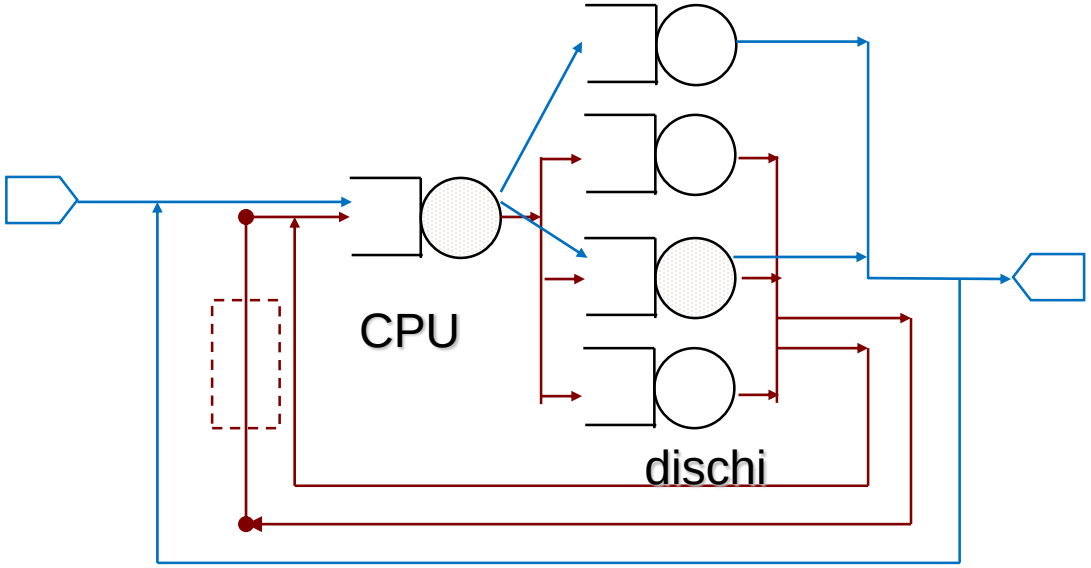
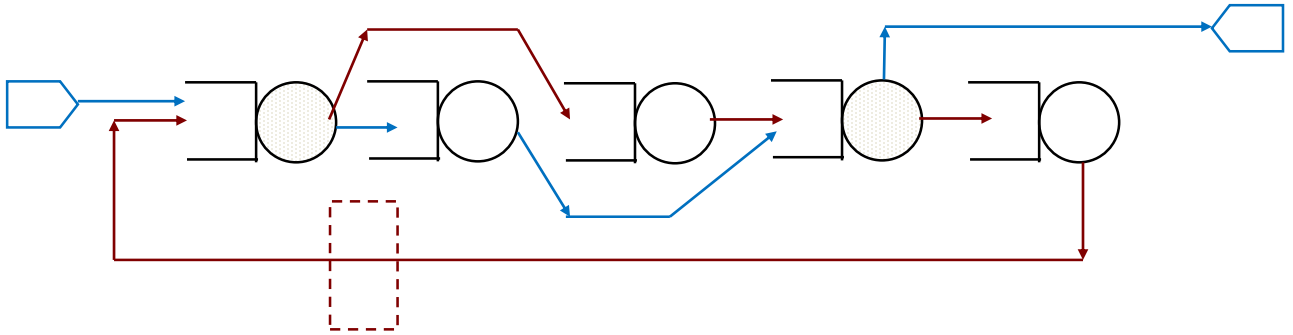
Esercizi su reti di code

Analisi operazionale e soluzioni approssimate

Modello misto



vista
“rete di code”



vista
“operazionale”

Modello misto

carico		CPU	disco1	disco2	disco3	disco4
	s		12	8	7,5	10
A	D	100				
	V		0	25	20	25
B	D	100				
	V		11	0	6	0

- (tempi in msec)
- A: carico batch
- B: carico transazionale

Modello misto

	CPU	disco1	disco2	disco3	disco4	tot
D (A)	0,100	0,000	0,200	0,150	0,250	0,700
D (B)	0,100	0,132	0,000	0,045	0,000	0,277

- Domande di servizio (in secondi)
- $D = s \times V$
- È presente il solo carico A, applicazione di tipo batch, di cui bisogna determinare:
  - l'andamento asintotico del tempo di risposta al crescere della popolazione N di job attivi;
  - le grandezze di prestazione (R, X, U) dei job batch con N = 3 job attivi (metodo esatto MVA).

Modello misto

	CPU	disco2	disco3	disco4	tot	CPU	disco2	disco3	disco4	
N	Ri				R	code				X
1	0,100	0,200	0,150	0,250	0,700	0,143	0,286	0,214	0,357	1,429
2	0,114	0,257	0,182	0,339	0,893	0,256	0,576	0,408	0,760	2,240
3	0,126	0,315	0,211	0,440	1,092					2,747

	CPU	disco2	disco3	disco4
N	Utilizzi			
1	0,143	0,286	0,214	0,357
2	0,224	0,448	0,336	0,560
3	0,275	0,549	0,412	0,687

- Il disco4 è il collo di bottiglia, perciò X(max) = 1/D(disco4) = 4 transazioni /sec.
- L'asintoto del tempo di risposta è quindi la retta: R = N / X(max) = N $\times$ 0.25
- La curva del tempo di risposta in funzione di N è perciò contenuta nella porzione di piano compresa fra la retta R = N $\times$ D(tot) = N $\times$ 0.7, la retta R = D(tot) e quella asintotica.
- Il punto di ascissa N* = D(tot) $\times$ X(max) = 2.8 rappresenta il valore di N oltre il quale si formano necessariamente accodamenti nel sistema.

Modello misto

CPU	disco2	disco3	disco4
Utilizzi			
0,196	0,392	0,294	0,491

Immaginiamo che il carico A sia eseguito da:
20 utenti con Z = 9.1 e R= 1.092

- Prestazioni del carico interattivo (A)
- Il throughput X = 1.962 dalla legge del tempo di risposta $(R+Z) \times X = N$;
- I 20 utenti si ripartiscono fra il sistema, mediamente: $R \times X = 2.14$ e il nodo “terminali” (20 - 2.14 = 17.86);
- utilizzi (dalla formula $U(i) = D(i) \times X$)
- La retta asintotica del tempo di risposta è data da:
- $R = N / (Xmax) – Z = N / 4 – 9.1$
- $N^* = (D + Z) \times X(max) = 39.2$
- Ovviamente X(max) e D non sono cambiati dalla soluzione batch, perciò la retta asintotica è parallela a quella già calcolata ma traslata verso il basso di un segmento pari a Z.

Modello misto

	utilizzo carico B				
X (B)	CPU	disco1	disco2	disco3	disco4
1,000	0,100	0,132	0,000	0,045	0,000
2,000	0,200	0,264	0,000	0,090	0,000
3,000	0,300	0,396	0,000	0,135	0,000
4,000	0,400	0,528	0,000	0,180	0,000
5,000	0,500	0,660	0,000	0,225	0,000
6,000	0,600	0,792	0,000	0,270	0,000
7,000	0,700	0,924	0,000	0,315	0,000
7,576	0,758	1,000	0,000	0,341	0,000

- Prestazioni del carico transazionale B (utilizzo dei componenti in funzione di X(B)
- Il collo di bottiglia per il carico B è il disco1 (che è usato solo da tale classe di carico)

Modello misto

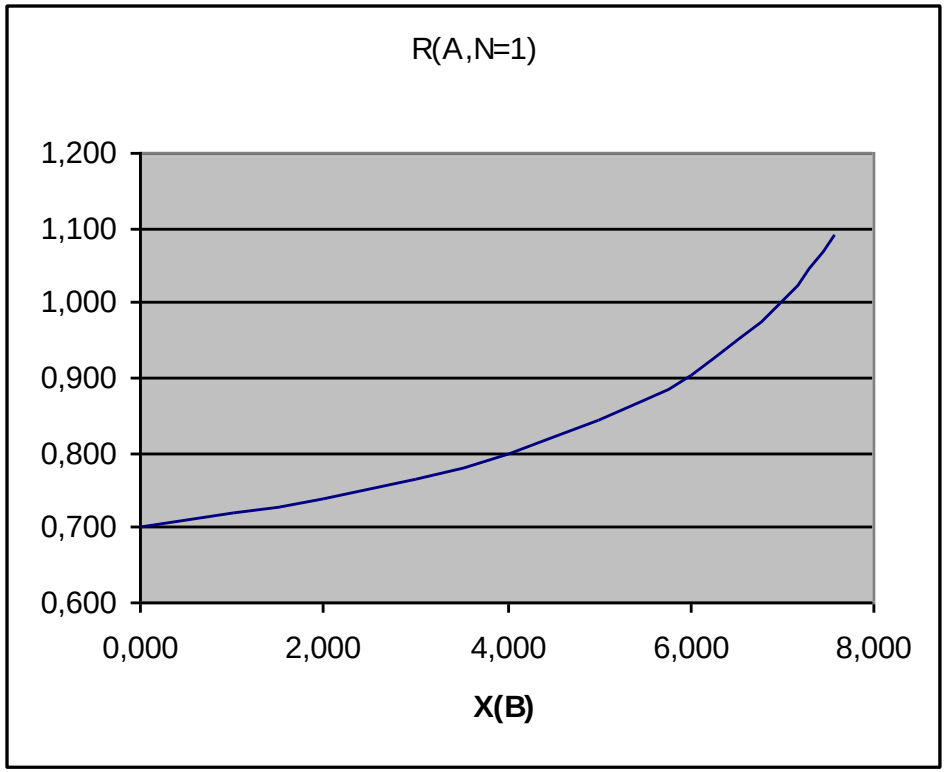
- Prestazioni carico misto
- Al carico batch A (chiuso) precedentemente esaminato, viene aggiunto il carico B di tipo transazionale (aperto).
  In particolare calcolare:
- il tempo di risposta del batch quando N = 1, al variare del carico transazionale, tracciandone anche un grafico indicativo;
- l'andamento asintotico del carico A quando il carico B è prossimo alla saturazione.

Modello misto

	domande D* del carico A					
X (B)	CPU	disco1	disco2	disco3	disco4	tot
1,000	0,111	0,000	0,200	0,157	0,250	0,718
2,000	0,125	0,000	0,200	0,165	0,250	0,740
3,000	0,143	0,000	0,200	0,173	0,250	0,766
4,000	0,167	0,000	0,200	0,183	0,250	0,800
5,000	0,200	0,000	0,200	0,194	0,250	0,844
6,000	0,250	0,000	0,200	0,205	0,250	0,905
7,000	0,333	0,000	0,200	0,219	0,250	1,002
7,576	0,413	0,000	0,200	0,228	0,250	1,090

- Il comportamento del carico A in presenza di B si ottiene ricordando che esso, dal punto di vista delle prestazioni (tempi di risposta e throughput) si comporta come se le sue domande di servizio fossero D(i)* = D(i) / (1- U(i,B)), dove D(i) è la domanda originaria di A e U(i;B) l'utilizzo del nodo (i) da parte di B; la tabella riporta le domande D* e quella totale che è pure il tempo di risposta di A in funzione di X(B) se N = 1

Modello misto

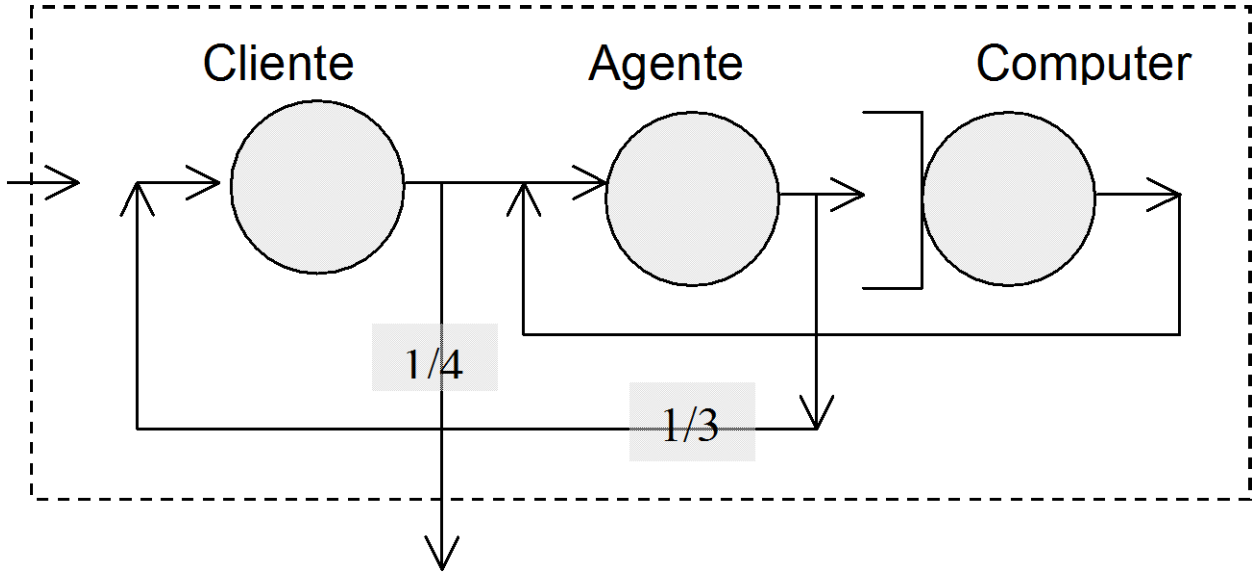


- Tempo di risposta del carico A (con N = 1) in presenza di B (variabile)

Modello misto

- L’andamento asintotico di A, in presenza di B dipende dalla capacità rimanente dei componenti. Quando X(B) = max = 7.576, il collo di bottiglia risulta essere, per la classe A, la CPU. Il metodo più immediato per calcolare il massimo valore X(A), dato B, è quello di calcolare l’inverso della massima D(i)*.
- Concludendo allora per X(B) = 7.576, il massimo X(A) vale: 1/ 0.413 = 2.42 e la corrispondente retta asintotica del carico batch A sarà: R = N / 2.42.
- Il grafico della pagina precedente mostra in funzione di X(B) il tempo medio di risposta R(A, N=1).
- In generale se U(i,O) è l’utilizzo del nodo (i) da parte delle classi aperte, a quelle chiuse resta una capacità 1-U(i,O), detta D(i,C) la domanda di una classe chiusa al nodo (i), (1-U(i,O))/ D(i,C) è il suo massimo throughput attraverso (i), allora il massimo valore X della classe C sarà determinato dal nodo a cui corrisponde il massimo D(i,C)/(1-U(i,O)).

Modello «call center»



L'equilibrio globale richiede che:
il flusso in entrata nel sistema sia uguale a quello in uscita
L'equilibrio locale richiede che:
il flusso in entrata in un sottosistema sia uguale a quello in uscita

Modello «call center»

- Fra le linee telefoniche e gli agenti vi è un ACD (Automatic Call Distribution) switch, il cui ruolo è di distribuire le chiamate fra gli agenti competenti in modo uniforme.
- L'agente si considera occupato (= utilizzato) durante la durata dell'intera chiamata, cioè da quando viene stabilito il collegamento a quando esso si chiude.
- Ovviamente, un agente serve una chiamata alla volta mentre il computer (mono-CPU) è condiviso fra tutti gli agenti.
- Della transazione, che rappresenta il flusso generato dalla chiamata, conosciamo i tempi di servizio medi per ogni visita, e cioè:
  - S(Cliente) = 9 sec.; S(Agente) = 7 sec.; S(Computer) = 0.01 sec.
  - e le probabilità di "instradamento" che sono indicate in figura.

Modello «call center»

- Calcolo del numero di visite ai nodi del sistema e delle domande di servizio

- detti X_0, X_1, X_2, X_3 i flussi che, rispettivamente, attraversano il sistema, il Client, l'Agente, il Computer; se si tiene conto delle probabilità di instradamento, l'equilibrio operazionale si traduce nel seguente sistema lineare (o in altro equivalente):

- $¼\ X_1 = X_0$
- $¾\ X_1 = 1/3\ X_2$
- $2/3\ X_2 = X_3$
- che risolto fornisce:
- $X_1 = 4\ X_0$; $X_2 = 9\ X_0$; $X_3 = 6\ X_0$
- Perciò i coefficienti sono le visite $V_i = X_i / X_0$ e le domande sono i prodotti $D_i = V_i \times S_i$

Modello «call center»

- Allora:

	Cliente	Agente	Computer	Totale
V	4	9	6	
D	36	63	0.06	99.06

- dove la domanda totale (99.06 secondi) è il tempo di risposta medio della transazione da parte del Call Center in assenza di contesa sul Computer, nel caso generale il tempo medio di risposta vale 99 sec. + tempo di residenza nel computer (che non conosciamo, ma sappiamo $\geq$ 0.06 sec.).

Modello «call center»

- il massimo tasso di chiamate nell'unità di tempo, che il *Call Center* è in grado di servire con il computer attuale è determinato dalla saturazione del collo di bottiglia (computer) se si ipotizza che il numero N di agenti possa crescere indefinitamente:
- $X_{max} = 1/D_3 = 1/0.06 = 16.667$ trans/sec.

Modello «call center»

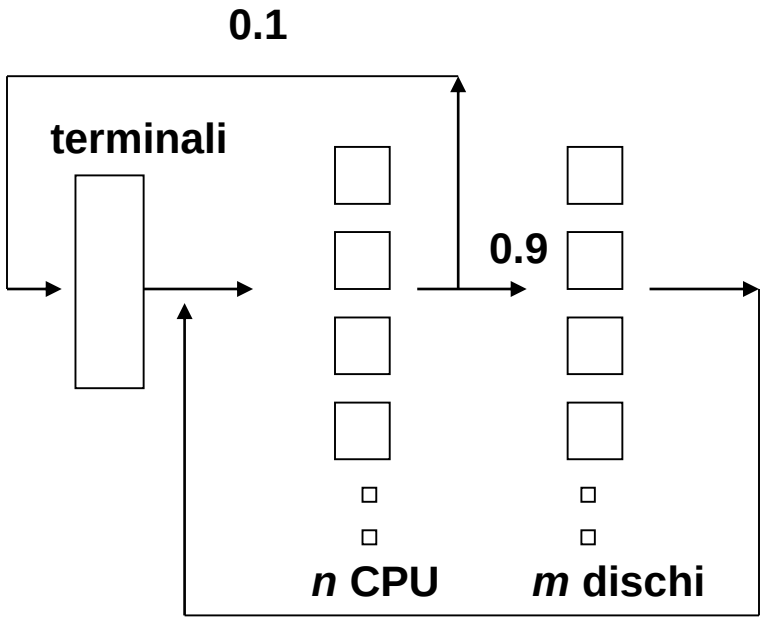
- Il tempo per servire una chiamata Ct (tempo di ciclo) è la somma, come si è già detto, del tempo di residenza nel Cliente + Agente (Z = 99) e del tempo di residenza *r* nel computer, perciò il valore asintotico di *r* vale:
- $r = N / X_{max} - Z = N \times 0.06 - 99$

- In modo ancora più diretto, dato che il massimo tasso di elaborazione di chiamate vale X_{max}, la legge di Little dice:
- $N = Ct \times X_{max}$
- Il valore N* si ottiene ponendo r = 0.06 oppure Ct = 99.06, nelle formule riportate (N* = 1651).

Modello «call center»

- Nel caso in cui il tempo medio (asintotico) per servire una chiamata sia di 120 secondi, si vuole sapere il corrispondente numero N di Agenti in servizio:
- se Ct = 120, dalla formula asintotica il corrispondente valore del numero di agenti è N = 120 / 0.06 = 2000. Il valore di r è, ovviamente, r = 21.
- perché la produttività del centro sia la stessa ma con gli agenti occupati al 80%, bisogna che siano:
- N’ = N / 0.8 = 2500

Sistema con numero di nodi variabile



- Un sistema interattivo consiste di **N = 500** utenti, n CPU e m dischi. I “thinktime” degli utenti sono **Z = 10** secondi.
- Al termine del tempo Z una transazione viene inviata a una CPU scelta a caso con uguale probabilità (**1/n**)
- Al completamento della visita alla CPU ritorna all’utente con probabilità **0.1** oppure va a un disco, scelto anch’esso a caso con uguale probabilità (**1/m**).

- Al termine della visita al disco la transazione va ad una CPU scelta sempre a caso.
- Il tempo di servizio per visita alla CPU vale SCPU = **0.01** secondi e quello al disco Sdisco = **0.02** secondi.

Sistema con numero di nodi variabile

1. È possibile raggiungere un throughput di sistema di almeno 50 transazioni al secondo?
2. Se togliamo il vincolo $N = 500$, cioè N può variare in modo arbitrario, quale è il modo migliore di configurare il sistema (cioè quante CPU e quanti dischi occorrono perché il massimo throughput del sistema Xmax sia di almeno 40 transazioni al secondo?

Sistema con numero di nodi variabile

- Domanda 1

  Conoscendo le probabilità con cui vengono scelti i rami della rete si calcola agevolmente:

  - $V_{CPU} = 10$ e $V_{dischi} = 9$, visite rispettivamente al cluster di CPU e a quello dei dischi.
  - Perciò $D = D_{CPU} + D_{dischi} = 10 \times 0.01 + 9 \times 0.02 = 0.1 + 0.18 = 0.28$.
  - Le domande alla singola CPU e al singolo disco sono allora rispettivamente:
    0.1 / n e 0.18 / m
  - Supponendo dapprima che il collo di bottiglia siano le CPU (*m* sufficientemente grande), allora:
  - Db = 0.1/*n* o anche 1/Db = 10 *n* N* = 10.28 / (0.1/*n*) = 102.8 *n*.
    Se $n \leq 4$, N* < 500 e Xmax ≤ 10 *n* < 50

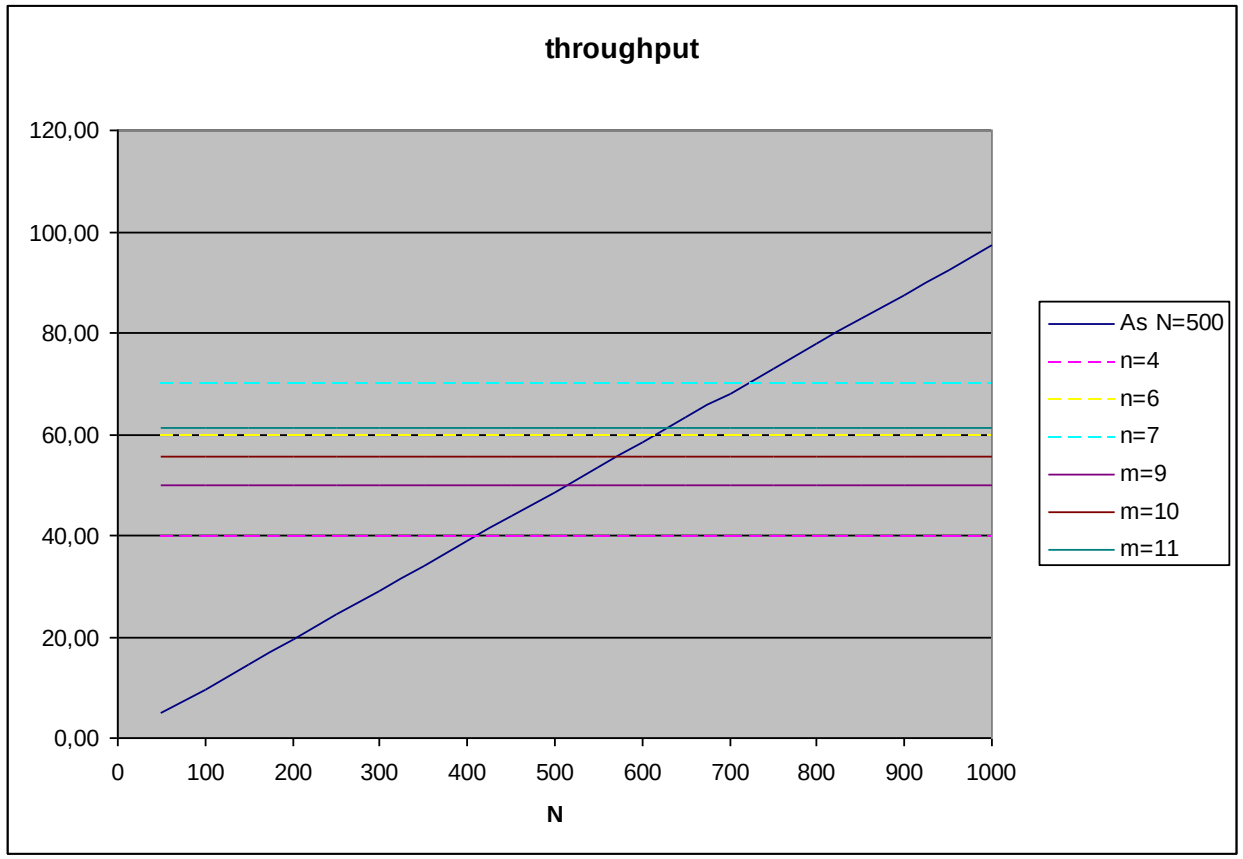
Sistema con numero di nodi variabile

- se $n > 4$, N* > 500 e il throughput è limitato superiormente da N / (D + Z) = 500 / 10.28 = 48.64 che ancora è minore di 50.
- In modo analogo se il collo di bottiglia sono i dischi (m sufficientemente grande), allora:
- Db = 0.18/m o anche 1/Db = 5.56 m N* = 10.28 / (0.18/m) = 57.11m. Se $m < 9$, N* < 500 e 18 m < 50,
- se $m \geq 9$, N* > 500 e il throughput è limitato superiormente da N / (D + Z) = 500 / 10.28 = 48.64 che ancora è minore di 50. (**la risposta alla domanda 1 è negativa**).

Sistema con numero di nodi variabile

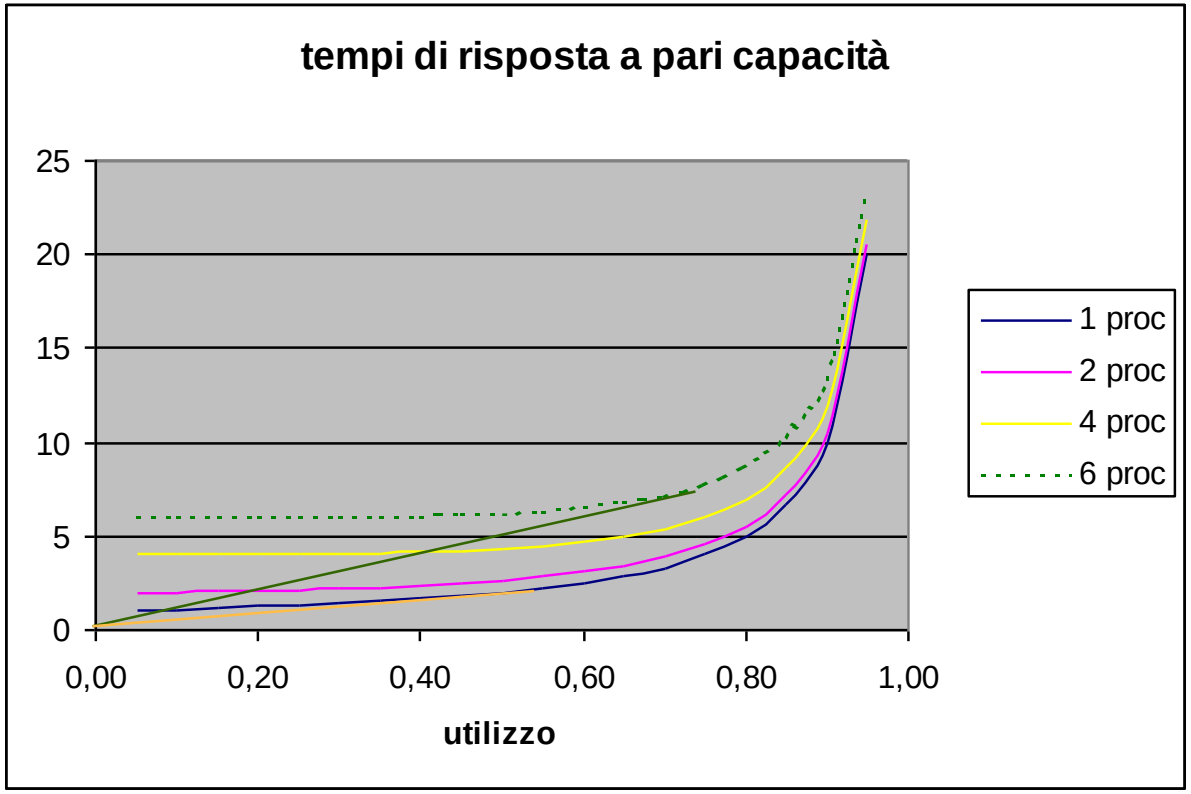
- Domanda 2
  - Per raggiungere un throughput massimo di almeno 40 transazioni al secondo Db $\leq$ 1/40. Allora se
  - Perciò se il collo di bottiglia è la CPU
    Db = 1/10n $\leq$ 1/40 perciò $\boldsymbol{n \geq 4}$.
    se il collo di bottiglia è il disco
    Db = 1/5.56m $\leq$ 1/40 perciò $\boldsymbol{m \geq 8}$.

Sistema con numero di nodi variabile



Riassunto grafico

Scalabilità



i tempi di risposta a parità di carico sono *maggiori* al crescere del numero di processori ma le curve sono *"più regolari"*

all'aumentare del carico la distanza fra le curve diminuisce: cioè per elevati utilizzi la differenza fra soluzioni con un diverso numero di processori tende a ridursi

la capacità totale della famiglia di curve è la stessa (es: 6 processori hanno ciascuno 1/6 della potenza del processore singolo). I punti con la stessa ascissa sono relativi perciò allo stesso carico globale

Scalabilità

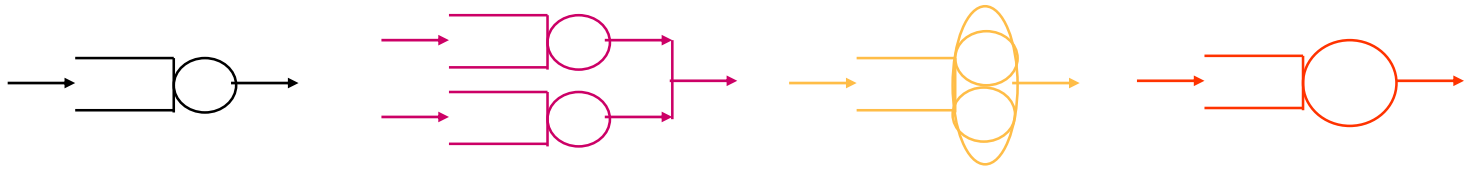
- (trascuriamo l'effetto overhead dovuto al multiprocessing)
- la scalabilità presenta due aspetti principali:
  - distribuzione delle capacità fra diversi processori
  - incremento del carico e conseguente necessità di aumentare la capacità elaborativa (il carico e la capacità crescono di n volte)

si tratta di un caso che si può ricondurre al precedente *solo con drastiche semplificazioni valide nella realtà in modo approssimato (se non esistono altri colli di bottiglia fisici o logici / applicativi)*

- 0) n processori identici indipendenti ciascuno dei quali elabora 1/n del carico totale;
- 1) un solo processore di potenza $\times$ n
- 2) con un n-processore (in tutti i casi l'utilizzo del singolo processore resta inalterato)
- r_0, r_1, r_n sono rispettivamente i tempi medi di risposta nei tre casi considerati (caso iniziale, soluzione a 1-proc e a n-proc)

$$r_1 = r_0 / n \quad ; \quad r_0 / n < r_n < r_0 \quad ; \quad r_n \to r_0 / n \quad (\text{per } u \to 1)$$

Scalabilità



M/M/1
b

M/M/1
1

M/M/2
2

M/M/1
3

- vogliamo confrontare i tempi di risposta dei sistemi 1, 2, 3 con quello del sistema b (base) ***a parità di traffico X***
- i sistemi b, 1, 2 hanno tempo di servizio s, il sistema 3 invece s/2
- sia u l'utilizzo del sistema b u = Xs

$$r(b) = \frac{s}{1 - sX} = \frac{s}{1 - u}$$

$$r(1) = \frac{s}{1 - \frac{1}{2}sX} = \frac{s}{1 - \frac{u}{2}}$$

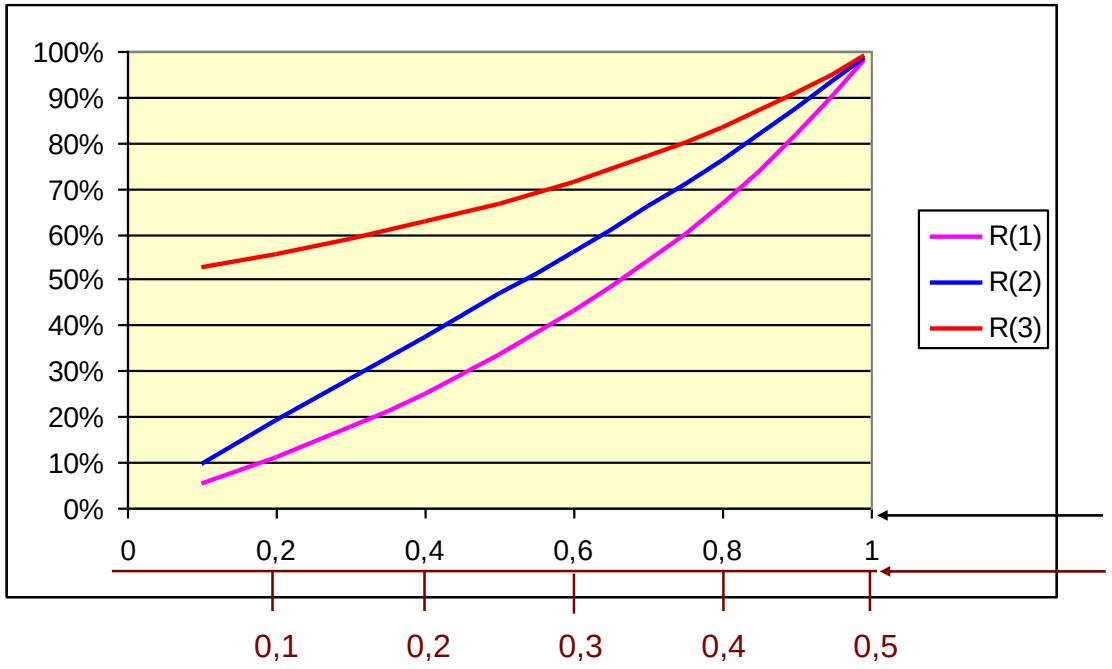
$$r(2) = \frac{s}{1 - \left(\frac{sX}{2}\right)^2} = \frac{s}{1 - \left(\frac{u}{2}\right)^2}$$

$$r(3) = \frac{\frac{1}{2}s}{1 - \frac{1}{2}sX} = \frac{s}{2 - u}$$

$r(b) \geq r(1) \geq r(2) \geq r(3)$

Scalabilità

- per confrontare i tre sistemi 1, 2, 3 calcoliamo la riduzione del tempo di risposta relativamente al sistema base:
- R(x) = (r(b)-r(x))/r(b)



$R(3) \geq R(2) \geq R(1)$

$$R(1) = \frac{u}{2-u}$$

$$R(2) = \frac{u(4-u)}{(2-u)(2+u)}$$

$$R(3) = \frac{1}{2-u}$$

Serventi occupati

- la scelta di un modello per rappresentare un fenomeno reale comporta sempre semplificazioni di cui bisogna valutare la portata - un caso reale:
  - la misura di un multiprocessore (a 6 vie) in un periodo di carico stazionario fornisce i seguenti dati:

coda	0	1	2	3	4	5	6	5+	6+
freq.%	0,0%	0,7%	3,8%	7,5%	9,8%	11,4%	10,9%	66,8%	55,9%

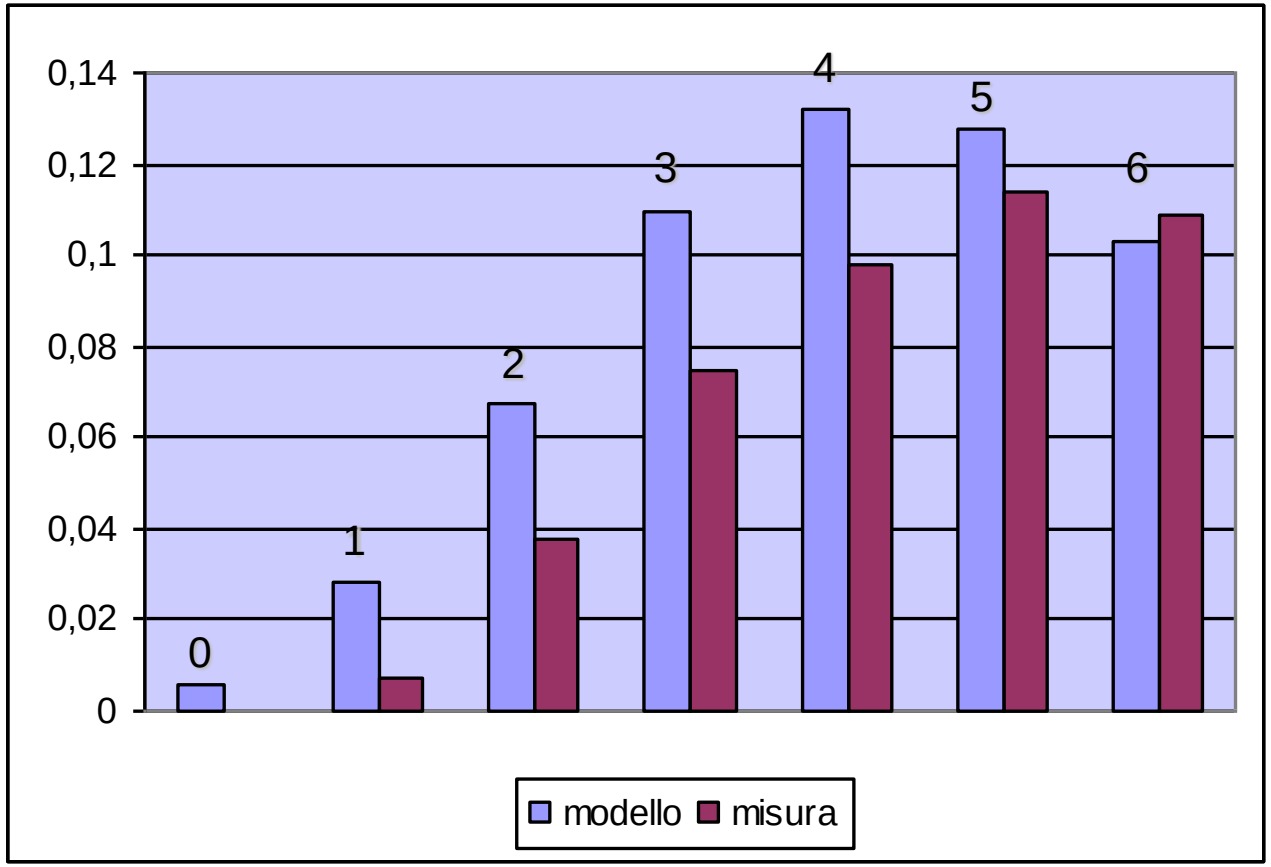
- la seconda riga riporta le frequenze (ottenute per campionamento) in cui si sono trovati nel sistema rispettivamente 0, 1, 2 ...., 6 utenti *"ready"* cioè in servizio o in attesa di servizio;
- la misura dà un'indicazione della *opportunità* di aumentare il numero di processori
  - nel nostro caso (100 - 0.0 - 0.7 -...-10.9) = 55.9
  - cioè: nel 56% del tempo avremmo potuto sfruttare ulteriori processori

Serventi occupati

- sappiamo anche che i processori sono mediamente utilizzati al 80.58% (dato di misura)
- interpretiamo l'utilizzo medio (80.58%) come la probabilità di trovare un server (scelto a caso) occupato:
  - se i server fossero indipendenti (6 processori con la loro coda)
  - $U^6 = 0.8058^6 =$ 27.38% = probabilità che siano tutti occupati
- considerando il multiprocessor come una coda M/M/6
  - C(c,u) = C(6,4.83) = 52.96%
- il complemento della somma dei primi 6 valori (5+ della tabella)
  - (100 - 0.0 -...-11.4)= 66.8% è la stima della probabilità che tutti i 6 processori siano busy

- i valori da confrontare (probabilità che tutti i 6 processori siano occupati) sono perciò:
  - 27.4 (indipendenza); 53.0 (M/M/6); 66.8 (misura)

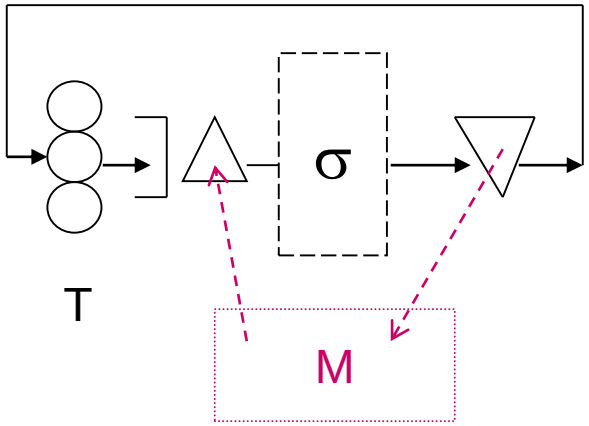
Probabilità che n serventi «busy»



P(n>6)
modello: 0.43
misura: 0.56

Scomposizione - aggregazione

- le reti che ammettono una soluzione in *forma di prodotto* sono separabili in modo esatto, in altri casi (quasi separabili) la soluzione che si ottiene è solo approssimata
- un esempio è lo studio (approssimato) dell'effetto del blocco del massimo numero di utenti attivi contemporaneamente in memoria



M contiene un numero fisso di «gettoni»

Scomposizione - aggregazione

- il metodo di soluzione gerarchico può essere usato per risolvere in modo analitico mediante due o più passi modelli che presentano risorse passive, quali un fissato livello massimo di multiprogrammazione e che, per questa ragione, non sono risolubili in modo esatto. Si opera nel seguente modo:
- 1. si risolve più volte, per diversi livelli di multiprogrammazione (da 1 al massimo previsto) il *modello interno cortocircuitato* cioè senza terminali, nel senso che la transazione, finita la sua elaborazione, ritorna in input al sistema;
  - il modello interno fornisce, in funzione della multiprogrammazione, la produttività (tasso di servizio).
- 2. Infine viene risolto il *modello esterno* che comprende i terminali e una *coda equivalente* rappresentante la parte interna caratterizzata da una "*potenza" di elaborazione* variabile in funzione della lunghezza della coda, che resta costante per valori della lunghezza superiori o uguali al massimo livello permesso di multiprogrammazione
  - $X_{eq}(n) = 1/D_{eq}(n)$

Scomposizione - aggregazione

misure

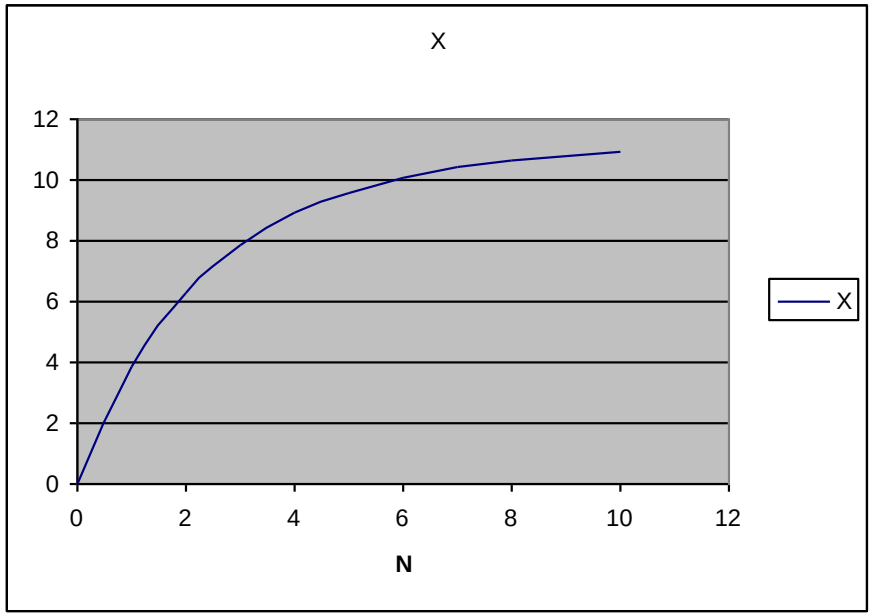
	X	s	U	Z
terminali	3,492778			19,81288
CPU			0,188556	
disco1	14,869	0,004		
disco2	10,441	0,004		
disco3	17,046	0,004		
disco4	6,979	0,029		
disco5	5,058	0,007		
disco6	11,599	0,027		

	domande
terminali	19,8128772
CPU	0,05398441
disco1	0,01702828
disco2	0,01195725
disco3	0,01952143
disco4	0,05794557
disco5	0,01013692
disco6	0,08966302

parametri modello interno σ

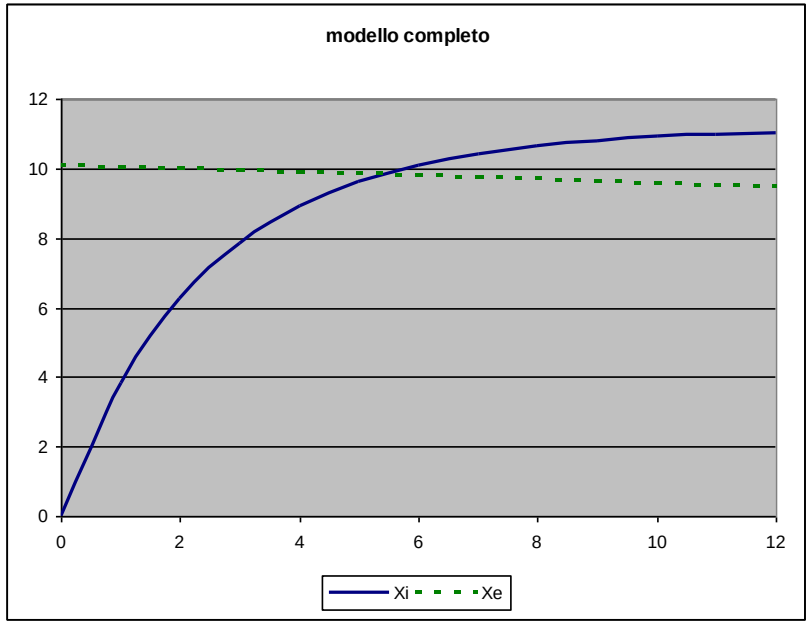
Scomposizione - aggregazione

N	R	X
1	0,260237	3,842653
2	0,318754	6,274436
3	0,382014	7,853108
4	0,449459	8,899592
5	0,520564	9,604974
6	0,594862	10,08638
7	0,671937	10,41765
8	0,751411	10,64664
9	0,832934	10,80518
10	0,916183	10,91485



$X_{\sigma}(N)$ throughput di (σ) in funzione della lunghezza N della coda
(parametro del modello esterno)

Scomposizione - aggregazione



utenti in σ	tasso verso i terminali	tasso dai terminali	tempo di risposta
0	0,00	10,09	
1	3,84	10,04	0,26
2	6,27	9,99	0,32
3	7,85	9,94	0,38
4	8,90	9,89	0,45
5	9,60	9,84	0,52
6	10,09	9,79	0,59
7	10,42	9,74	0,67
8	10,65	9,69	0,75
9	10,81	9,64	0,83
10	10,91	9,59	0,92
11	10,99	9,54	1,00
12	11,04	9,49	1,09

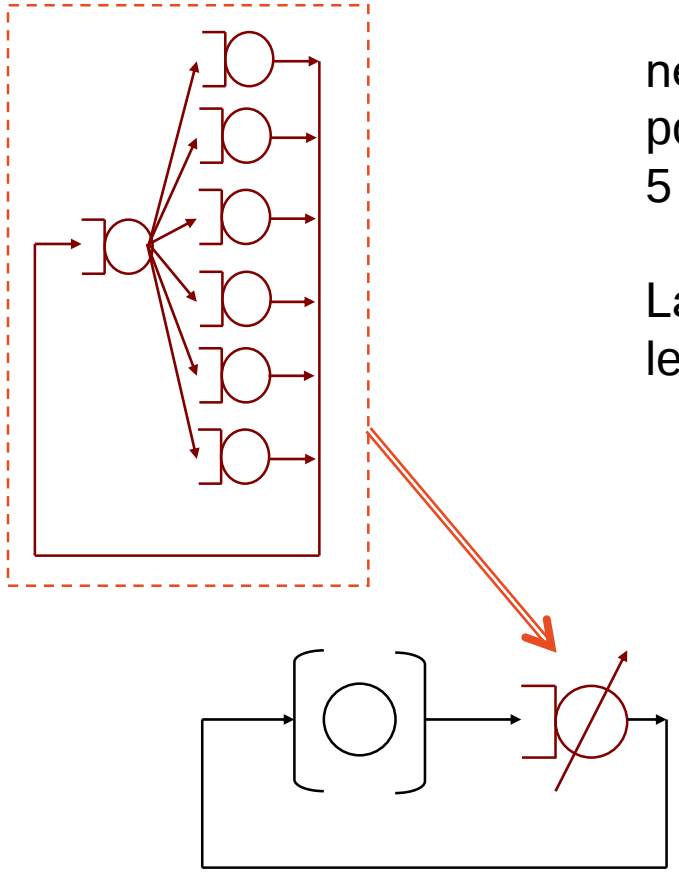
La soluzione grafica è relativa a N=5.44,
La soluzione completa tiene conto di tutti
i valori N con la loro probabilità di verificarsi

- può essere trovata dall'intersezione - nel piano (X, N_σ) della soluzione del modello σ con la retta $X = (N_{tot} - N_\sigma) / Z$ che rappresenta il tasso di arrivo delle transazioni dai terminali
- la soluzione grafica per $N_{tot} = 200$ vale: $N_\sigma = 5.44$; $X = 9.82$; $R = 0.58$ - quella esatta (modello completo MVA) è invece: $X = 9.66$; $R = 0.89$

Come studiare le contese di software

- un modo per studiare la contesa a livello software è quello di imporre un massimo livello di parallelismo per un certo insieme di componenti hardware
- esempio:
  - se il modello interno delle pagine precedenti può contenere al più 5 processi, questo effetto viene studiato risolvendo un modello esterno che fornisce un throughput pari a quello calcolato per $0<N\leq5$ e pari a quello ottenuto per N=5 per $N>5$
  - in altri termini la domanda, in funzione di N è calcolata da:
  - $D(N) = 1/X(N)$

Come studiare le contese di software



nel sistema “centrale”
possono entrare al massimo
5 utenti contemporaneamente

La coda “load dependent” ha
le seguenti domande di servizio:

N_σ	$1/D(N_\sigma)$
1	3.842
2	6.274
3	7.853
4	8.899
5+	9.605

Soluzione (da JMT)

N	X	R
100	4.952	0.383
150	7.354	0.585
200	9.293	1.709

Come studiare le contese di software

Ni	N	X	R
1	77	3,8427	0,2602
2	126	6,2744	0,3188
3	159	7,8531	0,3820
4	180	8,8996	0,4495
5	195	9,6050	0,5206
6	196	9,6050	0,6247
7	197	9,6050	0,7288
8	198	9,6050	0,8329
9	199	9,6050	0,9370
10	200	9,6050	1,0411
11	201	9,6050	1,1452
12	202	9,6050	1,2494
Z =	19,8129		

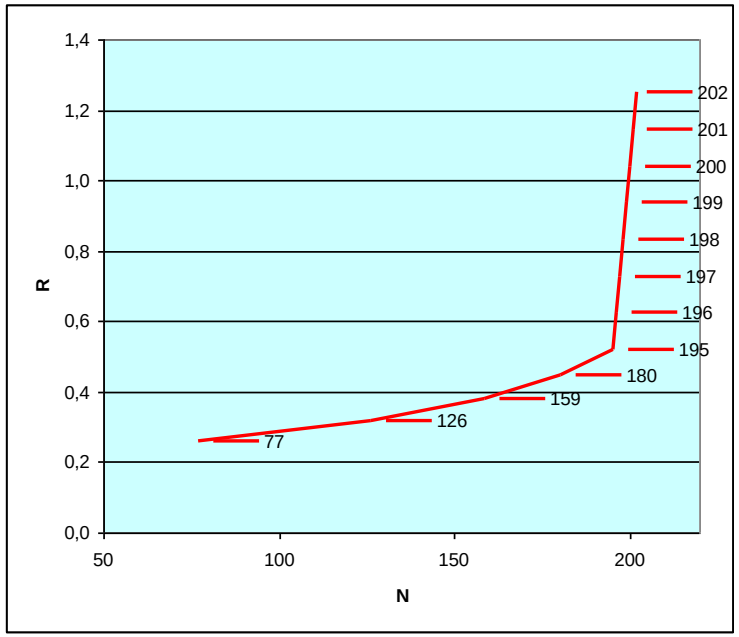
il sistema “centrale interno” può contenere al massimo 5 utenti

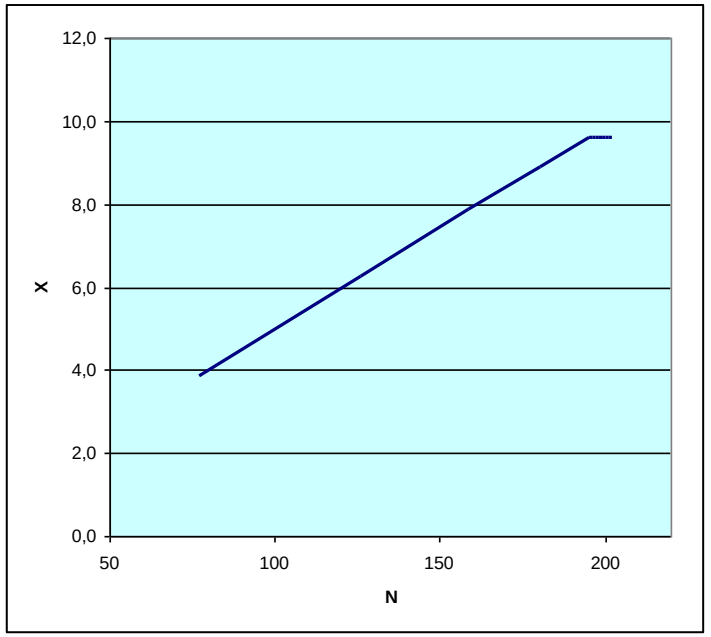
la colonna “X” riporta i valori ottenuti dalla soluzione del modello interno in funzione di Ni
(a partire da Ni = 5, X è costante)

R è dato da $R=Ni/X$ (Legge di Little)

N è ottenuto da $N = Ni + X \times Z$
sostituendo R in $N = (R + Z) \times X$

Come studiare le contese di software





Le differenze fra la soluzione JMT e quella approssimata sono dovute al fatto che quest'ultima suppone una situazione stazionaria "fissa" mentre il modello più corretto tiene conto delle fluttuazioni attorno allo stato medio (cioè se gli utenti sono N = 200, Ni = 10 solo in media)

Soluzione approssimata di una rete chiusa

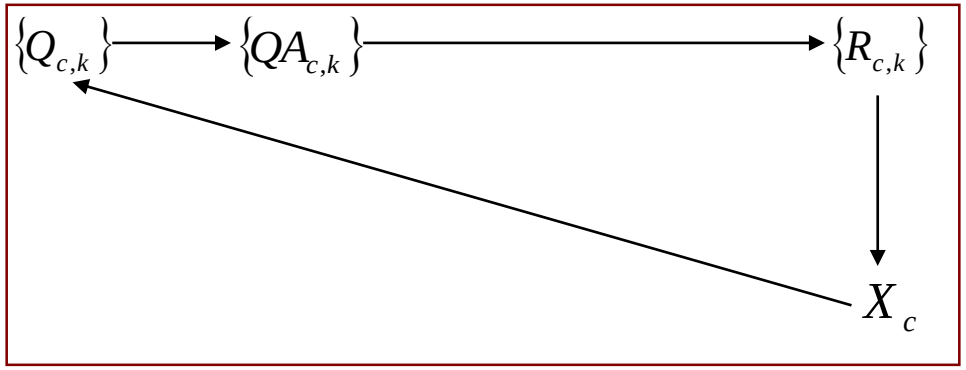
$$QA_k(N) = Q_k(N-1) \cong \frac{N-1}{N} Q_k(N)$$

$$Q_{c,k}(\vec{N}) = X_c(\vec{N}) \cdot R_{c,k}(\vec{N})$$

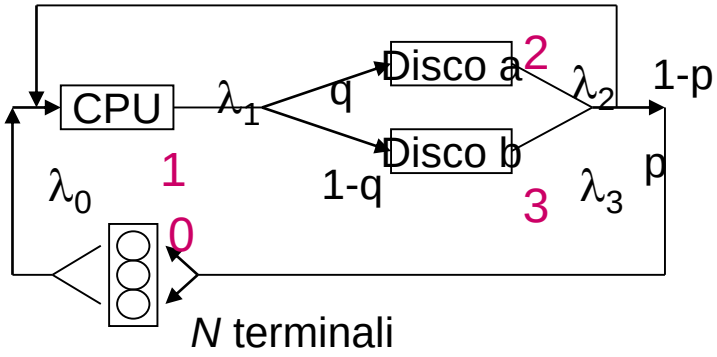
$$Q_{,k}(\vec{N}) = \sum_c Q_{c,k}(\vec{N})$$

Questa approssimazione, valida per N “grande” permette di evitare di risolvere il modello per tutti i valori di N fino a quello finale richiesto

secondo la seguente procedura



Soluzione approssimata di una rete chiusa



nodo	0	1	2	3
serv	20	0,008	0,01	0,01
domanda	**20**	**0,2**	**0,15**	**0,1**
visite	**1**	**25**	**15**	**10**

Soluzione esatta con N = 100

nodo	0	1	2	3	0	1	2	3				
N	lunghezza coda				tempi Residenza				CT	X	U(CPU)	Resp
100	0	6,608186	2,025055	0,818554	20	1,459595	0,447288	0,1808	22,08768	4,52741	0,905482	2,087682
		0,977995	0,733496	0,488998	20							
1		1,892426	1,244769	0,71348	20	0,393643	0,258924	0,148411	20,80098	4,807466	0,961493	0,800978
2		2,726258	1,58845	0,809455	20	0,5747	0,334848	0,170634	21,08018	4,743792	0,948758	1,080183
3		3,472287	1,811171	0,845478	20	0,739799	0,385885	0,180136	21,30582	4,693553	0,938711	1,30582
4		4,129854	1,949537	0,85482	20	0,887513	0,418959	0,183702	21,49017	4,653289	0,930658	1,490174
5		4,702515	2,030817	0,853103	20	1,017711	0,439506	0,184627	21,64184	4,620678	0,924136	1,641845
6		5,196358	2,074579	0,847412	20	1,131098	0,451576	0,184457	21,76713	4,594083	0,918817	1,767132
7		5,618798	2,094455	0,840817	20	1,228879	0,458075	0,183894	21,87085	4,572297	0,914459	1,870848
8		5,977751	2,099699	0,834552	20	1,312522	0,461027	0,183241	21,95679	4,5544	0,91088	1,956789
9		6,281066	2,096445	0,829038	20	1,383595	0,461805	0,182621	22,02802	4,539673	0,907935	2,028021
10		6,536189	2,088654	0,824351	20	1,443651	0,461322	0,182075	22,08705	4,52754	0,905508	2,087048

iterazioni

valori di partenza: es.: 0.978 = 0.2 × 100 /(20+0.2+0.15+0.1)

Memoria virtuale

- Le risorse che gli utenti consumano non dipendono solo dal carico di lavoro *produttivo* che questi svolgono, nel qual caso gli utilizzi dei componenti sarebbero legati linearmente al throughput. Esistono infatti *risorse passive*, come ad esempio la memoria, la cui occupazione è legata alla presenza degli utenti e non solo alla attività che questi svolgono.

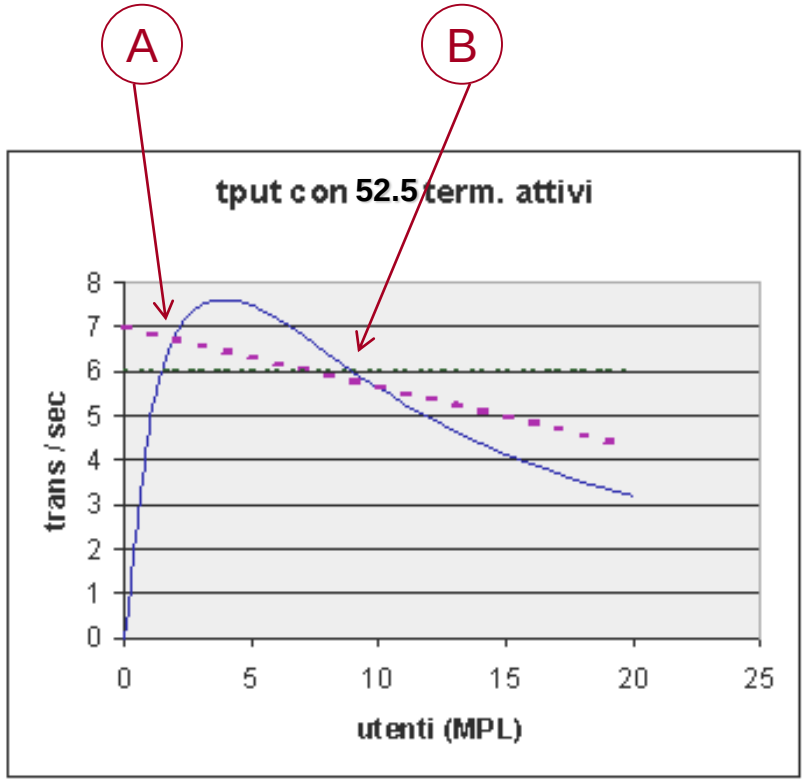
- Se il numero di utenti presenti cresce, ciascuno di essi avrà a disposizione un minor numero di pagine di memoria centrale. Ne segue che il fenomeno indotto della paginazione (cioè delle operazioni di I/O su memoria ausiliaria che verranno compiute per completare la transazione) è *non omogeneo* ma varia in proporzione al numero di utenti presenti.

- Introducendo un tempo di paginazione per transazione proporzionale al quadrato del numero degli utenti presenti si ottiene la soluzione riportata nel grafico. *La curva X = X(n) non presenta più un asintoto orizzontale ma ha un massimo per un certo valore di n. Per valori di n maggiori la produttività tende a diminuire. Se viene fissato il valore massimo di n, per n > nmax il valore X resta costante.*

- (I parametri del modello a cui si riferiscono le prossime pagine sono diversi da quelli usati in precedenza. Sono stati ottenuti da misure reali, ma non vengono riportati in dettaglio in quanto si vuole mostrare il fenomeno soprattutto dal punto di vista qualitativo).

Memoria virtuale

- la presenza di memoria virtuale produce *paginazione* che è un tipico fenomeno non omogeneo in quanto dipende dallo stato del sistema
- il page_life_time (intervallo di CPU fra due operazioni di page-in successive) ha il seguente andamento: $cost \cdot (WS)^k$
  - dove WS: working set
  - $(1.5 < k < 3)$
- al crescere di N $\Rightarrow$ WS diminuisce e il numero di page-in aumenta perciò la produttività raggiunto un massimo comincia a decrescere
- il valore massimo MPL viene controllato dinamicamente



esempio di soluzione di un modello interattivo (simulazione)

Memoria virtuale

- usando il metodo descritto in precedenza scopriamo che la retta mandata dal punto (N,0) nel piano (n,X), interseca generalmente la curva rappresentativa in tre punti:
- A: nel tratto di curva ascendente,
- B: nel tratto di curva discendente,
- C: nel tratto rettilineo.
- Il punto A è un punto stabile, nel suo intorno infatti se n crescesse (diminuisse) la produttività X crescerebbe (diminuirebbe) riducendo (facendo crescere) il corrispondente numero di transazioni attive, perciò il valore n tenderebbe a tornare nella posizione di partenza n(A).
- Il punto B è invece punto di instabilità perché, comportandosi il sistema nel suo intorno in modo opposto che in A, ogni fluttuazione tenderebbe ad essere amplificata.
- Il punto C è invece ancora stabile. Naturalmente in funzione del valore N si danno casi in cui esiste solo l'intersezione stabile A o C.
- La distribuzione della lunghezza delle code, ottenuta risolvendo il modello con una simulazione, mostra dei massimi in corrispondenza delle intersezioni della retta con la curva rappresentativa.

Memoria virtuale

- per MPL = 6 si ha un unico punto (A) di stabilità del sistema: le frequenze hanno un massimo per n = A
- per MPL = 9 il sistema diviene instabile e si formano code più lunghe con prestazioni inferiori a quelle del caso precedente: le frequenze hanno due massimi per n = A ed n = B

QL	MPL = 6	MPL = 9
0	0.06491	0.03130
1	0.09123	0.04371
2	0.09184	0.04468
3	0.08477	0.03950
4	0.07518	0.03554
5	0.06977	0.03078
6	0.06364	0.02856
7	0.05956	0.02670
8	0.05446	0.02647
9	0.04824	0.02880
10	0.04270	0.03143
11	0.03956	0.03440
12	0.03434	0.03499
13	0.02987	0.03504
14	0.02656	0.03636
15	0.02276	0.03745
16	0.01942	0.03775
17	0.01633	0.03651
18	0.01332	0.03748
19	0.01117	0.03514

(da un modello risolto con la simulazione)