Multiserver-Job Systems

Isaac Grosof

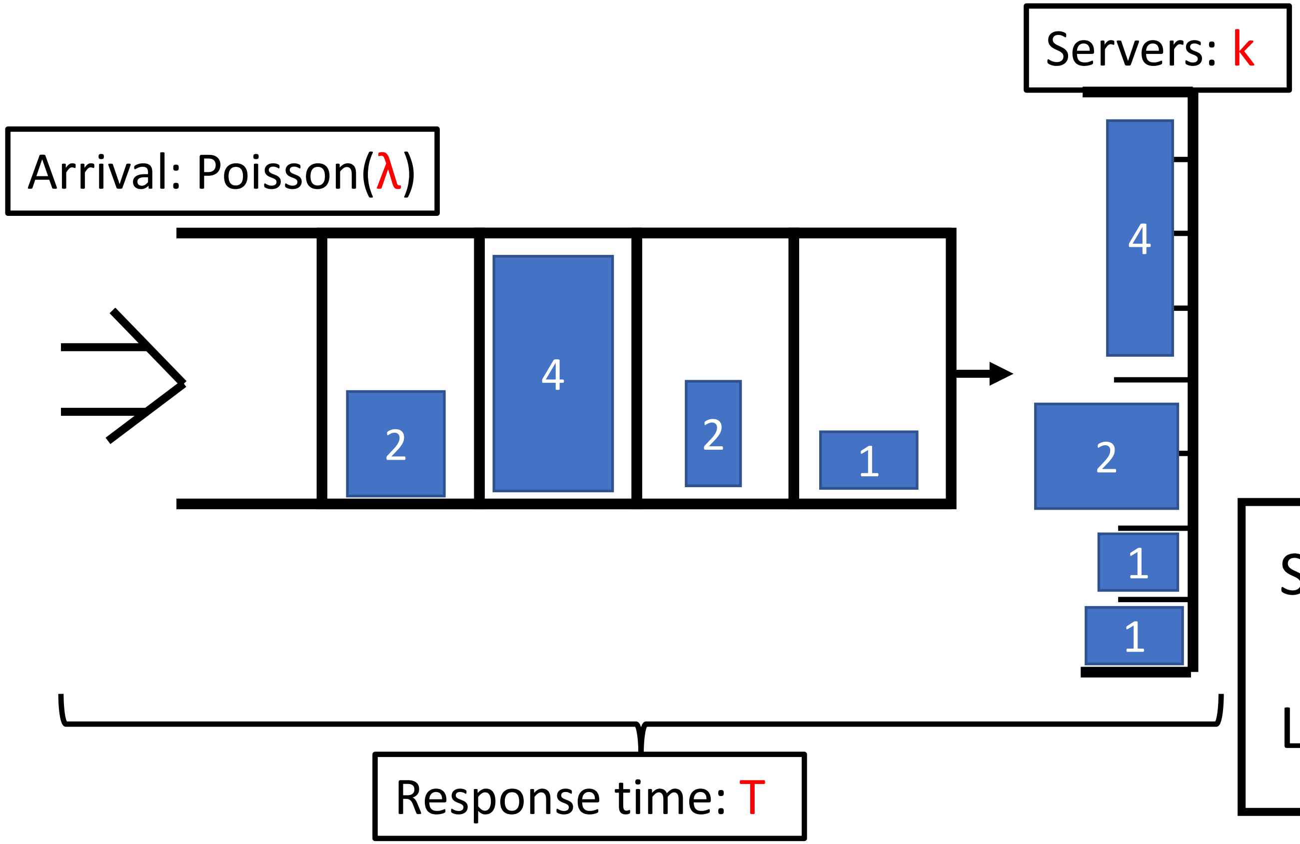
Mor Harchol-Balter

Alan Scheller-Wolf

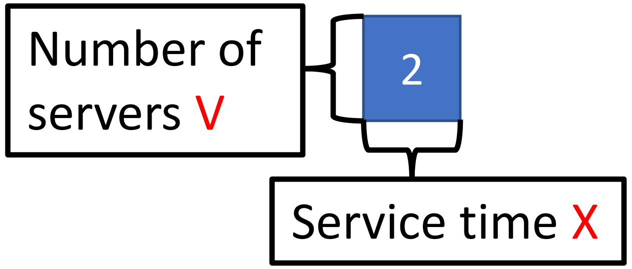
Carnegie Mellon University

INFORMS 2021

Multiserver-Job Model



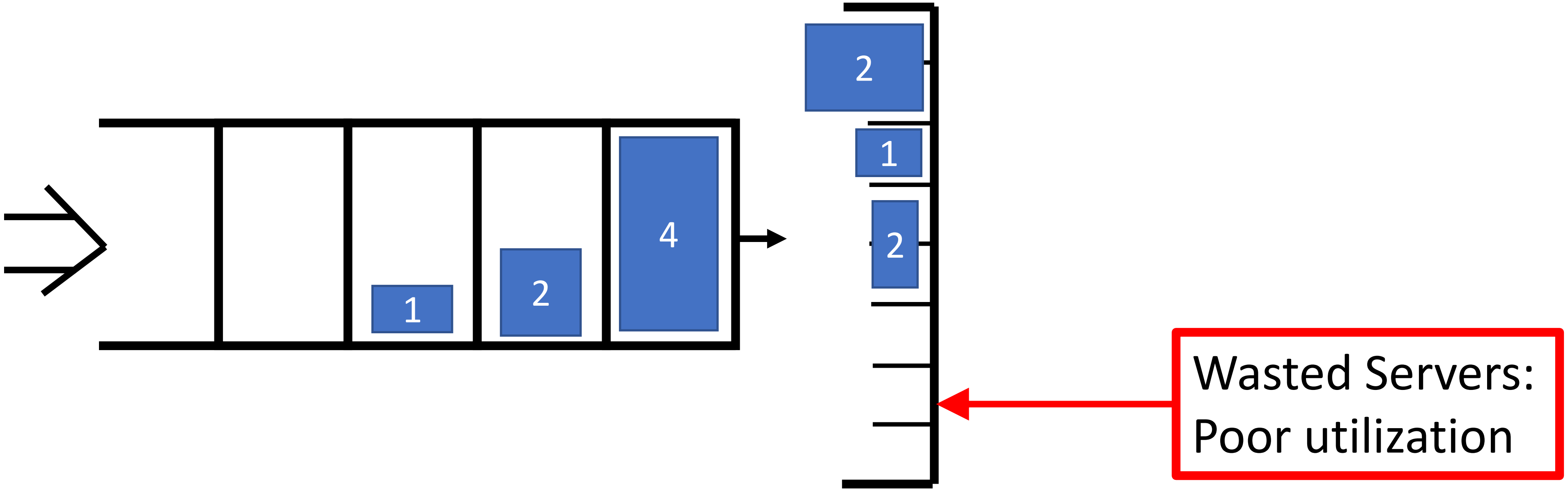
Two job requirements, i.i.d.:



Size S = VX/k

Load/Utilization $\rho = \frac{\lambda E[VX]}{k} = \lambda E[S]$

Default: FCFS Service



FCFS only stable for $\rho < \rho^* < 1$

Goal: Better Scheduling

Want a scheduling policy with:

1. Full stability region: stable for any $\rho < 1$
2. Simple implementation
3. Mean response time E[T] analysis

Challenge: No response time analysis known for **any** multiserver-job policies.

Challenge: Full stability region not possible for some V.

Results: “The Finite-Skip Method for Multiserver Analysis”

Introduce ServerFilling Policy

Bound mean response time E[T]

First analytic bounds on E[T] for any Multiserver-Job Policy

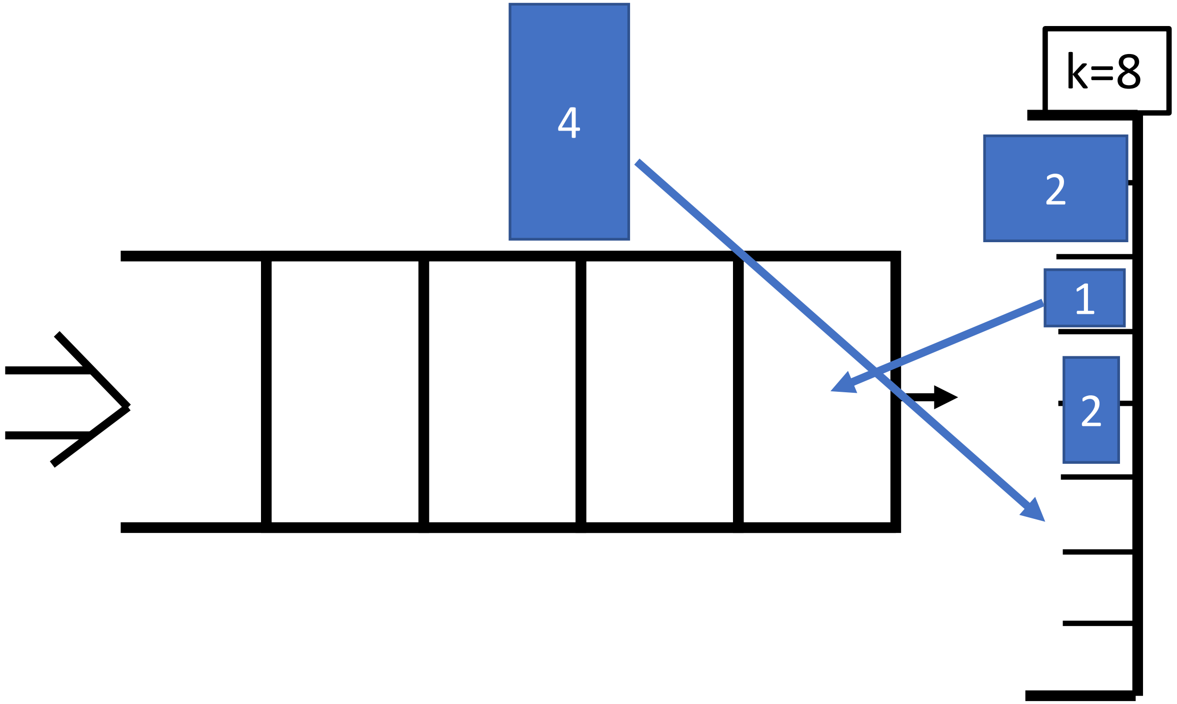
Tight heavy-traffic characterization: additively-tight bounds

First simple policy with full stability region

Policy: ServerFilling

Key idea: Fill all servers whenever possible, using preemption.

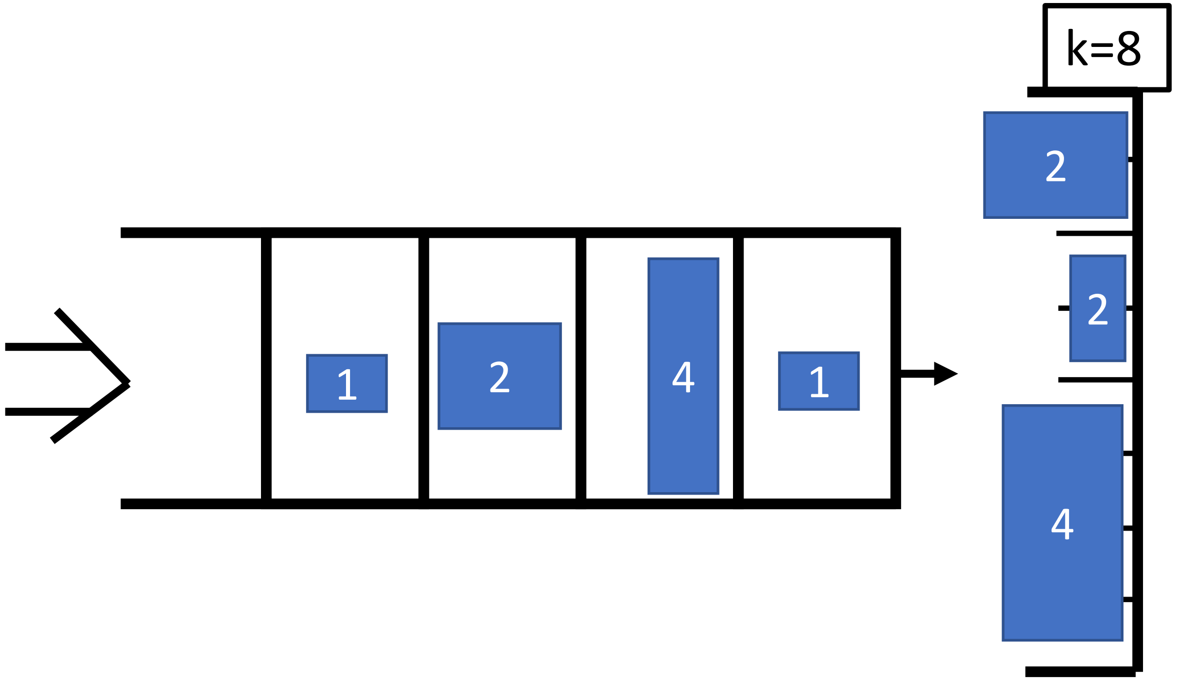
Setting: Server requirement V always power of 2, k power of 2.



Policy: ServerFilling

Key idea: Fill all servers whenever possible, using preemption.

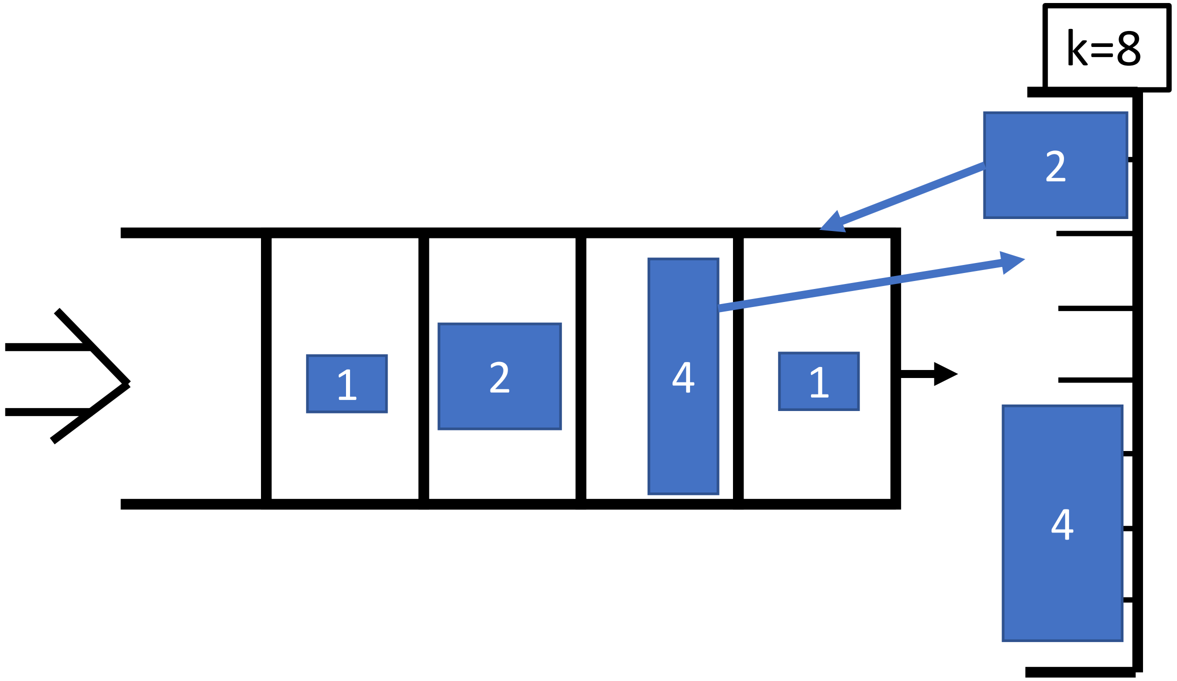
Setting: Server requirement V always power of 2, k power of 2.



Policy: ServerFilling

Key idea: Fill all servers whenever possible, using preemption.

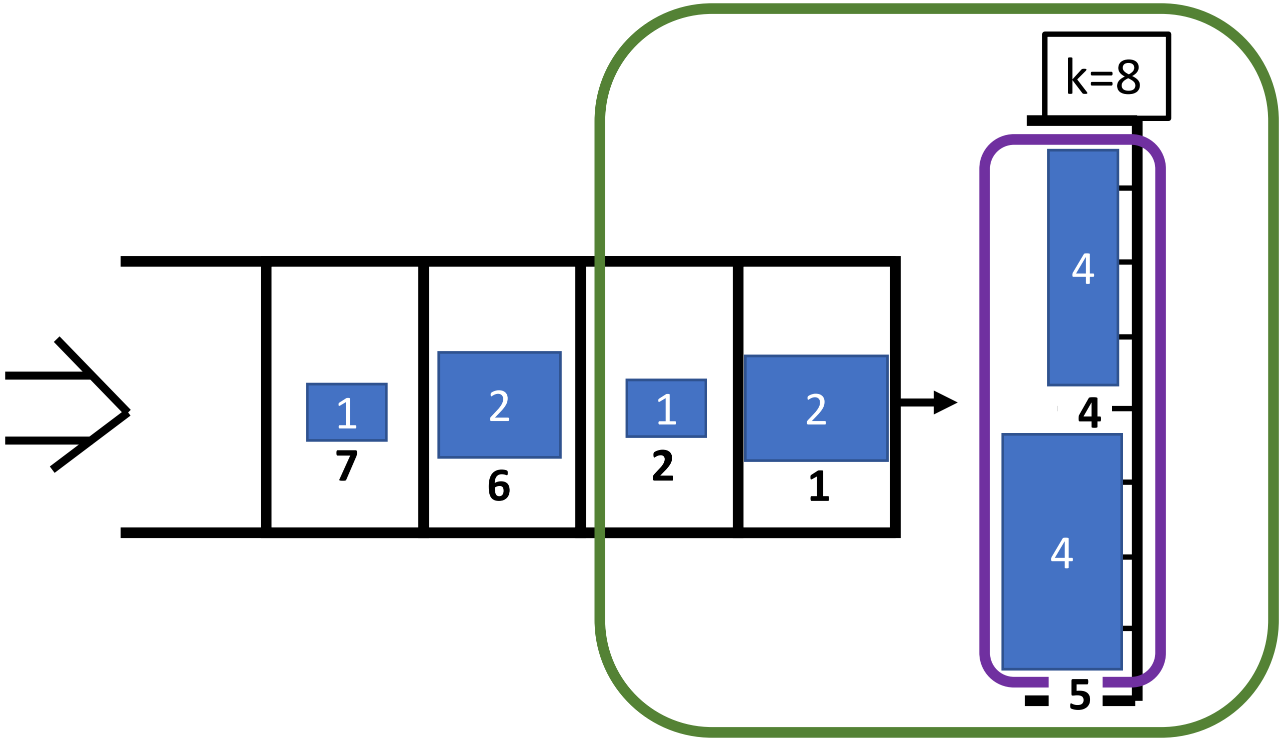
Setting: Server requirement V always power of 2, k power of 2.



Policy: ServerFilling

Key idea: Fill all servers whenever possible, using preemption.

Setting: Server requirement V always power of 2, k power of 2.

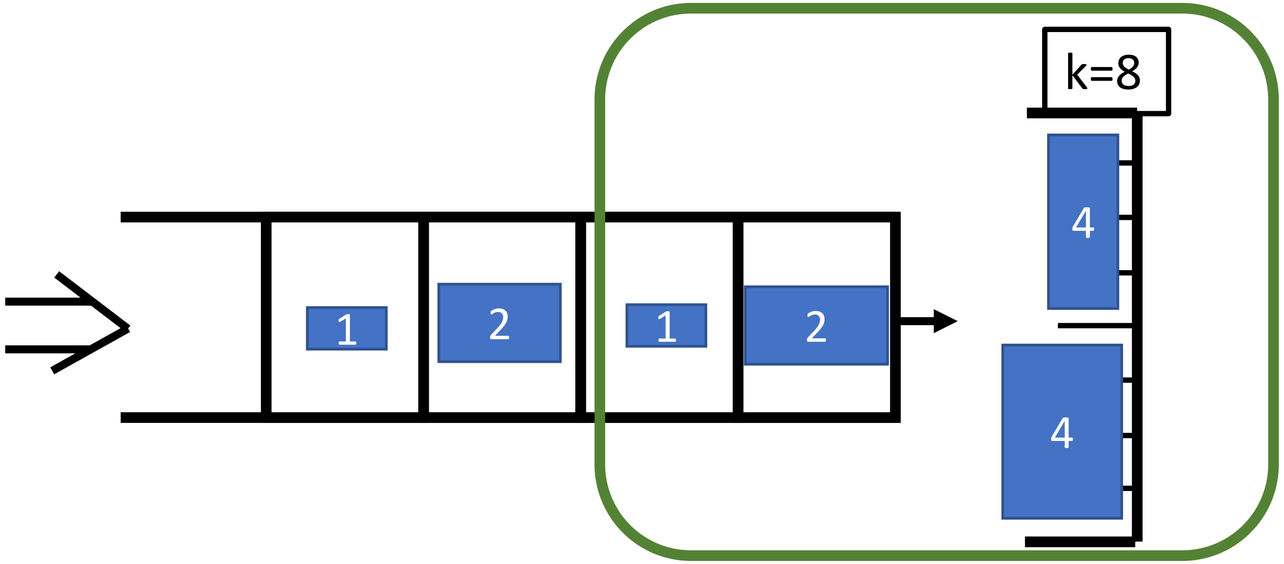


ServerFilling Details:
Looking in arrival order, find minimal set of old jobs requiring $\geq k$ servers.
Always exists subset requiring exactly k.
Serve that subset, break ties FCFS.

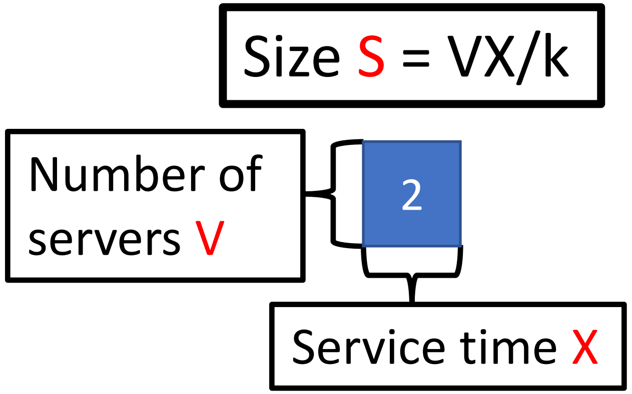
Response Time Analysis

Key insights:

1. Jobs leave in near-FCFS order: Completions in old-jobs set.
2. "Work conserving": Fills all k servers whenever a job is in queue.



Response Time Analysis



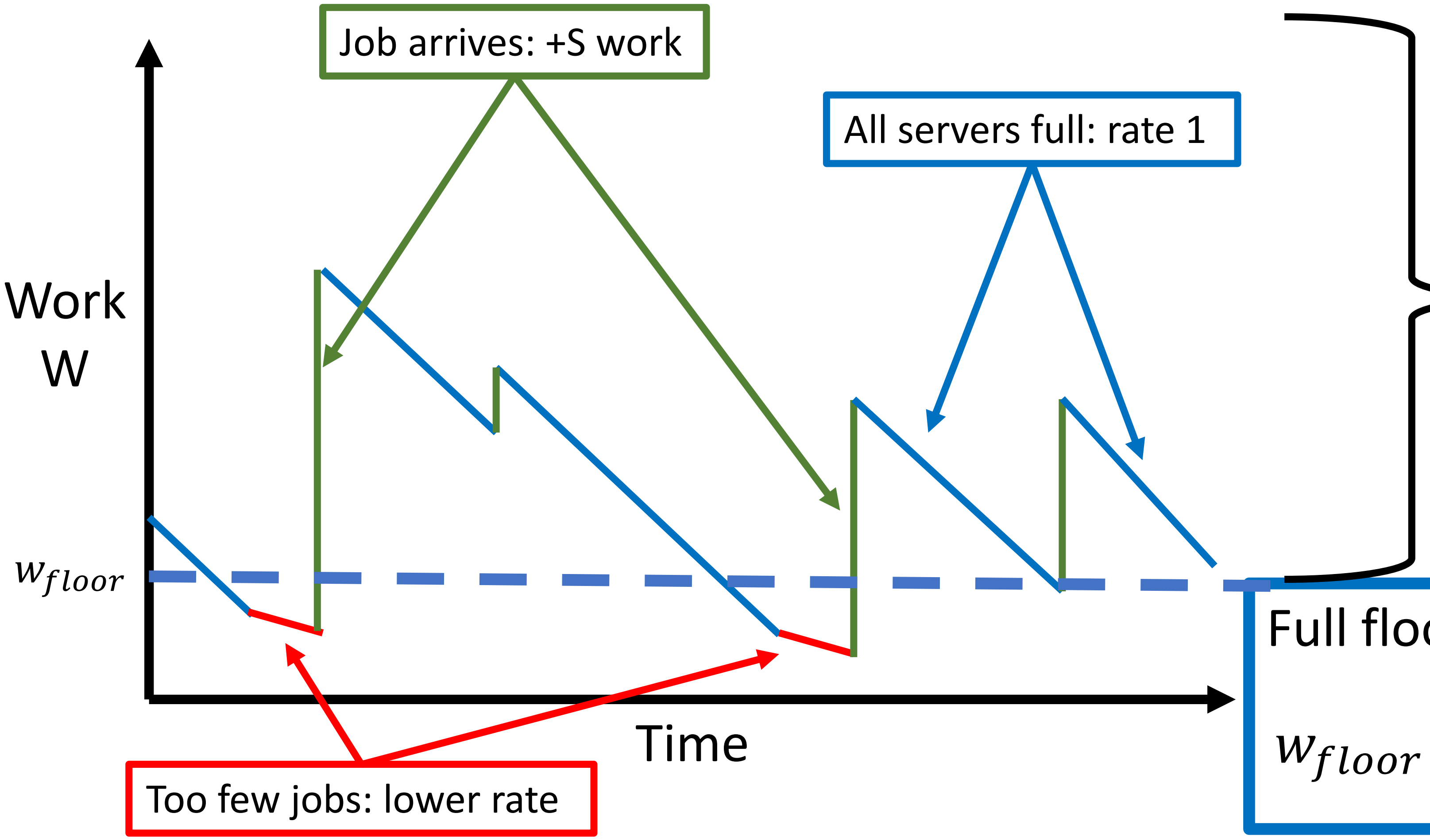
Work-based analysis:

Work W is the total remaining size of all jobs.

If all servers are full, W falls at rate 1.

1. Bound E[T] relative to E[W] using near-FCFS property.
2. Bound E[W] using work-conserving property.

Work Analysis



Identical to M/G/1!

Full floor w_{floor}: If $W > w_{floor}$, rate 1.

$$w_{floor} \leq \left(1 - \frac{1}{k}\right)(\max \mathrm{E}[\text{rem time}])$$

Bound work relative to M/G/1

We bound work:

$$\frac{\lambda E[S^2]}{2(1-\rho)} \leq E\left[W^{ServerFilling}\right] \leq \frac{\lambda E[S^2]}{2(1-\rho)} + w_{floor}$$

Recall that $\mathrm{E}\left[\mathrm{W}^{\mathrm{M/G/1}}\right] = \frac{\lambda E\left[S^2\right]}{2(1-\rho)}$.

Intuitively: ServerFilling has essentially the same work as M/G/1.

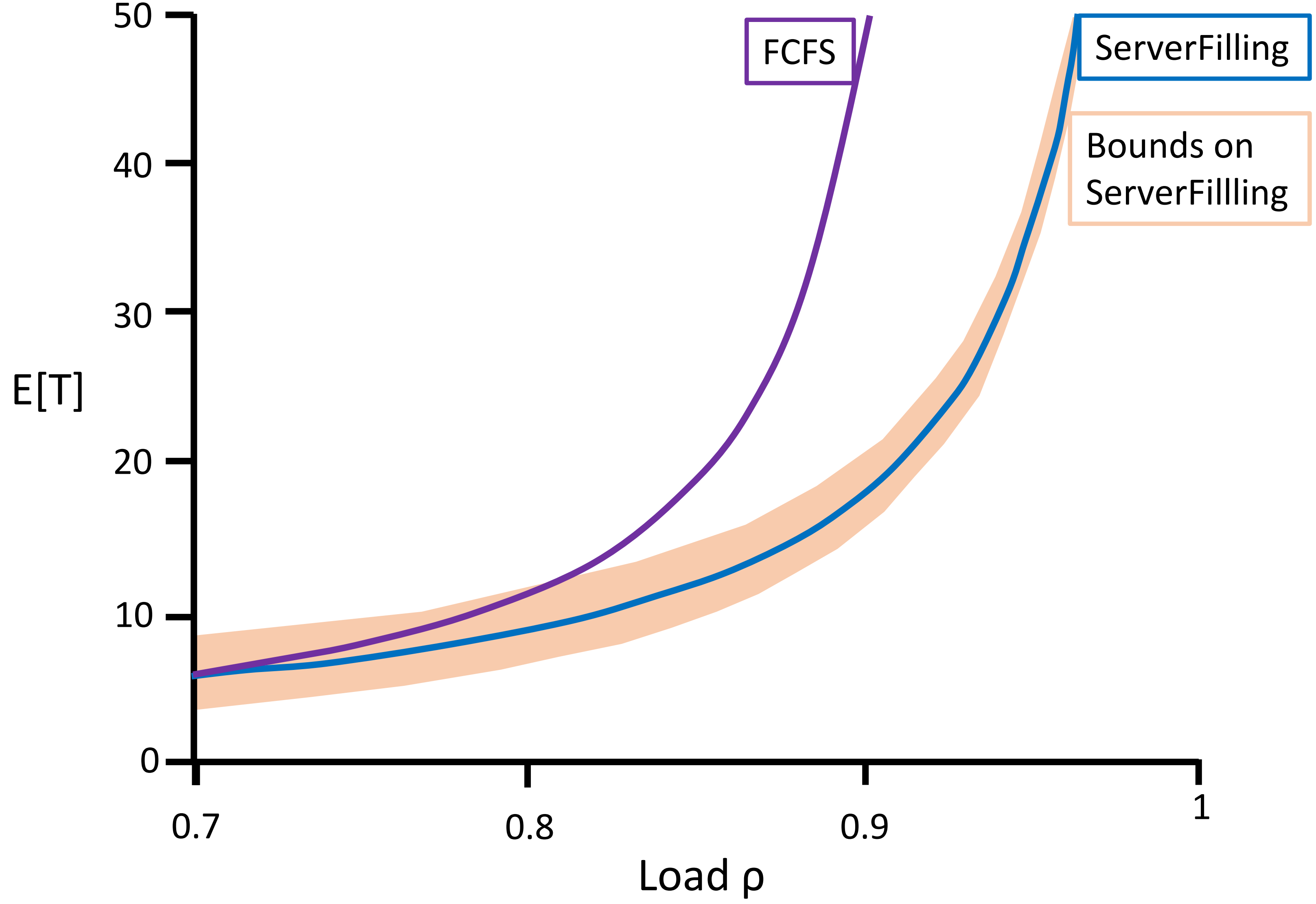
Response time bounds

$$E[T] \leq \frac{\lambda E[S^2]}{2(1-\rho)} + w_{floor} + kE[S]$$
$$E[T] \geq \frac{\lambda E[S^2]}{2(1-\rho)} - w_{floor} + E[S]$$

First bounds on mean response time for any Multiserver-Job policy.

Additively tight bounds.

Plot of Simulation and Bounds



Setting: k=4 servers,
(V, X)=
Prob 1/3: (1, Exp(5))
Prob 1/3: (2, Exp(2))
Prob 1/3: (4, Exp(1/2))

Future Work

- Analyze policies which are “near-SRPT” rather than “near-FCFS”?
- Analyze non-work-conserving policies, such as FCFS?
- Characterize tail behavior of ServerFilling?

Conclusion

New policy for multiserver-job scheduling: ServerFilling

First mean response time bounds on any multiserver-job policy

Tight in heavy-traffic: Additively-tight bounds

Key properties: near-FCFS, work-conserving