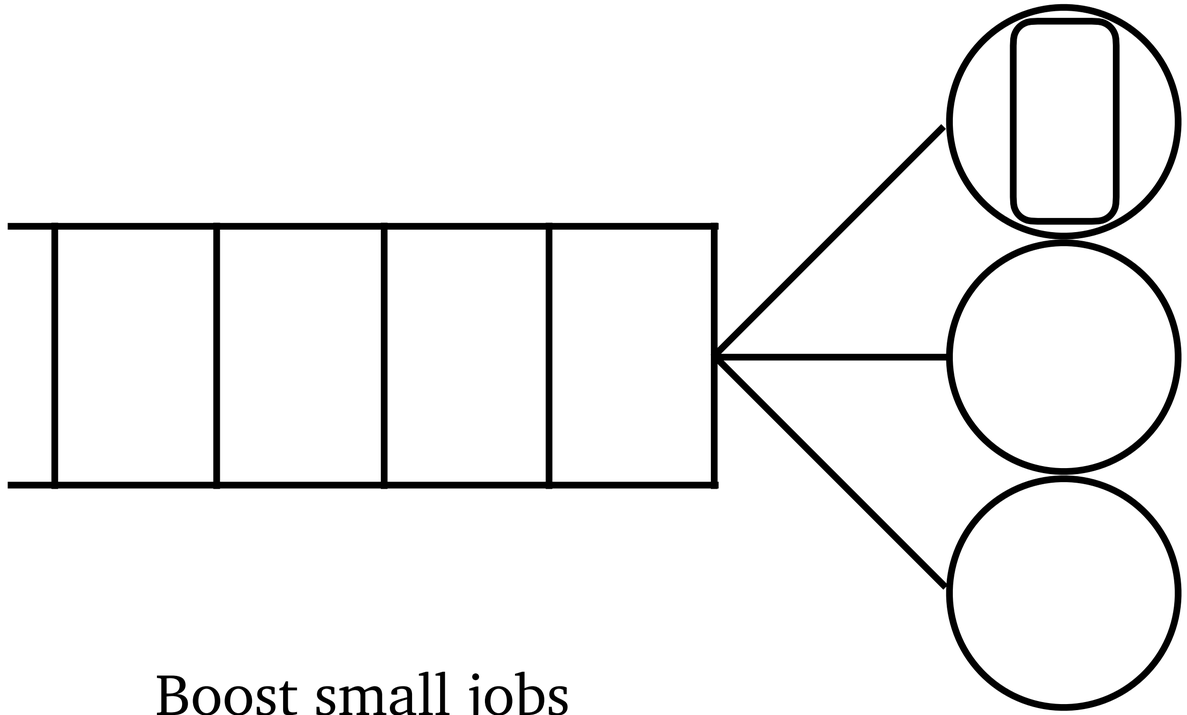


Boost small jobs

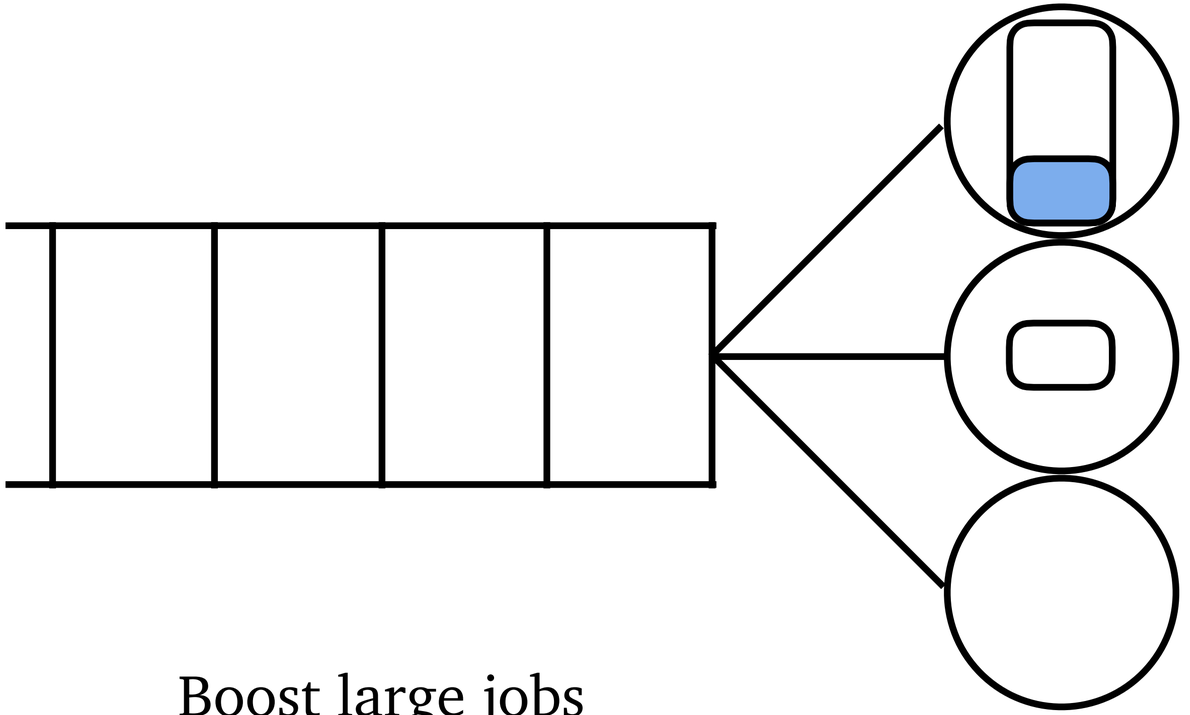


Boost large jobs

Why might SizeBoost make sense at low load?

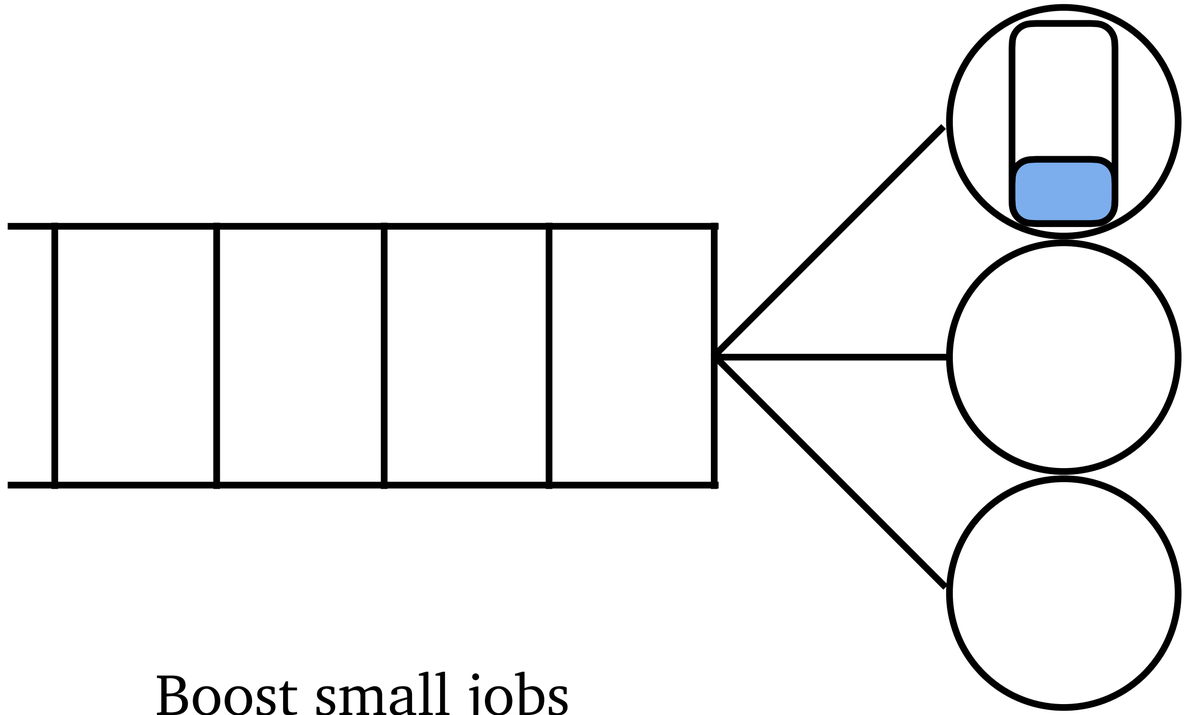


Boost small jobs

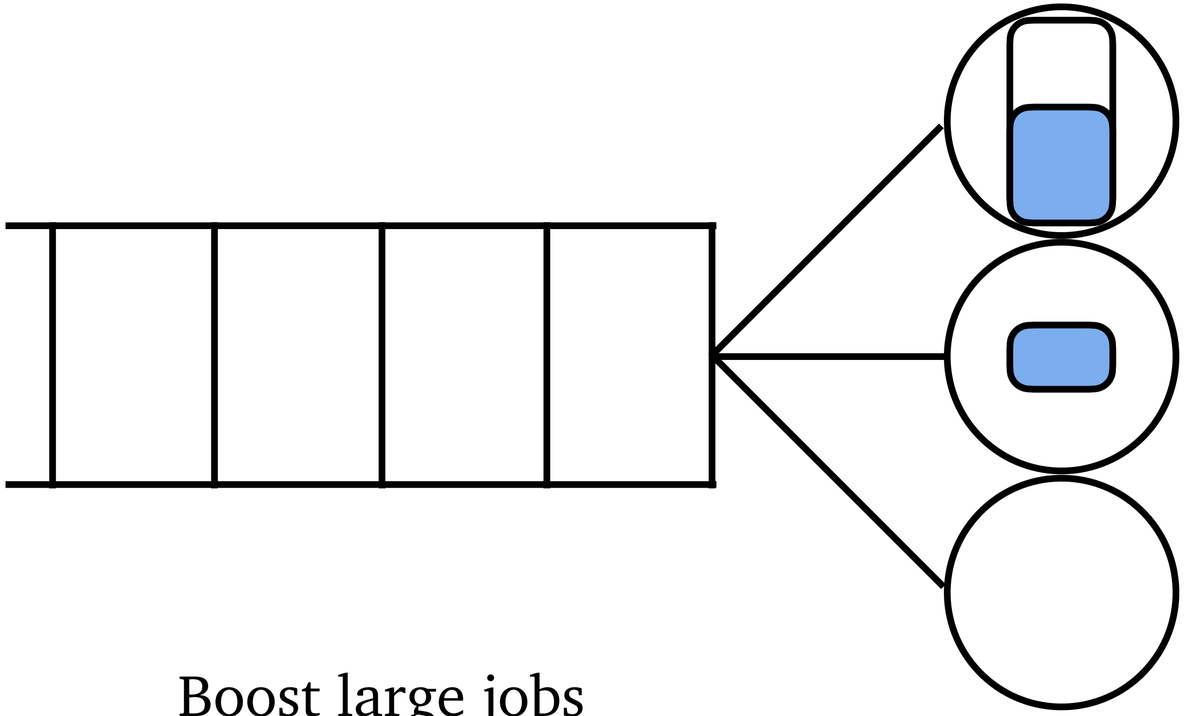


Boost large jobs

Why might SizeBoost make sense at low load?

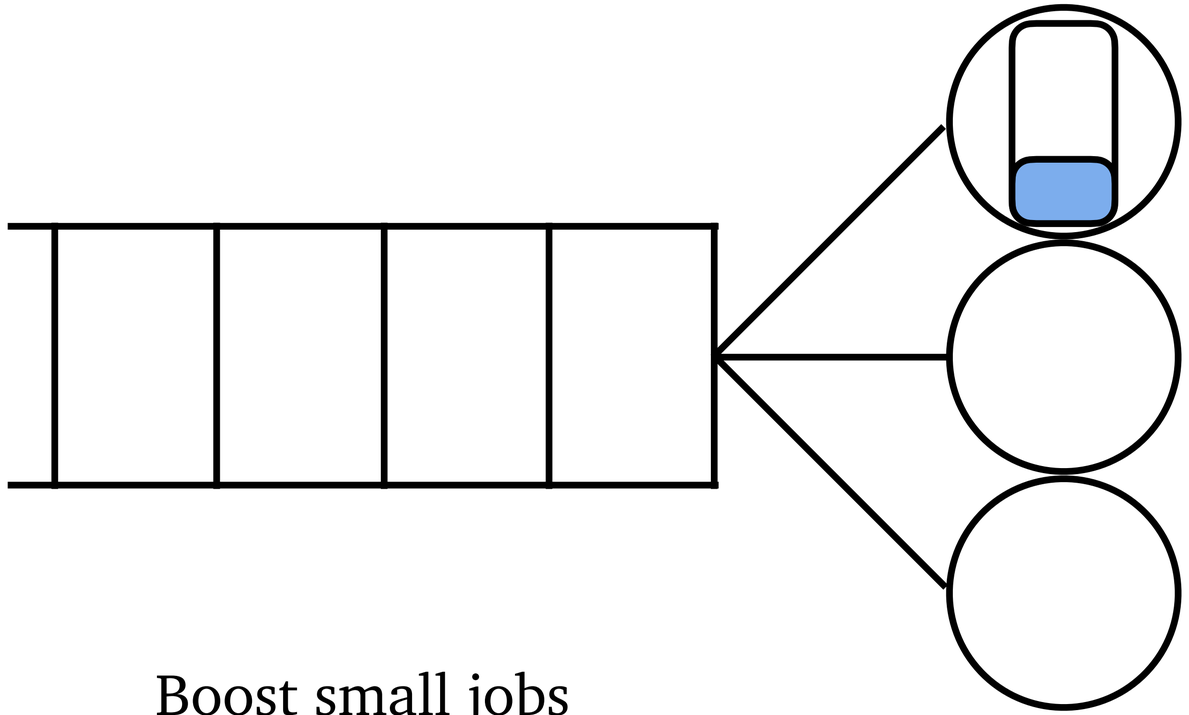


Boost small jobs

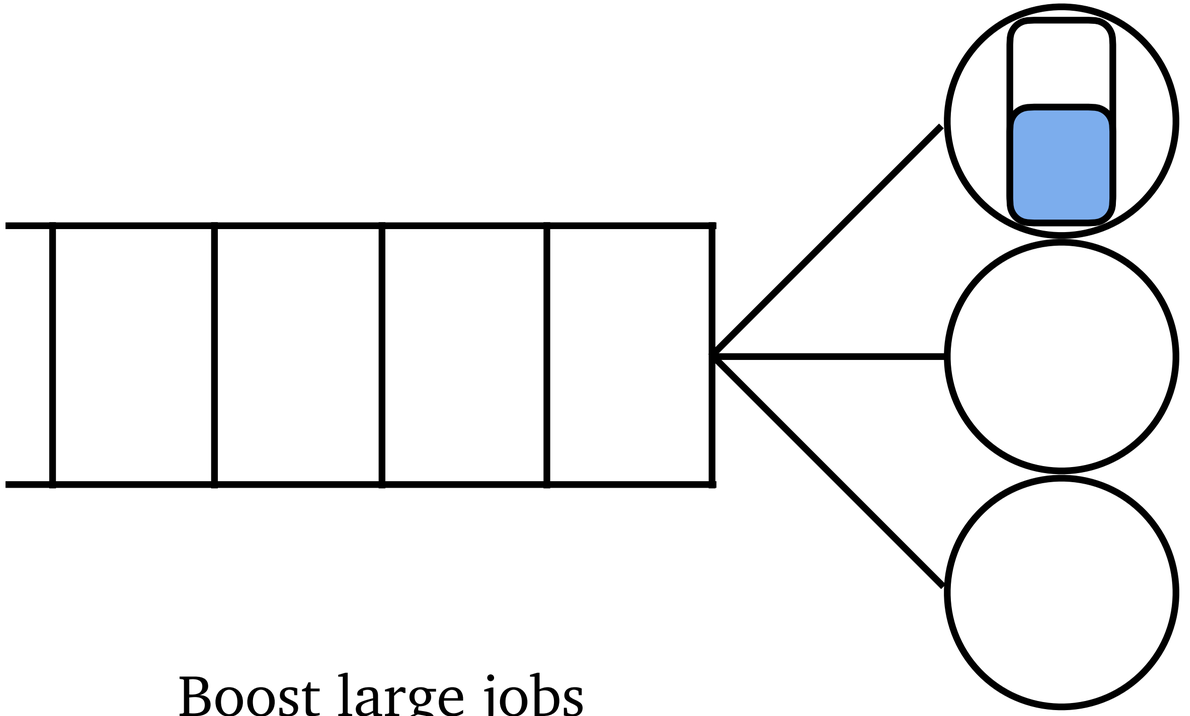


Boost large jobs

Why might SizeBoost make sense at low load?

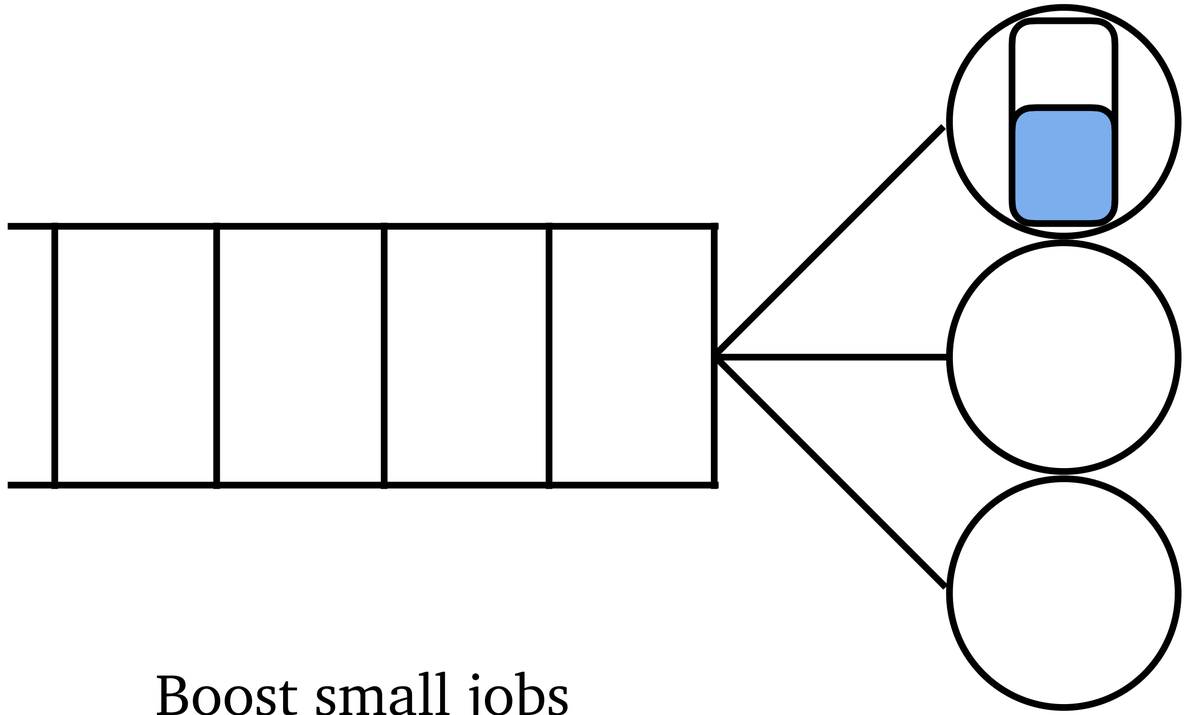


Boost small jobs

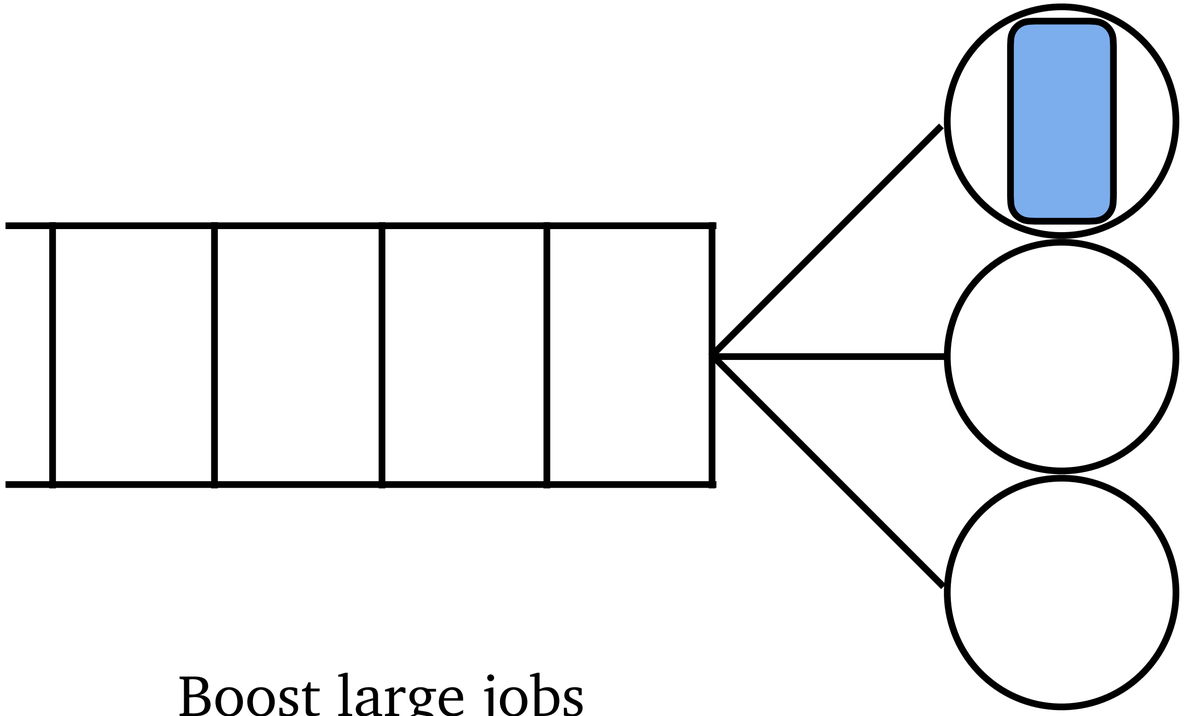


Boost large jobs

Why might SizeBoost make sense at low load?

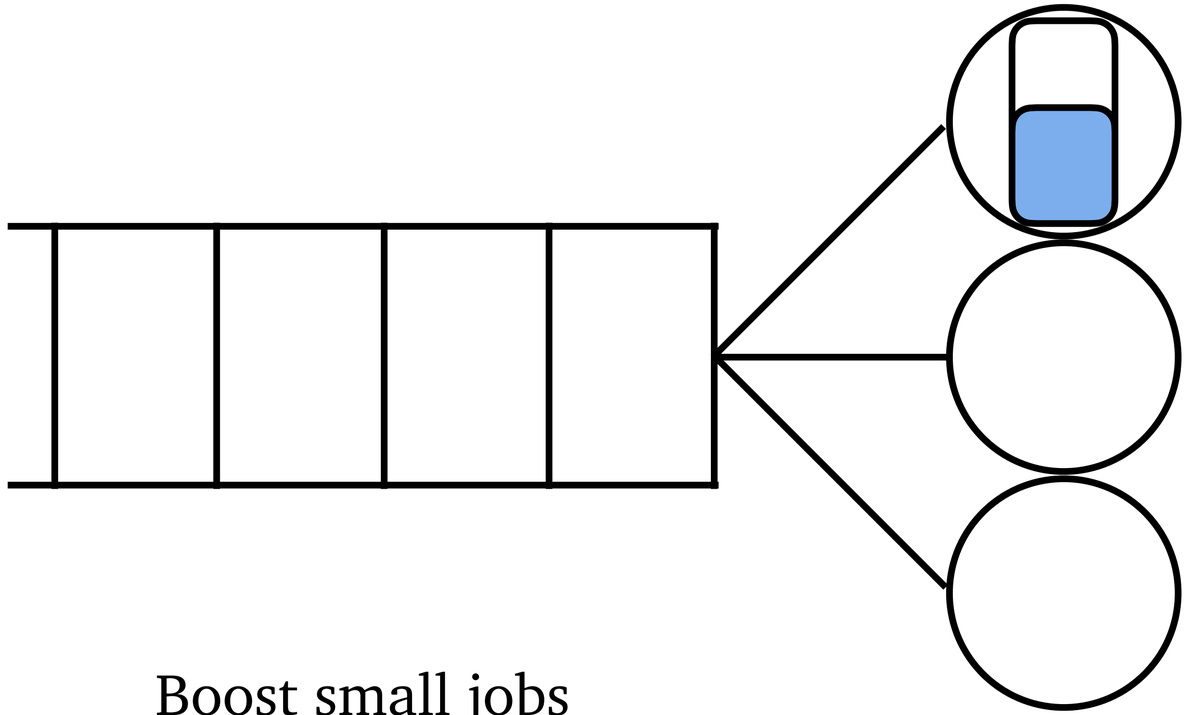


Boost small jobs

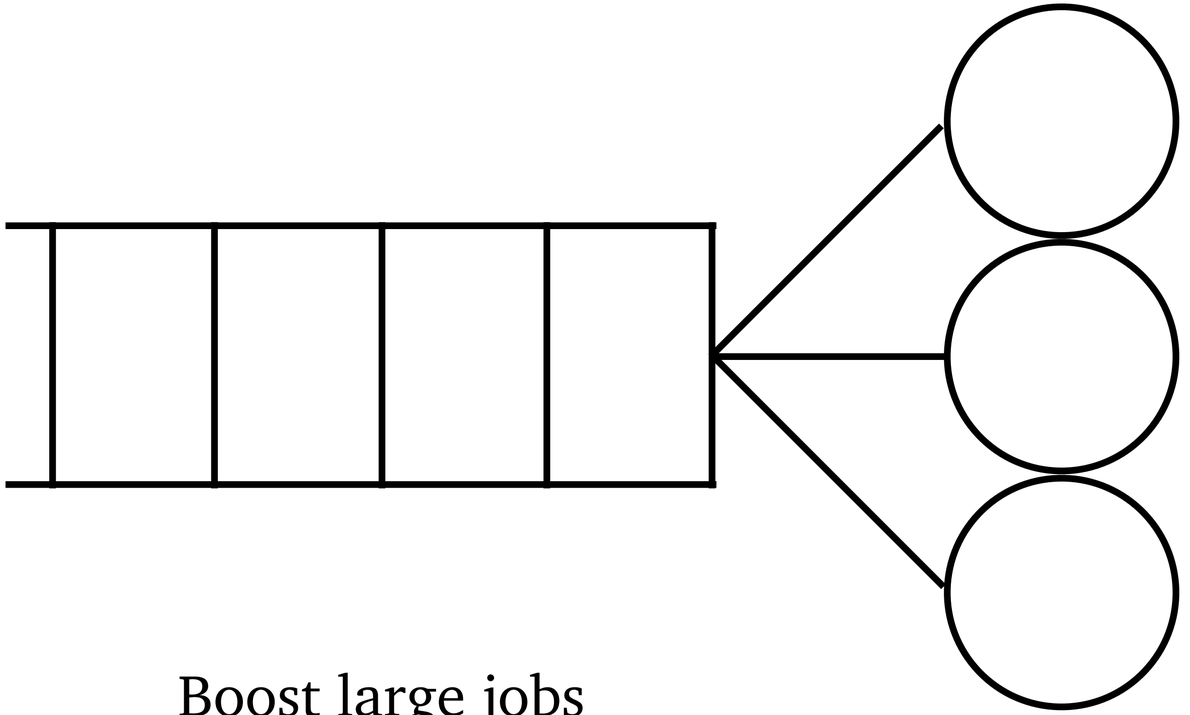


Boost large jobs

Why might SizeBoost make sense at low load?

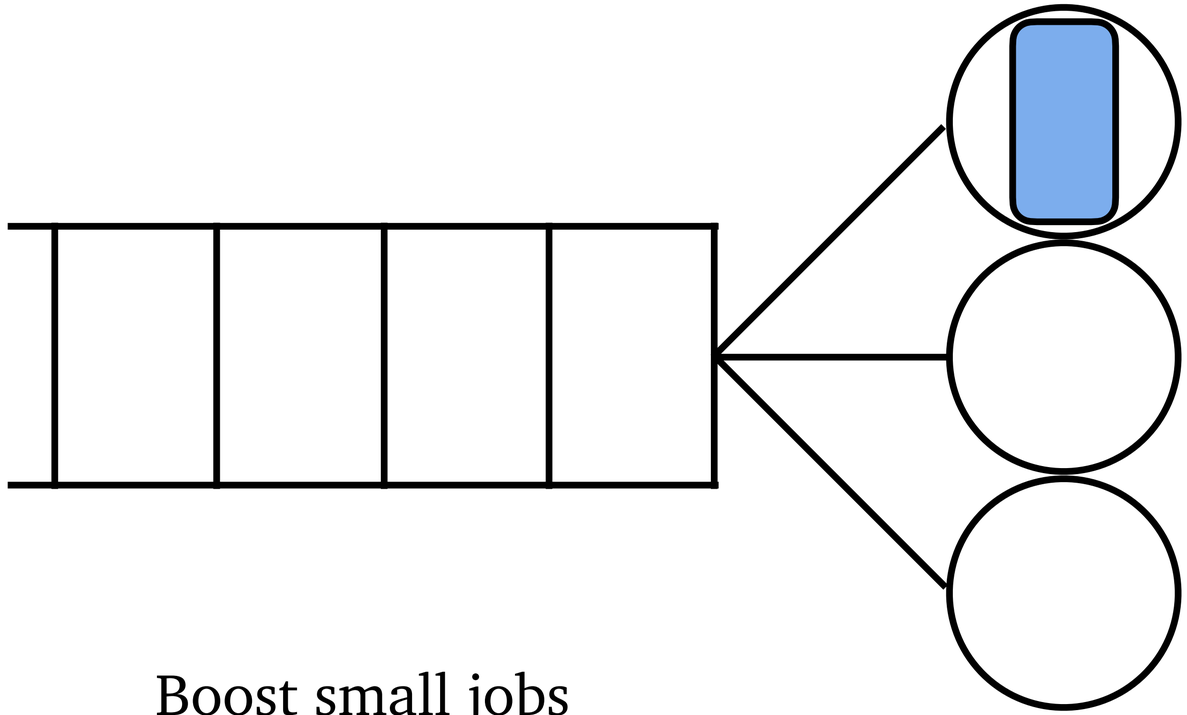


Boost small jobs

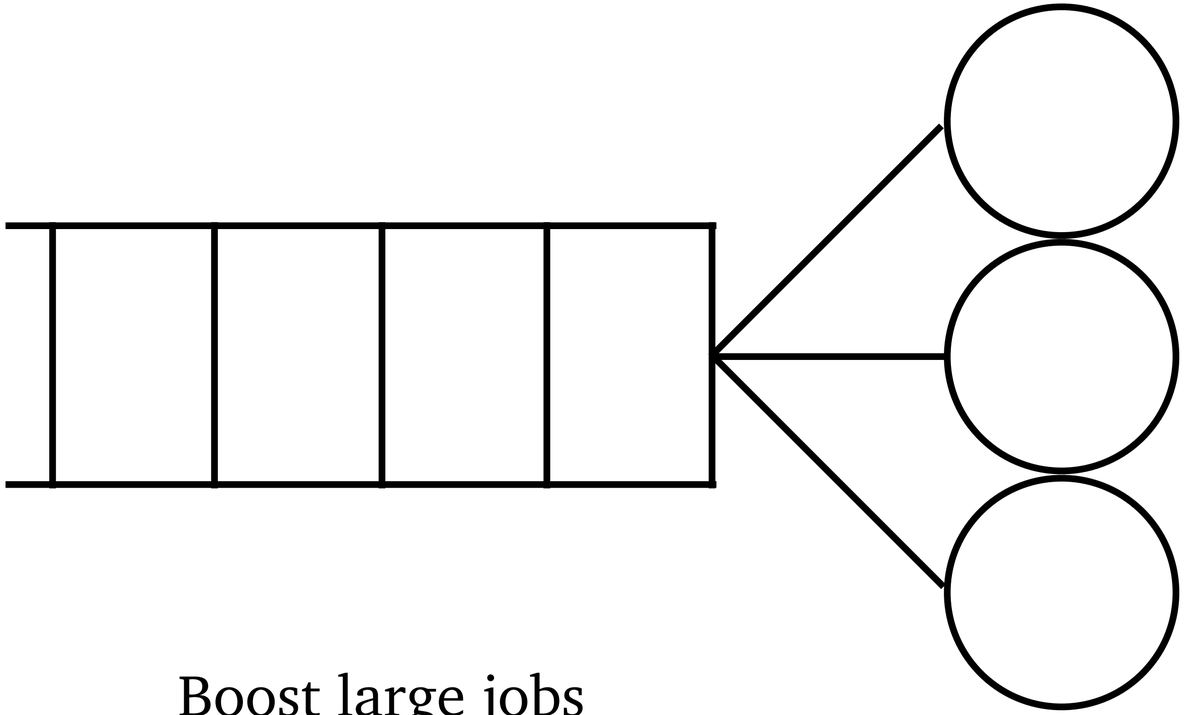


Boost large jobs

Why might SizeBoost make sense at low load?

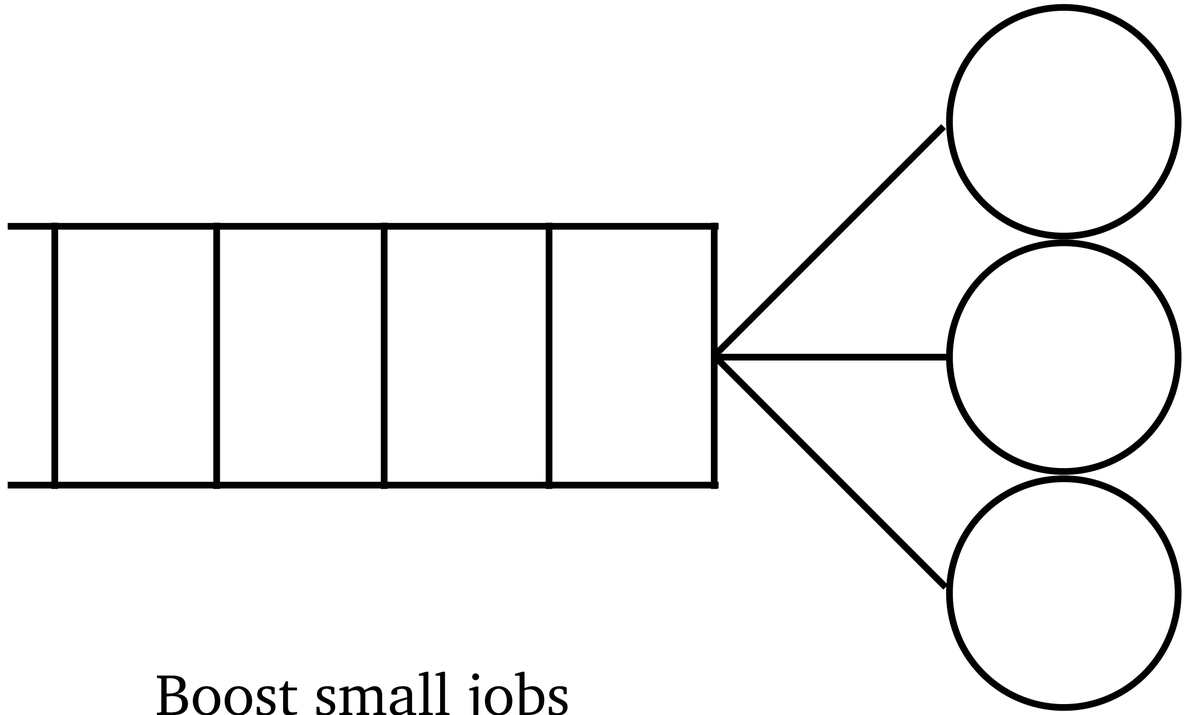


Boost small jobs

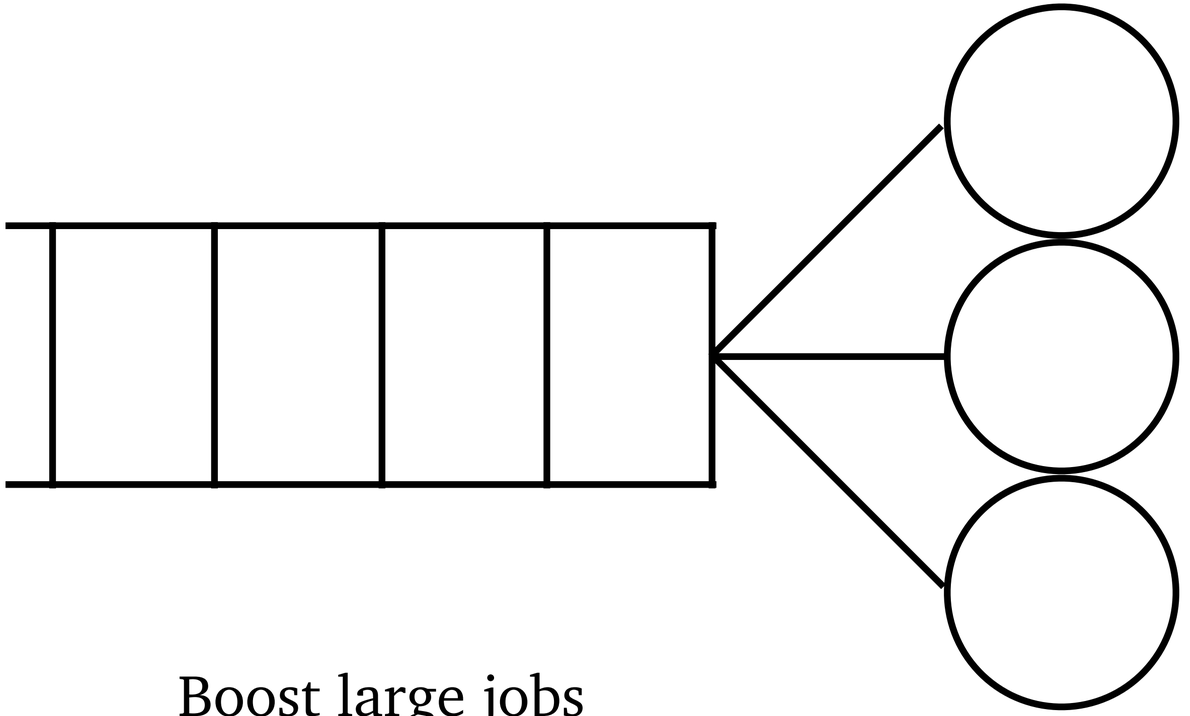


Boost large jobs

Why might SizeBoost make sense at low load?

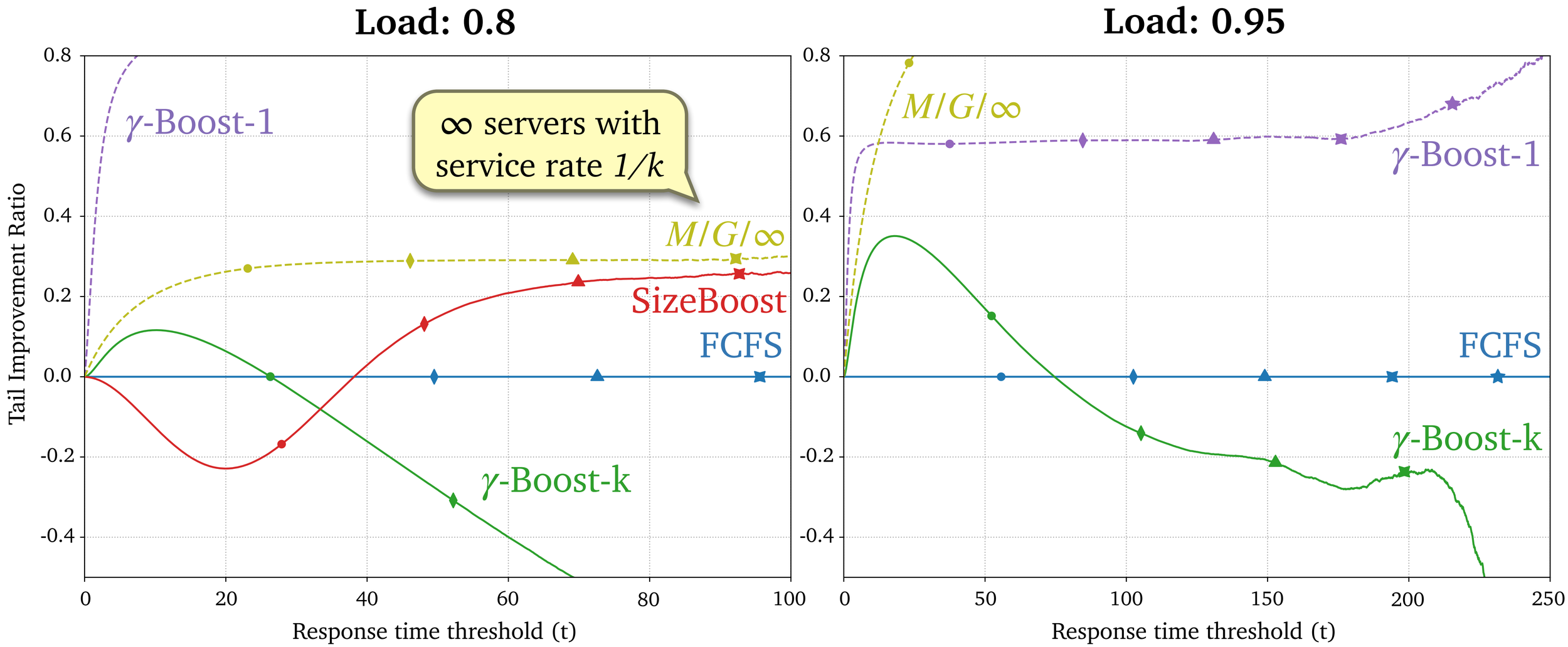


Boost small jobs



Boost large jobs

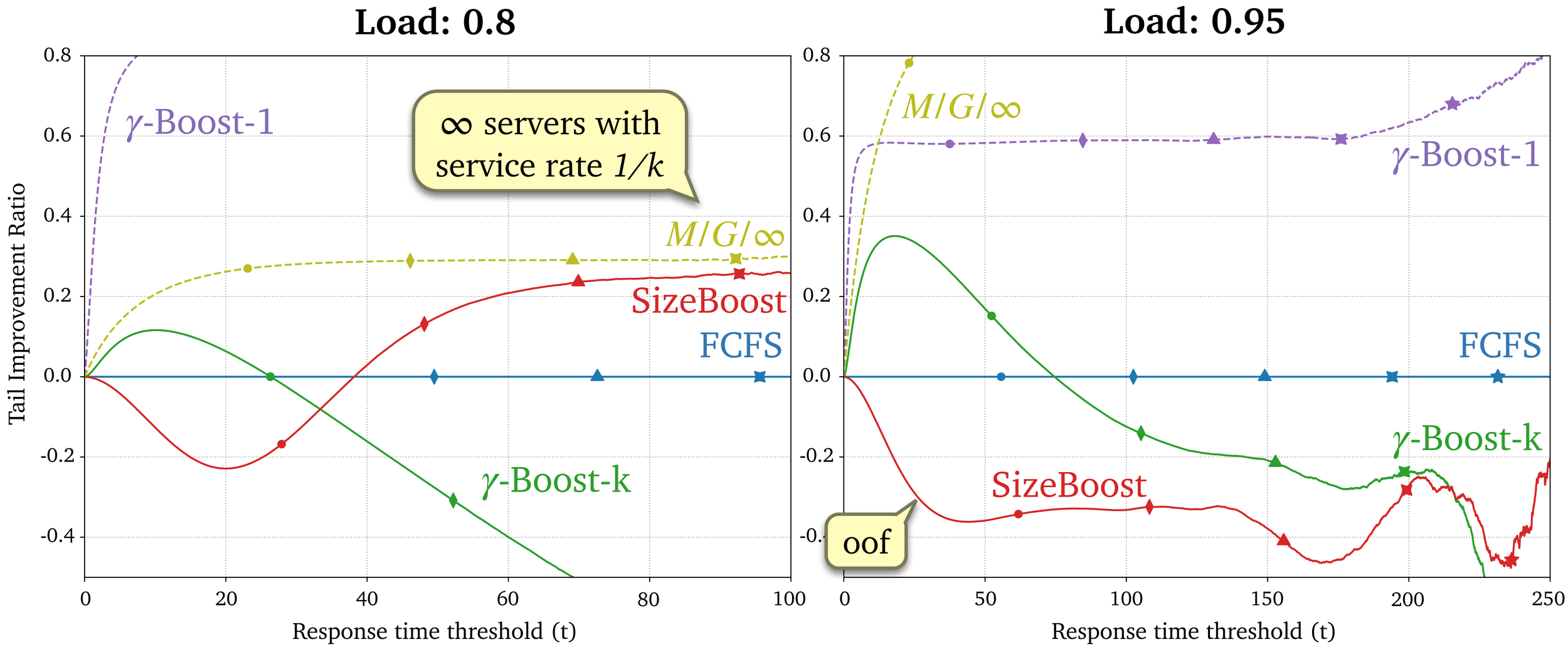
γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

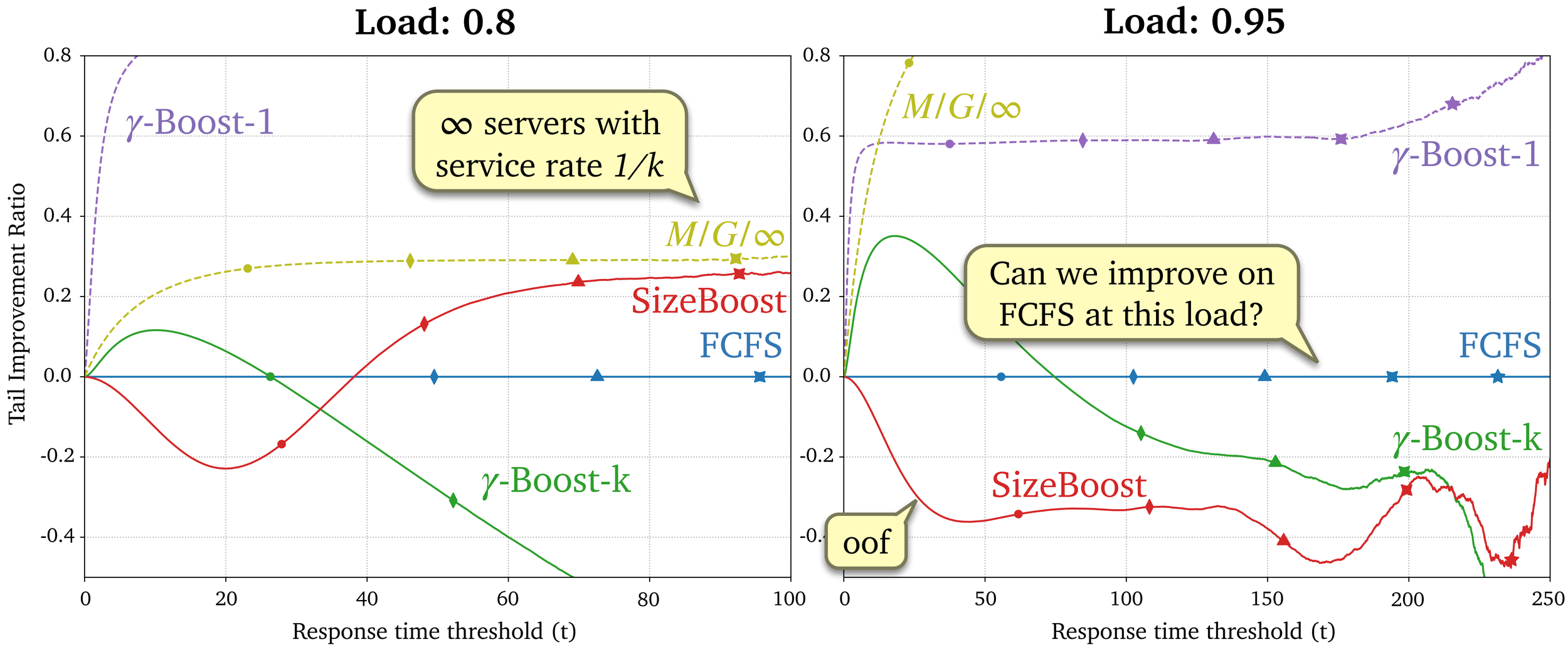
γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

• 90th percentile ♦ 99th percentile ▲ 99.9th percentile ✖ 99.99th percentile ★ 99.999th percentile

γ-Boost performance in lighter traffic



10 servers,
Exp(1) job sizes

90th percentile · 99th percentile · 99.9th percentile · 99.99th percentile · 99.999th percentile

Where does γ-Boost come from?

Where does γ-Boost come from?

Queueing Problem

$$\text{minimize} \quad C_\pi = \lim_{t\to\infty} e^{\gamma t}\,\mathbf{P}[T_\pi > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T_\pi}]$$

(by final value theorem for Laplace transforms)

Where does γ-Boost come from?

Queueing Problem

$$\text{minimize} \quad C_\pi = \lim_{t\to\infty} e^{\gamma t}\,\mathbf{P}[T_\pi > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T_\pi}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T_\pi}]\text{”}}$$

(by final value theorem for Laplace transforms)

Where does γ-Boost come from?

Queueing Problem

$$\text{minimize } C_\pi = \lim_{t\to\infty} e^{\gamma t}\,\mathbf{P}[T_\pi > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T_\pi}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T_\pi}]\text{”}}$$

(by final value theorem for Laplace transforms)

Batch Problem

$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i}$$

minimized by γ-Boost in M/G/1

Where does γ-Boost come from?

Queueing Problem

$$\text{minimize} \quad C_\pi = \lim_{t\to\infty} e^{\gamma t}\,\mathbf{P}[T_\pi > t] = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T_\pi}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T_\pi}]\text{”}}$$

(by final value theorem for Laplace transforms)

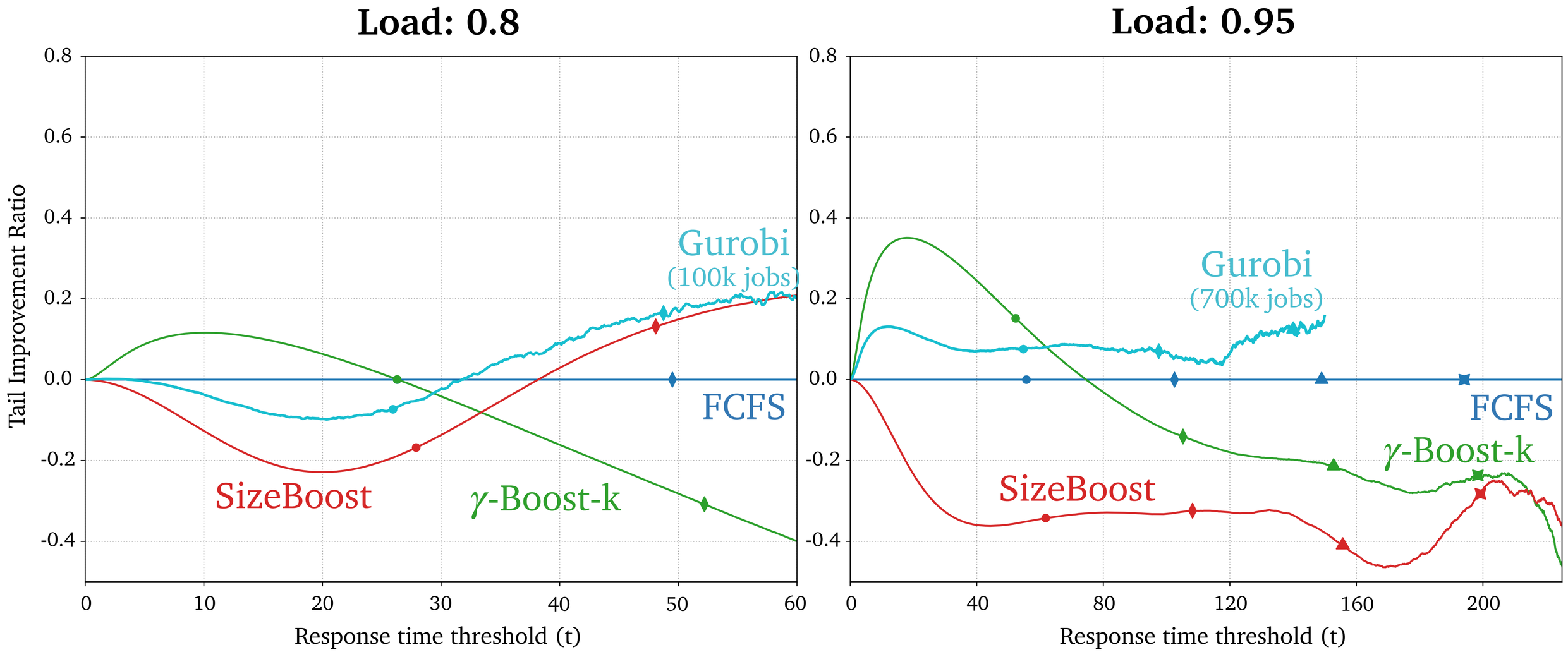
Batch Problem

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i}$$

minimized by γ-Boost in M/G/1

Numerically optimize $\frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i}$ in M/G/k after each arrival.

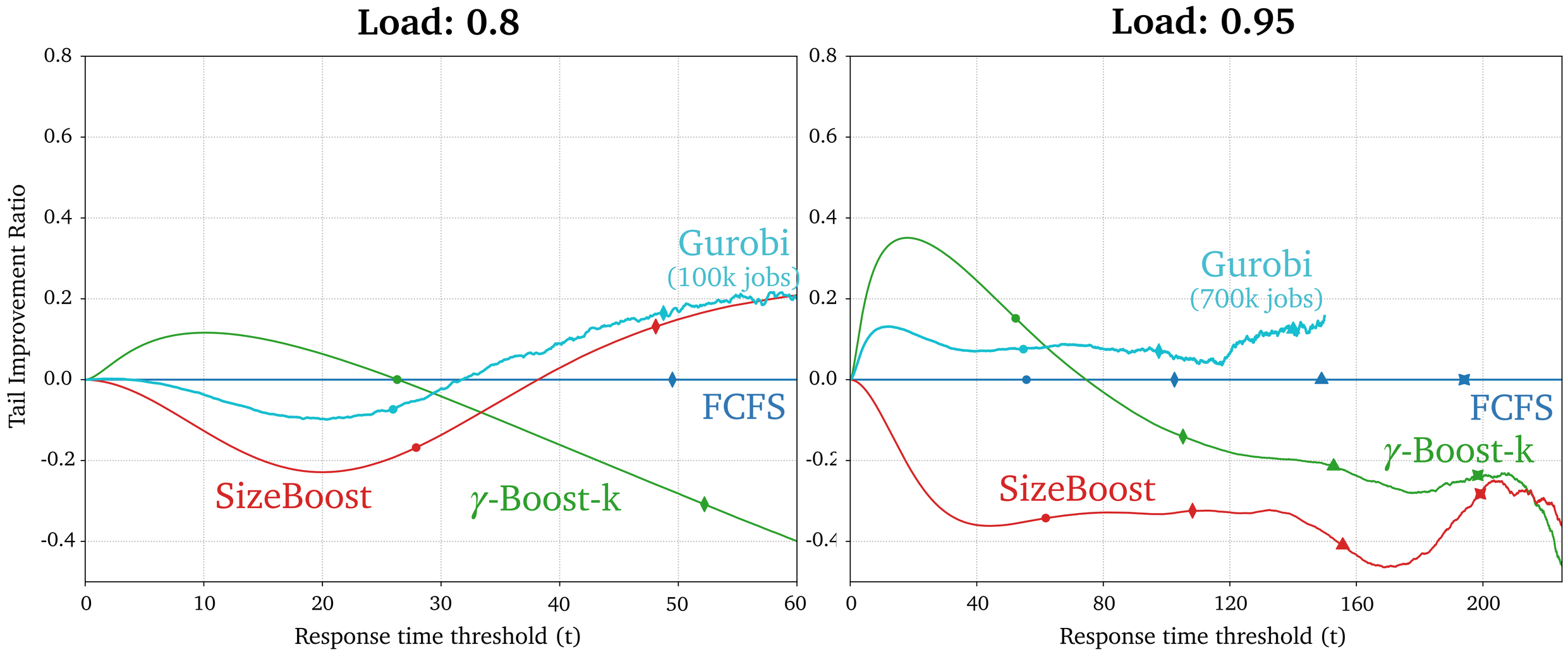
Numerically optimizing the batch problem in the M/G/k



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

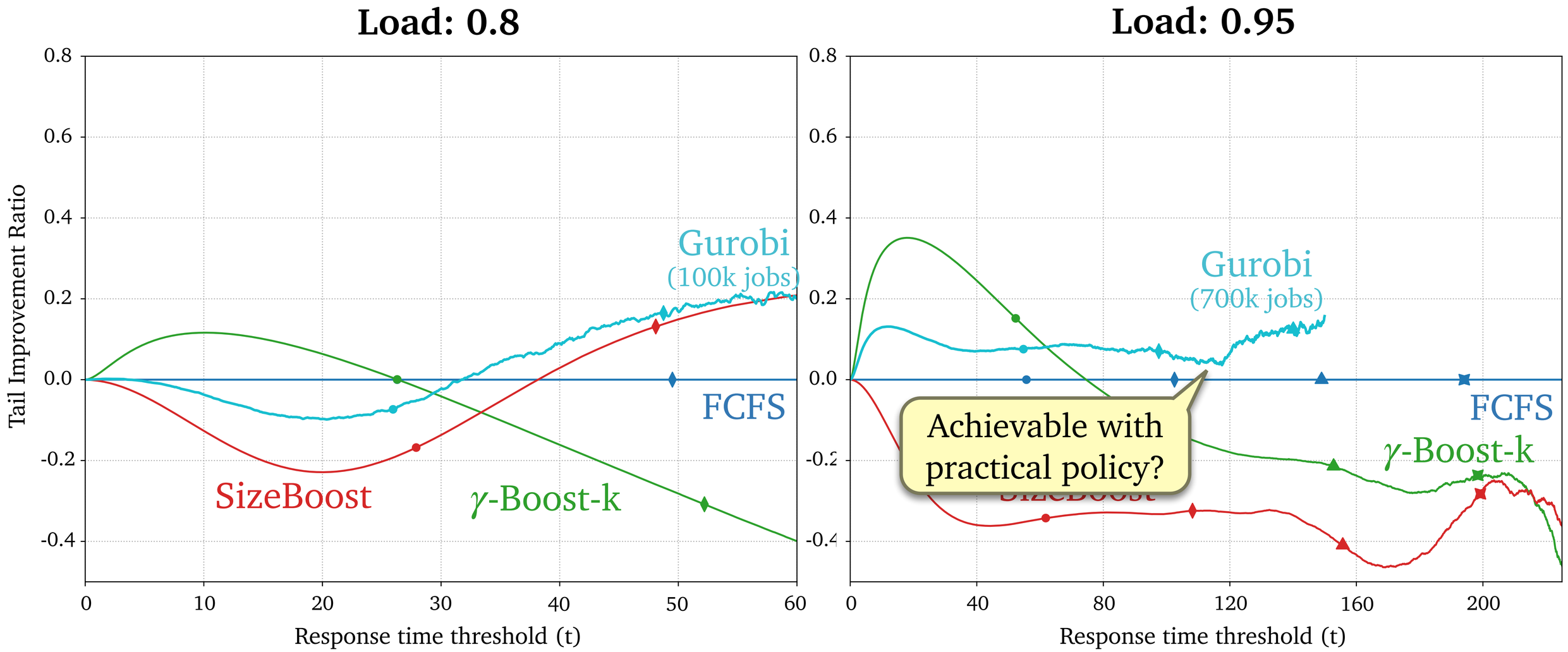
One week before submission…



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

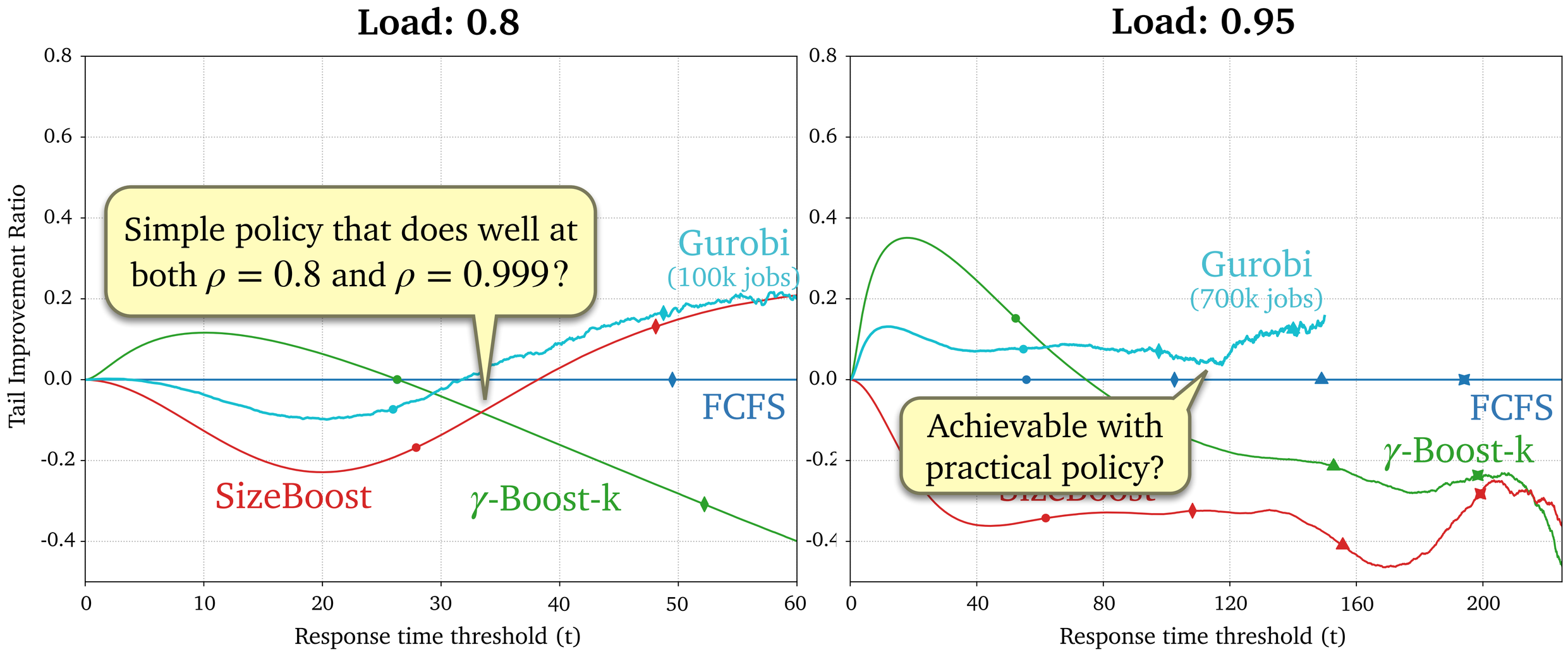
One week before submission…



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

One week before submission…



10 servers,
Exp(1) job sizes

90th percentile | 99th percentile | 99.9th percentile | 99.99th percentile | 99.999th percentile

Two days before submission…

Two days before submission…

Dr. Ralph Scully
(Professor at Harvard Med)

Two days before submission…

Dr. Ralph Scully
(Professor at Harvard Med)

a.k.a. Ziv's dad

Two days before submission…

Is there some way to combine γ-Boost and SizeBoost?

Dr. Ralph Scully

(Professor at Harvard Med)

a.k.a. Ziv's dad

Two days before submission…

Is there some way to combine γ-Boost and SizeBoost?

Dr. Ralph Scully
(Professor at Harvard Med)

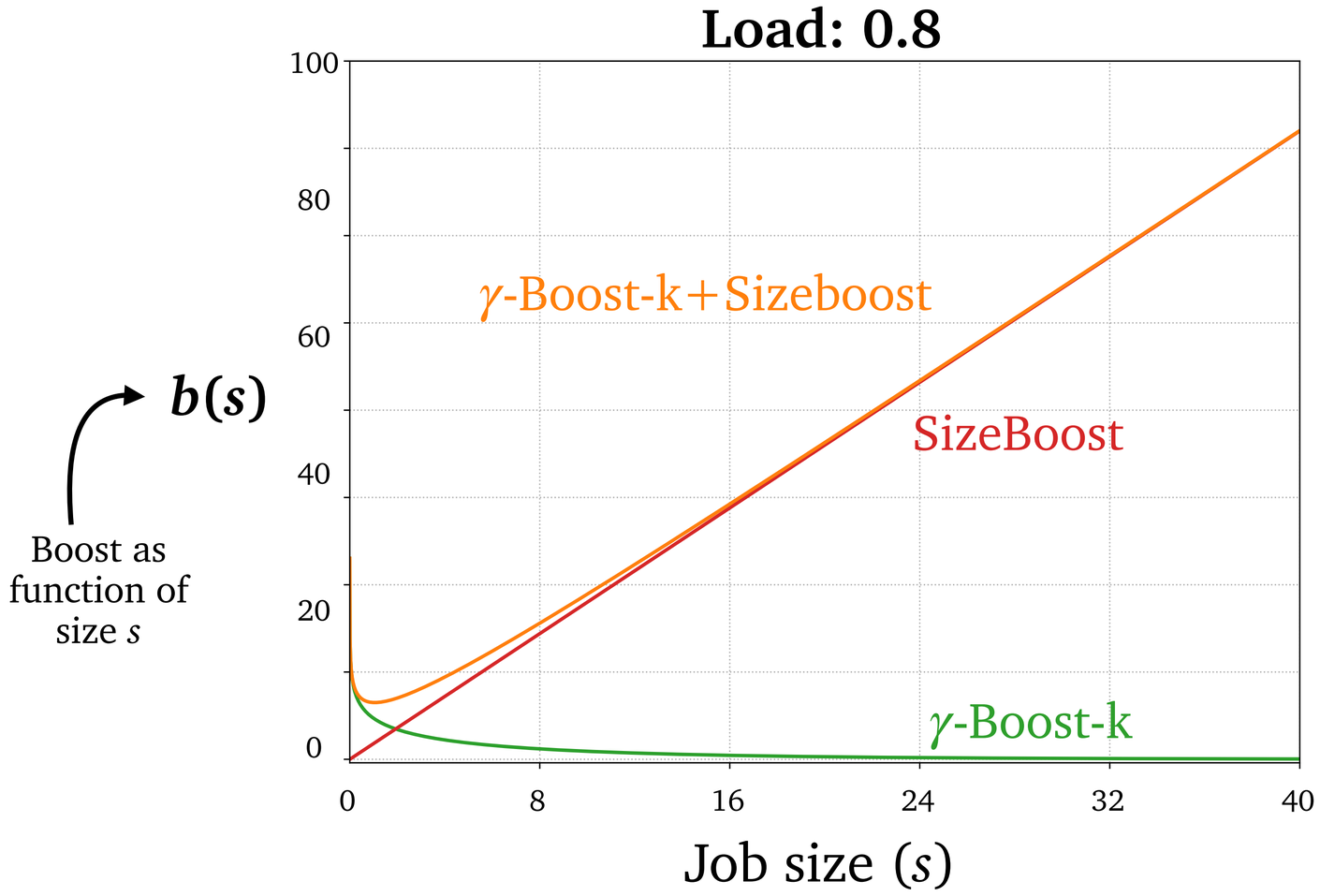
a.k.a. Ziv's dad

Add the boosts from both policies together!

Why does adding the boosts make any sense?

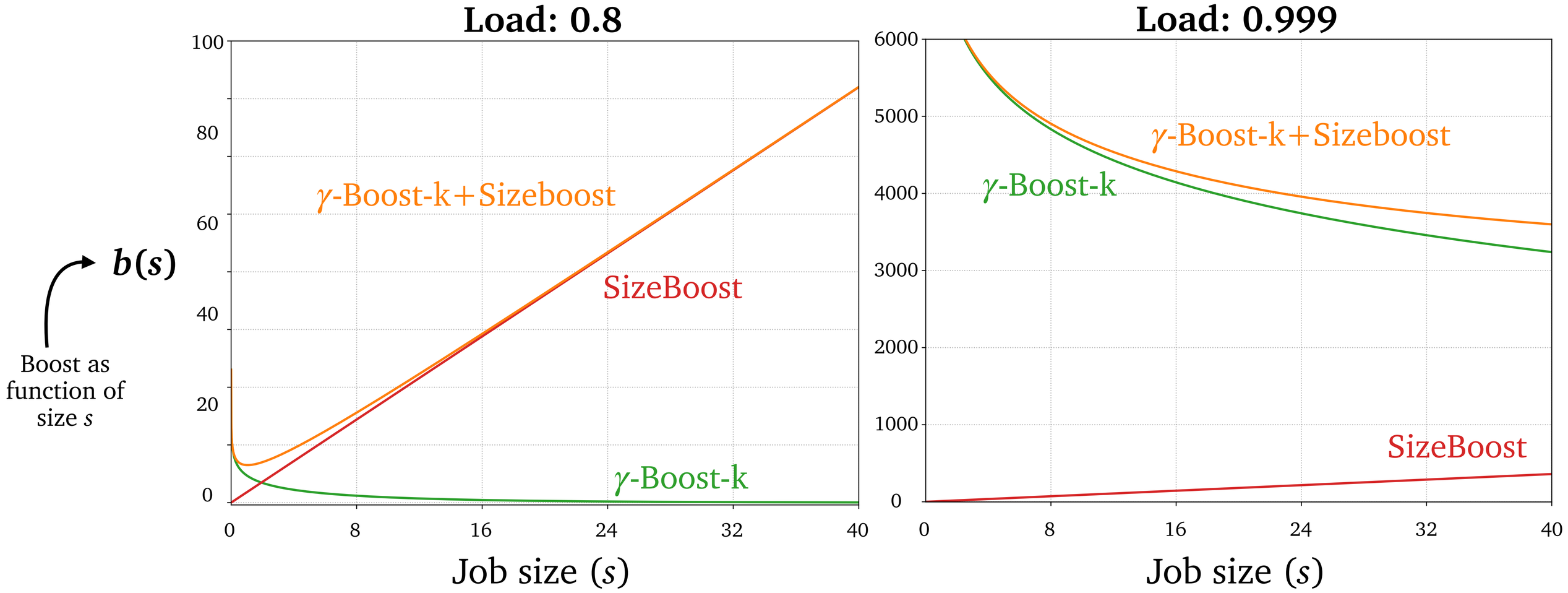
10 servers,
Exp(1) job sizes

Why does adding the boosts make any sense?



10 servers,
Exp(1) job sizes

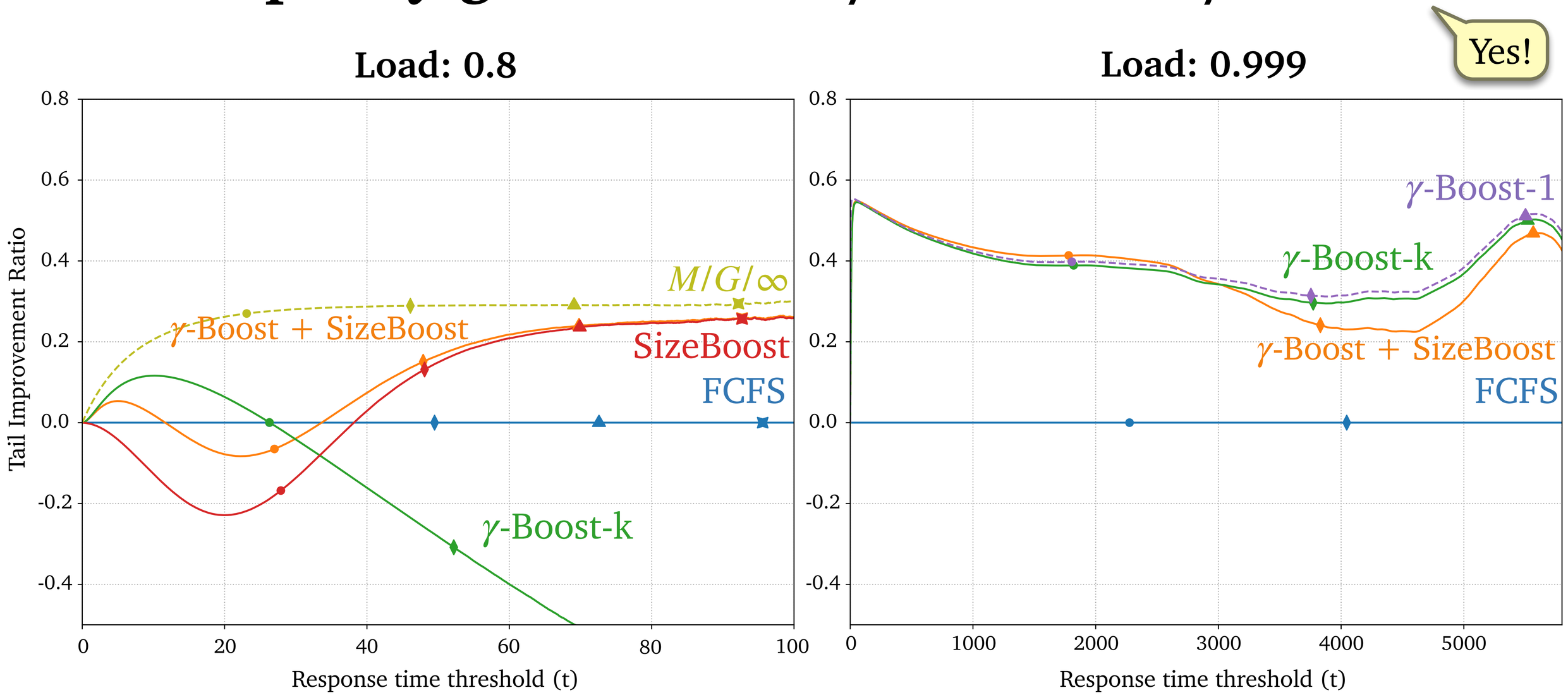
Why does adding the boosts make any sense?



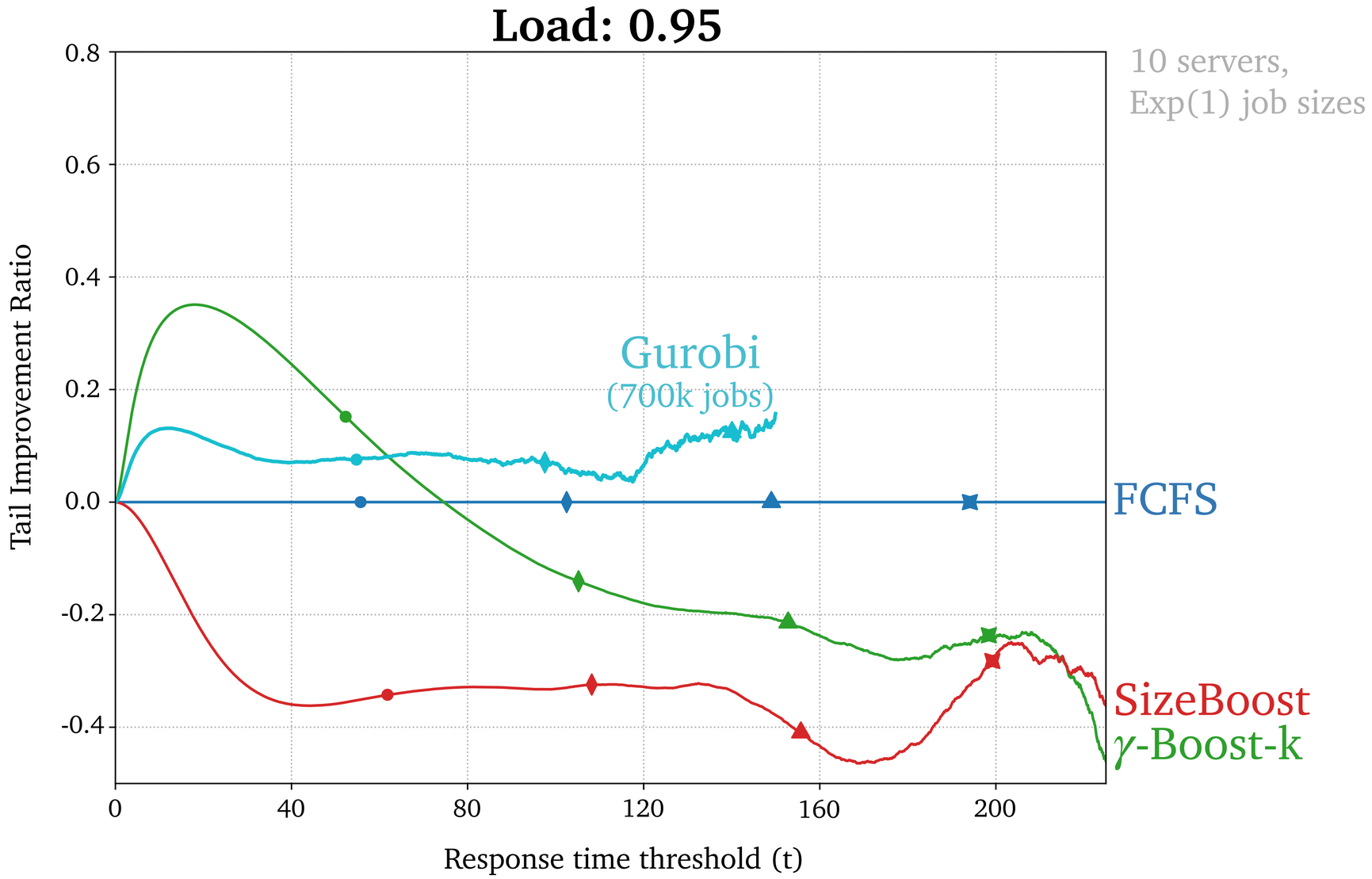
10 servers,
Exp(1) job sizes

Is this policy good at both $\rho = 0.8$ and $\rho = 0.999$?

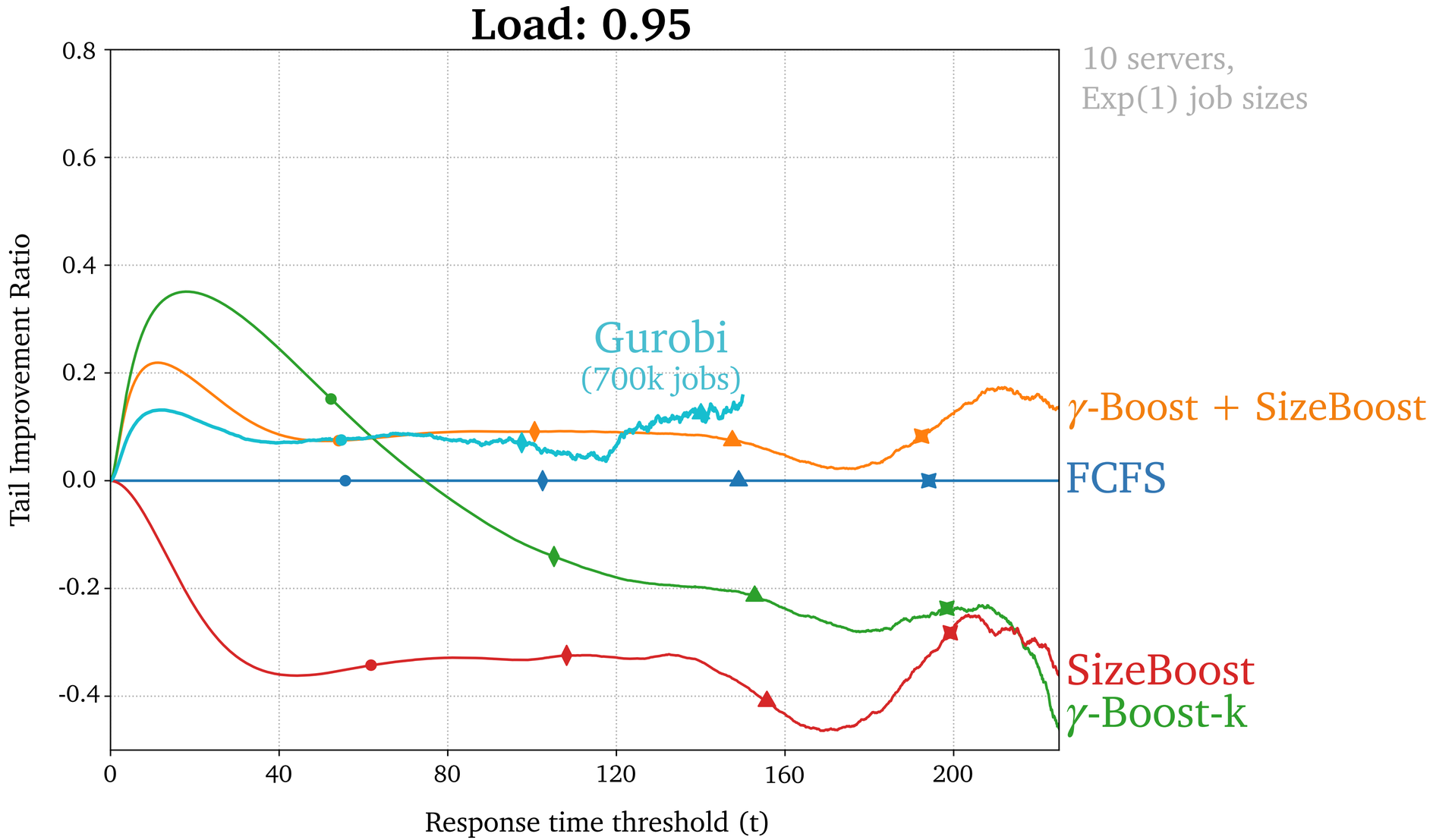
Is this policy good at both $\rho = 0.8$ and $\rho = 0.999$?



How does this policy perform at $\rho = 0.95$?

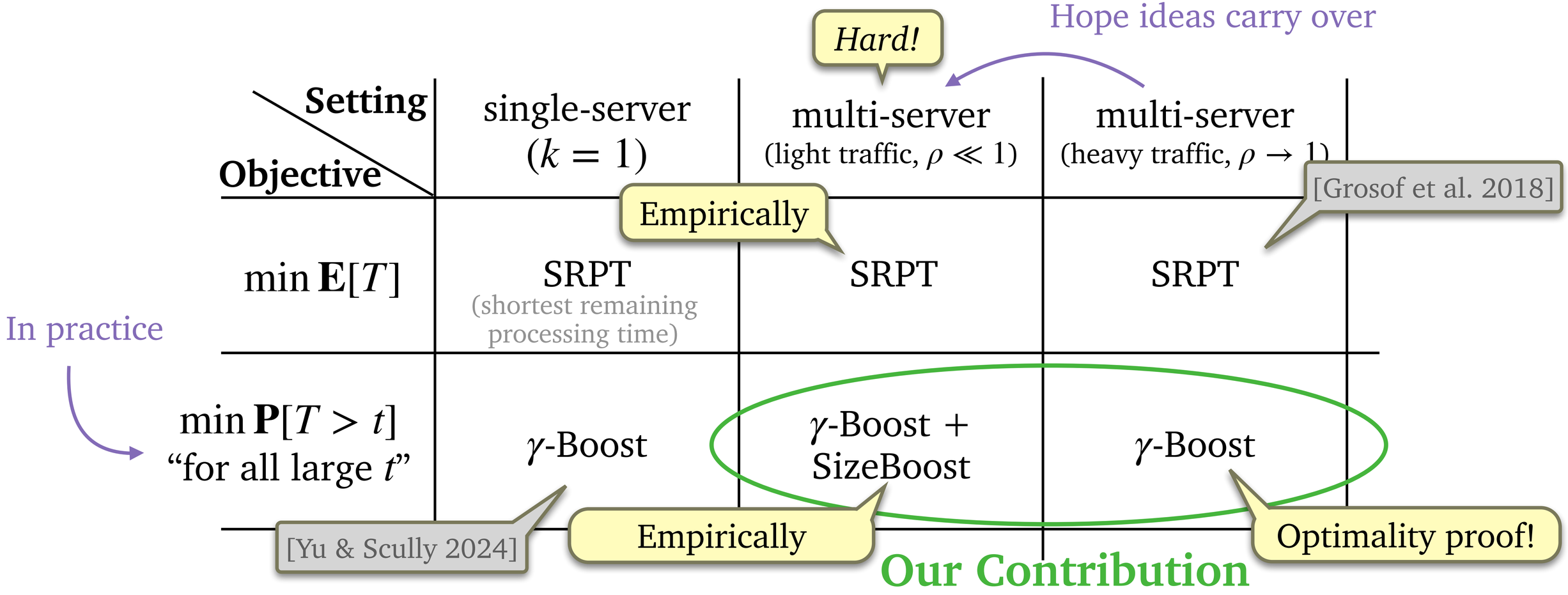


How does this policy perform at $\rho = 0.95$?

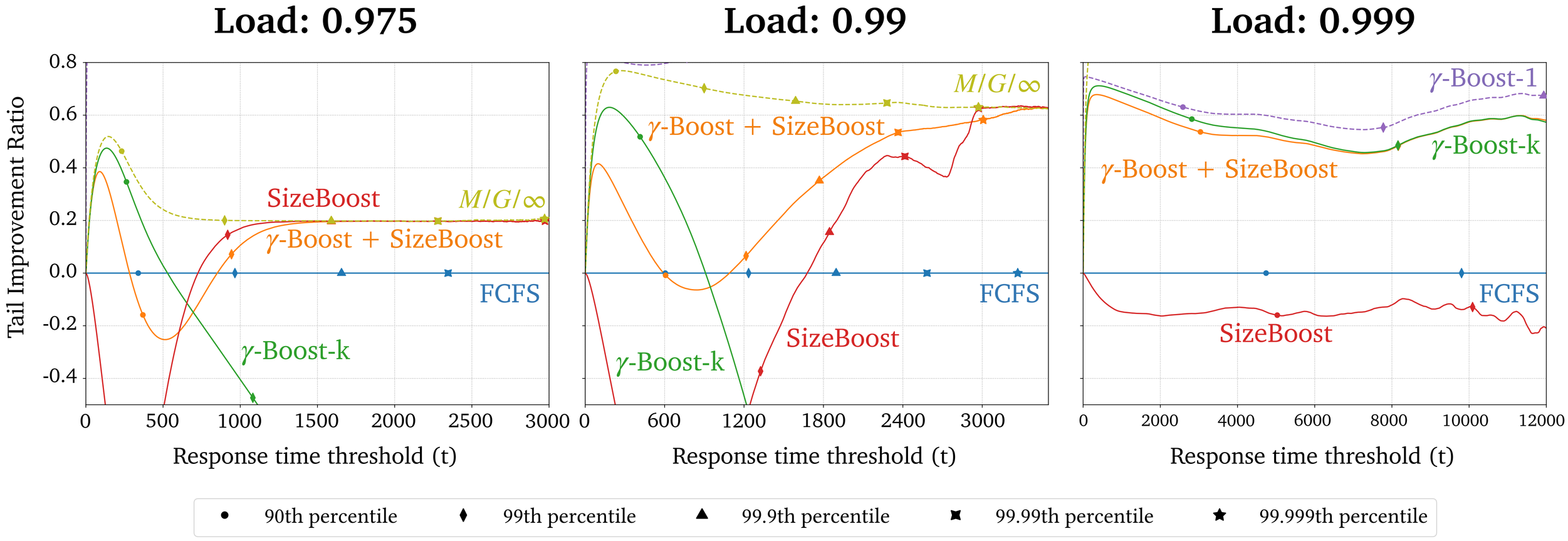


How does this [image] m at $\rho = 0.95$?

Summary


Hard!
Hope ideas carry over
Setting
Objective
single-server
$(k = 1)$
multi-server
(light traffic, $\rho \ll 1$)
multi-server
(heavy traffic, $\rho \to 1$)
Empirically
[Grosof et al. 2018]
$\min \mathbf{E}[T]$
SRPT
(shortest remaining processing time)
SRPT
SRPT
In practice
$\min \mathbf{P}[T > t]$
"for all large t"
γ-Boost
γ-Boost + SizeBoost
γ-Boost
[Yu & Scully 2024]
Empirically
Optimality proof!
Our Contribution

100 servers



100 servers,
Hyperexponential job sizes