

Sotto una rete di code abbiamo astrazioni diverse. Quando abbiamo visto Erlang, essa proveniva da un processo di Markov. La rete di code è una visione ad alto livello della rete di Markov. Quando definiamo come si comporta la rete di code, definisco Markov.

esempio reti chiuse:

Il centro 1 distribuisce parte del suo throughput in 2 e 3, mentre l'altra parte è feedback.

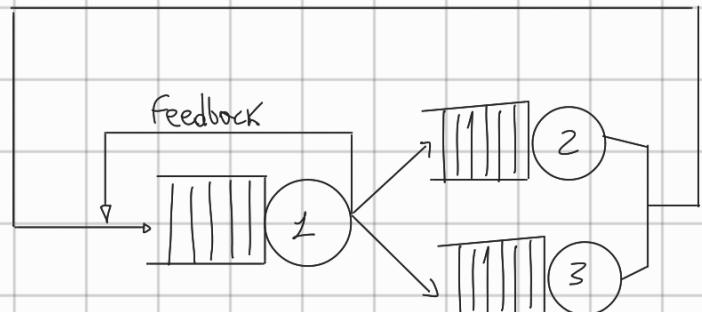
Le risorse 2 e 3 sono in alternativa, e quando terminano tornano come flusso in 1.

$M = \text{numero centri} = 3 ; n = \text{popolazione} = 2$

Poichè chiuso, circolo continuamente nel sistema.

Ipotizziamo scheduling FIFO e tasso esponenziale, quindi  $M/M/1$  FIFO.

Il tasso  $\mu_1 = 4$  mentre  $\mu_2 = \mu_3 = 2$  (il primo è due volte più veloce)



Introduco la matrice di routing (di probabilità) che indica qual'è la percentuale uscente da 1 verso 2, oppure verso 2 oppure verso 3.

Ovviamente  $P_{2,1} = P_{3,1} = 1$  nel senso che tutto ciò che esce da centri 2 e 3 va in unica direzione.

Come si possono distribuire i 2 job?

Centro1 Centro2 Centro3

$s_1 (2 \ 0 \ 0)$

$s_2 (1 \ 1 \ 0)$

$s_3 (1 \ 0 \ 1)$

$s_4 (0 \ 2 \ 0)$

$s_5 (0 \ 1 \ 1)$

$s_6 (0 \ 0 \ 2)$

Questi sono "stati". In generale  $s = (n_1, n_2, \dots, n_M)$ . (singolo stato,  $n_i$  dice dove sono i job)

STATO DEGLI SPAZI  $E = \{s \mid n_i \geq 0, \sum_{i=1}^M n_i = N\}$

↳ ha  $N$  job nel sistema, ogni stato ne ha sempre  $M$ .

La cardinalità  $|E| = \binom{N+M-1}{M-1}$

Nel nostro esempio era semplice trovarli, ma in casi più complessi?

Uso il seguente algoritmo: parto da stato  $(N, 0, \dots, 0)$  cioè tutti nel primo centro.

Poi passo a  $(N-1, 1, 0, \dots)$ ,  $(N-1, 0, 1, \dots)$  cioè distribuisco il primo job nei centri.

Poi passo a  $(N-2, 2, 0, \dots)$ ,  $(N-2, 0, 2, \dots)$  e così via.

Associo per ogni  $s_i$  un tempo di vita  $T_{s_i} = \text{tempo di vista in stato } s_i = \text{tempo passato da quanto sto in quello stato a quando esco da tale stato.}$

La distribuzione dei job nella rete appena fatto ci porta a definire un processo stocastico sotto alla rete, con stati e tempi di vista nello stato.

I nostri processi sono Markoviani, e possiamo dire che un processo lo sia se:

1) tutti gli stati della rete sono rappresentabili come un insieme di stati :

- mutualmente esclusivi: NON POSSO TROVARMI IN DUE STATI SIMULTANEAMENTE

- collettivamente esaustivi: L'INSIEME DI STATI DESCRIVE COMPLETAMENTE IL SISTEMA, copre tutte le condizioni possibili.

2) Lo stato futuro del processo sia dipendente SOLO dallo stato presente (cioè memoryless)

Possiamo dire che un processo è markoviano se sono valide queste condizioni!

Nell'ipotesi che i tempi di servizio siano esponenziali, allora il passaggio da uno stato ad un altro stato è un tempo di servizio esponenziale. Se il centro avesse distribuzione generale, la notazione di stato sarebbe  $(\eta_i, \xi)$  dove ' $\xi$ ' = fase in cui si trova.

Quindi la fase potrebbe cambiare, e il job rimanere dove si trovava.

L'ipotesi esponenziale ci permette quindi di attuare tecniche non applicabili in altri casi.

Ad ogni stato ' $s$ ' associo tempo di vita. Nell'ipotesi sufficiente-non necessaria di tempi

servizio esponenziale ho:  $E(t_{v_i}) = \frac{1}{\alpha_i}$  dove al denominatore ho il tasso di uscita =  $\mu_i = \alpha_i$   
(in questo caso!)

Se prendo  $s_1 = (2 \ 0 \ 0)$ , allora il tasso di uscita è come sopra. Ma come ci esco da questo stato?

Dipende dal centro1, poiché centro2 e centro3 sono vuoti.

Se prendo  $s_2 = (1 \ 1 \ 0)$ , ho un job in centro1 e centro2. Vi esco se termina job in centro1 o job in centro2. In che stati posso arrivare?

posso arrivare in  $(0 \ 2 \ 0)$ , perché job in centro1 va in centro2,

posso arrivare in  $(2 \ 0 \ 0)$ , perché job in centro2 ritorna in centro1,

posso arrivare in  $(0 \ 1 \ 1)$ , perché job in centro1 va in centro3.

Si entra in  $s_2$  se un job nel centro2 è già presente, e ne arriva uno in centro1 o viceversa.

Simultaneamente è molto improbabile e non lo considero. (in  $s_2$  un job 'fermo', arriva l'altro).

Si esce da  $s_2$  per una partenza da centro1 o centro2 (come abbiamo visto).

Abbiamo quindi che:  $\alpha_2 = \mu_1 + \mu_2$ ,  $E(t_{v_2}) = \frac{1}{\mu_1 + \mu_2}$

Ho passaggio tra stati solo cambiando un componente alla volta, non ne cambio due insieme!

$P_{ij}$  = Probabilità di transizione da stato i a stato j. Sempre facendo riferimento all'esempio:

(110) -> (200). Nel sistema termina centro2 che manda flusso a centro1.

Poichè lavoriamo in contesto markoviano possiamo dire che:

$r(i,j) = \frac{\text{tasso di uscita dal nodo} * \text{probabilità di routing tra i nodi corrispondenti alla transizione}}{\text{tasso di uscita dallo stato}}$

al denominatore ho tutto il tasso di uscita, al numeratore una parte di questo tasso che provoca la transizione moltiplicata per la probabilità di routing di quella transizione.

Nell'esempio da  $s_2 \rightarrow s_1$ :

$$\begin{array}{c} \text{esce da } s_2 \\ \text{indici di stati: (prob da stato 2 a stato 1)} \end{array}$$

$$\begin{array}{c} M_2 \cdot P_{2,1} \\ = R_{2,1} \\ \uparrow \downarrow \\ M_1 + M_2 \\ \text{indici di nodi} \end{array}$$

$$\begin{array}{c} (1,0) \quad (2,0,0) \\ \text{in node 2} \\ \text{in node 1} \end{array}$$

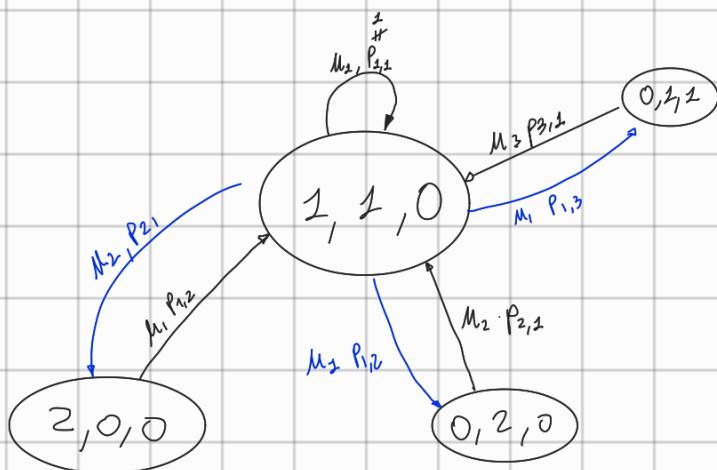
stato manda stato<sub>1</sub> + stato<sub>2</sub>

Definiamo  $q(i,j)$  = frequenza di transizione dallo stadio  $S_i$  a  $S_j$ , corrisponde a:  $Q_i \cdot \underbrace{r_{i,j}}_{\substack{\text{prob.} \\ \text{transizione} \\ \text{tra 2 stati: } (i \rightarrow j)}}$

attenzione:  $q_{i,j} = \alpha_i \cdot r_{ij}$  ~~MARKOV~~  $= \alpha_i \cdot$  tasso uscita nodo. . . .  
tasso uscita da i

ovvero il processo di markov è definito sulla rete, poichè al numeratore (ciò che resta) descrivo ciò che succede nella rete. Allora la frequenza di transizione sotto queste condizioni è data dalla frequenza di transizione tra i due centri che provocano questo cambiamento di stato.

Disegniamo il GRAFO DI TRANSIZIONE DEL PROCESSO nel caso (1, 1, 0)



$M_i$  = centro che lascia partire il job

$$P_{ij} = \begin{cases} i = \text{partenza job} \\ j = \text{arrivo job} \end{cases}$$

Esiste BILANCIAMENTO FLUSSO : tasso uscita da Si = tasso di ingresso in Si. (stazionarietà).

Potremmo definire la probabilità istantanea, e che poi tende ad una stazionarietà se:

- **irriducibile**: ogni stato raggiungibile in un numero di passi finiti. Basta far vedere che ogni centro sia raggiungibile da ogni altro.
- **aperiodicità**: posso tornare ad uno stato  $S_i$  in ogni istante  $t$ , non ho periodicità.
- **ricorrenza positiva**: il tempo di ritorno in uno stato è finito.

Le prime due danno l'esistenza della probabilità limite  $\pi(S_i) = \lim_{t \rightarrow \infty} \pi(S_i, t)$

La terza, insieme alle prime due, mi dice che è UNICA ed è NON BANALE (diversa da 0, perchè sennò potrei dire che cambia continuamente e quindi la probabilità è 0, ma è appunto banale).

Prima abbiamo lavorato sul grafo, in generale l'equazione è:

$$\pi(S_i) \cdot (\text{flusso in uscita}) = \sum_{\substack{\forall S_j \in E \\ \text{stati dove posso andare}}} \pi(S_j) \cdot (\text{flusso in entrata in } i) ;$$


Sommo a destra e a sinistra  $\pi(S_i)(1 - \text{flusso uscita})$

$$\pi(S_i) = \pi(S_i)(1 - \text{flusso uscita}) + \sum_{\substack{\forall S_j \in E}} \pi(S_j) \cdot (\text{flusso in entrata in } i) \quad \text{eq. bilanciamento globale}$$

Lo scrivo in forma matriciale:

$$\overline{\pi} = \overline{\pi} \cdot Q$$

(tutti gli stati)

ha elemento  $q_{ij}$   
(freq. transiz. da stato  $i$  a  $j$ )

$$\overline{\pi} = [\pi(S_1), \pi(S_2), \dots, \pi(S_{|E|})], \quad Q = \begin{bmatrix} q_{11} & & & \\ q_{21} & \ddots & & \\ \vdots & & \ddots & \\ q_{|E|1} & & & q_{|E||E|} \end{bmatrix}, \quad \text{tassi ingresso in } i^{\text{a}}$$

colonne = tassi ingresso allo stato

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,|E|} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,|E|} \\ \vdots & \vdots & \ddots & \vdots \\ q_{|E|,1} & q_{|E|,2} & \cdots & q_{|E|,|E|} \end{bmatrix}$$

righe = tassi uscita  
somma riga = 1

Poiché  $\overline{\pi} Q - \overline{\pi} = \emptyset \rightarrow \overline{\pi}(Q - I) = \overline{\pi} S$

generatore del processo

quando faccio  $-I$ , tolgo  $1 - \sum(q_{i,1}, q_{i,2}, \dots, q_{i,|E|}) = 0$

A questo punto, posso imporre condizione di normalizzazione, sostituendo in una colonna qualsiasi di  $S$  la condizione di normalizzazione (ovvero colonna di tutti '1'), ottenendo  $S'$ .

Ottengo  $\overline{\pi} \cdot S' = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$  cioè la probabilità di tutti gli stati \* prima colonna (tutti 1) = 1, condizione di normalizzazione.

La soluzione finale è:  $\overline{\pi} = [1, 0, \dots, 0] \cdot S'^{-1}$ , la soluzione sta nella prima linea in questo caso!

Partendo da eq. bilanc. globale, ho sommato a dx e sx  $\pi(S_i)$  per scriverlo in forma matriciale  $\overline{\pi} = \overline{\pi} \cdot Q$  per poi arrivare a  $\overline{\pi}(Q - I) = \emptyset$  (togliendo l' '1' in viola dalla diagonale), mi sono messo in condizione di normalizzazione (da  $S$  a  $S'$  cambiando una colonna). Ho poi trovato  $\overline{\pi} = [1 0 \dots] \cdot S'^{-1}$ , nella prima riga ho la soluzione!

# Lezione 10 maggio

Dopo aver visto come schematizzare le reti di code, vorremo anche valutarle.

Supponiamo un sistema stabile, aventi le probabilità  $\pi(\underbrace{m_1, m_2, \dots, m_M}_S)$

Definiamo la **probabilità marginale della lunghezza della coda** come  $P_i(m) = \sum_{S: M_i=n} \pi(S)$

se  $i=2, n=3 \rightarrow \sum$  probabilità degli stati  
aventi:  $m_2=3$  (3 job in centro 2)

Calcolate queste, ricavo la **popolazione media al centro 'i'**:

$$E(m_i) = \sum_{M_i=0}^M m_i \cdot P_i(m_i)$$

"valore"  
"peso"

Siamo interessati anche all'utilizzazione  $U$ , che dipende dai serventi:

- **centro a servente singolo**:  $U_i = 1 - P_i(\emptyset)$

- se il centro è a servente multiplo, con 'm' centri:  $U_i = \frac{E(c_i)}{m_i}$  rispetto ERLANG-C (è chiuso)  
=  $\sum_{j=1}^{m-1} \frac{j \cdot P_i(j)}{m_i}$  +  $\sum_{j=m_i}^N \frac{P_i(j) \cdot m_i}{m_i}$  < 1  
SUBITO SERVITI CODA

- se il centro è **infinite server** allora non ho coda (centro di ritardo), lo immagino come un multiserver che può accogliere tutti i centri senza creare coda, ovvero  $m_i \rightarrow N$

$$U_i = \frac{E(c_i)}{N} = \sum_{j=1}^N \frac{P_i(j)}{N}$$

Passiamo adesso ai throughput: in un sistema aperto stazionario, corrisponde ad un parametro di input (traffico in ingresso), ma NEL SISTEMA CHIUSO NO, qui ho una popolazione  $N$  che circola in modo indefinito nella rete. Calcoliamo throughput ' $X$ '

- nel **servente singolo**:  $X_i = \mu_i \cdot U_i$

è un'applicazione di Little ( $\rho = \lambda E(s)$ ), cioè il throughput 'i' è il suo tasso  $\mu_i$  per la probabilità di non essere vuoto.

- nel **multiserver**:  $X_i = m_i \cdot \mu_i \cdot U_i = m_i \cdot \mu_i \cdot \frac{E(c_i)}{m_i} = E(c_i) \cdot \mu_i$   
tasso singolo servente  
quanto sono pieni in media.

osserviamo che:  $X_i = \sum_{j=1}^{m-1} \frac{j \cdot P_i(j)}{m_i} + \sum_{j=m_i}^N P_i(j) \cdot m_i \cdot \mu_i$  ovvero è una somma pesata!



- nell'**infinite server**, come abbiamo visto, ho  $N$ :  $X_i = N \cdot \mu_i \cdot U_i = \sum_{j=1}^N j \cdot \mu_i \cdot P_i(j)$

ovvero throughput è il numero di job\*tasso singolo job\*probabilità che ci sia un certo numero di job, che è la 'solita media pesata'.

Ora ho tutti gli indici locali  $U_i, X_i, E(m_i)$ , calcolo l'ultimo con Little  $E(t_i) = \frac{E(m_i)}{X_i}$ , dipende dal centro!

In una rete chiusa un job potrebbe passare per un centro più volte. Il tempo di risposta deve essere ben definito.

- **Tempo di risposta**: da quando arrivo a quando me ne vado dal sistema.(NO per reti chiuse)

- **Residence Time**: tempo di risposta del centro 'i' per tutte le visite.

Vediamo gli indici globali:

Supponiamo  $P(j,k)$  la probabilità di andare nel centro  $k$ .

Se volessi vedere il throughput rispetto ad  $i$  (quanti job escono da  $i$ )  $= X_i$

Se invece lo vedo per l'arco  $A$ :  $X_A = X_j \cdot P_{j,k}$

Definiamo la legge del tempo di risposta:  $E(t_{r,i}) = \sum_{j=1, j \neq i}^M E(t_{j,i}) \cdot V_{j,i}$

sto in centro 'i', esce un job 'x', dopo quanto tempo ritorna nella coda del centro 'i'? faccio medio per più job)

n° volte che il job visita "i" dopo "j"



(se conto  $E(T_{si}) = \text{tempo di ciclo}$ )

Il numero medio di visite lo calcolo con le equazioni di traffico:  $\bar{Y} = \bar{Y}P$

$P$  è la matrice di routing, e ci dice ad esempio come il carico si ripartisce tra i 3 centri, anche qui, come per le equazioni di flusso quello che esce = quello che entra.

$$\text{centro } 2 : Y_1 = Y_1 P_{1,1} + Y_2 + Y_3 \quad Y_2 = Y_1 P_{1,2} \quad Y_3 = Y_1 P_{1,3}$$

sono linearmente dipendenti e vengono detti 'throughput relativi'.

$$\text{Definiamo le visite come: } V_{j,i} = \frac{Y_j}{Y_i} \quad ; \quad \text{se } Y_i = 1 \rightarrow \begin{cases} Y_2 = P_{1,2} \\ Y_3 = P_{1,3} \end{cases} \rightarrow V_{1,2} = \frac{Y_1}{Y_2} = \frac{1}{P_{1,2}}$$

Se il sistema fosse aperto (posso uscire dalla rete chiusa) potrei definire:

- la **probabilità di uscire**, a partire dal centro 'i', come:  $P_{i,\emptyset} = 1 - \sum_{j=1}^M P_{i,j}$

- il **flusso che arriva** in 'i' come:  $\lambda_i = \gamma + \sum_{j=1}^M \lambda_j \cdot P_{j,i} \doteq X_i$  throughput vero (stazionario)

$$- \text{le } \text{visite in } i: V_i = \frac{\lambda_i}{\gamma} = \frac{X_i}{\gamma} \leftarrow \begin{array}{l} \text{throughput "i" vero} \\ \text{throughput "rete" vero} \end{array}$$

## ESERCIZIO

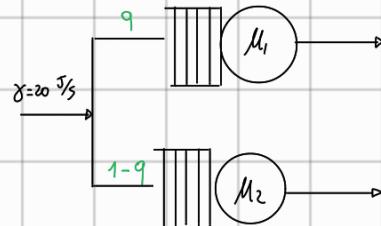
In un contesto di allocazione di file, ho due dischi eterogenei (diversi). Questi due dischi hanno tempi di accesso diversi, il disco1 ha 30 ms, il disco2 ha 46 ms. Il flusso è 20 j/s

Vorrei trovare 'p' e 'q' ottimali per migliorare il tempo di allocamento e minimizzare il tempo  $E(T_s)$

Svolgimento:

In questo tipo di esercizi vogliamo stessa utilizzazione per entrambi.

Se così non fosse avrei un collo di bottiglia, non devo massimizzare il disco più veloce.



Dai dati ottengo:  $\mu_1 = \frac{1}{0.035} = 33,3 \text{ j/s}$ ;  $\mu_2 = \frac{1}{0.045} = 22,2 \text{ j/s}$  (più lento)

Imposto  $P_1 = P_2$ , servono equazioni traffico!  
I dischi non comunicano, allora  
la matrice di routing è 0. (ogni  $P_{ji} = p$ )

$$P_i = \frac{\lambda_i}{\mu_i} \rightarrow \frac{\lambda_1}{\mu_1} \equiv \frac{\lambda_2}{\mu_2}$$

Non li ho

Vedo se i dati mancanti sono recuperabili dalle visite:

$$\begin{cases} V_1 = \frac{\lambda_1}{\gamma} = q \\ V_2 = \frac{\lambda_2}{\gamma} = 1 - q \end{cases} \rightarrow P_1 = \frac{q \cdot \gamma}{\mu_1} \equiv P_2 = \frac{(1-q) \cdot \gamma}{\mu_2} \Leftrightarrow q = 0.6052, \text{ disk2 riceve } 60\%.$$

Possiamo anche valutare il tempo necessario per attraversare l'intera rete:

$$E(T_s) = \underbrace{q \frac{1}{\mu_1 - \lambda_1}}_{0,02851} + \underbrace{(1-q) \frac{1}{\mu_2 - \lambda_2}}_{0,02851} = 0,570286$$

NB: i tempi sono esponenziali, KP usabile.

$$\rightarrow \frac{q}{\mu_1 - \lambda_1} = \frac{q}{\mu_1 - \cancel{\lambda_1} + \cancel{\lambda_1}q} = \frac{q}{\mu_1 - p_1 \mu_1} = \frac{q}{\mu_1 [1-p_1]} = \frac{q}{\mu_1} \frac{1}{1-p_1} = \frac{q E[S_1]}{1-p_1}$$

$E[S_1]$  in  $\text{ms}$

$0,02851$

$E(T_s)$  in  $\text{ms}$

da  $V_1$

da  $p_1 = q \delta$