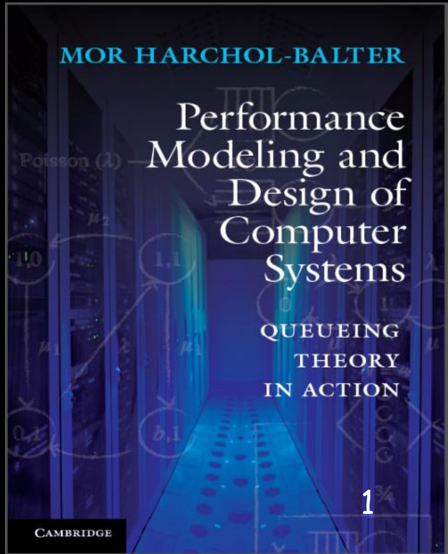
The most common queueing questions asked by computer systems practitioners

Mor Harchol-Balter
Ziv Scully

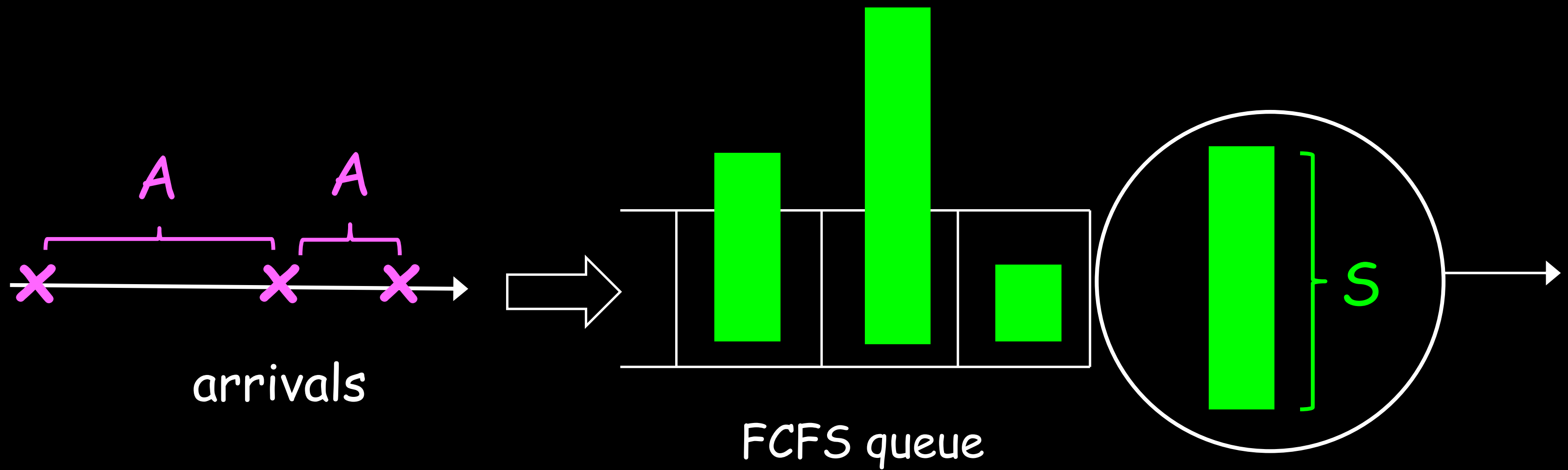
Computer Science Dept.
Carnegie Mellon University



Question 1:

"My system utilization is low, so why are job delays so high?"

Kingman's Approximation



A: interarrival time

S: job size (service time)

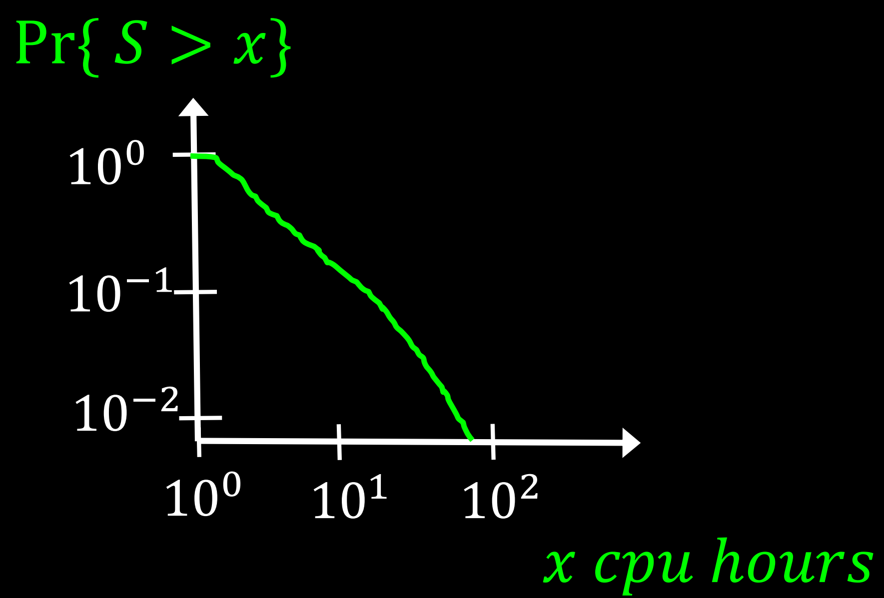
Utilization

$$\frac{Var(S)}{E[S]^2}$$

$$\boxed{E[Delay] \approx \frac{\rho}{1-\rho} \cdot \left(\frac{C_A^2 + C_S^2}{2}\right) \cdot E[S]}$$

Empirical Job Size Distribution

UNIX jobs. [Harchol-Balter, Downey – SIGMETRICS 1996]



$S = Job\ Size$

$S \sim BoundedPareto(\alpha \approx 1)$

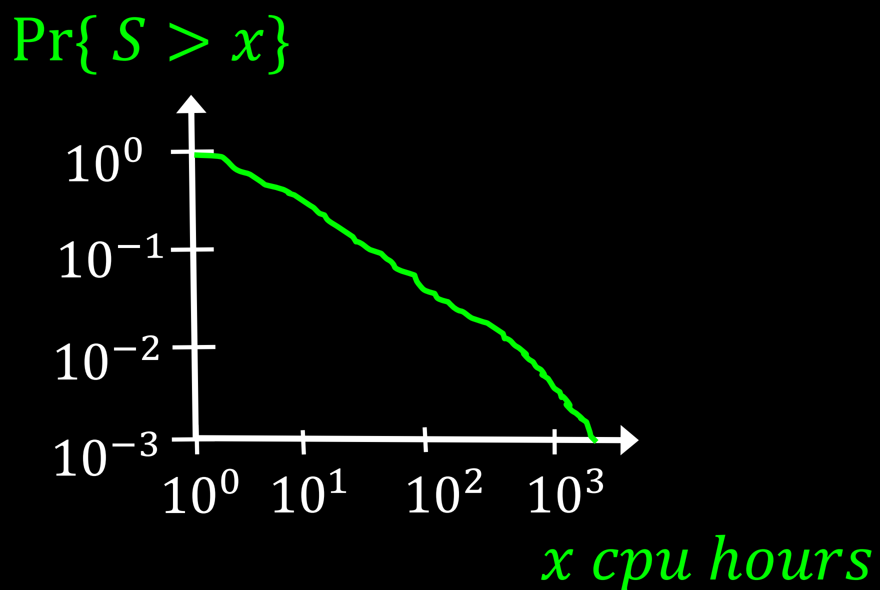
$\boxed{C_S^2 = 50}$

$Top\ 1\%\ of\ jobs \approx 50\%\ of\ load$

$$\boxed{E[Delay] \approx \frac{\rho}{1-\rho} \cdot \left(\frac{C_A^2 + C_S^2}{2}\right) \cdot E[S]}$$

Empirical Job Size Distribution

Borg Scheduler at Google [Tirmazi et al., EUROSYS 2020]



<u>$S = Job\ Size$</u>

$S \sim BoundedPareto(\alpha = 0.69)$

$\boxed{C_S^2 = 23{,}000}$

$Top\ 1\%\ of\ jobs \approx 99\%\ of\ load$

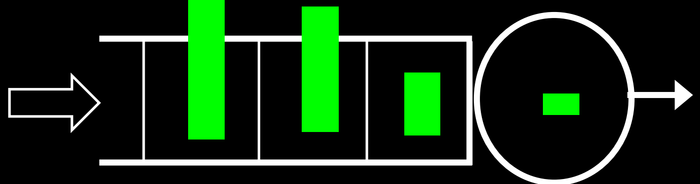
$$\boxed{E[Delay] \approx \frac{\rho}{1-\rho} \cdot \left(\frac{C_A^2 + C_S^2}{2}\right) \cdot E[S]}$$

Question 2: "How can I lower job delay?"

3 solutions:
All based on lowering the effect of job size variability

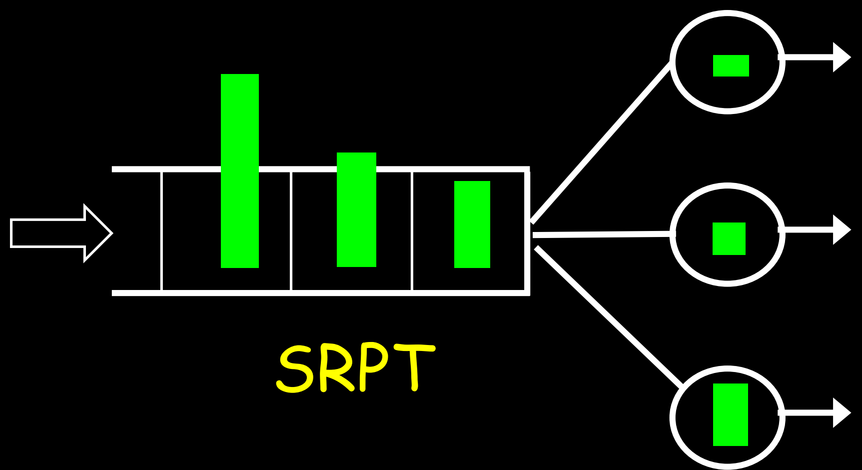
Solution 1: Schedule to favor smalls

SRPT = Shortest Remaining Processing Time



SRPT

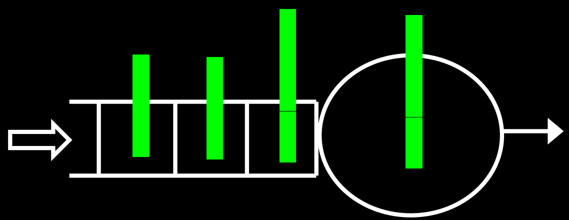
At all times run the job with shortest remaining time.



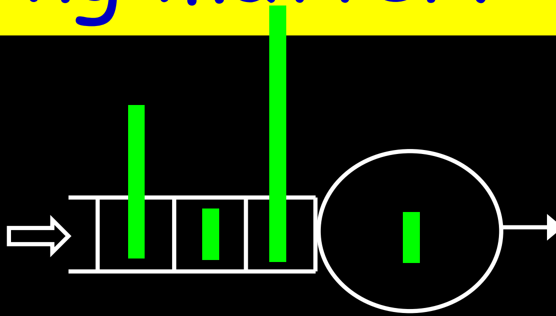
SRPT

At all times run the 3 jobs with shortest remaining times.

How much does scheduling matter?

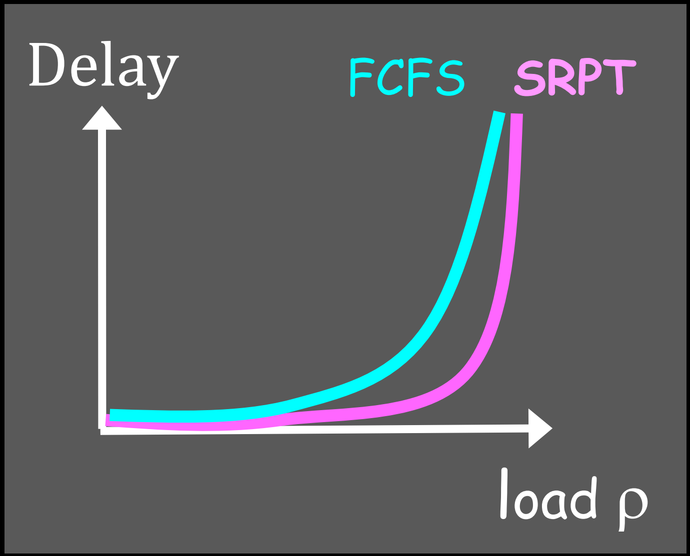


$C_S^2 = 1$

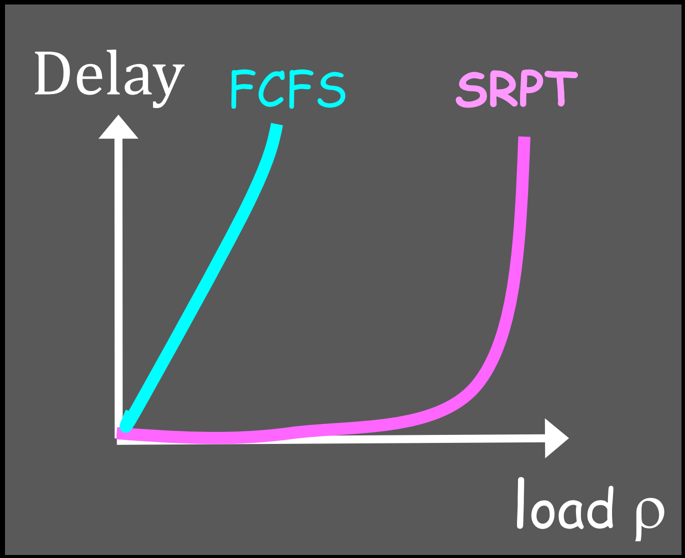


$C_S^2 = 100$

Low variability



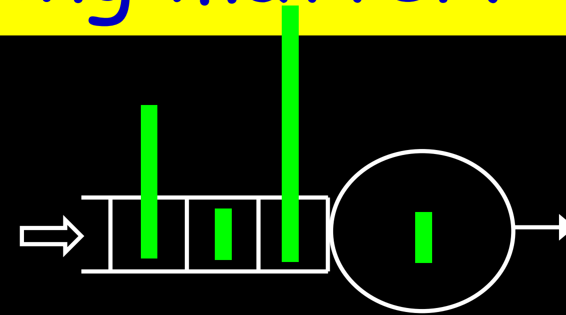
High variability



How much does scheduling matter?

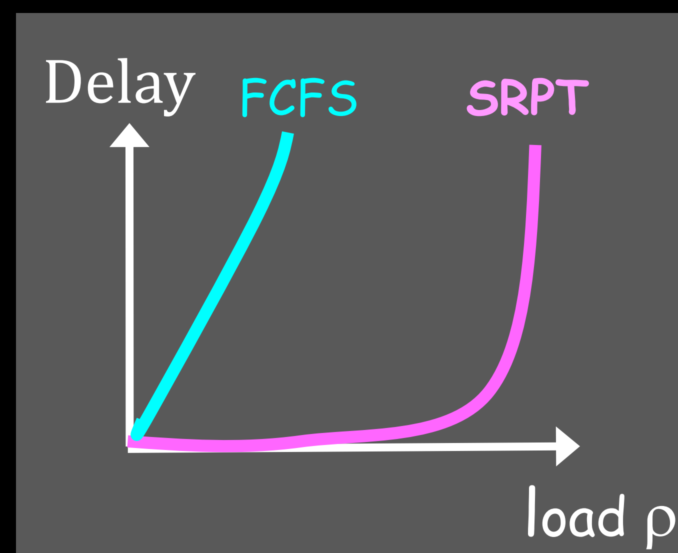
But wait! Doesn't SRPT starve big jobs?

No.
"All Can Win Theorem"
[Bansal, Harchol-Balter, Sigmetrics '01]

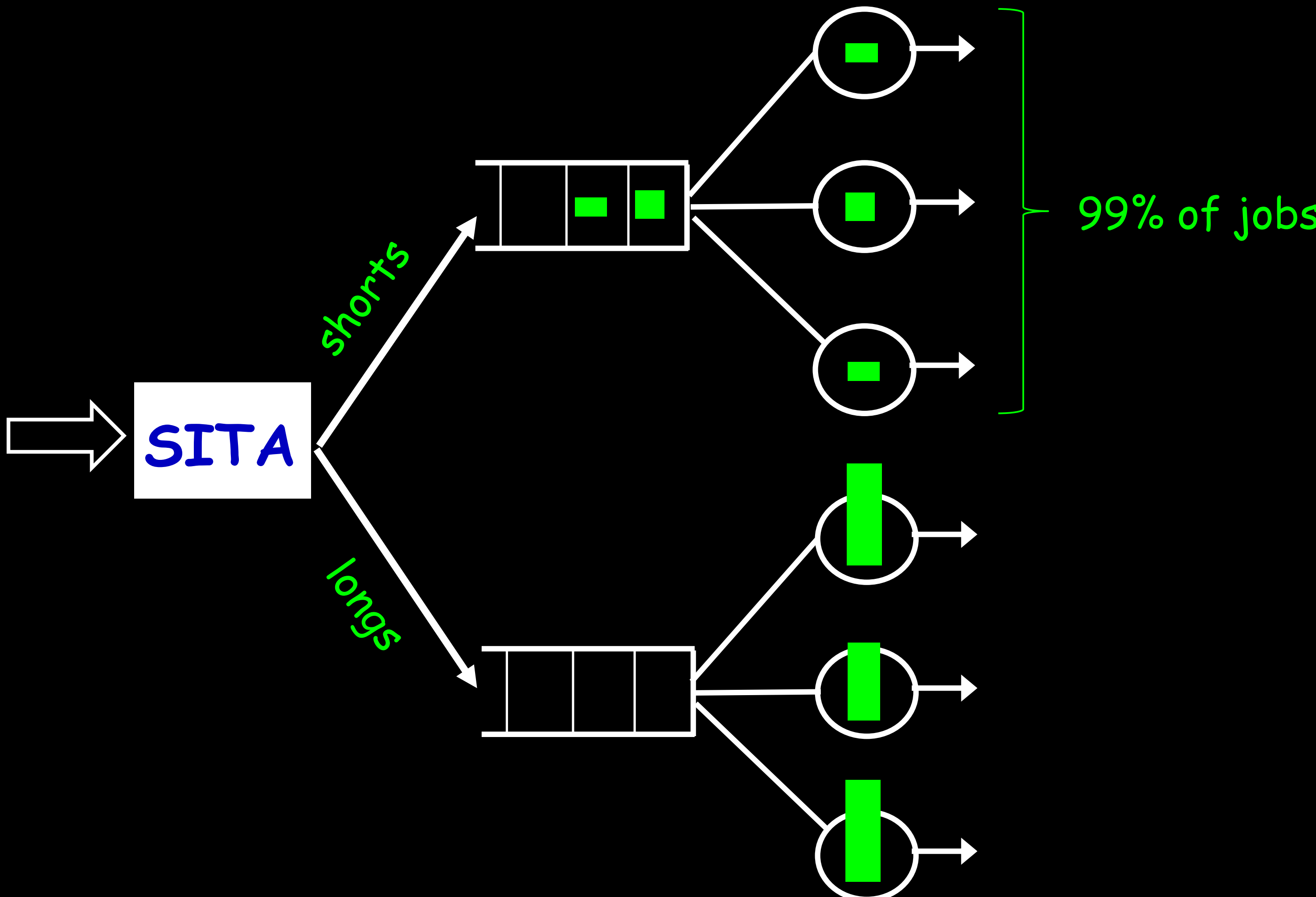


$C_S^2 = 100$

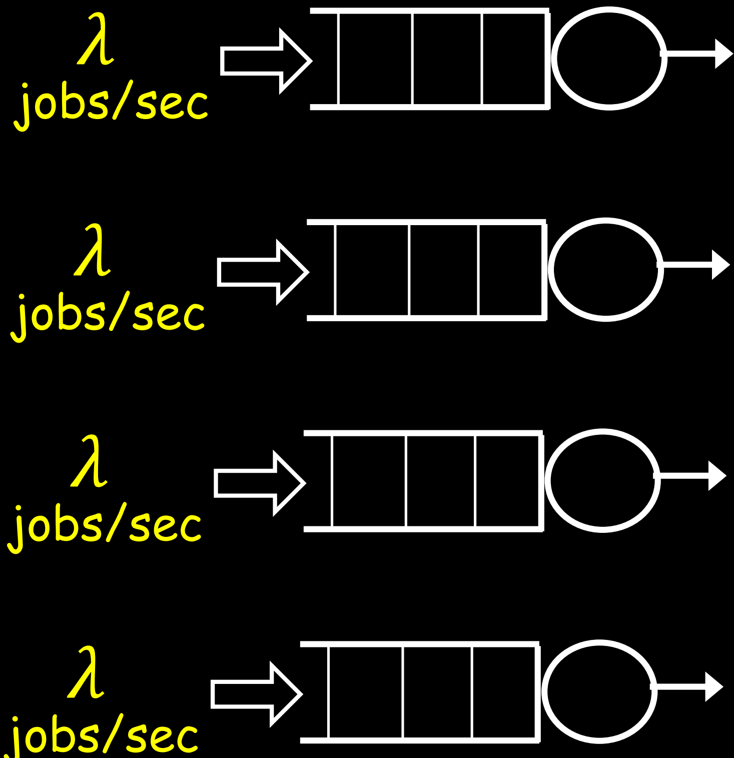
High variability



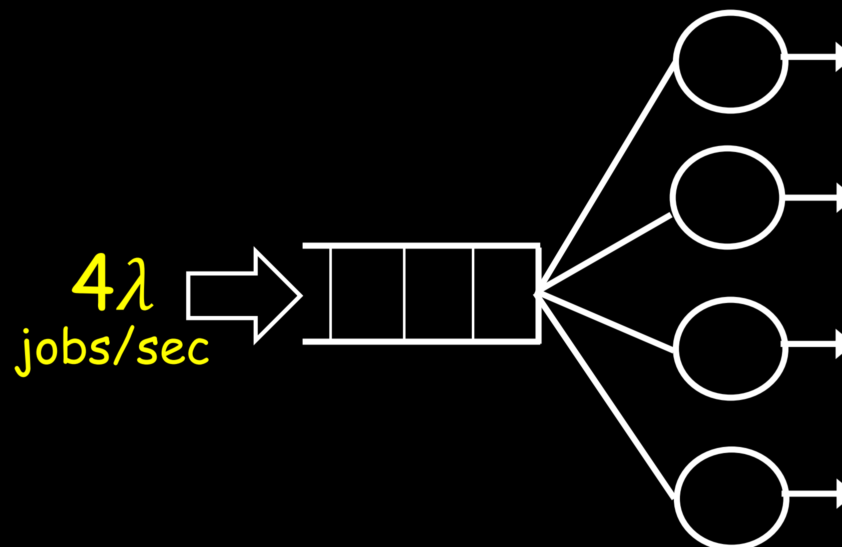
Solution 2: Isolate smalls via SITA



Solution 3: Pooling



- Pooled system has same utilization.
- but MUCH lower delay



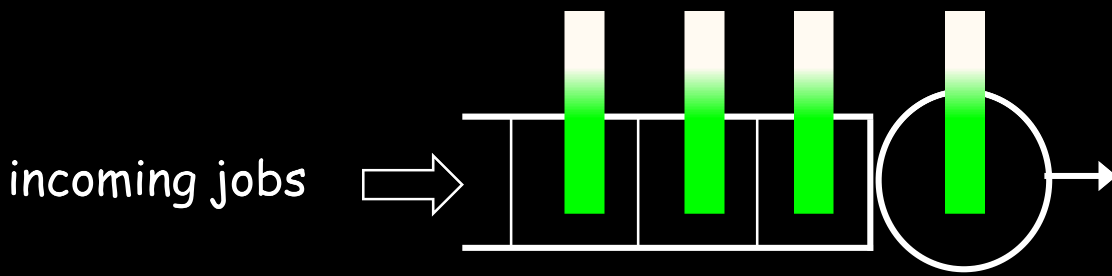
- Pooling allows short jobs to circumvent long ones.

Question 3:

"How can I schedule better when I don't know job size?"

Unknown job size

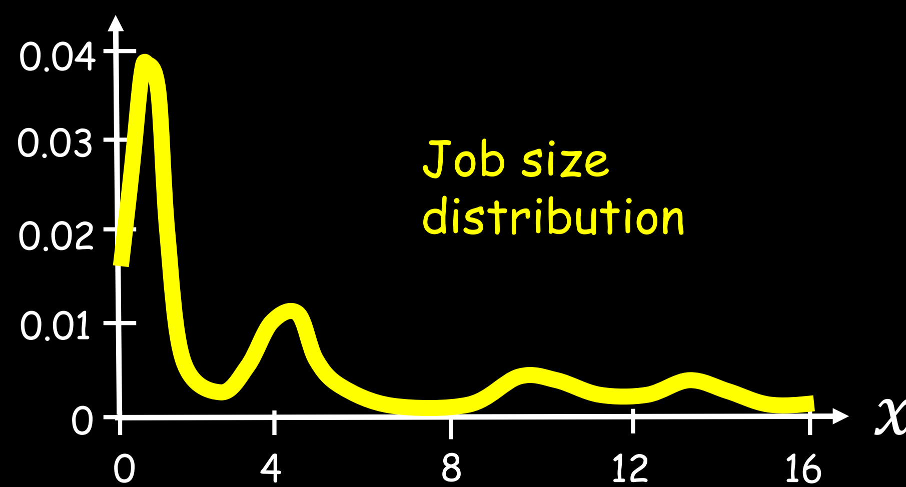
Do NOT know job size



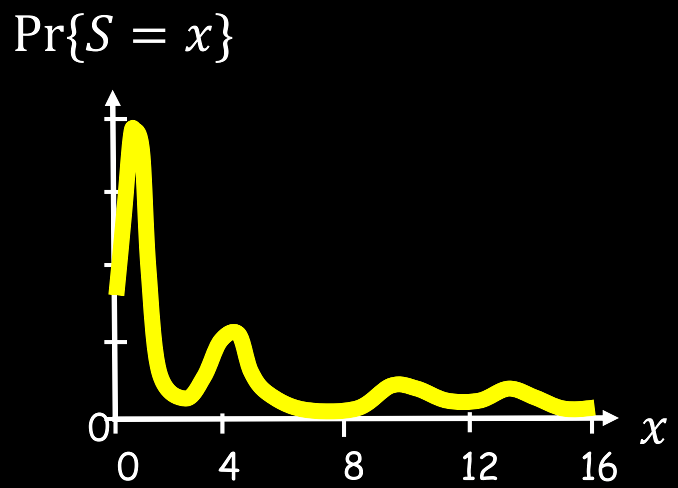
KNOW job age (time served so far)

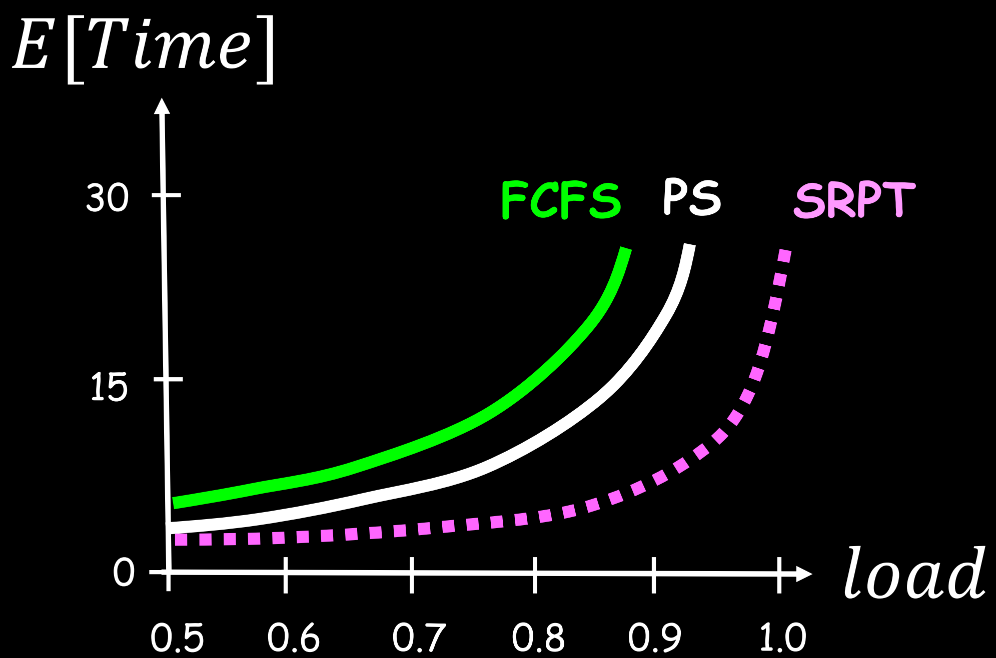
KNOW job size distribution

$$\Pr\{S = x\}$$



Unknown job size

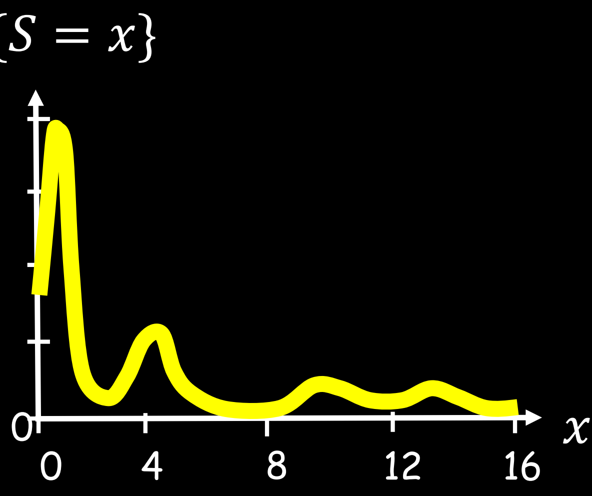


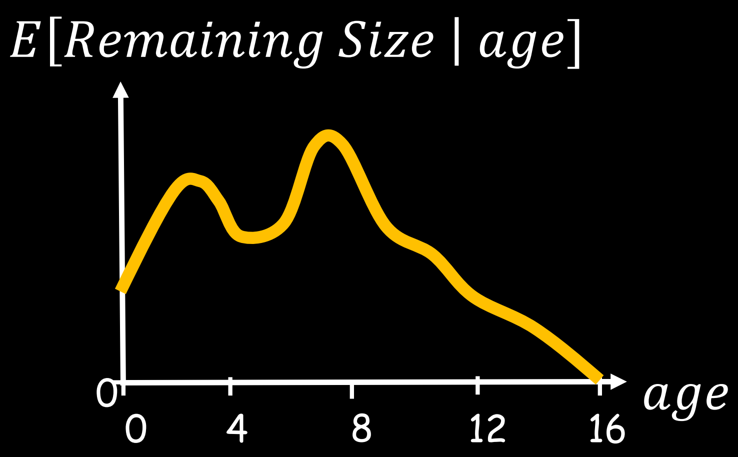


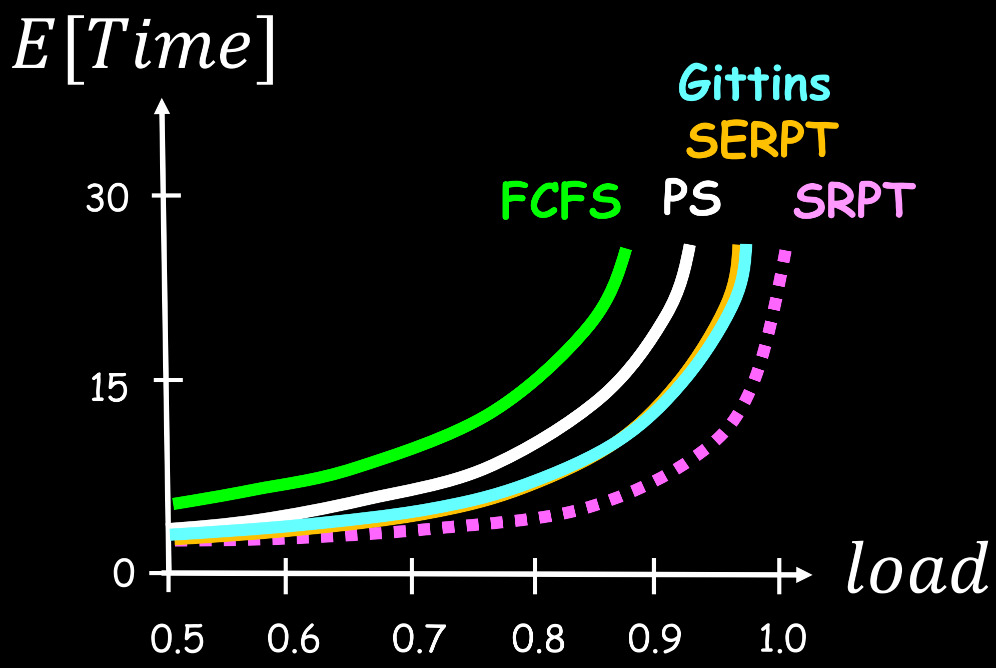
Processor-Sharing (PS)

allows shorts to complete more quickly

Unknown job size







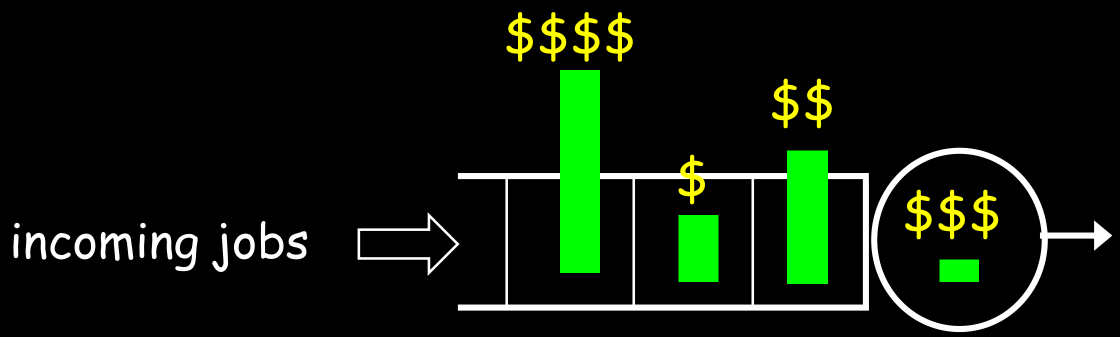
Shortest-Expected-Remaining-Processing-Time (SERPT)

Gittins Index is true optimal when job sizes not known.

Question 4:

"How to schedule jobs which differ in size and value?"

Jobs differ in size & value



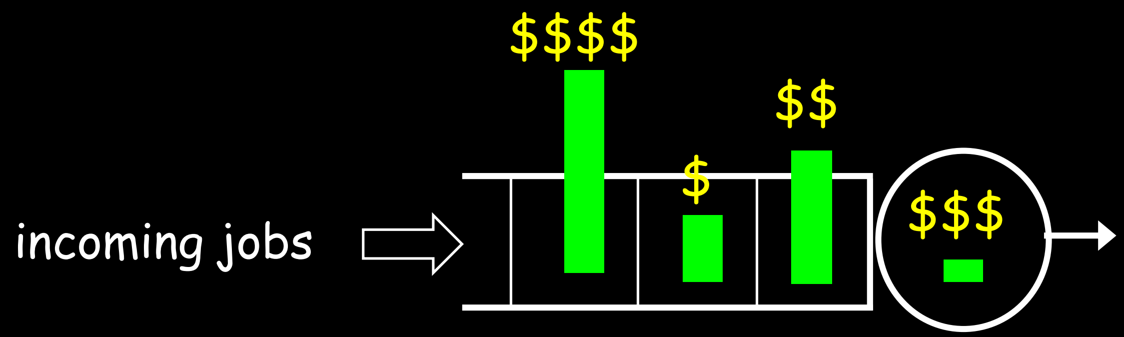
$$$ Holding cost of job = dollar cost for every hour that this job is not done

Size of job = hours of work needed to get job done

Every hour, there's a "total holding cost" – summed cost over all jobs

GOAL: Minimize time-average total holding cost

cμ-Rule



$$$ <u>Holding cost of job</u> = dollar cost for every hour that this job is not done

<u>Size of job</u> = hours of work needed to get job done

$$Index(job) = \frac{Holding\ cost\ of\ job}{Remaining\ size\ of\ job}$$

Schedule jobs Highest Index First.

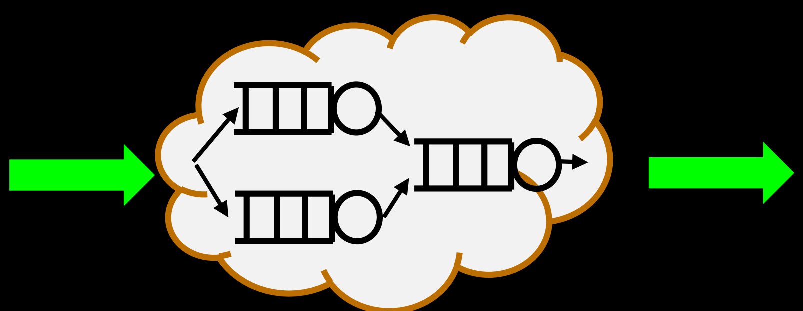
Question 5:

"How do answers change for closed-loop system configurations?"

Closed versus Open Models

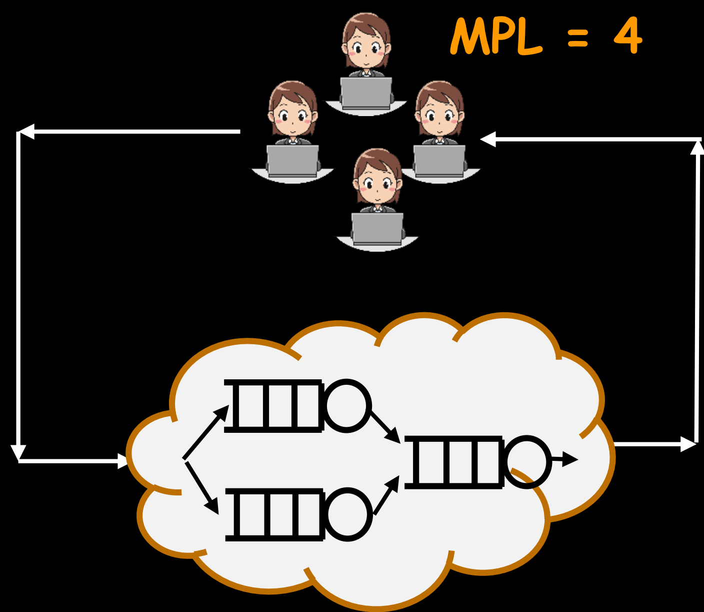
Open System

New job arrivals are exogenous to the system



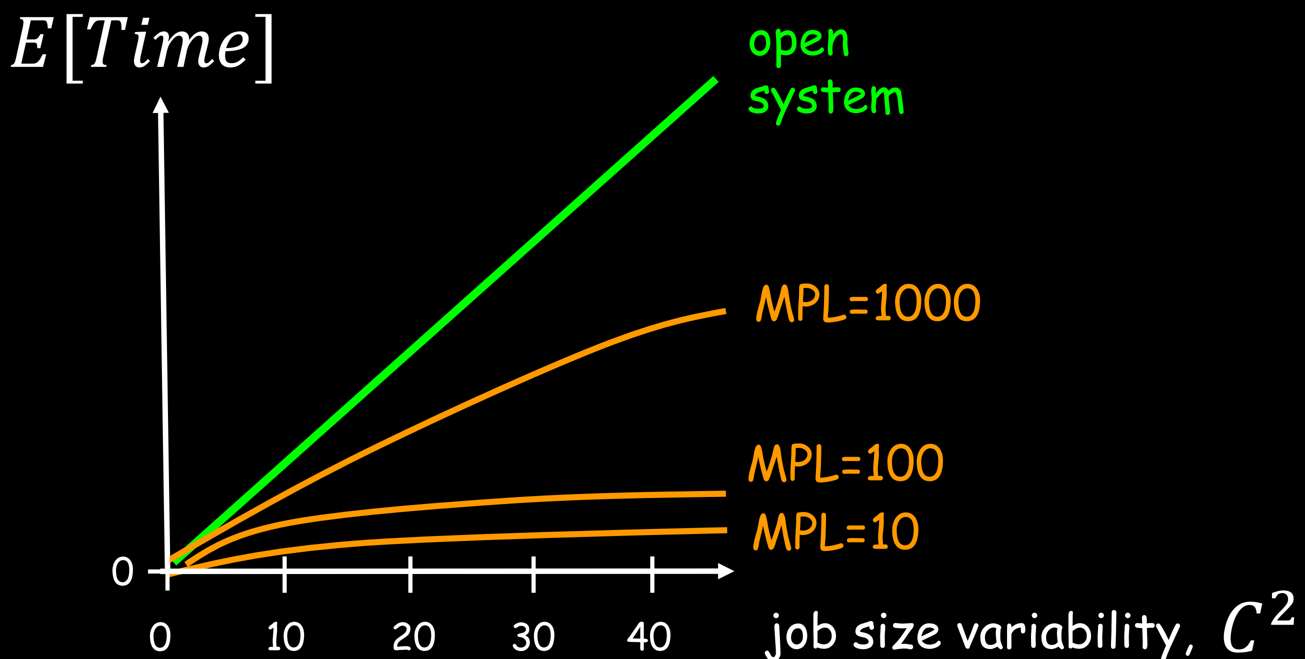
Closed System

New job arrivals are triggered by job completions



Closed systems don't feel variability

Operate open system & closed system, both with the same avg. utilization



Conclusion

Q1: My system utilization is low, so why are my delays so high?

Q2: How can I lower job delay?

Q3: How can I schedule when I don't know job size?

Q4: How to schedule jobs with different values?

Q5: How do answers change for closed-loop system configurations?

Thank you!