Sistema a Coda M/M/1 con arrivi rallentati

Leonardo Serrentino 180043
Luca Corticelli 176087

Il seguente progetto simula la gestione del traffico e di elaborazione dei pacchetti di un sistema a coda di tipo M/M/1 (notazione di Kendall) con arrivi rallentati.
I parametri caratteristici sono infatti la distribuzione degli arrivi, la descrizione del servizio ed il numero di servitori, definiti nel tipo di sistema, mentre Y (capacità del sistema), N (Cardinalità potenziale) e Z (Disciplina della coda) essendo non specificate, assumono i valori di default (infinito, infinito, FIFO).

La simulazione App.java lavora con un oggetto di tipo MM1, nel quale è presente una LinkedList di capienza infinita di oggetti di tipo Pacchetto, ognuno dei quali contiene un id incrementale (da 0 quando si avvia il simulatore) e un campo per il contenuto informativo.
Il generatore produce ed instrada in modo asincrono $n=\frac{\alpha}{k+1}$ pacchetti al secondo con rispetto dello stato k in cui si trova al tempo t, che verranno linkati alla coda (LinkedList) per essere poi elaborati (tramite politica FIFO) dal Server, a gruppi di mu pacchetti al secondo.
Le uniche variabili da cui dipende l'intero sistema sono le scelte iniziali di **α** e **μ**.

Il simulatore lavora per un tempo di 20 secondi, discretizzato a 1 secondo, nel quale produce ed elabora gli output del sistema:
- il vettore **Pk** (probabilità di essere nello stato k) di alpha elementi;
- **Ls** (numero medio di pacchetti nel sistema | **E{k}**);
- **Lq** (numero medio di pacchetti in coda | **E{q}**);
- **Wq** (tempo medio di attesa nel sistema)
- **Ws** (tempo medio di attesa in coda)

Vengono inoltre forniti in output valori impliciti, ad hoc per il sistema in questione:
- **ρ** (fattore di utilizzo)
- matrice **Q** (matrice delle frequenze di transizione)

La matrice Q è una matrice tridiagonale che contiene nella sottodiagonale i valori di mu per ogni k, nel nostro caso $\mu(k)=\mu$; nella sopradiagonale i valori di $\lambda(k)=\frac{\alpha}{k+1}$;

infine la diagonale contiene l'opposto del tasso di nascite + il tasso di morti di pacchetti del sistema allo stato k. Una considerazione interessante (ed a posteriori quasi scontata) è la correlazione tra la dimensione della matrice di transizione Q e l' α scelto come parametro caratterstico, infatti il numero massimo di pacchetti nel sistema, ovvero il massimo di lambda è esattamente uguale al valore che assume quando il sistema è vuoto, esattamente all'avvio del sistema di elaborazione

$$max(\lambda)=\frac{\alpha}{k+1}, k=0$$

quindi il valore di k che massimizza la funzione λ(k) è proprio 0, quindi insieme vuoto.

Funzionamento del simulatore

Il simulatore prende come parametri di ingresso α e μ.

In ogni secondo il thread calcola all'inizio il valore del λ corrente in funzione dello stato in cui si trova, e genera al ritmo di 1 pacchetto ogni $\frac{1}{\lambda(k)}$ secondi. Essendo discretizzato a runtime è possibile che la generazione del pacchetto avvenga nel Thread successivo, e a lungo andare la propagazione di questo errore diventa troppo evidente, quindi, per venire incontro al computer sul quale viene simulato il sistema, diamo un tempo di 0.5 secondi per generare tutti i pacchetti, e aspetta in attesa lo scoccare del secondo; ci sarà comunque propagazione dell'errore ma in questo modo è contenuta e possiamo avere dei grafici molto simili a quelli teorici.

Il simulatore lavora per conteggio, quindi non vi sono altre formule applicate oltre al

int lambdaCalculation(int k);

Nella fase di analisi, invece, facciamo lavorare un simulatore teorico che, basato su iterazioni macchina e formule di teoria, produce gli stessi grafici e valori di output del simulatore "realistico", ma con risultati didatticamente più affidabili.

Il calcolo del Pk viene fatto attraverso la notazione esponenziale

$$e^{-\frac{\alpha}{\mu}} * \left(\frac{\alpha}{\mu}\right)^{k * \frac{1}{k!}}$$

e poi calcolando

$$Ls = \frac{\alpha}{\mu}$$

dal quale ricaviamo Ws tramite la **relazione di Little**

$$Ws = \frac{Ls}{E[\lambda]}$$

con

$$E[\lambda] = \rho * \mu \quad , \quad \rho = (1 - P0) = (1 - e^{-\frac{\alpha}{\mu}})$$

dal quale poi ricaviamo

$$Wq = Ws - Tx = Ws - \frac{1}{\mu}$$

calcolando infine

$$Lq = Wq * E[\lambda] = \frac{\alpha - \rho\mu}{\mu}$$

Nel plot finale dei vari grafici di output vediamo infatti che il grafico teorico è molto smooth, mentre quello del simulatore a threads ed eventi è caratterizzato da del "rumore", dovuto all'asincronia del sistema di generazione e di elaborazione.

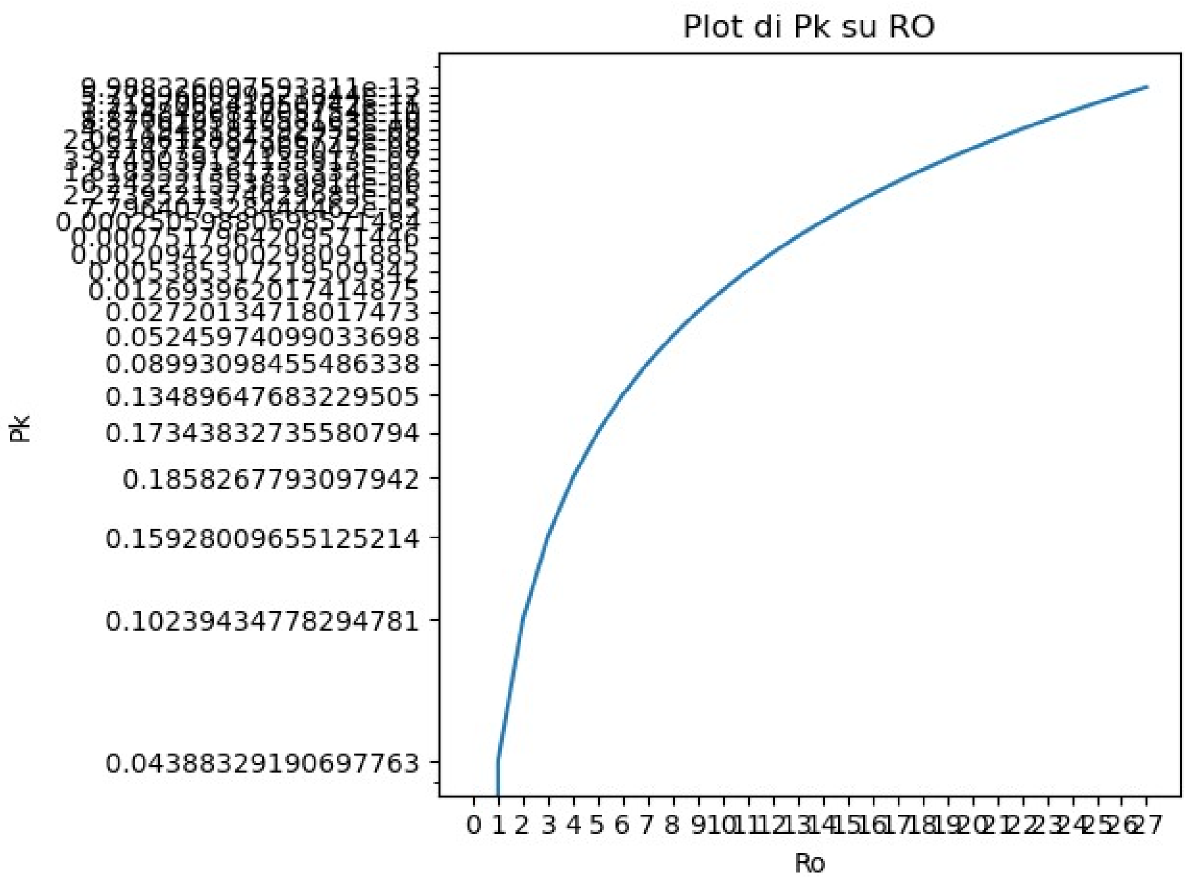
OUTPUT DELLA SIMULAZIONE

Di seguito gli output per una simulazione di 20 secondi di un sistema M/M/1 impostando un α per gli arrivi rallentati α=28 e un μ=6

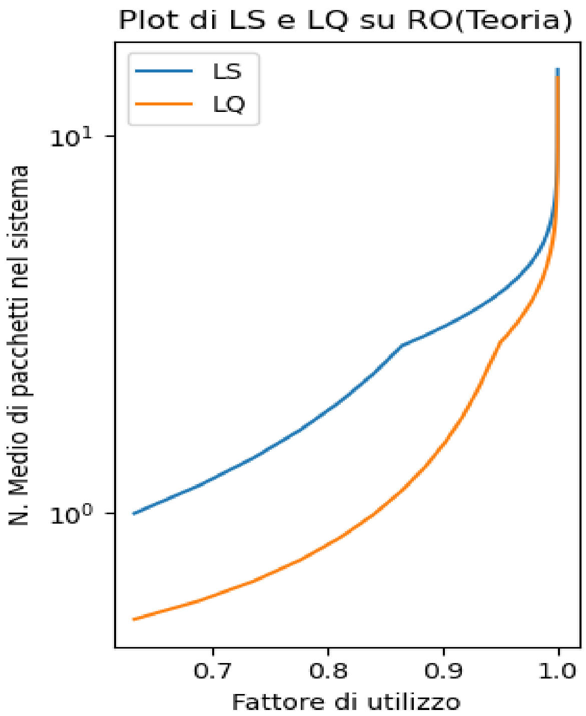
Matrice di Transizione di stato **Q**

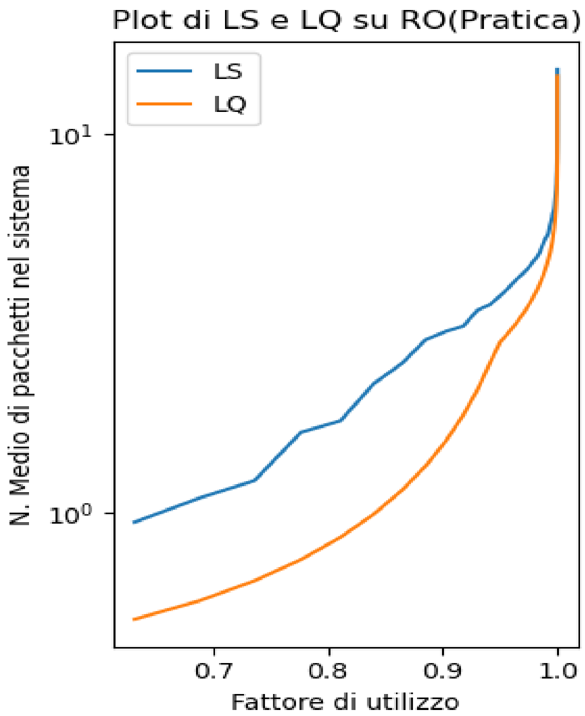
-28	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.0	-20.0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	6.0	-15.0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	6.0	-13.0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	6.0	-11.0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	6.0	-10.0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	6.0	-10.0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	6.0	-9.0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	6.0	-9.0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	6.0	-8.0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	6.0	-8.0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	6.0	-8.0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	6.0	-8.0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6.0	-8.0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0	-7.0	1

Grafico del Pk in funzione di ρ

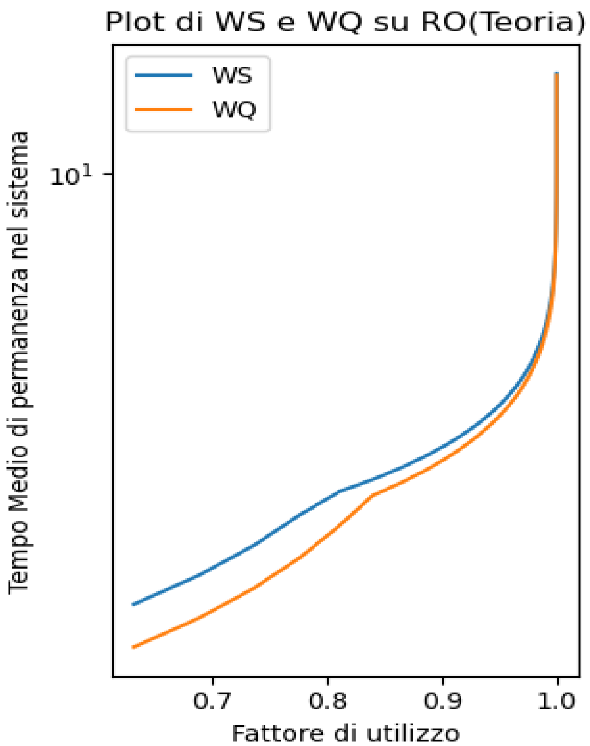


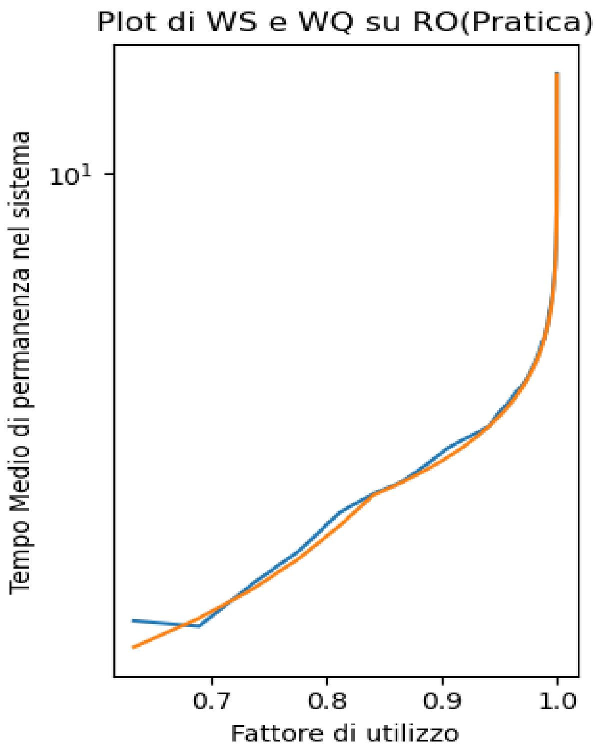
Confronto tra il Plot teorico di Ls e Lq e quello dato in output dal simulatore





Medesimo confronto tra plot teorici e pratici di Ws e Wq in funzione di ρ





Tracciamento dei pacchetti nel tempo

t=tempo, GT=GeneratiTotali, GA=GeneratiAdesso, DC=DimensioneCoda, ET=ElaboratiTotali, EA=ElaboratiAdesso

t	GT	GA	DC	ET	EA
0	28	28	0	6	6
1	29	1	22	12	6
2	30	1	17	18	6
3	32	2	12	24	6
4	35	3	8	30	6
5	39	4	5	36	6
6	46	7	3	42	6
7	51	5	4	48	6
8	58	7	3	54	6
9	63	5	4	60	6
10	70	7	3	66	6
11	75	5	4	72	6
12	82	7	3	78	6
13	87	5	4	84	6
14	94	7	3	90	6
15	99	5	4	96	6
16	106	7	3	102	6
17	111	5	4	108	6
18	118	7	3	114	6
19	123	5	4	120	6

CONSIDERAZIONI E CONCLUSIONI

Lavorando con questo tipo di sistema abbiamo notato che scegliendo un α ed un μ anche parecchio distanti avremo una coda molto lunga nel primo secondo, il cui q (numero di pacchetti in coda, ovvero nel nostro caso $q=k-1$) andrà via via stabilizzandosi fino ad arrivare ad un valore costante, o in un suo intorno, con una varianza bassissima. Questo è dovuto al fatto che il lambda varia in modo inversamente proporzionale allo stato k, quindi all'aumentare dei pacchetti nel sistema ne verranno generati meno, mentre allo svuotarsi del sistema ne vengono prodotti di più.

A differenza dei sistemi nei quali λ e μ sono costanti nel tempo, il sistema ad arrivi rallentati non sarà mai instabile: avendo un sistema complesso di aggiornamento del λ si adatta alla situazione corrente per la presa in carico dei pacchetti nella coda, quindi la coda del sistema non divergerà mai nel tempo, al contrario, arriverà ad avere un numero costante di pacchetti nella coda o, come già detto, un intorno di quel valore. Al contrario, nei sistemi con λ e μ costanti nel tempo, la differenza tra λ e μ scelti inizialmente comporta la possibile instabilità del sistema, arrivando a divergere nel caso λ sia troppo più grande di μ.

È pur sempre vero, però, che nei sistemi non rallentati il valore di λ non rimane realmente costante, infatti segue una distribuzione poissoniana, quindi non sarà mai davvero instabile, nella realtà il numero di pacchetti che circolano al di fuori del sistema che devono essere serviti tende e a decrescere nel tempo, quindi ad un certo punto la coda si svuoterà. Didatticamente possiamo di re che il numero N di cardinalità massima è infinito, ma nella realtà è un numero finito che la maggior parte della volte tende a decrescere nel tempo, seguendo una distribuzione poissoniana appunto.