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

1

1996	Dogpile	Lancio	Attivo, Aggregatore
	Inktomi	Fondazione	Inattivo, acquisito da Yahoo!
	HotBot	Fondazione	Attivo
	Arianna	Fondazione	Inattivo, inglobato in Libero
	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 *Startpage*
	MSN Search	Lancio	Inattivo, diventato Bing
	empas	Lancio	Inattivo (fusa con NATE)
1999	Alltheweb	Lancio	Inattivo (URL reindirizzato a Yahoo!)
	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
	Guruji.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

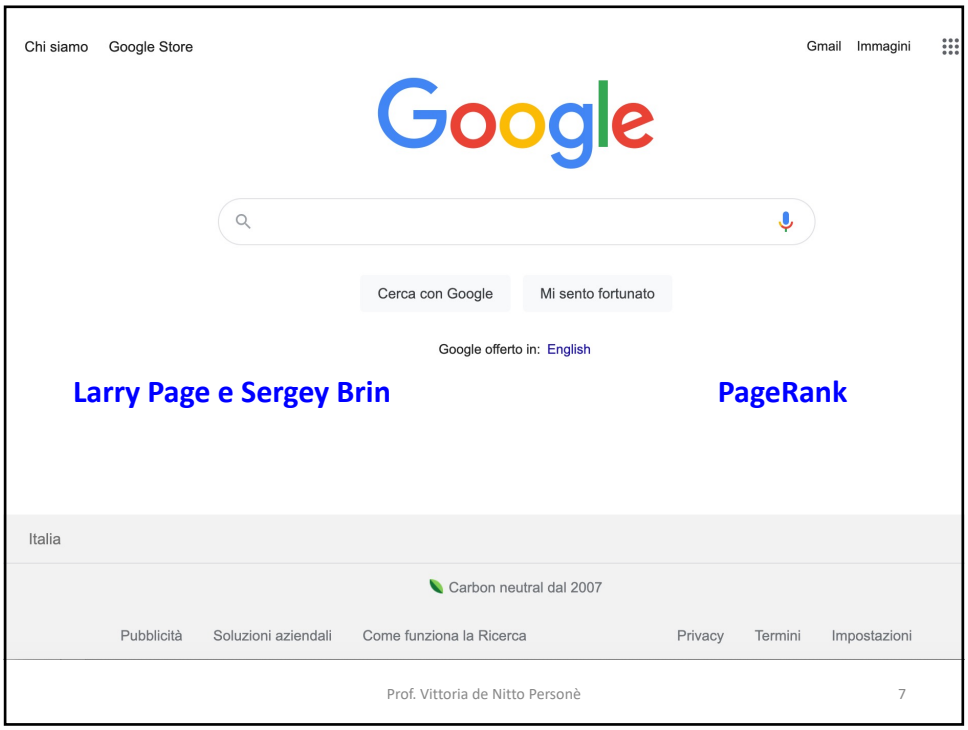
4

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 vietnamita
	Egerin	Lancio	Attivo, motore di ricerca in curdo / Sorani
2014	Searx	Lancio	Attivo, metamotore di ricerca
	Swisscows	Lancio	Attivo
2017	Xaphir	Beta	Inattivo, acquisito da Qwant[3]

5

1996	Dogpile	Lancio	Attivo, Aggregatore
	Inktomi	Fondazione	Inattivo, acquisito da Yahoo!
	HotBot	Fondazione	Attivo
	Arianna	Fondazione	Inattivo, inglobato in Libero
	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 *Startpage*
	MSN Search	Lancio	Inattivo, diventato Bing
	empas	Lancio	Inattivo (fusa con NATE)
1999	Alltheweb	Lancio	Inattivo (URL reindirizzato a Yahoo!)
	GenieKnows	Lancio	Inattivo
	Naver	Lancio	Attivo
	Teoma	Fondazione	Inattivo, reindirizza Ask.com
	Vivísimo	Fondazione	Inattivo, IMB
	superEva	Lancio	Inattivo, solo analisi trend

6



7

recursively

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

rank j ≅ probability π_j

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

8

Google's page rank algorithm

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
3. Solve the DTMC; page are ranked based on their limit probabilities

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

10

Processo stocastico

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

Spazio degli stati

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

Catena di Markov

$$P\{X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \ldots, X(t_0) = x_0\} =$$

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

11

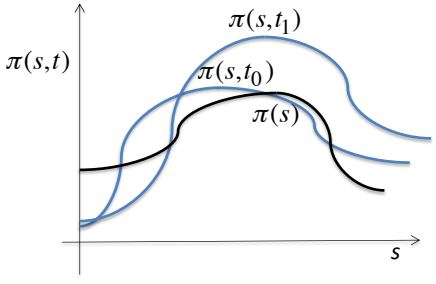
Probabilità stazionaria

Distribuzione di probabilità istantanea

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

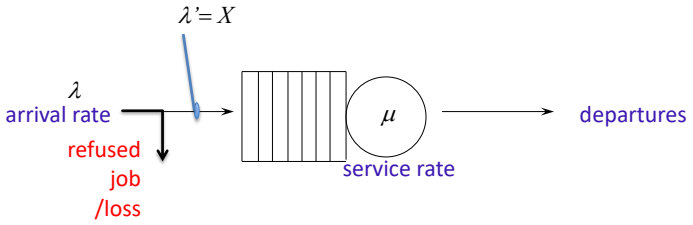
- Spazio finito
- Processo irriducibile e ergodico

$$\lim_{t \to \infty} \pi(s_i, t) = \pi(s_i)$$



12

Single server center with finite buffer



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

Which is the throughput?

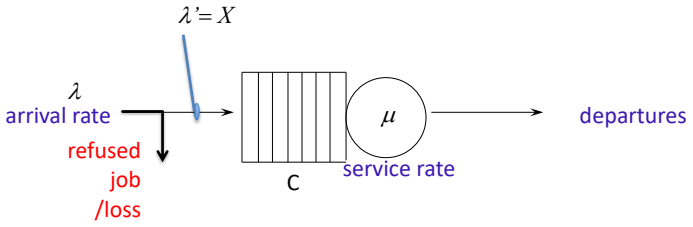
$X = \lambda$ $\quad \rho = \lambda / \mu$

No!
On the contrary

$X < \lambda$

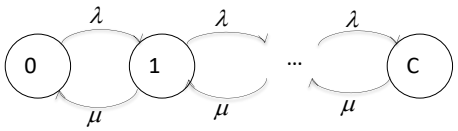
13

Single server center with finite buffer

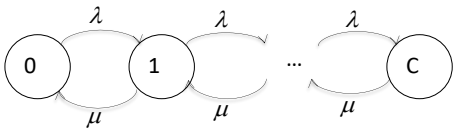


X(t) ≅ n° di job nel centro

$E = \{0,1,2,\ldots,C\}$



14



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

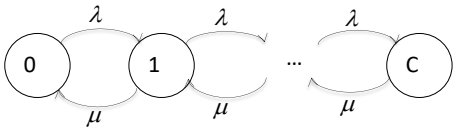
$$\pi_1(\lambda+\mu) = \pi_0\lambda + \pi_2\mu$$

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

.
.
.

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

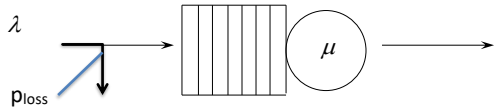
15

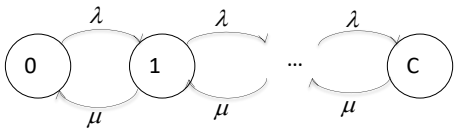


$$\sum_{i=0}^{C} \pi_i = 1$$

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

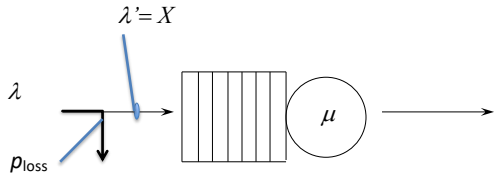
16





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

17



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

$\rho = \lambda' / \mu$

Es.:

C=4

$\lambda = \mu = 5$ j/s

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

$$\pi_1 = \pi_2 = \pi_3 = \pi_4 = \pi_0 = \frac{1}{5}$$

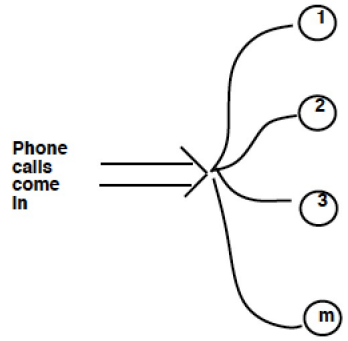
$$p_{loss} = \pi_4 = \frac{1}{5}$$

$\lambda' = 5\,(1 - 1/5) = 4 = X$

$\rho = 0{,}8$

18

M/M/m/m
the m-server loss system



Phone calls come in

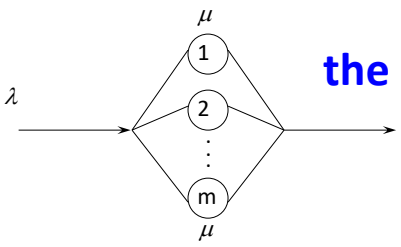
Phone call may be picked up and serviced by any one of the m circuits.

However, if none is free, the phone call is lost.

Duration of phone call ~ Exp()

19

M/M/m/m
the m-server loss system

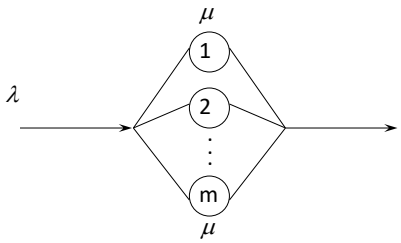


$$\pi_i = \left(\frac{\lambda}{\mu}\right)^i \frac{1}{i!}\pi_0$$

$$\sum_{i=0}^{m} \pi_i = 1 \qquad \pi_0 = \frac{1}{\sum_{i=0}^{m}\left(\frac{\lambda}{\mu}\right)^i \frac{1}{i!}} \qquad \pi_i = \frac{\left(\frac{\lambda}{\mu}\right)^i \Big/ i!}{\sum_{j=0}^{m}\left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}}$$

20

The *Erlang*-B formula



$$\pi_m = \frac{\left(\frac{\lambda}{\mu}\right)^m \Big/ m!}{\sum_{j=o}^{m} \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}}$$

21

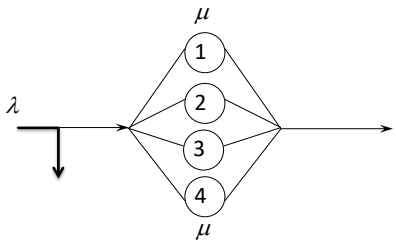
The *Erlang*-C formula

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

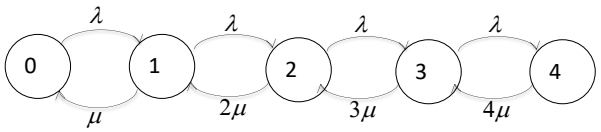
$$= \frac{\left(\frac{\lambda}{\mu}\right)^m}{m!(1-\rho)} p(0) \quad > \quad \pi_m = \left(\frac{\lambda}{\mu}\right)^m \frac{1}{m!} \pi_0$$

$$p(0) \quad < \quad \pi_0$$

22



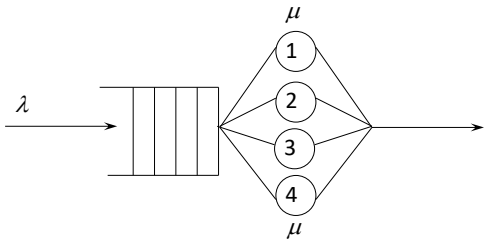
λ =5 j/s, μ = 1/300 j/ms =3.33333 j/s



$p_{loss} = \pi_4 = 0.048$ $\lambda' = \lambda (1 - p_{loss}) = 4.76$ j/s

$\rho = \lambda' / m\mu = 4.76 / 13.333333 = 0.357$

23



λ =5 j/s, μ = 1/300 j/ms =3.33333 j/s

24

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.

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.