

NOTE PMCSN

$$E[T_a] = \frac{\lambda}{\frac{1}{2} E[S^2]} \quad \text{GENERALI}$$

SOLO per M/M/1 ho $E[T_a] = \frac{\rho E[S]}{1-\rho}$

al numeratore ho sempre $E[S_{\text{REM}}]$

Per avere $E[T_s]$ sommo $E[T_a] + E[S]$,

nel caso esponenziale $E[T_s] = \frac{1}{\mu - \lambda}$, questo

lo si trova spesso nei problemi

in cui viene chiesto $E[T_s]$.

Per Little $E[N_q] = \lambda \cdot E[T_a]$, applicabile

in coda, nel centro o nel servente.

$$(E[N_s] = \lambda E[T_s]) \quad (\rho = \lambda E[S])$$

CASO PARTICOLARE e' L'HYPREEXP:

$$\frac{\rho E[S]}{2(1-\rho)} \left(1 + g(p) \right) \quad \text{con } g(p) = \frac{1}{2\rho(1-\rho)} - 1$$

Attenzione all' Hyperexponential, dove ho un SERVENTE che lavora a 2μ con probabilità "p" e $(1-p)\mu$ con probabilità " $1-p$ ".

Concepto espresso chiesto è lo **SLOWDOWN**, generalmente espresso come $E[Sd(x)] = \frac{E[T_q]}{x} + 1$, se ho ≥ 2 code devo scegliere la coda in cui "x" code! Spesso viene confrontato con il Processor Sharing, che è $E[AU] \approx$

$$E[Sd(x)] = \frac{1}{1-p} \quad \forall x$$

Per il PS si ha $E[T_s] = \frac{E[S]}{1-p}$

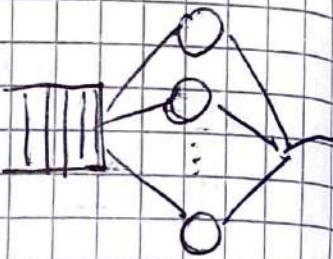
Nei casi M/M/1 (1 servente) le formule usate provengono dalla KP.

Nei casi M/M/m (m serventi) le formule usate provengono dalla ERLANG-C

ERLANG C

Postazioni al Multiserv con CODA M/M/m

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



usata in $P_{02} = \frac{(mp)^m}{m!(1-p)} P(0) = \frac{\left(\frac{\lambda}{m}\right)^m}{m!(1-p)} P(0)$

Inoltre $E[S_i] = \frac{1}{\mu}$ del server "i"

$$E[S] = \frac{E[S_i]}{m}; p_i = \left(\frac{\lambda}{m}\right) = \frac{\lambda}{m\mu} e$$

$p = \frac{\lambda}{(m\mu)}$ sono uguali anche se ci danno due "visioni diverse"

$$E[T_a] = \frac{P_a \cdot E[S]}{1 - P}$$

si libera un server qualsiasi, al job non interessa entrare in un server specifico, aspetta che si liberi il primo.

$$E[T_g] = E[T_a] + E[S_i]$$

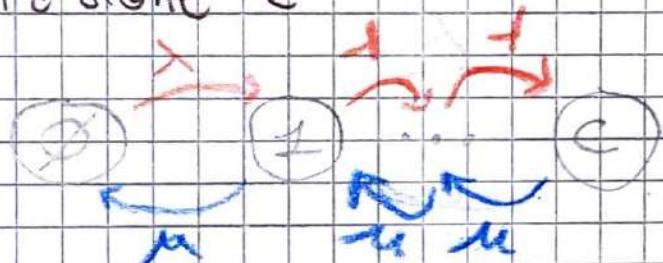
↳ "scelto un server" deve completare lì dentro il servizio.

Fino ad ora abbiamo sempre analizzato
con una coda infinita, quindi al 100%
un job riceverà il servizio, sempre.

Potremo introdurre il **buffer finito**
di capacità C ($C = 1$ arrivante + $C-1$ posti
in coda). La rappresentazione è
fatta con Markov:

M/M/1/C $\xrightarrow{\lambda} \boxed{M}$

\uparrow \downarrow Ploss



Sia π_i = prob di stare nello stato " i ",
devo scrivere le **equazioni di flusso**:

$$\begin{cases} \pi_0 \lambda = \pi_1 \mu \\ \pi_1 (\lambda + \mu) = \pi_0 \lambda + \pi_2 \mu \\ \sum \pi_i = 1 \end{cases}$$

cioè che esce = ciò che entra,
lo scrivo in funzione di π_0 ,
si troverà sempre.

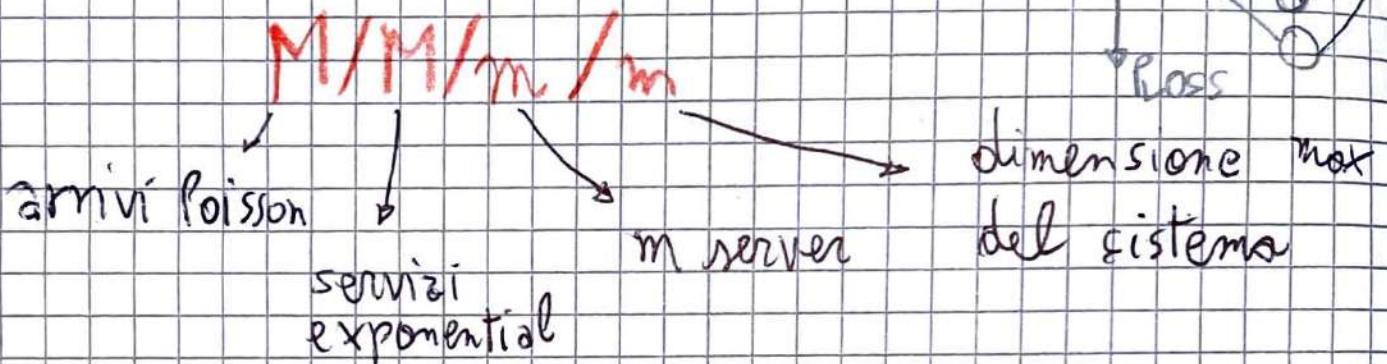
$$\pi_i = \left(\frac{\lambda}{\mu}\right)^i \pi_0$$

π_0 lo trovo grazie alla **NORMALIZZAZIONE** = $\frac{1}{\sum_{i=0}^C \left(\frac{\lambda}{\mu}\right)^i}$

A me interessa $\pi_C = \text{Ploss} = \left(\frac{\lambda}{\mu}\right)^C \cdot \pi_0 \sim \lambda^C = \lambda(1 - \text{Ploss})$

Con capacità finita il sistema è sempre stazionario.

MULTISERVER SENZA CODA λ



se dim. max del sistema = m server allora

non ho coda. Come combia da $M/M/1/\infty$?

Qui ogni nuovo job va direttamente nel server

Libero (re de). La catena è:

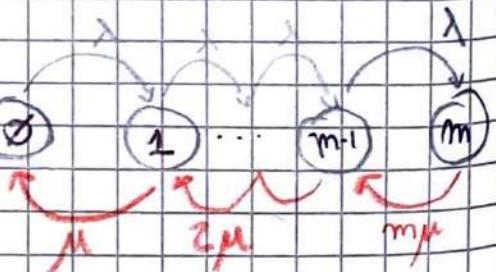
cio' che combia e' la parte

del servizio, dove c'e' $\mu, 2\mu, 3\mu, \dots m\mu$ in quanto

ogni job che esce libera un server.

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

ERLANG - B



quindi cambiano "solo" i! e j!

erlang B: n° chiamate perse, senza coda.

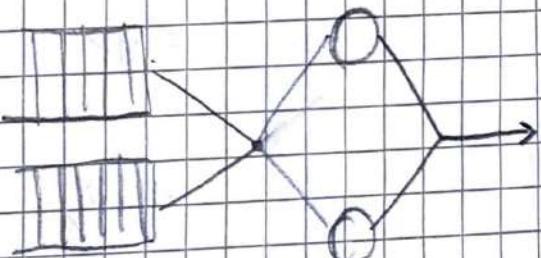
erlang E: n° chiamate in attesa, cioè in coda.



MULTISERVER MULTICODA

Visto nel corso di 2 code

e 2 server. La coda 1 è



prioritaria con P_1 , poi c'è P_2 .

CON PRELAZIONE

$$E[T_q] = P_1 \cdot \frac{P_2(P_1) \cdot E[S]}{1 - P_1} + P_2 \cdot \frac{P_1(P) E[S]}{(1 - P)(1 - P_1)}$$

per avere $E[T_s]$ notiamo $E[S_{\text{rit}_k}] = \frac{ECS}{\sum_{i=1}^k p_i}$

SENZA PRELAZIONE

$$E[T_q] = P_1 \cdot \frac{P_2 \cdot E[S]}{1 - P_1} + P_2 \cdot \frac{P_1 E[S]}{(1 - P)(1 - P_1)}$$

per avere $E[T_s]$ notiamo $E[S]$

PRIORITÀ

Tutto ciò che abbiamo visto prevedeva sempre al massimo una coda.

L'idea è quello di dividere le code per avere livelli di priorità diversi.

Dobbiamo SEMPRE chiederci:

- conosco la Taglia dei job?
- un job "importante" può prendere il posto di uno "meno importante" che nel mentre era servito?

La prima domanda è associata alla proprietà **SIZE BASED / ABSTRACT PRIORITY**.

La seconda è **PREEMPTIVE / NOT PREEMPTIVE**

Vediamo "caso per caso":

ABSTRACT + NON PREEMPTIVE

Se sono un job più importante, "sono avanti" nello scheduling viene servito. Se un job poco importante viene servito, devo aspettare.

$$E[T_{a_k}] = \frac{\frac{1}{2} E[S^2]}{(1 - \underbrace{\sum_{i=1}^k p_i}_{\text{Job della mia classe in Fila prima di me}}) \left(1 - \underbrace{\sum_{i=1}^{k-1} p_i}_{\text{Job di classi più importanti}}\right)}$$

NP-priority $E[S_{\text{REM}}]$
("non preemptive")

Per avere $E[T_{S_k}]$ valore nommo $E[S]$ ("abstract")

$E[T_{a_1}] \leq E[T_{a_2}] \leq \dots$ ugual per $E[T_{S_1}] \leq E[T_{S_2}] \dots$

Per $E[T_a]$ o $E[T_s]$ GLOBALI devo pesare i tempi per un peso p_k che spesso è da trovare per avere un certo QoS.

- Se $\text{EXP } \frac{1}{2} E[S^2] = P E[S]$.
 - uno clone K
 - uno generale
- generalmente, per ABSTRACT, $P_K = \bar{P} \cdot P \rightsquigarrow$ peso clone K

Miglioro i tempi?

- LOCALMENTE, miglioro clam priou forie, peggioro le meno importanti.
- GENERALMENTE, NON miglioro rispetto allo KP, poiché sono 2 scheduling ASTRATTI (UGUALI).

ABSTRACT + PREEMPTIVE

Cio' che combia e' che un job importante puo' scolzare uno meno importante anche se era sta venendo servito. Quindi cio' che mi aspetto che combi e' solo il numeratore, poiche' non aspetto piu' tutti, ma solo job piu' importanti di me!

$$E[T_{0K}] = \frac{\frac{1}{2} E[S^2] \sum_{i=1}^K \lambda_i}{\left(1 - \sum_{i=1}^K P_k\right) \left(1 - \sum_{i=2}^{K-1} P_k\right)}$$

ovvero ora aspetto λ in funzione di ' K ', prima li aspettavo sempre TUTTI.

Va da sé che ogni classe viene migliorata,

prima avevo $\frac{1}{2} E[S^2] \cdot \lambda$, ora $\frac{1}{2} E[S^2] \sum \lambda_k$ che solo nell'ultima classe coincidono!

($\sum \lambda_k = \lambda$ se sommo tutte le classi!)

Per $E[ET_{Sk}]$? ogni job deve considerare anche il tempo "rubato" da job più importanti.

Allora $E[ET_{Sk}] = E[T_{ak}] + \frac{E[S]}{1 - \sum_{i=1}^{K-p} p_i}$

SERVIZIO VIRTUALE

SULLE PRESTAZIONI?

LOCALMENTE: $E[T_{ak}] \leq E[T_{ak}]$

GENERALMENTE $E[T_a]$ ^{P priority} _(vedi numeratore) $\leq E[T_a]$ ^{NP priority} $= E[T_a]$

NON POSSO CONFRONTARE $E[TS_k]$ ^{P prior} VS $E[TS_k]$ ^{NP prior}

(perché $E[S] \leq E[S_{virt}]$, ma $E[T_{ak}]^P \leq E[T_{ak}]^NP$)

• SE EXP perdo ogni vantaggio, e' come KP, colpa della Memoylen.

• Ciò che ho scritto vale per **ARRIVI RANDOM** (**DISTR POISSON**)

SIZE BASED + NO PRELAZIONE

Mentre nell'abstract NON C'ERA un criterio specifico per ordinare i job, qui li ordiniamo in base alla loro SIZE.

Essendo SENZA PRELAZIONE, un job di qualsiasi classe aspetta la fine del servizio dei job precedenti.

Nella size based si usano gli INTEGRALI.

La divisione in classi avviene in base alle dimensioni dei job, mettendo due classi, con tempi servizio $[0, E[S]]$ e $(E[S], +\infty)$.

Le formule usate sono:

$$E[S_k] = \frac{1}{F(x_k) - F(x_{k-1})} \int_{x_{k-1}}^{x_k} t \cdot f(t) dt$$

P_k — cedo in questo "mposio".

Poi trovo $\lambda_k = \lambda \cdot P_k$

Moltiplicamente $P_k = \lambda_k E[S_k]$.

Non posso fare $P_k = P \cdot p_k$, questo è OK solo nell'ABSTRACT !!

Le formule sono come prima!

$$E[T_{a_k}] = \frac{\lambda E[S^2]}{[(1 - \sum_{i=1}^{k-1} p_i) \underbrace{(1 - \sum_{i=1}^k p_i)}_{\lambda \int_0^{\infty} t \cdot f(t) dt}]} \text{ se } \exp$$

CAMBIA SOLO
COME TROVO P_k

GLOBALMENTE, moltiplico per i tempi in coda per la probabilità p_k . Per $E[T_{S_k}]$ devo sommare a $E[T_{a_k}]$ il rispettivo $E[S_k]$!

LOCALMENTE $E[T_{a_k}]_{SB-NP} \leq E[T_{a_k}]_{abstract-NP}$

(combi come calcolo i p_k), mentre confrontando $E[T_{S_k}]_{SB-NP}$ vs $abstract-NP$ NON POSSO DIRLE

NULLA (posso avere $E[S_k]_{SB-NP} > E[S_k]_{ab-NP}$, magari nelle ultime classi, perché in 'ab' non ordino)

GLOBALMENTE $E[T_s]^{SB-NP} \leq E[T_s]^{ab}$ perchè

$$E[T_a]^{SB-NP} \leq E[T_a]^{ab-NP} \quad (\text{detto min})$$

$$E[S]^{SB-NP} = E[S]^{ab-NP}, \text{ i job quello sono!}$$

—. —. —. —. —.

SIZE BASED + PRELAZIONE

Caso più completo, dove non basta che il Job che vuole entrare abbia priorità superiore a quello in servizio; bensì il suo tempo di servizio deve essere MINORE del tempo di servizio RIMANENTE del job che vuole sostituirlo

$$E[T_{a_k}] = \frac{\lambda}{2} \left[\left(\int_0^{x_k} t^2 dF(t) \right) + (1 - F(x_k)) x_k^2 \right] \\ \left(1 - \sum_{i=1}^k p_i \right) \left(1 - \sum_{i=1}^{k-1} p_i \right)$$

$$E[S_{virz-k}] = \frac{E[S_k]}{1 - \sum_{i=1}^{k-1} p_i}$$

passando alle globali devo sempre pensare per
 $P_k = F(X_k) - F(X_{k-1})$

Spero lo mi confronta con S.Job. First No Preemp.,
ma sono incomparabili, allora le unisco in
Shortest Remaining Processing Time.

Completo, infatti al massimo lo mi calcolo
per uno specifico size.

Normalmente queste formule vengono FORNITE.

TEORIA delle V.A.

la sequenza di arrivi $A_i, A_{i+1}, A_{i+2}, \dots$, e' visibile come $A_i = A_{i-1} + R_i$, dove R_i e' il tempo passante (di interarrivo) tra A_{i-1} ed A_i . Se questi R_i sono INDIP. ed IDENT. DISTRIBUITI con $E[R_i] = \frac{1}{\lambda}$ ALLORA

La sequenza di arrivi A_1, A_2, \dots e' un processo di arrivi STAZIONARIO con rate λ .

Se λ varia? NON STAZIONARIO.

Gli interarrivi sono sequenza i.i.d di $\text{Exp}(\frac{1}{\lambda})$, da cui la sequenza di arrivi di Poisson(λ). Il singolo arrivo A_i e' Erlang($i, \frac{1}{\lambda}$)

RETI CODE e MARKOV (Routing)

Abbiamo ora che fare con come M/M/2 legate tra loro, quindi il flusso può andare da un centro all'altro, disperdersi etc..

Abbiamo CASO CHIUSO e CASO APERTO

CASO CHIUSO (non c'è flusso da/verso l'esterno).

• A centro ricevo eq. traffico : $= \text{FLOW IN}$

Sono linearmente indipendenti (throughput relativi), ne fingo uno e trovo gli altri.

• Calcolo le visite $V_{i,j} = \frac{Y_i}{Y_j} \rightarrow$ "fisso il riferimento"

Dati N job, M centri, sistema chiuso, si procede con Mean Value Analysis, algoritmo ricorsivo, si parte da $N=0$ job fino ad N .

Per definizione, a centro i , con $N=0$ job si ha $E[n_i | N=0] = 0$

MVA per reti chiuse! (Product Form)

Poi si passa a $N=1$ job, e si calcola

$$\underline{E[t_i(N)] = E[S_i] \left(1 + E[n_i(N-1)] \right)}$$

"tempo medio risposta entro"

(tempo di
un job nel
centro "i" con
 N job nel sist.)

Colpiscono al centro le entrate λ_i ($N=1$)

$$\lambda_i(N) = \frac{N}{\sum v_{j,i} E[t_j(N)]}$$

ed infine throughput
entro "i"

$$\underline{E[n_i(N)] = \lambda_i(N) \cdot E[t_i(N)]}$$
 da Little

Devo arrivare al n° di job N clienti.

Il tempo di risposta di un centro "i" è dato da:

$$\underline{E[t_{i,i}(N)] = \sum_{j \neq i} v_{j,i} E[t_j(N)]}$$
 cioè visite e

tempi negli ALTRI centri.

Il tempo di ciclo rispetto "i" è come
prima, ma considera anche il centro "i".
Vale $R_i(N) = V_i E(t_i(N)) = \underbrace{E[S_i]}_{Di} \cdot N_i \cdot (1 + E[n_i(N-1)])$

CASO APERTO

• anche qui, per ogni centro, scrivo $\lambda_i = \text{FLOW IN}$, procedendo SEMPRE per sostituzione, mai per adegnozione di una variabile. Normalmente sono esercizi "reti di Jackson".

• A centro calcolo $E[T_{S_i}] = \frac{1}{M_i - \lambda_i}$ (se $M_i > \lambda_i$)
e calcolo le visite $V_{i,j} = \frac{\lambda_i}{\lambda_j}$ (normalmente j è il flusso esterno)

Infine A centro calcolo $E[t_{r_i}] = \sum V_{j,i} \cdot E(t_j)$

Se arrivano, $E[m_i] = \lambda_i \cdot E(t_i)$ $\rightarrow E[T_{S_i}]$
è anche il throughput (STAZ)

Nel sistema aperto i tempi di risposta sono del tipo $V_i \cdot E(t_i)$, indipendenti da N .

Nel chiuso (vedi formula a minuti) uno $P_i \cdot (1 + E(m_i(N-1)))$, dipendente da N .

In questi esercizi con MIMIZ e arrivo di Poisson Vale Burke: ciò che esce da un centro va in quello successivo a seconda di come è definito (non ho "perdite" inutile)

Burke vale se NON C'È FEEDBACK, perché solo grazie all'indipendenza le probabilità dello stato congiunto è calcolabile come prodotto delle probabilità.

Con feedback + sist aperto risolve come nel caso aperto.

ANALISI OPERAZIONALE

Si usano dati misurati come parametri

È importante capire i dati forniti.

T = tempo osservazione

B = tempo sist. occupato

A = arrivi in T

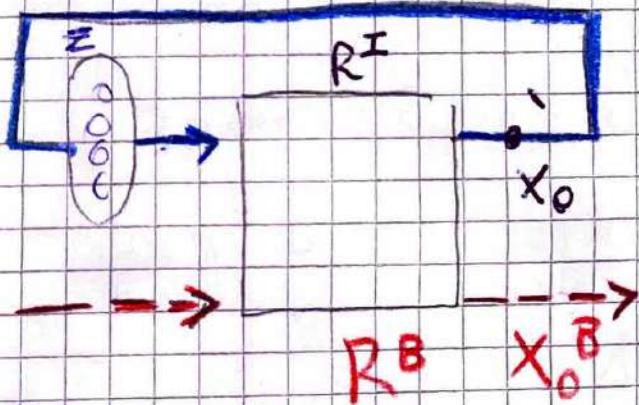
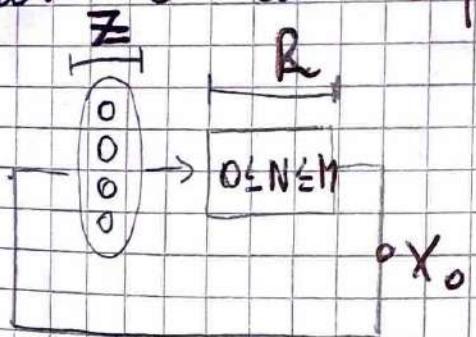
C = completamenti

$$\lambda = \frac{A}{T}, \quad \frac{C}{T} = \text{freq. usata } X, \quad U = \frac{B}{T}, \quad S = \frac{B}{C}$$

(\approx throughput)

$$U = \frac{C}{T} \cdot \frac{B}{C} = X \cdot S \quad (\text{legge utilizzazione})$$

Troviamo due tipi di sistemi :



Vediamo le leggi governanti tali sistemi

- Legge flusso forzato $X_i = \frac{V_i X_o}{\underbrace{SINGOLO}_{\text{CENTRO}} \underbrace{\downarrow}_{\text{TOT}} \underbrace{X_o}_{\text{SISTEMA}}}$

- Per sistemi INTERATTIVI, VISITE

il response time è $R = \frac{M}{X_o} - Z$ dove
INTERATTIVO

M è n° terminali, Z è think time.

- Il numero di thinker è $\#think = Z \cdot X_o$

- Legge utilizzazione è $U_i = X_i \cdot S_i$
 (vale per i singoli centri, non per X_o)

- Se il sistema è MISTO dev'essere considerate
 separate le parti batch ed interattive, cioè:

$$X_{disk} = X_{disk}^B + X_{disk}^I, \text{ ovvero misre i dati}$$

in modo corretto!

N.B.: $X_o = \frac{X_i}{V_i}$ $\frac{FLUSSO}{FORZATO}$ $\frac{UTILIZZ}{Vi}$ $\frac{(U_i) \cdot \frac{1}{V_i}}{S_i} = \frac{U_i}{D_i}$

DOMANDA

$$N_i = X_i \cdot R_i \text{ (Little)}$$

$$X^I_{disk} = X_{disk} - X^B_{disk} \text{ e' minimo se}$$

X_{disk} realizza $P \rightarrow 1$ cioè $X_{disk} = \frac{1}{S_{disk}}$
(completa più job ne il disco lo uso al minimo,
cioè ne opera in ETS_{disk})

- Se non si parla esplicitamente di sistemi interattivi, il tempo di risposta è: $R = \sum_i v_i R_i$
- Se abbiamo T, C_0 allora $X_0 = \frac{C_0}{T}$, devo vedere i dati che ho!

BOUND & Bottleneck

$$D_i = V_i \cdot S_i, \quad V_i = \frac{X_i}{X_0}, \quad D_i = \frac{X_i}{X_0} S_i = \frac{(C_i)(-)}{T} \frac{B_i}{(C_0) C_i} \quad (T) \frac{B_i}{(C_0) C_i}$$

$$U_i = X_i S_i = (X_0 V_i) S_i = X_0 D_i$$

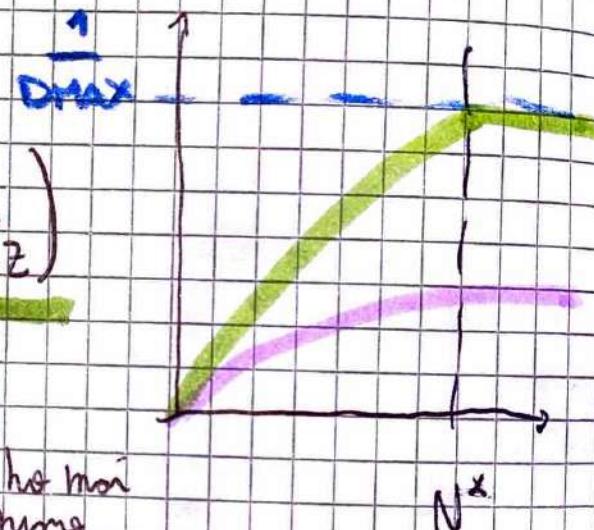
D_{MAX} = max tra le "D" domande.

$$D_{TOT} = D = \sum_i D_i \quad (i \in \text{centro})$$

THROUGHPUT

$$\frac{N}{ND+Z} \leq X(N) \leq \min\left(\frac{1}{D_{MAX}}, \frac{N}{D+Z}\right)$$

Sono l'ultimo
in coda sempre



dipende dai movimenti nel centro,
e via ne tutti reggono stessa persona, VERE altimenti

La soluzione la ho per $N^* = \frac{D+Z}{D_{MAX}}$, cioè

$$\text{impongo } \frac{1}{D_{MAX}} = \frac{N}{D+Z}$$

Per il tempo di Risposta R ?

caso peggiore: aspetto tutti: ND

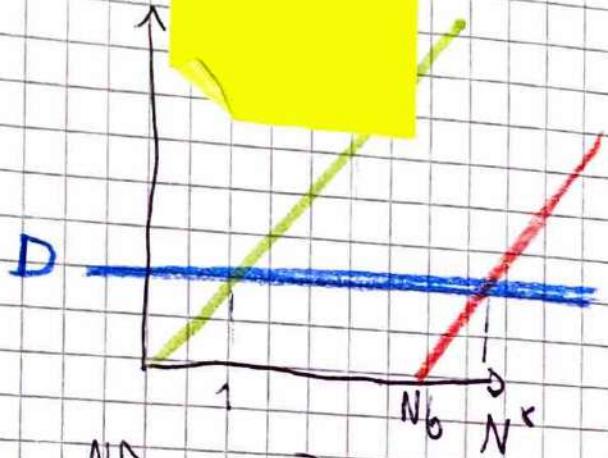
caso migliore:

aspetto solo me: D

$$\bullet \text{ se } R=0 \rightsquigarrow X_0 = \frac{1}{D_{\text{MAX}}} \rightsquigarrow \underline{\text{ND}_{\text{MAX}} - 2}$$

(non ci sono job)

$$\underline{\text{ND}} \geq R \geq \max \{ D, \underline{\text{ND}_{\text{MAX}} - 2} \}$$



Random Notes

servizi exp,
distribuzione ignota
(exp o k-erlg o...)

$$E[T_a] = \frac{P[E[S]]}{1-P} \cdot \left(\frac{1+C^2}{2} \right)$$

In Markov,

$$E[N_S] = \sum_{i=0}^c \pi_i \cdot i =$$

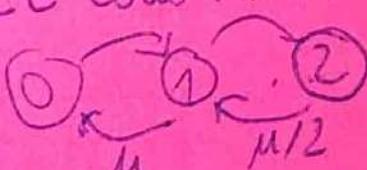
$$\pi_0 \cdot 0 + \pi_L \cdot 1 + \dots$$

MULTISER
MULTICODA
PREEMPTIVE,
per $E[T_{S_1}]$ sommo $E[S]$
(mai interrotto), per
 $E[T_{S_2}]$ uno $E[S_{\text{virt-2}}]$

SIZE BASED NP
con 2 classi, allora
non posso avere $E[S_K]$,
 I_K etc.. ma $\rho = \lambda E[S]$
è come uno KP

$$F_{\text{unit}} = \frac{x-a}{b-a}$$

Se Coda FINITA + PS



perché divido!

"Z" si misura in "op"

o se devo con
slowdown PS e
 $PS = \frac{1}{1-P}$, FIFO:

$$\text{var}(E[T_a]/x) +$$

$$\text{medio Req} = 0 \cdot \pi_0 + 1 \cdot \pi_1 + \dots$$

$$\sigma^2(\text{Req}) = E[\text{Req}^2] - \text{medio}^2$$

(1 · π₁ + 2² · π₂)

In un sistema interattivo

$$R = \frac{M}{X_0} - Z$$

$R \rightarrow$ se $X_0 \nearrow$, $X_0 \text{ MAX}$
se tutto ciò che arriva serve
cioè $P=1 \rightarrow X_0 = \frac{1}{\text{tempo servizio}}$

Se coda chiusa, e
la "Di" \rightarrow MVA ridotta, per tempo di risposta

$$V_i E(t_{i(N)}) < E[S_i] V_i / (1 + E(N-1))$$

$$R_i(N) = D_i (1 + E_i(N-1))$$

$$X_0(N) = \frac{N}{\sum R_i}$$

e $\sum R_i$ = tempo risp. totale