Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

The multi-server queue

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

1

P. 37 slide

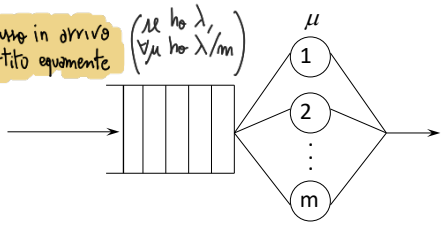
- t. servizio esp (μ) (continua) — M = "esponenziale continua memoryless" ↳ non invecchia
- arrivi Poisson (λ) (discreta)

Erlang, 1917

M/M/m abstract scheduling

multiserver

flusso in arrivo ripartito equamente ($\forall\mu$ ho λ, $\forall\mu$ ho λ/m)



$E(N_Q)_{Erlang}$

$p_{fifo} = p_i$, cambia solo il significato che gli do.

probabilità marginale di avere n job nel centro.

$$p(n) = \begin{cases} \frac{1}{n!}(m\rho)^n p(0) & for\ n = 1,\dots,m \\ \frac{m^m}{m!}\rho^n p(0) & for\ n > m \end{cases}$$

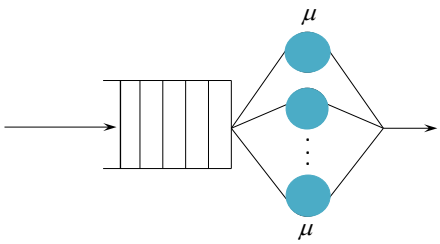
(si forma coda)

$$p(0) = \left[\sum_{i=0}^{m-1}\frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)}\right]^{-1}$$

probabilità che il centro sia vuoto.

2

The *Erlang*-C formula



Prob. tutti i server pieni

$$P_Q \cong \Pr\{n \geq m\} = \sum_{n=m}^{\infty} p(n)$$

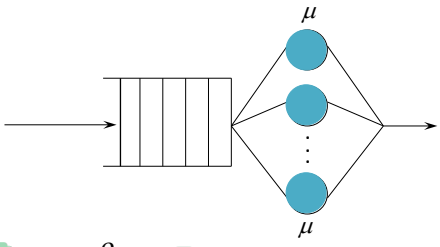
$$= \sum_{n=m}^{\infty} \frac{m^m}{m!}\rho^n p(0) = \frac{m^m}{m!} p(0) \sum_{n=m}^{\infty} \rho^n$$

$$= \frac{m^m}{m!} p(0) \sum_{n=0}^{\infty} \rho^{n+m} = \frac{m^m}{m!} p(0)\rho^m \sum_{n=0}^{\infty} \rho^n$$

$$\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho}$$

3

The *Erlang*-C formula



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

in coda

$E(N_Q)$ *Erlang* $= P_Q \frac{\rho}{1-\rho}$ ⟶ $E(N_S) = P_Q \frac{\rho}{1-\rho} + m\rho$

tutti server pieni

↑ pop. nel centro

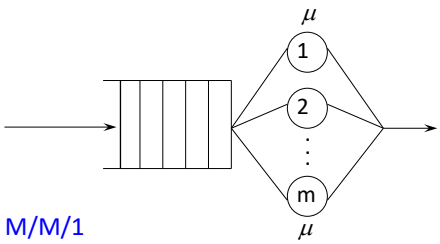
Little's law

$$E(T_Q) = \frac{E(N_Q)}{\lambda} \qquad E(T_Q) = P_Q \frac{\rho}{\lambda(1-\rho)} = \frac{P_Q E(S)}{1-\rho}$$

4

The *Erlang* formula

La erlang e' per l'esponenziale!



M/M/m

$$E(T_Q)_{Erlang} = \frac{P_Q E(S)}{1-\rho}$$

M/M/1

$$E(T_Q)_{KP} = \frac{\rho E(S)}{1-\rho} = \frac{E(S_{rem})}{1-\rho}$$

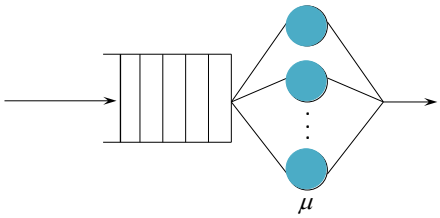
se ne libera uno qualsiasi

$$E(S) = \frac{E(S_i)}{m} = \frac{1}{m\mu}$$

5

The Multi Server Queue

c ≅ busy servers $(0,\dots,m)$



$$E(c) = \sum_{n=0}^{m-1} np(n) + \sum_{n=m}^{\infty} mp(n) = m\rho$$

probabilità di 'n'

server occupati sono sempre 'm'

P_Q (tutti pieni)

$$\rho = \sum_{n=0}^{m-1} \frac{n}{m} p(n) + \sum_{n=m}^{\infty} p(n) = \sum_{n=0}^{m-1} \frac{n}{m} p(n) + P_Q \quad \Rightarrow \quad \rho \geq P_Q$$

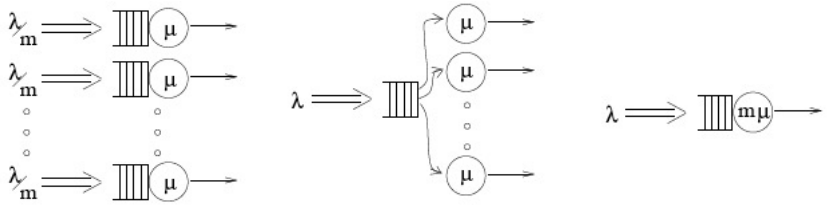
$$E(T_Q)_{Erlang} = \frac{P_Q E(S)}{1-\rho} \leq E(T_Q)_{KP} = \frac{\rho E(S)}{1-\rho}$$

6

P. 289 perf, p. 39 slide

Server Organizations

quale conviene? DIPENDE SEMPRE dal contesto!



• service rate $m\mu$ • $\rho = \frac{\lambda}{m\mu}$ per tutti

7

Communication systems

1
2
3
m

communication line

independent Poisson packet streams

each with an arrival rate of λ/m packets per second

the transmission time for each packet Exponential(1/μ)

- Frequency-division Multiplexing
- Statistical Multiplexing

8

Communication systems

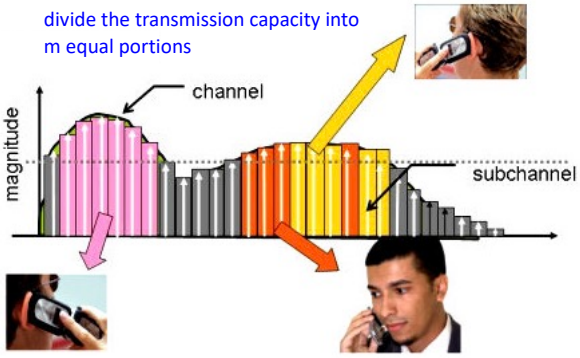
Frequency-division Multiplexing (separo canale, do garanzie!)

divide the transmission capacity into
m equal portions

1
2
3

m

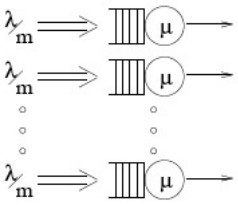
separated streams



9

Communication systems

Frequency-division Multiplexing



1° modello visto

10

Communication systems

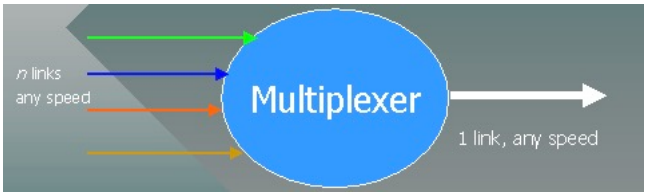
Statistical Multiplexing

keep the transmission capacity as a whole

1
2
3

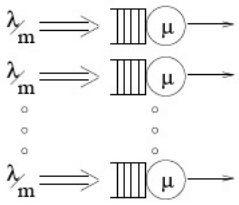
m

merge into a single stream



Communication systems

Frequency-division Multiplexing



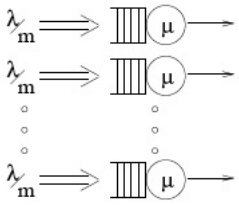
Statistical multiplexing

condivido la capacità comunicativa, c'è un "merge"

How do the two approaches compare with respect to mean response time?

Communication systems

Frequency-division Multiplexing



Statistical multiplexing

$$E(T_S) = \frac{\rho E(S)}{1-\rho} + E(S) = \frac{E(S)}{1-\rho}$$

$$E(T_S) = \frac{1}{\mu\left(1-\frac{\lambda}{\mu}\right)} = \frac{1}{\mu-\lambda}$$

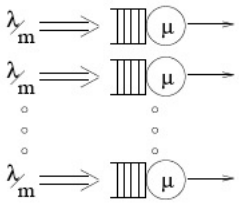
M/M/1

λ μ

Communication systems

(tutti e 3 i centri sono M/M/1)

Frequency-division Multiplexing



Statistical multiplexing

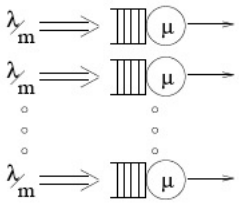
$$E(T_S)^{FDM} = \frac{1}{\mu - \frac{\lambda}{m}} = \frac{m}{m\mu - \lambda}$$

$$E(T_S)^{SM} = \frac{1}{m\mu - \lambda}$$

FDM shows a response time *m* times greater then for *SM* !

Communication systems

Frequency-division Multiplexing



QoS guarantee for each stream:
a specific service rate to each stream

Statistical multiplexing

No QoS guarantee

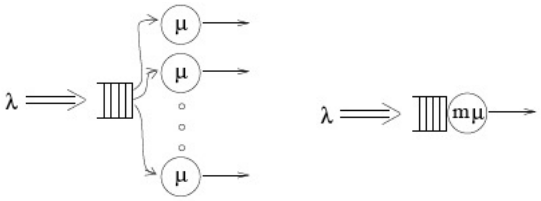
1

2 If the original m streams were very regular (not Poisson), i.e., they were much less variable than Poisson, by merging them, we introduce lots of variability into the arrival stream. This leads to problems if the application requires a low variability in delay, e.g., voice or video.

15

Server Organizations

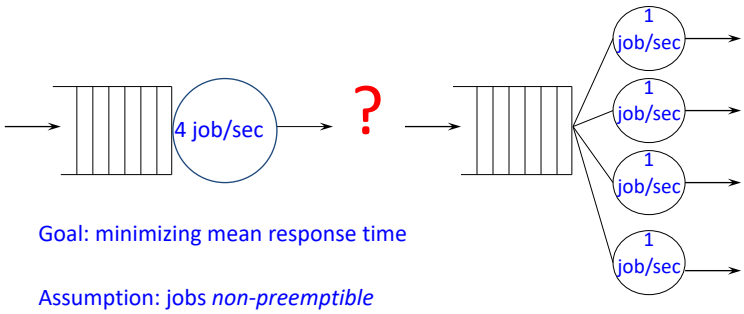
ancora da confrontare



16

Modelling power: Design tool

Design Example 4: One machine or many?



Goal: minimizing mean response time

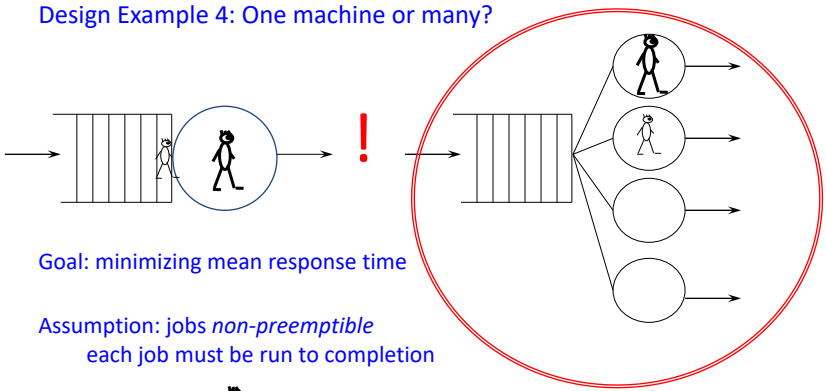
Assumption: jobs *non-preemptible*
each job must be run to completion

high variability 

17

Modelling power: Design tool

Design Example 4: One machine or many?



Goal: minimizing mean response time

Assumption: jobs *non-preemptible*
each job must be run to completion

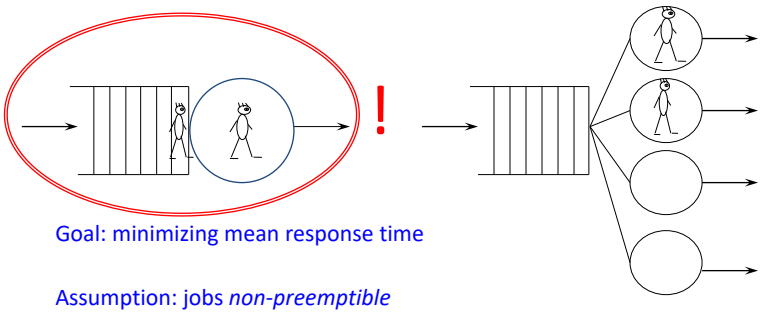
high variability 

Non voglio che i job più grandi blocchino i più piccoli

18

Modelling power: Design tool

Design Example 4: One machine or many?



Goal: minimizing mean response time

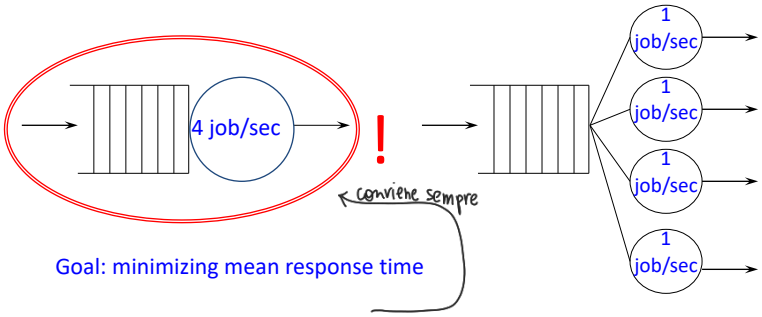
Assumption: jobs *non-preemptible*
each job must be run to completion

low variability 

19

Modelling power: Design tool

Design Example 4: One machine or many?



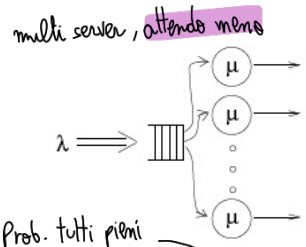
Goal: minimizing mean response time

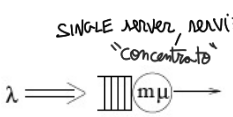
Assumption: jobs *preemptible*
each job can be stopped and restarted where they left off

Se c'è big job in corso, e arriva uno piccolo, posso stoppare big in favore di little

20

Server Organizations





Prob. tutti pieni

tempo attesa medio $E(T_Q)_{Erlang} = \frac{P_Q E(S)}{1-\rho}$

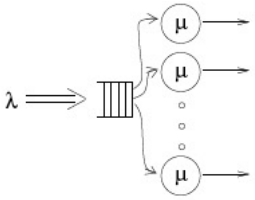
servente singolo

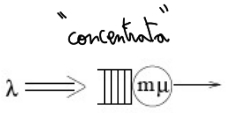
$E(T_Q)_{KP} = \frac{\rho E(S)}{1-\rho}$

$\rho \geq P_Q$ ⟹ from the waiting time perspective the distributed capacity solution produces an improvement in the user perceived QoS

Tempo servizio ciascuno

Server Organizations





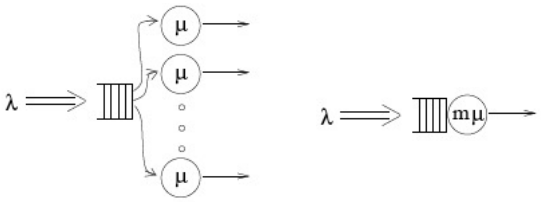
What about the response time perspective??

$E(T_S)_{Erlang} = \frac{P_Q E(S)}{1-\rho} + E(S_i)$ (servizio singolo $\frac{1}{\mu}$)

$E(T_S)_{KP} = \frac{\rho E(S)}{1-\rho} + E(S)$ ($E(S) = \frac{1}{m\mu}$)

$E(S_i) = \frac{1}{\mu} = m\frac{1}{m\mu} = mE(S)$

Server Organizations



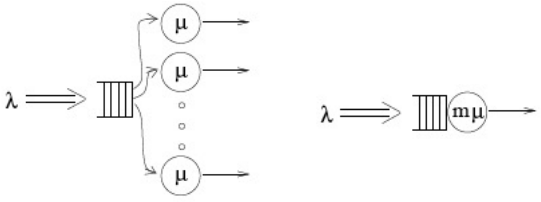
What about the response time perspective??

(distribuita) $E(T_S)_{Erlang} = \frac{P_Q E(S)}{1-\rho} + mE(S)$

Decreasing less than linear ∧ ∨ linear growth

$E(T_S)_{KP} = \frac{\rho E(S)}{1-\rho} + E(S)$ chi influisce di più?

Server Organizations



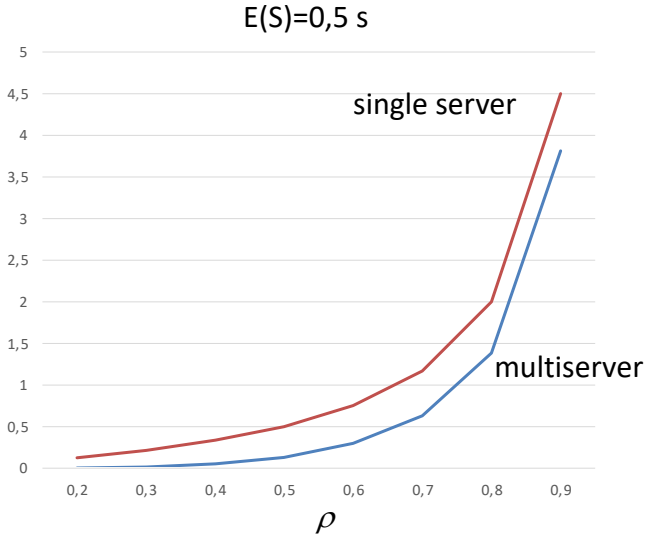
Performance goal:

Waiting time perspective — Distributed capacity

- $\rho \to 0$: distrib. capac. gives an *m* times slower organization
  (attesa non c'è quasi mai, meglio servizio breve)
- $\rho \to 1$: approximately the same response time
  (vanno a ∞ le attese nella coda)

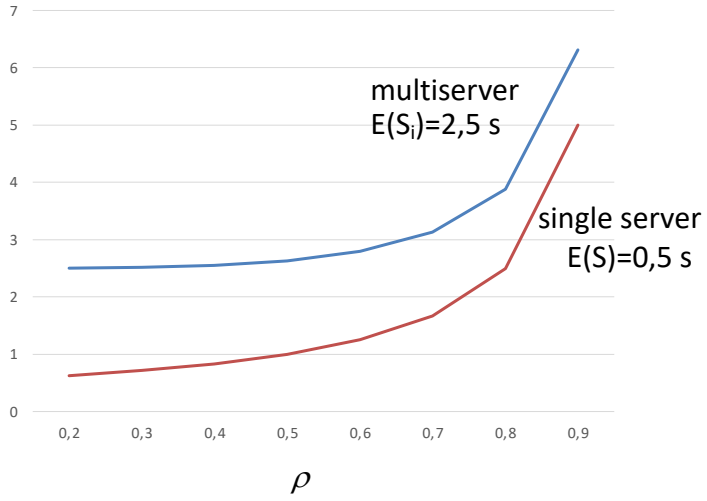
Response time perspective

Waiting time perspective



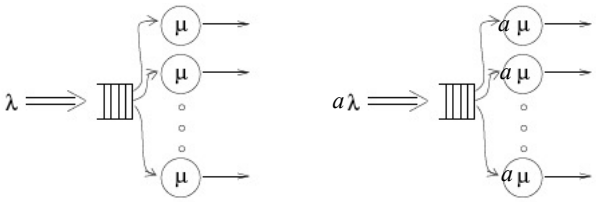
25

Response time perspective



26

Scaling factor

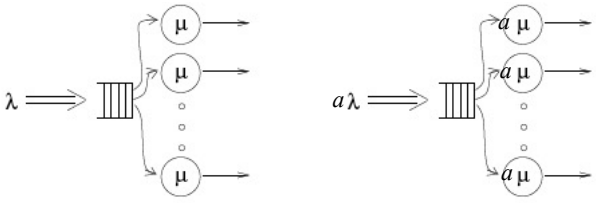


What about waiting and response time?

$$\rho = \frac{\lambda}{m\mu} \qquad E(S_i) = \frac{1}{\mu} \qquad E(S) = \frac{1}{m\mu}$$

$$\rho = \frac{a\lambda}{ma\mu} = \frac{\lambda}{m\mu} \qquad E(S_i) = \frac{1}{a\mu} \qquad E(S) = \frac{E(S_i)}{m} = \frac{1}{am\mu}$$

Scaling factor



Mean waiting time

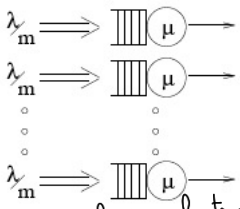
$$E(T_Q)_{m,a} = \frac{P_Q E(S)_{m,a}}{1-\rho} = \frac{1 \cdot P_Q}{ma\mu(1-\rho)} = \frac{1}{a}\frac{P_Q E(S)_{m,1}}{(1-\rho)} = \frac{1}{a}E(T_Q)_{m,1}$$

ERLANG — tempo servizio globale — serventi — fattori scala

$E(S) = \frac{E(S_i)}{m} = \frac{1}{am\mu}$

$E(T_S)_{m,a} = E(T_a)_{m,a} + \frac{1}{a\mu} = \frac{1}{a}\left(E(T_S)_{m,1}\right)$

Config. isolate



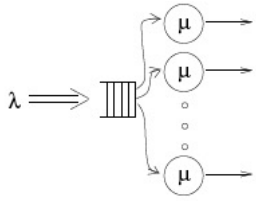
(è tutto esponenziale, per confronto multiserver)

$\lambda = 4$ j/s, m=4, μ=1.5 j/s E(S)=0.666667 s

$$\rho = 0.666667 = \left[\frac{\lambda}{m}\right]\cdot\frac{1}{\mu} = \frac{1}{1,5} < 1$$

$$E(T_S) = \frac{1}{\mu - \left(\frac{\lambda}{m}\right)} = 2\ s = \frac{1}{1,5-1}$$

$$E(T_Q) = \frac{\rho E(S)}{1-\rho} = 1.3336$$



$$E(S) = \frac{E(S_i)}{4} = 0.1667$$ (tempo medio che se ne liberi 1 di 4)

$$\rho = 0.666667$$

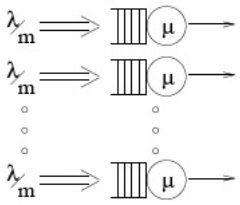
$$p(0) = \left[\sum_{i=0}^{3}\frac{(4\rho)^i}{i!} + \frac{(4\rho)^4}{4!(1-\rho)}\right]^{-1}$$

$$= \left[1 + 4\rho + \frac{(4\rho)^2}{2} + \frac{(4\rho)^3}{6} + \frac{(4\rho)^4}{24(1-\rho)}\right]^{-1} = 0.059857$$

p(0) = sistema vuoto in coda e in servizio
(da sist. stazionario, è vuoto il 6% del tempo di osservazione)

29

29

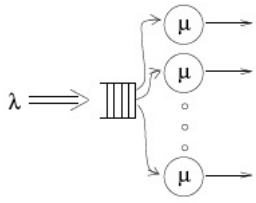


$\lambda = 4$ j/s, m=4, μ=1.5 j/s E(S)=0.666667 s

$$\rho = 0.666667$$

$$E(T_S) = \frac{1}{\mu - \lambda} = 2\ s$$

$$E(T_Q) = \frac{\rho E(S)}{1-\rho} = 1.3336$$



$$E(S) = \frac{E(S_i)}{4} = 0.1667$$

$E(S_i)$=0.666667 s

$\rho = 0.666667$ (occupazione ∀ server)

$$P_Q = \frac{(4\rho)^4}{4!(1-\rho)}p(0) = 0.37847$$ (TUTTI PIENI)

$$E(T_S) = \frac{P_Q E(S)}{1-\rho} + E(S_i) = 0.855992$$

└ singolo

$$E(T_Q) = 0.189292$$

(quando prendo servizio, uno E(Si) perché "colpo" su server con quel tempo

30

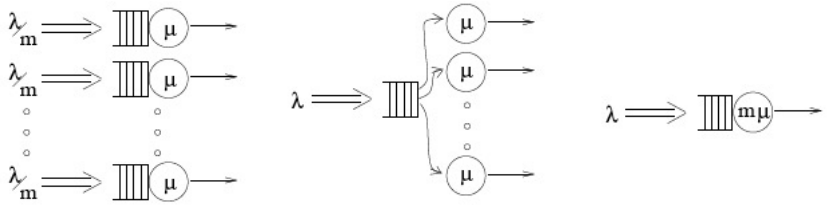
$\lambda = 4$ j/s, $m\mu = 4\text{x}1.5 = 6$ j/s E(S)=0.166667 s

$\rho = 0.666667$ (come prima, ma me lo aspettavo)

$$E(T_S) = \frac{1}{m\mu - \lambda} = 0.5$$

$E(T_Q) = 0.3334$ (tempo attesa $\frac{\rho E(S)}{1-\rho}$)

31



$\rho = 0.666667$

risposta minima

$$E(T_S) = \frac{1}{\mu - \lambda} = 2 \quad \geqslant \quad E(T_S) = 0.855992 \quad \geqslant \quad E(T_S) = \frac{1}{\mu - \lambda} = 0.5\ s$$

$$E(T_Q) = \frac{\rho E(S)}{1-\rho} = 1.3336 \quad \geqslant \quad E(T_Q) = 0.189292 \quad \leq \quad E(T_Q) = 0.3334$$

attesa minima multiserver

ESERCIZIO PER CASA: $\rho = 0,533334$ (20% in meno) $\rho = 0,8$ (20% in più)

32