







Anno	Motore	Evento	Stato attuale
1993	 W3Catalog	Lancio	Inattivo
	 Aliweb	Lancio	Inattivo
1993	 JumpStation	Lancio	Inattivo
	 WWW Worm	Lancio	Inattivo
	 WebCrawler	Lancio	Inattivo (Solamente un'interfaccia per Bing)
	 Go.com	Lancio	Inattivo
1994	 Infoseek	Lancio	Inattivo
	 Lycos	Lancio	Attivo
	 Spenki	Fondazione	Inattivo
	 AltaVista	Lancio	Inattivo, <a href="#">reindirizzato a Yahoo!</a>
	 Daum	Lancio	Attivo
1995	 Magellan	Lancio	Inattivo
	 Excite	Lancio	Inattivo, solo notizie
	 SAPO	Lancio	Inattivo, solo notizie
	 Yahoo!	Lancio	Attivo

Prof. Vittoria de Nitto Personè

1

	 Dogpile	Lancio	Attivo, Aggregatore
	 Inktomi	Fondazione	Inattivo, acquisito da Yahoo!
	 HotBot	Fondazione	Attivo
1996	 Arianna	Fondazione	Inattivo, inglobato in <a href="#">Libero</a>
	 Multisoft	Fondazione	Inattivo
	 Virgilio	Fondazione	Attivo, risultati Google
	 Ask Jeeves	Fondazione	Attivo (rimarchiato in ask.com)
1997	 Northern Light	Lancio	Inattivo
	 Yandex	Lancio	Attivo
	 Google	Lancio	Attivo
1998	 Ixquick	Lancio	Attivo anche come <i>Startpage</i>
	 MSN Search	Lancio	Inattivo, diventato Bing
	 empas	Lancio	Inattivo (fusa con NATE)
1999	 Alltheweb	Lancio	Inattivo (URL reindirizzato a <a href="#">Yahoo!</a> )
	 GenieKnows	Lancio	Inattivo
	 Naver	Lancio	Attivo
	 Teoma	Fondazione	Inattivo, reindirizza Ask.com
	 Vivísimo	Fondazione	Inattivo, IMB
	 superEva	Lancio	Inattivo, solo analisi trend

2

2000	 Baidu	Fondazione	Attivo
	 Exalead	Lancio	Attivo
	 Gigablast	Lancio	Attivo
2001	 Kartoo	Lancio	Inattivo
2003	 Info.com	Lancio	Attivo, risultati Bing
	 Scroogle	Lancio	Inattivo
2004	 Yahoo! Search	Lancio finale	Attivo (Solamente un'interfaccia per Bing)
	 A9.com	Lancio	Inattivo
	 Sogou	Lancio	Attivo
2005	 Windows Live Search	Lancio finale	Inattivo
	 GoodSearch	Lancio	Inattivo (cerca coupon con Google)
	 SearchMe	Lancio	Inattivo
2006	 Soso	Lancio	Attivo
	 Quaero	Fondazione	Inattivo
	 Search.com	Fondazione	Attivo
	 Ask.com	Lancio	Attivo
	 Windows Live Search	Lancio	Inattivo
	 ChaCha	Lancio beta	Inattivo
	 Guruii.com	Lancio beta	Inattivo

3

3

2007	 Wikiseek	Lancio	Inattivo
	 Sproose	Lancio	Inattivo
	 Wikia Search	Lancio	Inattivo
	 Blackle.com	Lancio	Inattivo (Solamente tema scuro di Google)
2008	 Cuil	Lancio (chiuso)	Inattivo
	 Powerset	Lancio	Inattivo
	 Picollator	Lancio	Inattivo
	 Viewzi	Lancio	Inattivo
	 Boogami	Lancio	Inattivo
	 LeapFish	Lancio	Inattivo
	 Forestle	Lancio	Inattivo (reindirizza a Ecosia)
	 Ecocho	Lancio	Inattivo
	 DuckDuckGo	Lancio	Attivo
2009	 Wolfram Alpha	Lancio	Attivo
	 Bing	Lancio	Attivo
	 Yebol	Lancio	Inattivo
	 Mugurdy	Lancio	Inattivo
	 Scout (Goby)	Lancio	Inattivo
	 Coozila!	Lancio	Inattivo
	 Ecosia	Lancio	Attivo (Bing + Algoritmi proprietari)
	 NATE	Lancio	Attivo

4





4

2010	 iAlgae	Lancio	Inattivo
	 Blekko	Lancio	Inattivo (venduto a IBM)
	 Cuil	Lancio	Inattivo
	 Yandex (Versione in Inglese)	Lancio	Attivo
2011	 YaCy	Lancio	Inattivo (online, non ricerca)
2012	 Volunia	Lancio (chiuso)	Inattivo
	 Ideao	Lancio beta	Inattivo
2013	 Istella	Lancio	Attivo
	 Qwant	Lancio	Attivo
	 Aoohe	Lancio	Inattivo
	 Coc Coc	Lancio	Attivo, motore di ricerca <a href="#">vietnamita</a>
2014	 Egerin	Lancio	Attivo, motore di ricerca in <a href="#">curdo</a> / <a href="#">Sorani</a>
	 Searx	Lancio	Attivo, metamatore di ricerca
	 Swisscows	Lancio	Attivo
2017	Xaphir	Beta	Inattivo, acquisito da <a href="#">Qwant</a> <sup>[3]</sup>

Prof. Vittoria de Nitto Personè

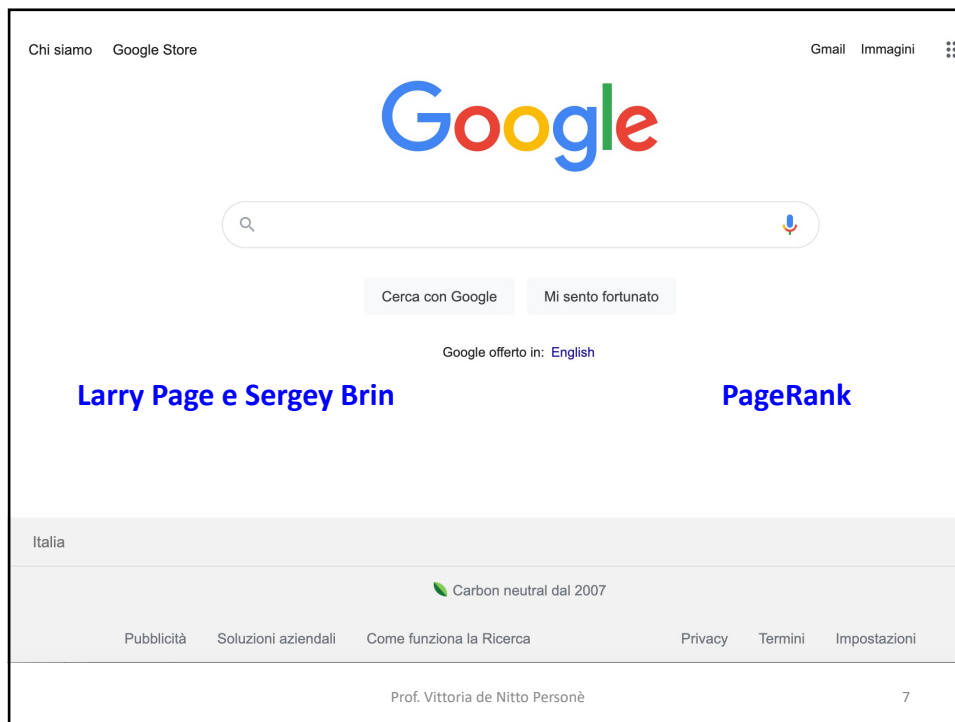
5

5

1996	 Dogpile	Lancio	Attivo, Aggregatore
	 Inktomi	Fondazione	Inattivo, acquisito da Yahoo!
	 HotBot	Fondazione	Attivo
	 Arianna	Fondazione	Inattivo, inglobato in <a href="#">Libero</a>
	 Multisoft	Fondazione	Inattivo
	 Virgilio	Fondazione	Attivo, risultati Google
	 Ask Jeeves	Fondazione	Attivo (rimarchiato in ask.com)
1997	 Northern Light	Lancio	Inattivo
	 Yandex	Lancio	Attivo
1998	 Google	Lancio	Attivo
	 Ixquick	Lancio	Attivo anche come <i>Startpage</i>
	 MSN Search	Lancio	Inattivo, diventato Bing
	 empas	Lancio	Inattivo (fusa con NATE)
1999	 Alltheweb	Lancio	Inattivo (URL reindirizzato a <a href="#">Yahoo!</a> )
	 GenieKnows	Lancio	Inattivo
	 Naver	Lancio	Attivo
	 Teoma	Fondazione	Inattivo, reindirizza Ask.com
	 Vivísimo	Fondazione	Inattivo, IMB
	 superEva	Lancio	Inattivo, solo analisi trend

6

Perché Google è il "migliore"? L'ordine dei risultati è fondamentale, in alto ci deve essere ciò che cerco. I creatori si sono basati sulla modellazione con probabilità, l'algoritmo usato è PageRank, con un'idea semplice dietro.



7

si basa su  
recursively

a page has high rank if the sum of the ranks of its backlinks is high

rank  $j$   $\equiv$  probability  $\pi_j$

↓

$$\pi_j = \sum_{i=1}^n \pi_i p_{ij}$$

$p(i,j)$  è prob che la pagina "i" punti a pagina "j"

si assegna un peso ad ogni pagina, in forma ricorsiva, in base al peso che hanno le pagine che puntano a questa pagina.  
Se siti con peso 'alto' puntano a me, allora avrò anche io peso alto.  
Il 'peso' è calcolato come sopra.

8

8

Devo evitare però dei "trucchi", potrei creare tante pagine che puntano alla mia per metterla in evidenza.

## Google's page rank algorithm

catena di markov a tempo discreto, esso è un grafo con nodi (stato del processo stocastico che attraversa, qui ad ogni pagina web ho uno stato) connessi con link (pagina che punta ad altra pagina)

1. Create a DTMC transition diagram where there is one state for each web page with connections for linked pages
2. If page  $i$  has  $k > 0$  links, state each probability to  $1/k$  (uniforme)
3. Solve the DTMC; page are ranked based on their limit probabilities

Prof. Vittoria de Nitto Personè

9

9

## Performance Modeling of Computer Systems and Networks

*Prof. Vittoria de Nitto Personè*

### Markov Process

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

Copyright © Vittoria de Nitto Personè, 2021  
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



10

Processo stocastico

collezione di variabili random, che variano nel tempo

$$\{X(t_1), X(t_2), \dots\}$$

Spazio degli stati

$$E = \{s_0, s_1, s_2, \dots\}$$

$s_0, s_1$  non hanno riferimento al tempo.  
E' solo lo spazio dei valori possibili.

Se tale insieme è finito, parlo di Catena di Markov

Catena di Markov ha memoryless property, stato futuro dipende solo dal presente, non da come ci arrivo.

$$P\{X(t_{n+1}) = x_{n+1} | \underbrace{X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0}_{s_0, s_1, \dots}\} =$$

$$= P\{X(t_{n+1}) = x_{n+1} | X(t_n) = x_n\}$$

dimostrazione di ciò che abbiamo appena detto.

Prof. Vittoria de Nitto Personè

11

11

## Probabilità stazionaria

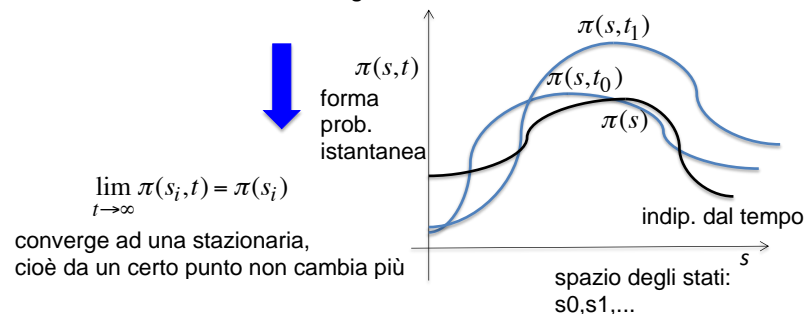
Distribuzione di **probabilità istantanea**

$$P\{X(t) = s_i\} = \pi(s_i, t)$$

- Spazio finito spazio dei valori possibili finito
- Processo irriducibile e ergodico

irriducibile: posso sempre raggiungere in un tot di passi uno stato partendo da un altro stato, non rimango mai bloccato.

ergodico: tempo in cui ritorno negli stati sono finiti, non periodici, quindi non capita che con periodi fissi torno in un determinato stadio. Passo per tutti i punti possibili di lavoro.



Prof. Vittoria de Nitto Personè

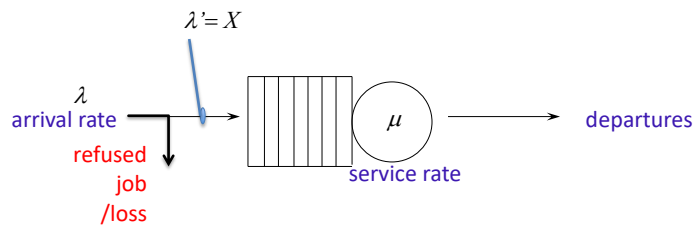
12

12

non posso usare KP, perchè non so dire tasso che entra

Analytical models  
basic laws

## Single server center with finite buffer



Each arrival when the queue is *full* will be lost

Which is the throughput?

~~$X \neq \lambda$~~   
 ~~$\rho = \lambda / \mu$~~   
No!  
On the contrary  
 $X < \lambda$

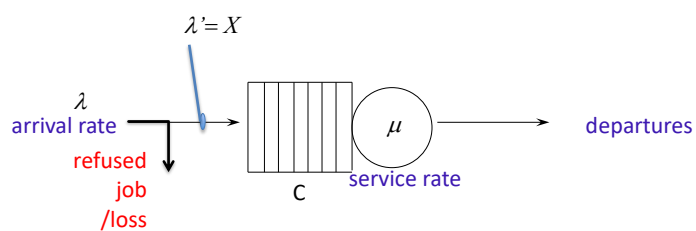
Prof. Vittoria de Nitto Personè

13

13

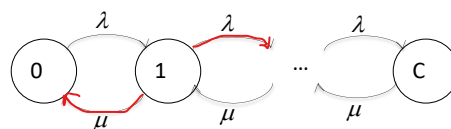
Analytical models  
basic laws

## Single server center with finite buffer



$X(t) \equiv$  n° di job nel centro Posso vederla solo con Markov

$E = \{0, 1, 2, \dots, C\}$  C+1 stati, può essere anche vuoto.

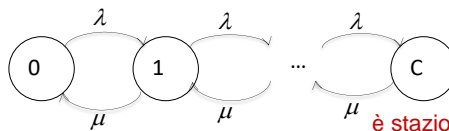


per gli stati nel mezzo posso andare avanti o indietro

Prof. Vittoria de Nitto Personè

14

14



equazione di bilanciamento globale

$$\pi_0 \lambda = \pi_1 \mu \Rightarrow \pi_1 = \frac{\lambda}{\mu} \pi_0$$

esco da stato 1 = entro in stato 1

$$\pi_1 (\lambda + \mu) = \pi_0 \lambda + \pi_2 \mu$$

$$\pi_2 \mu = \frac{\lambda}{\mu} (\lambda + \mu) \pi_0 - \pi_0 \lambda \Rightarrow \pi_2 = \left( \frac{\lambda}{\mu} \right)^2 \pi_0$$

...

$$\pi_C = \left( \frac{\lambda}{\mu} \right)^C \pi_0$$

$$\pi_i = \left( \frac{\lambda}{\mu} \right)^i \pi_0$$

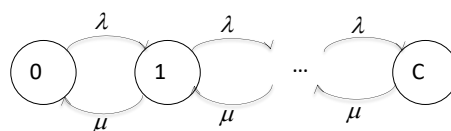
NON E' RHO, perchè non entra tutto lambda.

è stazionario perchè ha capacità finita, non può mai esplodere.

prob. stato 1 in funzione di stato 0.

15

il grafo è dipendente, c+1 variabili, equazioni dipendenti.



$$\sum_{i=0}^C \pi_i = 1$$

$$\sum_{i=0}^C \left( \frac{\lambda}{\mu} \right)^i \pi_0 = 1 \Rightarrow \pi_0 = \frac{1}{\sum_{i=0}^C \left( \frac{\lambda}{\mu} \right)^i}$$

16



The top diagram shows a queue system with an arrival rate  $\lambda$  and a service rate  $\mu$ . A blue arrow labeled  $p_{\text{loss}}$  indicates the probability of loss. The bottom diagram shows a Markov chain with states 0, 1, ..., C. Transitions are labeled with  $\lambda$  (forward) and  $\mu$  (backward).

La probabilità di perdita coincide con la probabilità di stare nell'ultimo stato, perchè lì se arrivano altri li perdo sicuri.

$$p_{\text{loss}} = \pi_C = \left(\frac{\lambda}{\mu}\right)^C \pi_0 = \frac{\left(\frac{\lambda}{\mu}\right)^C}{\sum_{i=0}^C \left(\frac{\lambda}{\mu}\right)^i}$$

Prof. Vittoria de Nitto Personè 17

17

The top diagram shows a queue system with a finite buffer of size C. The arrival rate is  $\lambda$  and the service rate is  $\mu$ . A blue arrow labeled  $\lambda' = X$  indicates the effective arrival rate. The bottom diagram shows a Markov chain with states 0, 1, ..., C. Transitions are labeled with  $\lambda$  (forward) and  $\mu$  (backward). A red arrow points to the state C transition, labeled "probabilità di stare nello stato C".

$\lambda' = \lambda (1 - p_{\text{loss}})$

$\rho = \lambda' / \mu$

se avesse coda infinita esplode perchè  $\rho = 1$

Es.:

$C=4$   
 $\lambda = \mu = 5 \text{ j/s}$   
 $\pi_i = \left(\frac{\lambda}{\mu}\right)^i \pi_0$   
 $\pi_1 = \pi_2 = \pi_3 = \pi_4 = \pi_0 = \frac{1}{5}$

$p_{\text{loss}} = \pi_4 = \frac{1}{5}$   
 $\lambda' = 5 (1 - 1/5) = 4 = X$   
 $\rho = 0,8$  perchè uso  $\lambda'$  e non  $\lambda$

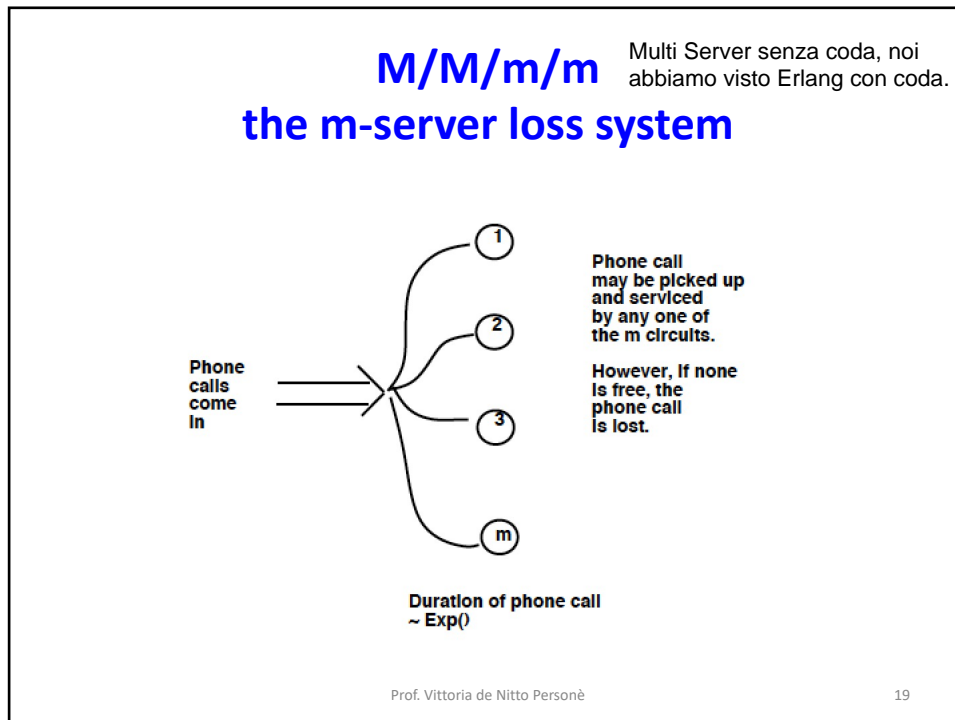
Serve processo di markov, in questo caso si parla di processo di nascita e morte, e ha una soluzione chiusa (una sola formula).

Prof. Vittoria de Nitto Personè 18

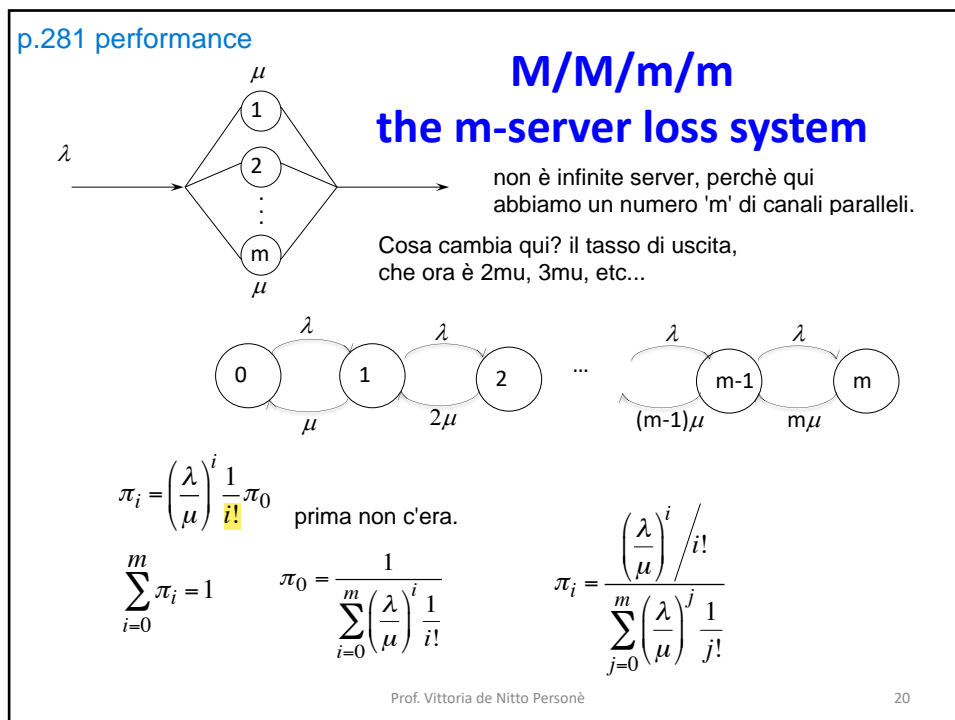
18

Le distribuzioni generali, modellabili con esponenziali, ci permettono di ricadere sempre in Markov.

Per risolvere la catena finita, ci vuole Markov. La erlang è la soluzione che si ottiene risolvendo un processo di Markov.

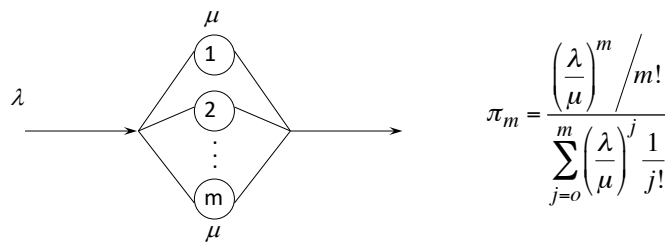


19



20

## The Erlang-B formula



Se lo confronto con quello con 'coda', quest ultimo ha maggior probabilità di essere pieno, perchè la coda è come un 'buffer' di arrivi pronti a chiedere servizio. Senza coda, dovrei trovarmi nel caso che, appena un servizio termina, arrivi un altro job a prenderne il posto (nella 'coda' invece li accumulo).

Prof. Vittoria de Nitto Personè

21

21

## The Erlang-C formula

## The Erlang-B formula

$$P_Q = \frac{(m\rho)^m}{m!(1-\rho)} p(0)$$

$$= \frac{\left(\frac{\lambda}{\mu}\right)^m}{m!(1-\rho)} p(0)$$

>

$$\pi_m = \frac{\left(\frac{\lambda}{\mu}\right)^m}{m!} \pi_0$$

$p(0)$

<

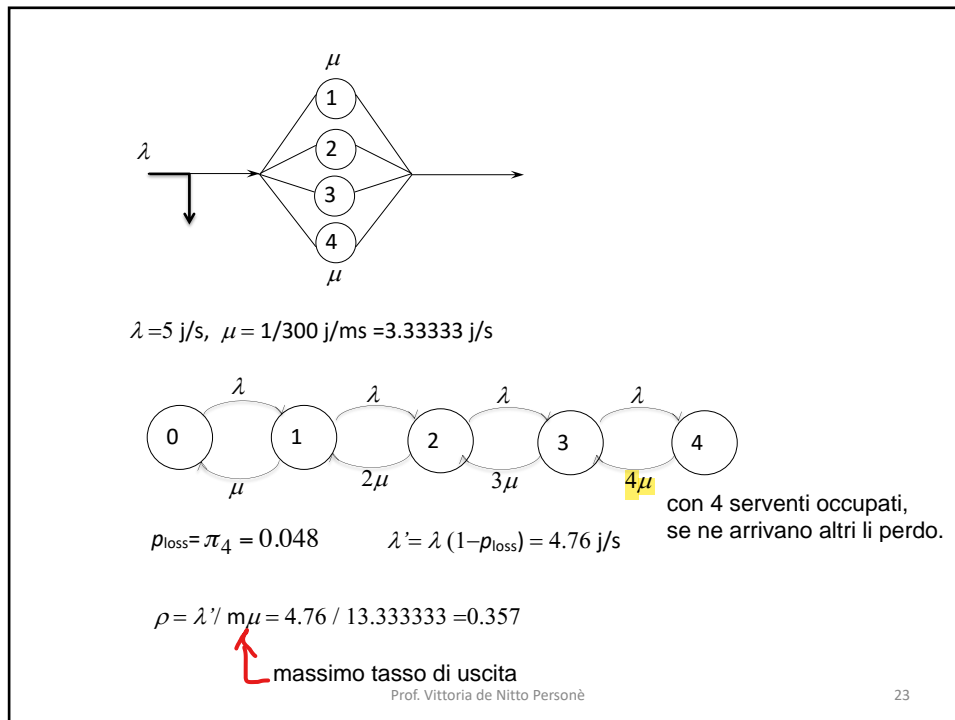
$\pi_0$

probabilità che sia vuoto con coda è più piccola, perchè la 'coda' è come un buffer, e quindi c'è qualcuno che aspetta il servizio. Nell'erlang B non ho questa sicurezza, dovrebbe arrivare qualcuno mentre ho finito un servizio.

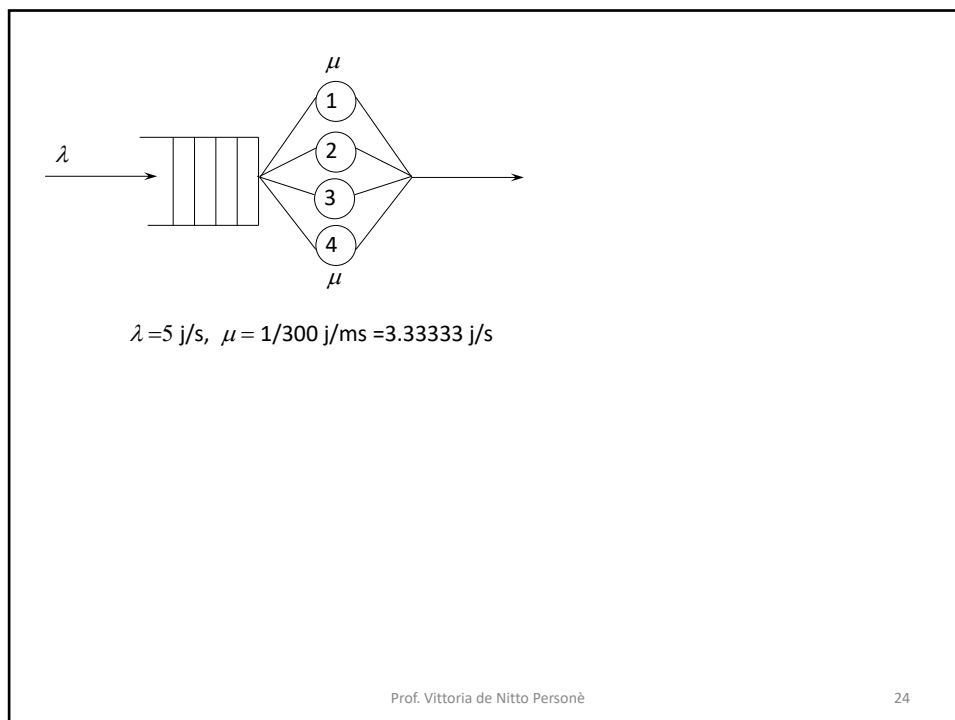
Prof. Vittoria de Nitto Personè

22

22



23



24

### esempio esercizio esame

Consider a single-core server hosting a web service. Requests arrive to the server according to a Poisson, with an average inter-arrival time of 200 ms. Knowing that the maximum buffer size is  $N = 4$  (including the jobs in service) and that each request requires on average 200 ms of processing time,

1. State if the system is stationary and explain the reason
2. compute the system utilization
3. compute the system throughput. 4 paralleli.

Consider a CPU upgrade to a slower quad-core processor, which can process a request in 300 ms using one of its processor cores. Compute the throughput of the upgraded system.

servente singolo che ospita web server, arrivo Poisson con interarrivo 200ms,  $N = C$  (noi l'abbiamo chiamata così) = 4, incluso job in servizio.  
 $E[S] = 200$  ms.

1 - Essendo a capacità finita è stazionario (anche calcolando  $\lambda$  si vede).

3 - il throughput è semplicemente quello che entra, se il sistema è stabile come in questo caso.