Performance Modeling
of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Memoryless property
and
probability distributions

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

1

P 27 slide, P. 308 discrete

Memoryless property as lifetime

esponenziale [≈ geometrico (memoryless nel discreto) ma nel caso continuo]

A random variable X is said to be memoryless if

$$Prob\{X > s + t \mid X > s\} = Prob\{X > t\} \quad \forall s, t > 0$$

il comportamento dipende solo dal presente

<u>Example</u>
X is the lifetime of a lightbulb. (lampadina)
The property says that the probability that the lightbulb survives for at least another t seconds before burning out, given that the lightbulb has survived for s seconds already, is the same as the probability that the lightbulb survives at least t seconds independent of s.

Con l'esponenziale, un lightbulb nuovo ha stesso tempo rimanente di lightbulb acceso x ore.

Does this seem realistic for a lightbulb???

2

Lifetime and failure rate

Considero distribuzioni NON memoryless:

Distributions for which $Prob\{X > s+t | X > s\}$ goes down as s goes up are said to have *increasing failure rate*. (s ↑, Prob ↘)
The device is more and more likely to fail as time goes on.

Example
A car's lifetime. The older a car is the less likely that it will survive another $t = 6$ years.

Distributions for which $Prob\{X > s+t | X > s\}$ goes up as s goes up are said to have *decreasing failure rate*. (s ↑, Prob ↗)
The device is less likely to fail as time goes on.

Example
- UNIX job CPU lifetimes. The more CPU a job has used up so far, the more it is likely to use up.
- The same for computer chips. If they're going to fail, they'll do it early. That's why chip manufacturers test them for a long while.

P. 29 slide, P. 234 Performance

Hazard rate function

funzione di distribuzione nel caso CONTINUO

Let X be a continuous random variable with probability density function $f(t)$ and cumulative distribution function $F(t) = Pr\{X < t\}$.
Then $r(t)$ is formally defined as: var. casuali discrete

$$\text{hazard} \quad r(t) \equiv \frac{f(t)}{\overline{F}(t)}$$

where $\overline{F}(t) = 1 - F(t) = Pr\{X > t\}$

Consider the probability that a t-year old item will fail during the next dt seconds:

$$\int_t^{t+dt} \dots dt$$

$$Pr\{X \in (t, t+dt) | X > t\} = \frac{Pr\{X \in (t, t+dt)\}}{Pr\{X > t\}} \approx \frac{f(t)dt}{\overline{F}(t)} = r(t)dt$$

the istantaneous failure rate

Hazard rate function

If $r(t)$ is constant then $f(t)$ must be exponential
Indeed for the exponential ← unica distr. ad avere tale proprietà

$$r(t) = \frac{f(t)}{F(t)} = \frac{\lambda \cancel{e^{-\lambda t}}}{\cancel{e^{-\lambda t}}} = \lambda \quad \text{(costante nel tempo)}$$

We use the failure rate concept when we study scheduling

5

P376 Perf, 30 slide

Why the remaining lifetime is so important?

CPU load balancing in a Network of Workstations

(sposto verso nodi meno carichi)

- It may help to migrate a job to a less-loaded workstation (one with fewer jobs) in order to improve mean response times
- migration can be expensive if the job has a lot of "state" that has to be migrated with the job (lots of memory). ↑ migrare processo in esecuzione, ha uno stato!

two types of migration used in load balancing techniques:

processi che non hanno ancora uno stato

- non-preemptive migration (NP) only relocates "newborn" processes (initial placement, or remote execution) ← sinonimi
- preemptive migration (P) migrates processes that are already active (running) (active process migration)

6

1. Should we bother with P migration, or is NP enough? serve?
2. If we are going to bother with P migration, which processes are worth migrating? (se serve, chi migro?)
   That is, what is a good migration policy?

terminology:
- a job's "size": its total CPU demand (domanda totale di CPU)
- a job's "age": its total CPU usage thus far (tempo già usato, processo già attivo)
- a job's "lifetime" refers to its total CPU requirement (richiesta TOT. CPU, coincide con size)
- a job's "remaining lifetime" refers to its remaining CPU requirement (tempo rimanente)

Observe that commonly, at any point in time, you don't know the job's remaining lifetime, just it's current CPU age.

7

Measurements of CPU requirements of Unix jobs

jobs lifetime for 3 months, only those jobs with CPU lifetimes greater than one second

$\overline{F}(x) = \Pr\{X > x\}$

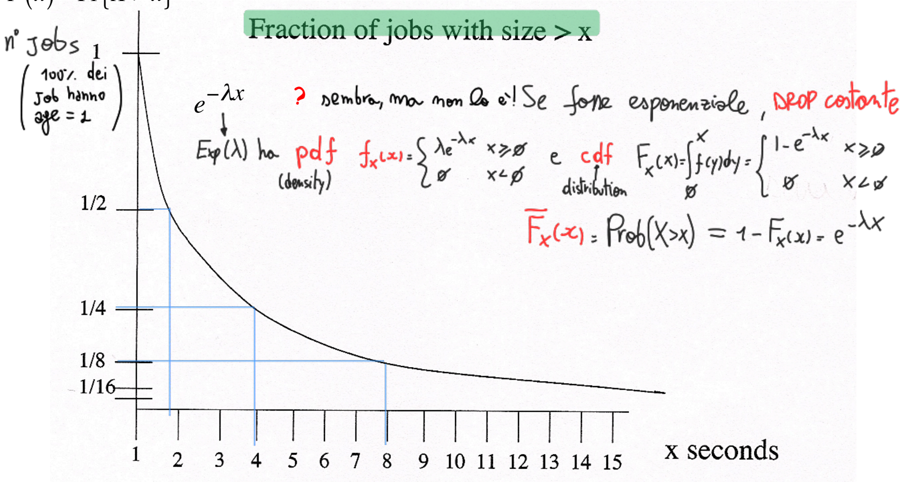
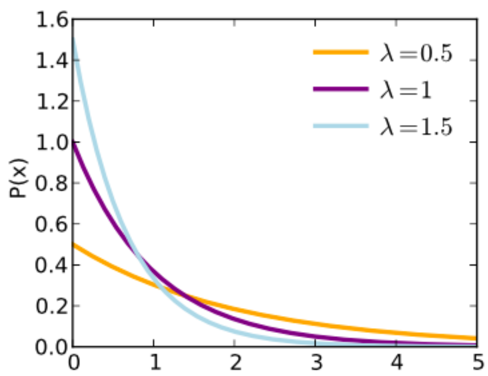


Figure 1: *Plot of measured distribution on non-log axes.*

8



$(0, \lambda)$ $(1, \lambda e^{-\lambda})$ $(2, \lambda e^{-\lambda \cdot 2})$ ...

Funzione densità PDF

$f(x) = \lambda e^{-\lambda x}$

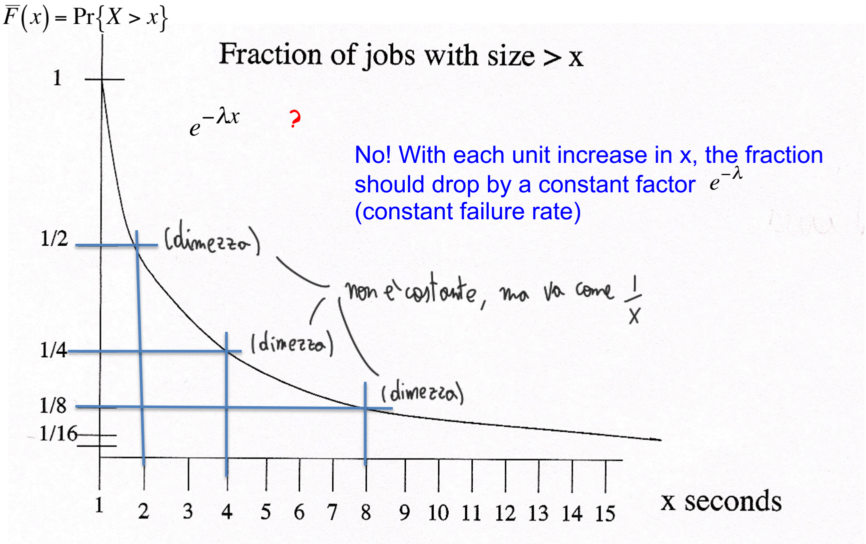
1. For each unit step, the "quantity" is reduced of $e^{-\lambda}$

λ is the *decay constant*

Measurements of CPU requirements of Unix jobs

jobs lifetime for 3 months, only those jobs with CPU lifetimes greater than one second

$\bar{F}(x) = \Pr\{X > x\}$



$e^{-\lambda x}$?

No! With each unit increase in x, the fraction should drop by a constant factor $e^{-\lambda}$ (constant failure rate)

Figure 1: *Plot of measured distribution on non-log axes.*

Analytical models

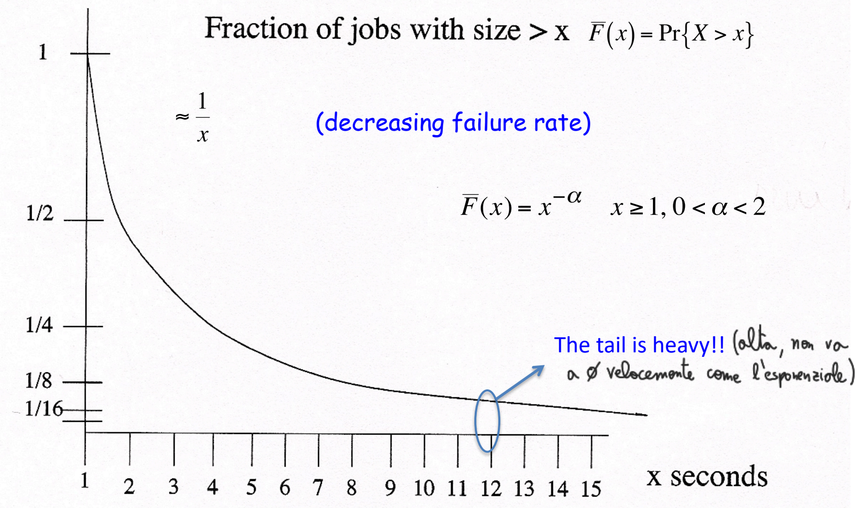


Figure 1: *Plot of measured distribution on non-log axes.*

11

Analytical models

Heavy tail distributions

Non è quindi exp, ma:

Pareto distributions

$$f(x) = \alpha k^{\alpha} x^{-\alpha-1} \quad k \leq x < \infty, 0 < \alpha < 2$$ (classe di distribuzioni)

α a measure of the distribution variability and of the "heavy-tailedness": (quanto coda si alza)

$\alpha \rightarrow 0$ +variability, +heavy

$\alpha \rightarrow 2$ -variability, -heavy

Problem: i-th moment is finite just for $\alpha > i$

$$E[X] = \frac{\alpha k}{\alpha - 1} \qquad \alpha > 1$$

$$var[X] = \frac{\alpha k^2}{(\alpha-1)^2(\alpha-2)} \qquad \alpha > 2$$

12

Properties of Pareto distributions

Decreasing Failure Rate (s ↗, prob ↗)

The more cpu you have used so far, the more you will continue to use.

completely different from the exponential distribution, where your cpu usage after any point in time is completely independent of the amount of cpu used up to that point (memoryless property)

Infinite Variance

"Heavy-Tail Property"

A miniscule fraction of the very largest jobs comprise half of the load on the system. (pochi job grandi occupano metà del sistema)

For example, when $\alpha = 1.1$, the largest 1% of the jobs comprise 1/2 of the load.

13

Bounded Pareto distributions (∃ riferimenti sul g-ex...)

the measured data has a *minimum* job lifetime and a *maximum* job lifetime.

Thus the measured data has all finite moments.

To model the measured data, we therefore want a Pareto distribution which has been truncated. (non voglio $k<x<\infty$, poiché i dati hanno momento finito, voglio togliere $x<\infty$)

$$f(x) = \alpha x^{-\alpha-1} \frac{k^{\alpha}}{1-\left(\frac{k}{p}\right)^{\alpha}} \qquad k \le x \le p,\ 0 < \alpha < 2$$

all of the moments are finite

The actual measured squared coefficient of variation values were (obviously) finite and were between 25 and 49 !!!

molto variabile $\quad C^2 = \frac{var}{mean^2}$, expo=1

14

1. Should we bother with P migration, or is NP enough?
2. If we are going to bother with P migration, which processes are worth migrating?
   That is, what is a good migration policy?

decreasing failure rate (s ↗, prob ↗), ha senso perchè sono propensi ad usare ancora molto la CPU

the DFR property leads us to think that it may pay to migrate old jobs.
The reasoning is that although an old job may have high migration cost, because it has accumulated a lot of memory, if the job is really old then it has a high probability of using a lot more cpu in the future, which means that the cost of migration can be amortized over a very long lifetime.

Sarebbe peggio migrare stato di un qualcosa che finisce. !!