Chapter 10
Heavy Tails:
the distributions of computing

Distributions we've seen so far

We've studied several continuous distributions so far:

- Normal(μ , σ^2)
- Uniform(a, b)
- Exp(λ)

Q: Which of these represents CS distributions?

- distribution of file sizes
- distribution of IP flow durations
- distribution of job CPU usage

Distributions we've seen so far

We've studied several continuous distributions so far:

- Normal(μ , σ^2)
- Uniform(a, b)
- Exp(λ)

Q: Which of these represents CS distributions?

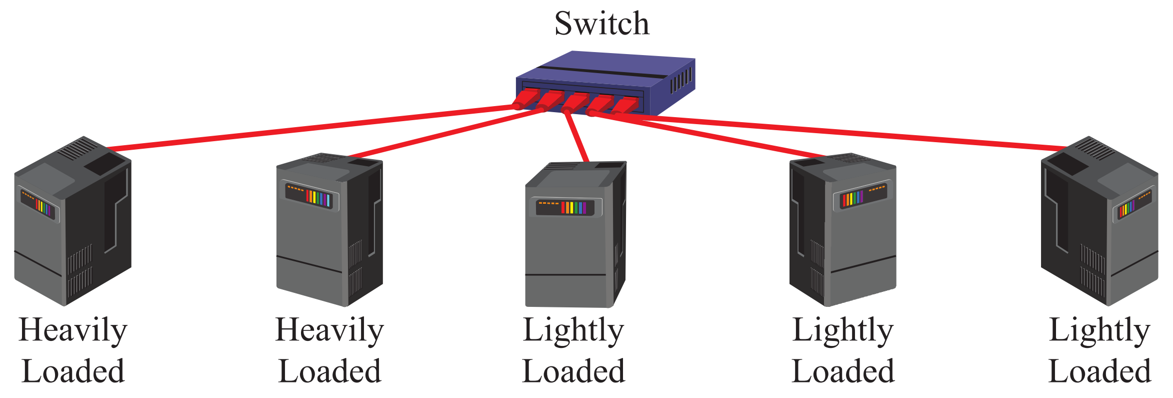
- distribution of file sizes
- distribution of IP flow durations
- distribution of job CPU usage

None!

Distributions of CS were studied in 1990's …

It all started with a computer science question…

[Harchol-Balter & Downey, "Exploiting Process Lifetimes for CPU Load Balancing," SIGMETRICS 1996]



CPU load balancing:

Migrate jobs from heavily-loaded to lightly-loaded machines

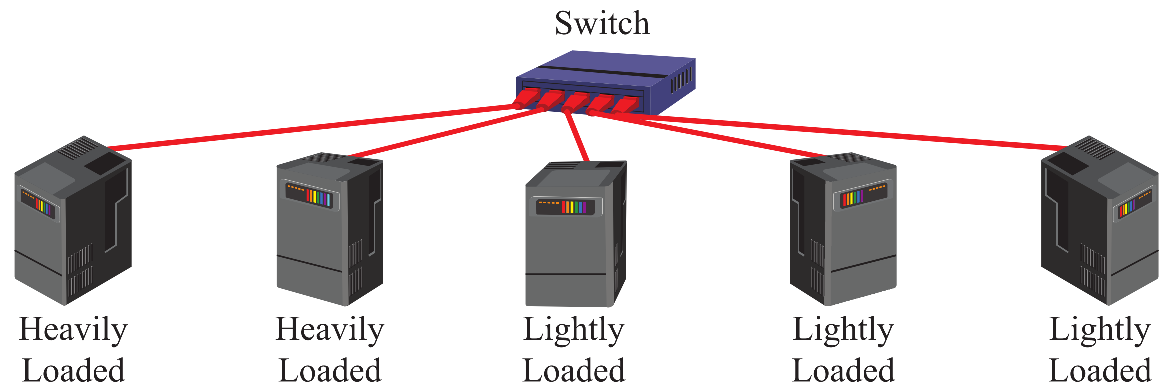
Q: In CPU load balancing, which kind of job migration makes sense?

P: Preemptive migration	vs.	**NP: Non-preemptive only**
Preempt/migrate jobs after they've started running. "Active process migration."		Don't preempt job once it starts running. Only load balance newborns.

Distributions of CS were studied in 1990's …

It all started with a computer science question…

[Harchol-Balter & Downey, "Exploiting Process Lifetimes for CPU Load Balancing," SIGMETRICS 1996]



CPU load balancing:
Migrate jobs from heavily-loaded to lightly-loaded machines

Q: In CPU load balancing, which kind of job migration makes sense?

P: Preemptive migration
Preempt/migrate jobs after they've started running. "Active process migration."

vs.

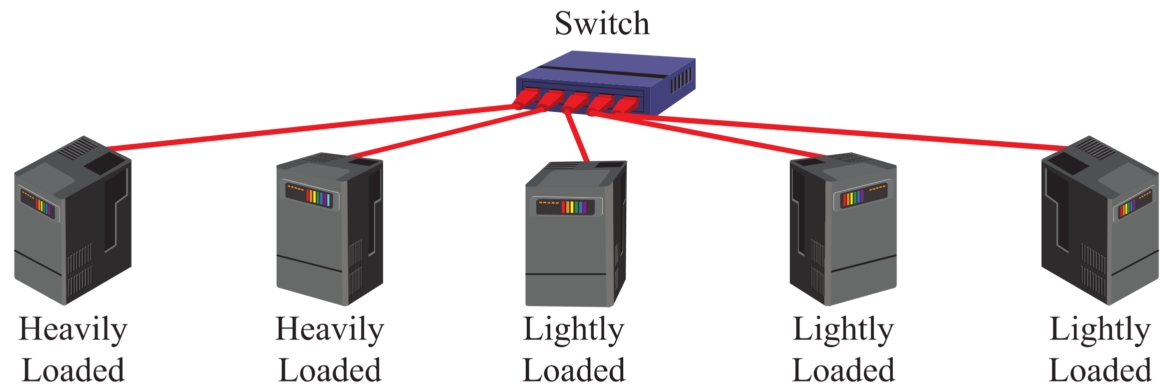
NP: Non-preemptive only
Don't preempt job once it starts running. Only load balance newborns.

Why is NP preferred?

Distributions of CS were studied in 1990's …

It all started with a computer science question…

[Harchol-Balter & Downey, "Exploiting Process Lifetimes for CPU Load Balancing," SIGMETRICS 1996]



CPU load balancing:

Migrate jobs from heavily-loaded to lightly-loaded machines

Q: In CPU load balancing, which kind of job migration makes sense?

P: Preemptive migration

Preempt/migrate jobs after they've started running. "Active process migration."

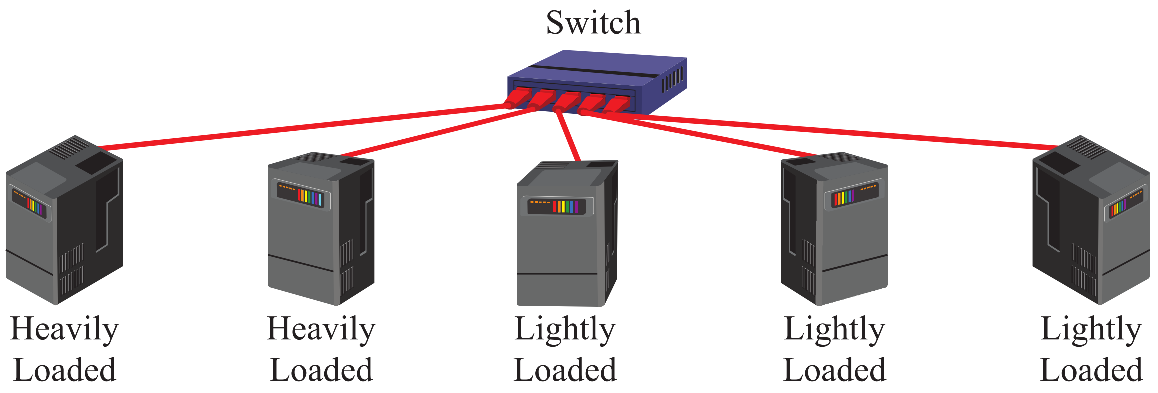
vs.

NP: Non-preemptive only

Don't preempt job once it starts running. Only load balance newborns.

NP is cheap!

P is costly!



CPU load balancing:

Migrate jobs from heavily-loaded to lightly-loaded machines

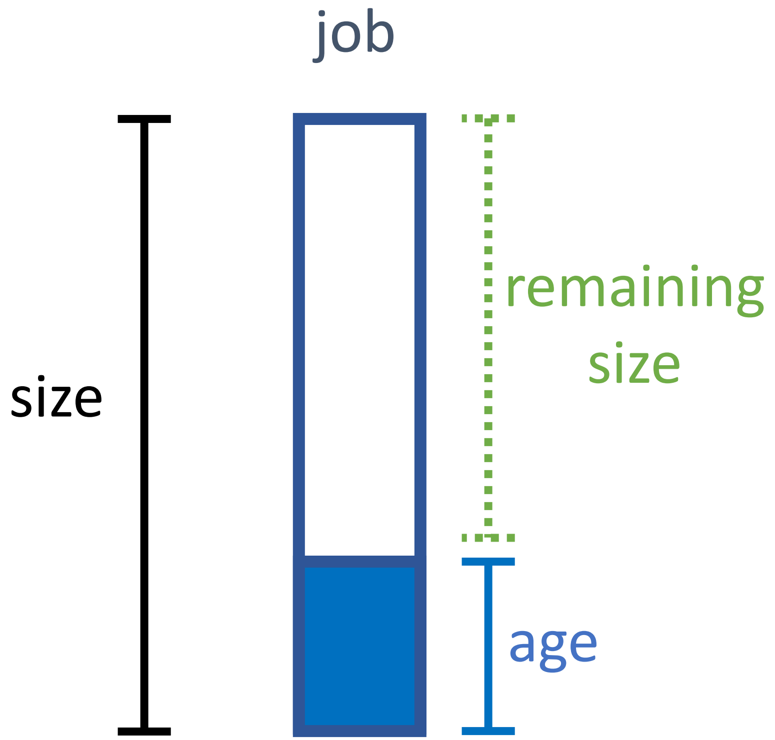
To better understand how to think about this question, let’s introduce some vocabulary …

Some vocabulary

A job's **size** is its total CPU requirement (a.k.a. CPU lifetime)

A job's **age** is its total CPU usage so far

A job's **remaining size** is its remaining CPU needs

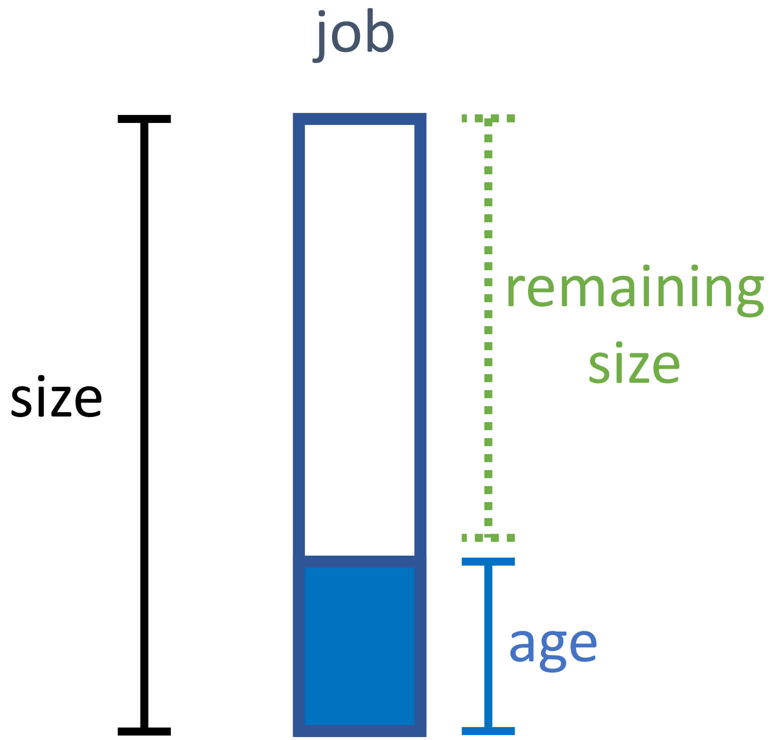


Some vocabulary

A job's **size** is its total CPU requirement (a.k.a. CPU lifetime)

A job's **age** is its total CPU usage so far

A job's **remaining size** is its remaining CPU needs



Q: Which of these do we know?

Q: Which of these do we NOT know?

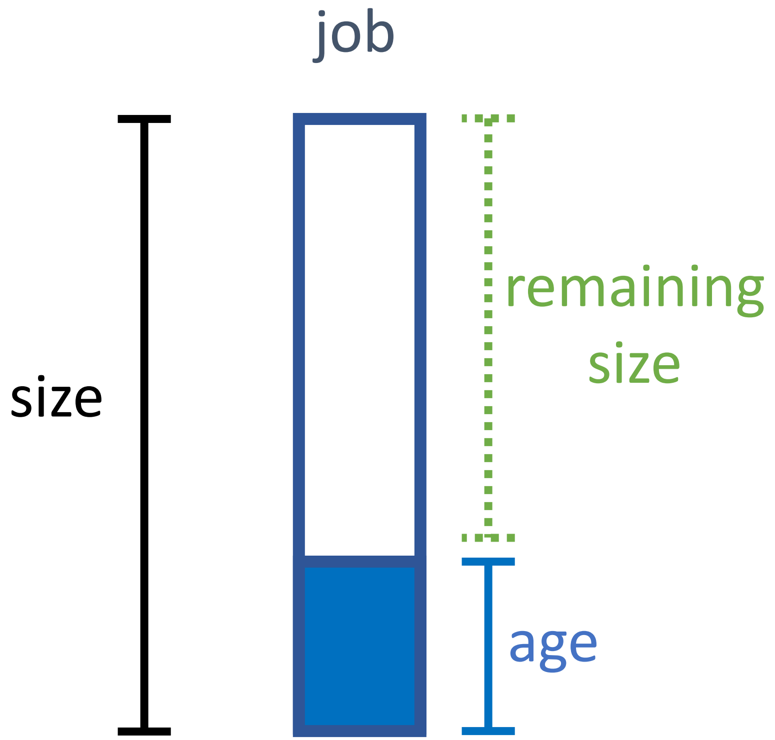
Q: Which of these is relevant to migration?

Some vocabulary

A job's **size** is its total CPU requirement (a.k.a. CPU lifetime)

A job's **age** is its total CPU usage so far

A job's **remaining size** is its remaining CPU needs

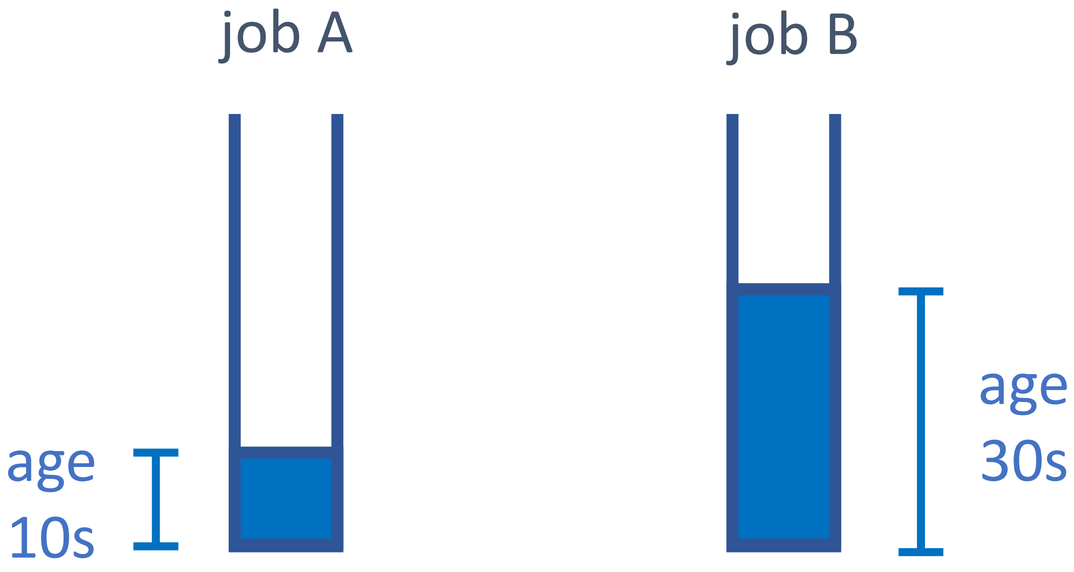


We only <u>know</u> a job's **age …**
But what we <u>need</u> is its **remaining size**.

- If **remaining size** is high, then pays to migrate, even if migration is costly.
- If **remaining size** is low, then doesn't pay to migrate.

What does age tell us about remaining size?

Q: Which of these jobs likely has higher remaining size?



$\mathbf{P}\{Size > x + a \mid Size > a\}$

Q: Does this increase with a?
or decrease with a?

Q: What would the answer be if $Size \sim Exp(\mu)$

Failure rate: informally

$$\mathbf{P}\{Size > x + a \mid Size > a\}$$

Increases with a

Decreasing Failure Rate

D.F.R.

The longer you've lived,
the longer you'll live …

Q: Examples?

Decreases with a

Increasing Failure Rate

I.F.R.

The longer you've lived,
the sooner you'll die …

Q: Examples?

Failure rate: informally

$$\mathbf{P}\{Size > x + a \mid Size > a\}$$

Increases with a

Decreasing Failure Rate
D.F.R.

The longer you've lived,
the longer you'll live …

- Time you've been friends with someone.
- Time you've lived in your home.

Decreases with a

Increasing Failure Rate
I.F.R.

The longer you've lived,
the sooner you'll die …

- Lifetime of a car.
- Lifetime of a washing machine.

Failure rate: definition

<u>Definition</u>: For continuous r.v. X, with p.d.f. $f_X(t)$ and tail $\overline{F_X}(t) = \boldsymbol{P}\{X > t\}$, the **failure rate function** for X is:

$$r_X(t) = \frac{f_X(t)}{\overline{F_X}(t)}$$

Looks like a conditional pdf, where we're conditioning on $X > t$

$$\boldsymbol{P}\{X \in (t\,, t + dt) \mid X > t\} = \frac{\boldsymbol{P}\{X \in (t, t + dt)\}}{\boldsymbol{P}\{X > t\}}$$

$$= \frac{f_X(t)dt}{\overline{F_X}(t)}$$

$$= r_X(t)dt$$

So $r_X(t)$ is the instantaneous death rate!

Failure rate: definition

$$\boldsymbol{P}\{X \in (t\,, t + dt) \mid X > t\} = r_X(t)$$

So $r_X(t)$ is the instantaneous failure rate!

$r_X(t)$ decreases with t

Decreasing Failure Rate
D.F.R.

- Lifetime of friendship.
- Time lived in home.

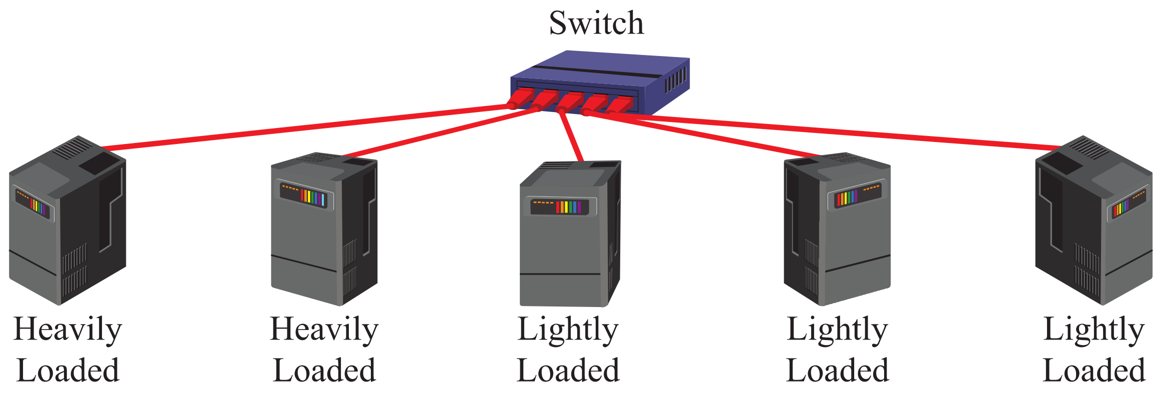
$r_X(t)$ increases with t

Increasing Failure Rate
I.F.R.

- Lifetime of a car.
- Lifetime of a washing machine.

Q: Is there a distribution with constant failure rate?

How does failure rate of job size affect P vs NP?



CPU load balancing:
Migrate jobs from heavily-loaded to lightly-loaded machines

Q: In CPU load balancing, which kind of job migration makes sense?

P: Preemptive migration
Preempt/migrate jobs after they've started running.
"Active process migration."

If Job Size (CPU reqt.) has DFR

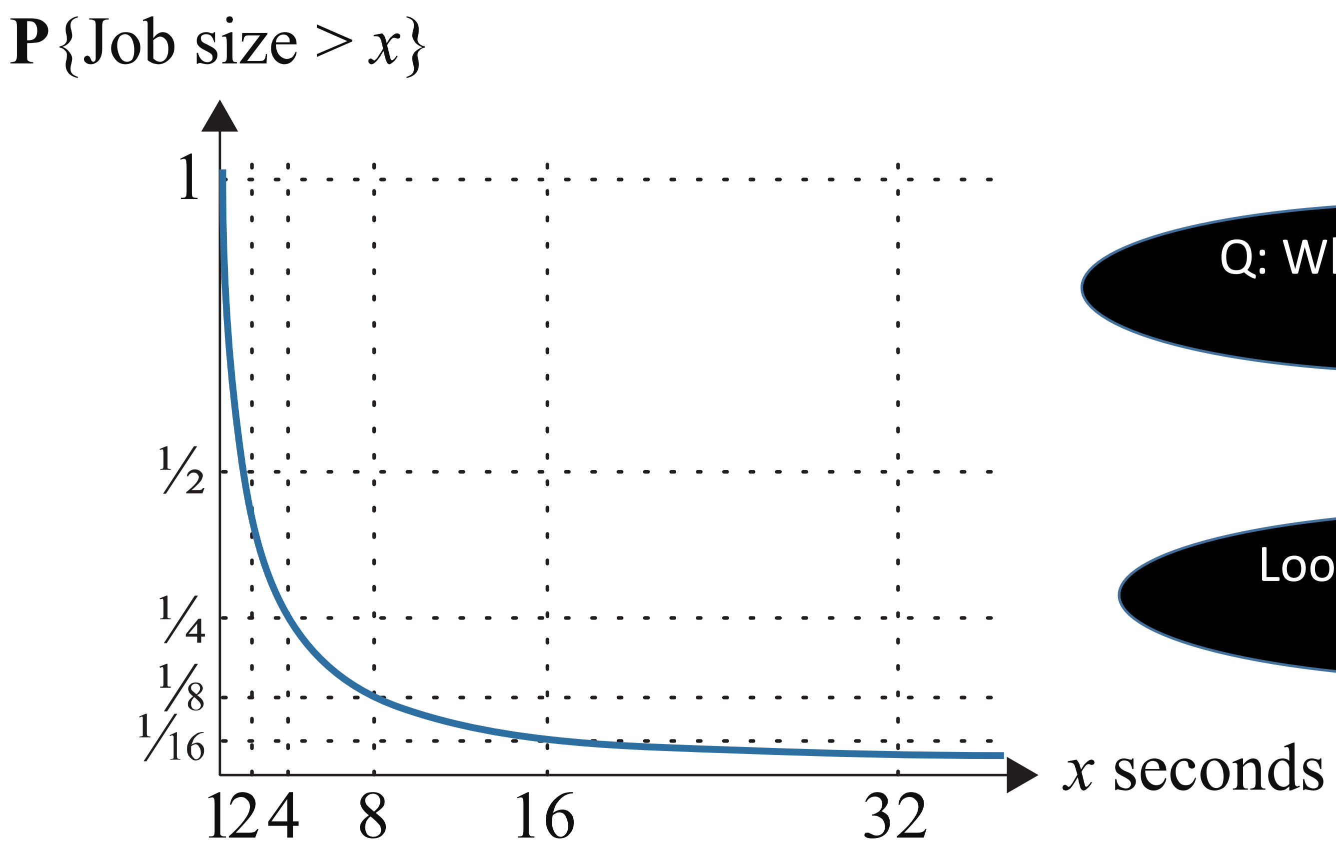
vs.

NP: Non-preemptive only
Don't preempt job once it starts running.
Only load balance newborns.

If Job Size (CPU reqt.) has IFR or CFR

So what is the distribution of job size?

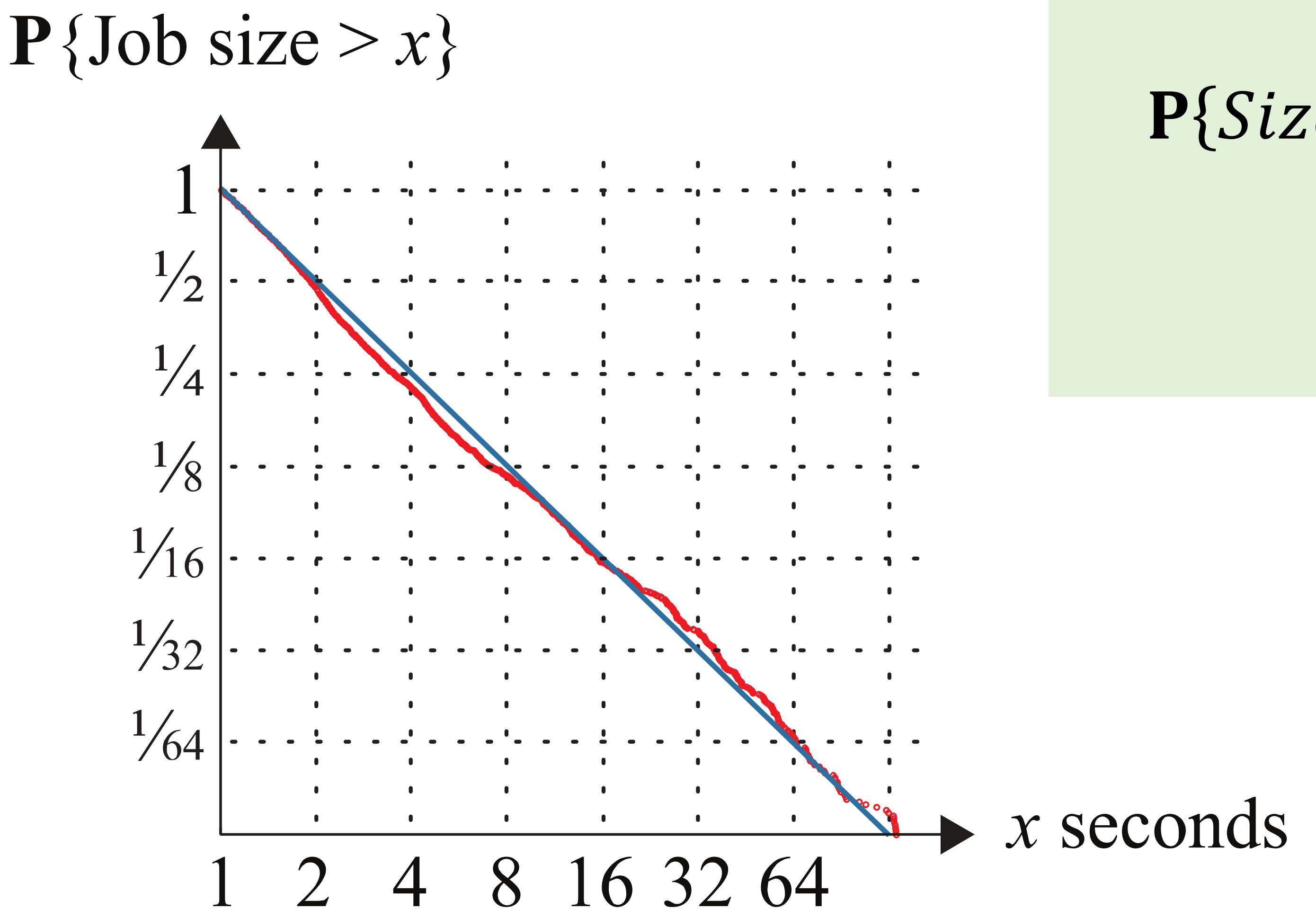
Results of measurements of millions of jobs [Sigmetrics 1996]



Q: What distribution is this?

Looks like Exponential, but NOT!

Let’s replot on a log-log scale



$$\mathbf{P}\{Size > x\} = \frac{1}{x}$$

$$(x \geq 1)$$

"Introduction to Probability for Computing", Harchol-Balter '24

Properties of the job size distribution

$$\mathbf{P}\{Size > x\} = \frac{1}{x} \quad \text{where} \quad x \geq 1$$

Q: is this a valid distribution?

$$\overline{F_X}(x) = x^{-1}$$

$$F_X(x) = 1 - x^{-1}$$

$$f_X(x) = x^{-2}$$

$$\int_1^\infty f_X(x)dx = \int_1^\infty x^{-2}dx = 1 \checkmark$$

Properties of the job size distribution

$$\mathbf{P}\{Size > x\} = \frac{1}{x} \quad \text{where} \quad x \geq 1$$

Q: What is the mean
of this distribution?

$$f_X(x) = x^{-2}$$

$$\boldsymbol{E}[X] = \int_1^\infty x \cdot f_X(x) dx$$

$$= \int_1^\infty x \cdot x^{-2} dx$$

$$= \int_1^\infty x^{-1} dx = \infty$$

All moments
are infinite!

Properties of the job size distribution

$$\mathbf{P}\{Size > x\} = \frac{1}{x} \quad \text{where} \quad x \geq 1$$

Q: What is the failure rate?

$$f_X(x) = x^{-2}$$

$$\overline{F_X}(x) = x^{-1}$$

$$r_X(x) = \frac{f_X(x)}{\overline{F_X}(x)} = \frac{1}{x}$$

D.F.R.

Properties of the job size distribution

$$\mathbf{P}\{Size > x\} = \frac{1}{x} \quad \text{where} \quad x \geq 1$$

Q: Doubling property?

$$\mathbf{P}\{X > 2a \mid X > a\} = \frac{\frac{1}{2a}}{\frac{1}{a}} = \frac{1}{2}$$

Job of age a will make it to age $2a$ with probability half.

Pareto Distribution

<u>Definition</u>: $X \sim Pareto(\alpha)$, if

$$\overline{F_X}(x) = x^{-\alpha}, \qquad x \geq 1$$

where $0 < \alpha < 2$.

Pareto was an economist in 1900.

Q: What is the distiribution of UNIX job sizes from 1996?

A: $Pareto(\alpha = 1)$

Pareto Distribution

Definition: $X \sim Pareto(\alpha)$, if

$$\overline{F_X}(x) = x^{-\alpha}, \qquad x \geq 1$$

where $0 < \alpha < 2$.

Pareto was an economist in 1900.

Properties of Pareto(α) distribution:

1) DFR
2) Infinite variance

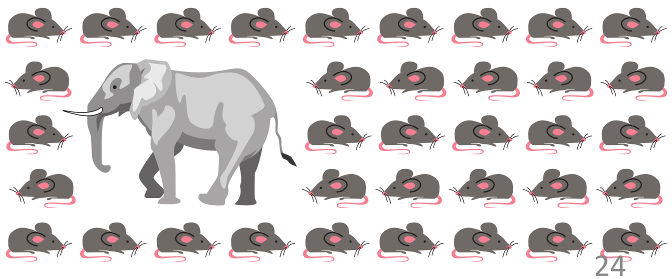
-- Note: $\boldsymbol{E}[X]$ is finite if $\alpha > 1$, but infinite if $\alpha \leq 1$.

-- All higher moments of X are infinite.

3) Heavy-tailed property: The top 1% of jobs comprise 50% of the load

-- For lower α we have a heavier tail.

-- Higher α results in less heavy tail.



Q: Where do heavy tails come up in economics?

Bounded-Pareto Distribution

Empirical distributions are always bounded.

The Bounded-Pareto distribution has a Pareto shape but is finite.

<u>Definition</u>: $X \sim BoundedPareto(k, p, \alpha)$,

lower limit

upper limit

$$f_X(x) = C \cdot \alpha x^{-\alpha-1}, \qquad k \leq x \leq p$$

where $0 < \alpha < 2$ and where C is a normalizing constant.

BoundedPareto has similar properties to Pareto:

-- (mostly) DFR

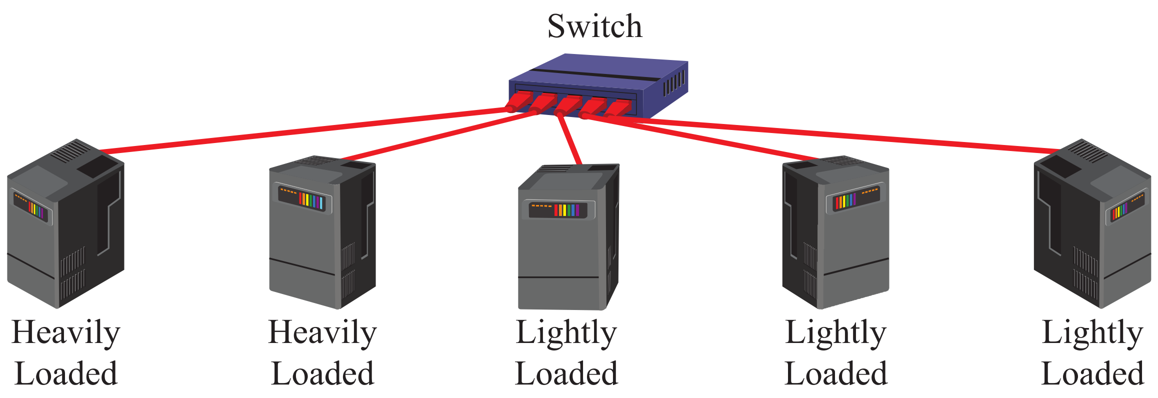
-- near-infinite variance

-- heavy-tailed property, assuming upper limit, p, is large.

What does all this mean for CPU load balancing?

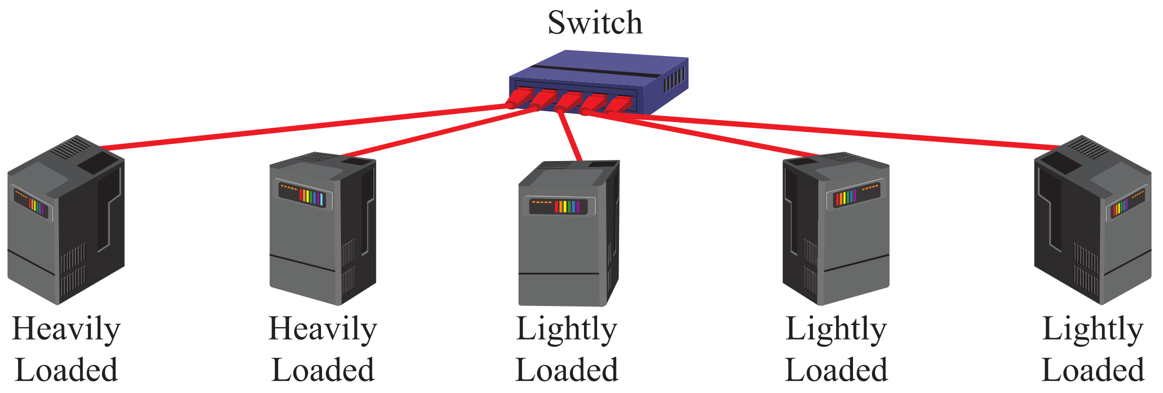
Properties of Pareto(α) distribution:

1) DFR
2) Infinite variance
3) Heavy-tailed property: The top 1% of jobs comprise 50% of the load



- DFR implies:
  - Older jobs have higher remaining sizes
  - Pays to migrate older jobs
- Heavy-tailed property implies:
  - Can get significant load balancing benefit from only migrating 1% of jobs

What does all this mean for CPU load balancing?



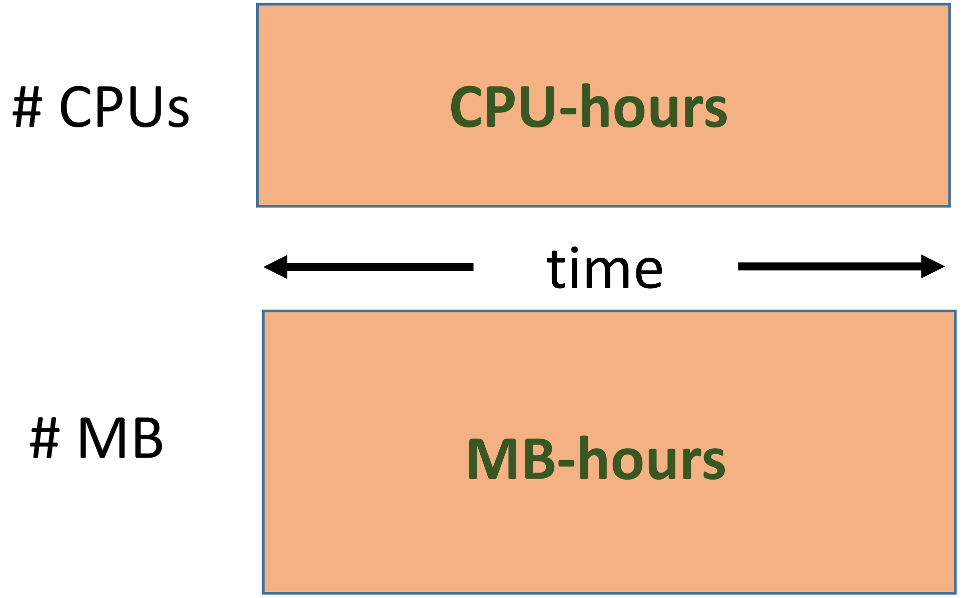
[Harchol-Balter, Downey "Exploiting process lifetime distributions for CPU load balancing." SIGMETRICS 1996 Best Paper.

- Measured CPU lifetimes of UNIX jobs: BoundedPareto($\alpha = 1$) job size distribution
- Very high squared coefficient of variation: $C^2 \approx 50$.
- Showed P-migration pays and is superior to NP-migration
- Achieved CPU load balancing by only migrating the 4% oldest jobs.

What do jobs look like today?

2020 study of jobs at Google run in Google Borg scheduler:

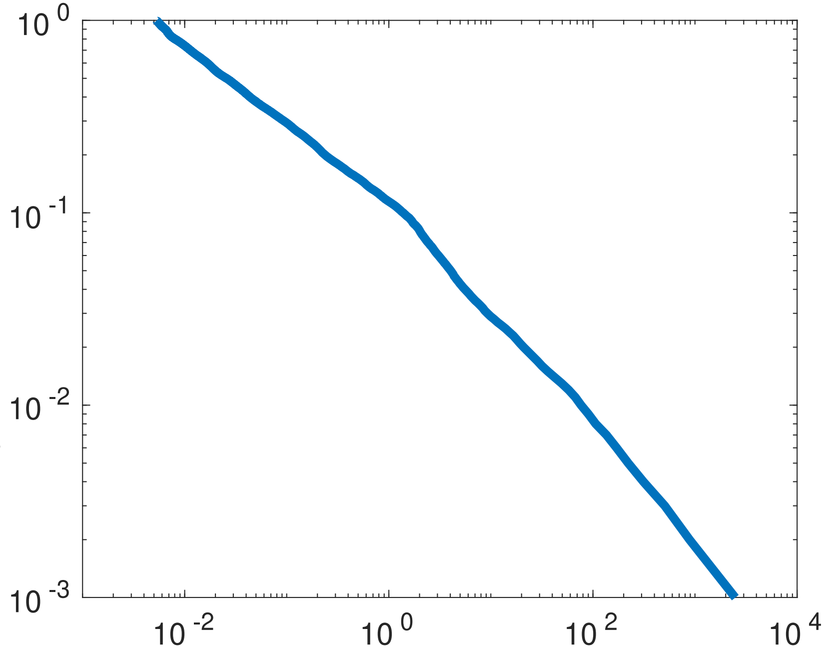
[Tirmazi et al., “Borg: The Next Generation,” USENIX 2020.]



Compute usage today

2020 study of jobs at Google run in Google Borg scheduler:

$\mathbf{P}\{\text{CPU–hours} > x\}$



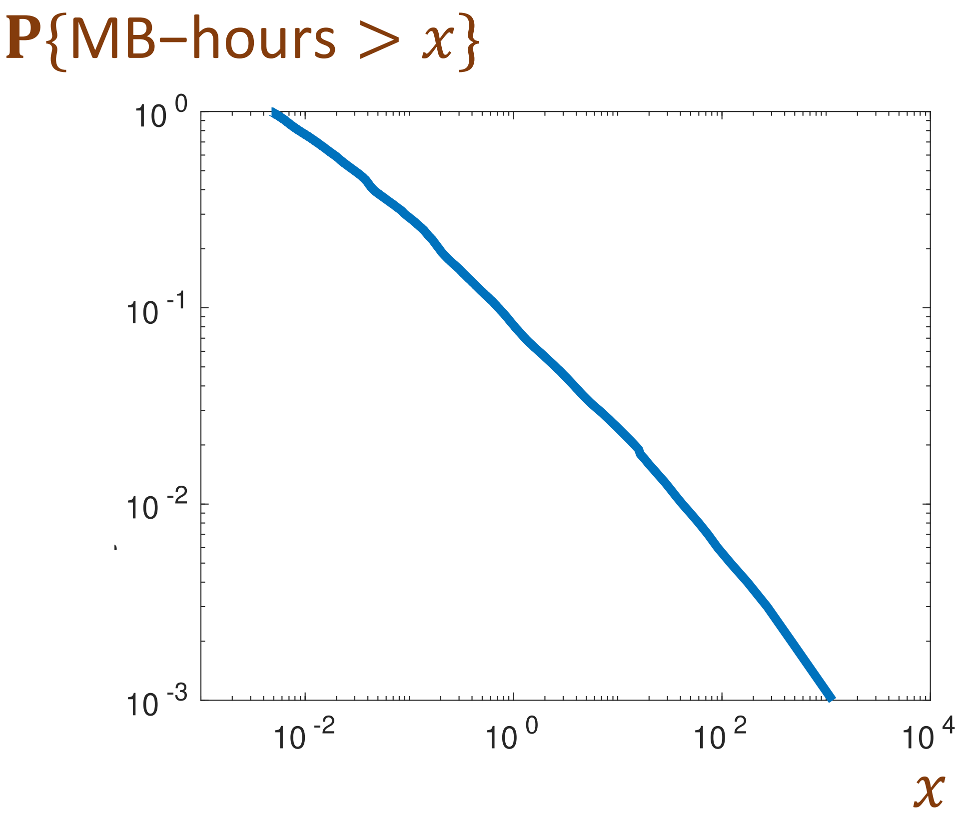
x

- CPU-hours used by jobs span 9 orders of magnitude
- Straight line on log-log scale fits Pareto($\alpha = 0.7$) distribution
- $C^2 = 23{,}000$
- Top 1% of jobs make up 99% of total CPU usage

* For privacy reasons, all numbers shown are normalized by unknown constant.

Memory usage today

2020 study of jobs at Google run in Google Borg scheduler:



- MB-hours used by jobs span 10 orders of magnitude
- Straight line on log-log scale fits Pareto($\alpha = 0.7$) distribution
- $C^2 = 43{,}000$
- Top 1% of jobs make up >99% of total CPU usage

* For privacy reasons, all numbers shown are normalized by unknown constant.

Pareto distributions are everywhere!

- Job compute usage
- Job memory usage
- Web file sizes
- IP flow durations
- Wireless session times
- Phone call durations
- National wealth
- Damage due to earthquakes
- Damage due to forest fires

The question is WHY?