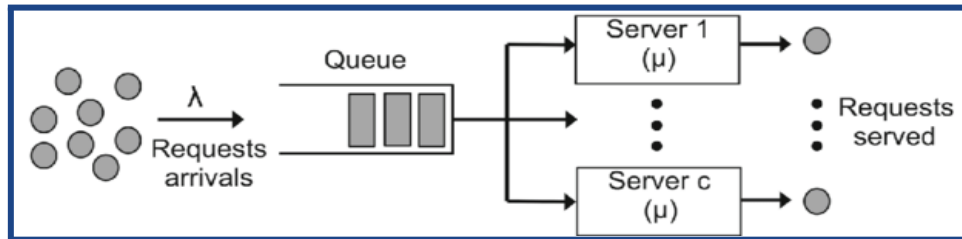


# Simulatore di un Sistema a Coda di Tipo M/M/C

## Scopo del Progetto:

Creare un simulatore ad eventi di un sistema a coda di tipo M/M/C.

## Schema Logico:



## Strumenti Utilizzati:

Il progetto è stato sviluppato tramite due software differenti:

**Visual Studio Code:** Utilizzato per la programmazione nel linguaggio **Java**.



Tramite Java è stata programmata tutta la parte algoritmica e di interfaccia utente per l'inserimento dei valori e per l'effettiva simulazione.

**MatLab:** Utilizzato per la programmazione nell'omonimo linguaggio.



Sfruttato per la veloce capacità di lettura ed elaborazione di dati da file: preleva i dati da file di testo generati dal programma Java, li elabora e genera i grafici.

## Metodo:

Per iniziare, si è diviso il problema in diversi blocchi:

1. Come implementare le componenti principali del sistema:
  - **Generatore:** simula l'arrivo dei pacchetti dalla rete con un tempo di interarrivo markoviano dato in input il parametro  $\lambda$  (tasso degli arrivi, pacchetti al secondo).
  - **Coda:** raccoglie tutti i pacchetti arrivati in attesa di essere serviti, trattandosi di un sistema M/M/C è prevista una coda infinita.
  - **Servitori:** simulano le entità ( $c$  servitori) che svolgono il servizio per i pacchetti con un tempo di servizio markoviano dato in input il parametro  $\mu$  (tasso delle partenze, pacchetti al secondo).
2. Quali dati sono necessari al fine della simulazione.
3. Come raccogliere i dati:
  - **Tempi medi di permanenza dei pacchetti:**
    - Tempo in Coda:  $W_Q$
    - Tempo nel sistema:  $W_S$
  - **Numero medio di pacchetti:**
    - In Coda:  $L_Q$
    - Nel Sistema:  $L_S$
  - **Probabilità di stato:**  $P_K$ , ossia, la probabilità di avere  $K$  pacchetti nel sistema.
4. Come interpretare i dati.
5. Come interfacciare il tutto per l'utilizzo da parte di un utente.

## Implementazione:

1. Per implementare le varie componenti del sistema sono state utilizzate delle classi Thread (Generatore, Coda 2, Servitore). Si è poi deciso di utilizzare due code ai fini implementativi.
  - **Generatore:** genera un tempo casuale:  $-\frac{1}{\lambda} \cdot \ln(1 - u(0,1))$ , attende per il tempo generato e infine, inserisce il pacchetto all'interno della coda.
  - **Coda 1:** l'effettiva coda è un Oggetto ConcurrentLinkedQueue dove vengono inseriti e prelevati i pacchetti.
  - **Coda 2:** la seconda coda chiama il primo servitore libero, per elaborare il primo pacchetto della Coda 1.
  - **Servitore:** preleva il primo pacchetto della Coda 1, genera un tempo casuale:  $-\frac{1}{\mu} \cdot \ln(1 - u(0,1))$  attende per il tempo generato e infine, torna "libero".

2. Sono stati ritenuti necessari i seguenti dati:
  - I parametri di  $\lambda$  e  $\mu$ , sopracitati.
  - Tempo e velocità della simulazione.
  - L'estrazione dei tempi di ogni pacchetto per il calcolo dei tempi medi di permanenza.
  - L'estrazione del numero di pacchetti in un dato istante, per il calcolo del numero medio di pacchetti e per le probabilità di stato.
3. La raccolta dei dati è stata affrontata simultaneamente all'esecuzione:
  - Per i tempi di permanenza: ogni pacchetto memorizza per sé i suoi tempi di permanenza in coda e nel server, il tempo di permanenza del sistema è calcolato come la somma dei due.
  - Per il numero di pacchetti: un thread esterno ogni secondo (in realtà va in base alla velocità di simulazione selezionata, ad esempio in x2 il campionamento sarà ogni 0.5 secondi e così via) campiona la dimensione della coda e del sistema (sommando alla dimensione della coda il numero di server pieni) per poi memorizzarli in due vettori.
4. Dai dati campionati durante la simulazione vengono calcolate:
  - Le diverse medie per trovare  $W_q$ ,  $W_s$ ,  $L_q$ ,  $L_s$ .
  - Le probabilità di stato, calcolate contando quante volte è stato campionato uno specifico stato e dividendolo per il numero di campionamenti fatti.
5. Tramite l'utilizzo dei JFrame, è stata implementata un'interfaccia nella quale inserire i valori di  $\lambda$ ,  $\mu$  e  $c$ , la durata della simulazione e la velocità della stessa. Successivamente si sceglie se eseguire una simulazione normale, oppure con animazione **FIFO** (consentita solamente con velocità: x1,x2). Al termine della simulazione un'interfaccia permette di visualizzare a schermo i dati e i relativi risultati.

### Raccolta Dati:

Per raccogliere i vari dati sono stati utilizzati tre diversi file di testo:

1. "**dati.txt**": contenente i tempi medi di permanenza dei pacchetti:  
Ogni entry contiene:  $T_{Coda}$   $T_{Server}$   $T_{Sistema}$
2. "**parametri.txt**": contenente i parametri  $\lambda$ ,  $\mu$ ,  $c$ ,  $L_q$  ed  $L_s$ .
3. "**probabilità.txt**": contiene le probabilità di essere negli **stati K**.

Successivamente, questi sono presi in analisi da parte di MatLab che fornisce i rispettivi grafici.

Circa il file "dati.txt" si utilizza la struttura readable, ottimale per la lettura di dati in colonne su file di tipo .txt, successivamente si creano rispettivamente gli array per le tre colonne.

Anche per il file "probabilità.txt" si utilizza la medesima struttura readable per poi creare un vettore contenente tutte le probabilità.

Inoltre, per facilitare il salvataggio dei dati, viene creato un file .zip che contiene i 3 file sopra-citati.

### Analisi Dati:

I grafici che seguono raffigurano nello specifico un sistema a coda M/M/4.

## Tempi medi di permanenza empirici rispetto ai rispettivi teorici

Le curve  $\mu c W_s$  e  $\mu c W_q$  rappresentano l'andamento teorico degli omonimi valori. I punti presenti sulle curve indicano i valori empirici trovati tramite 5 diverse simulazioni al variare di  $\lambda$  e  $\mu$ .

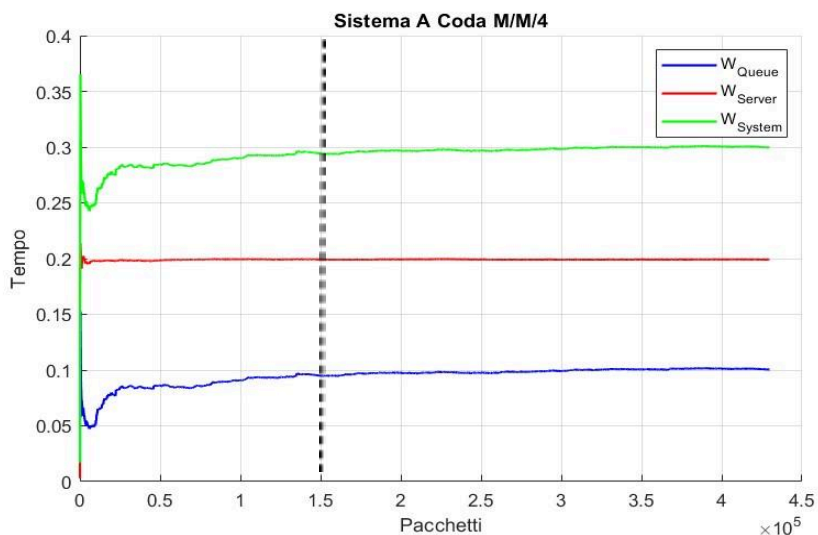
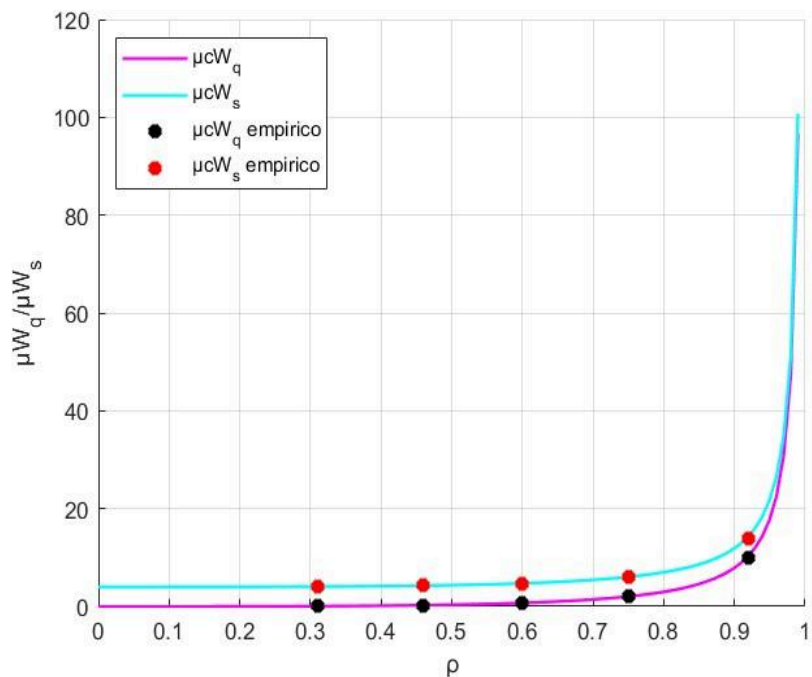
Per trovare le curve teoriche è necessario normalizzare le seguenti formule:

$$W_q = \frac{1}{c \cdot \mu} \cdot \frac{C(c, \rho)}{1 - \rho}$$

$$W_s = \frac{1}{c \cdot \mu} \cdot \left( c + \frac{C(c, \rho)}{1 - \rho} \right)$$

## Tempi medi di permanenza empirici rispetto al numero di pacchetti

In questo caso particolare (con una simulazione con  $\lambda = 15$ ,  $\mu = 5$  di 8 ore), si può notare che dopo circa 15000 **pkt**, i diversi valori empirici tendono tutti ai loro rispettivi valori teorici.



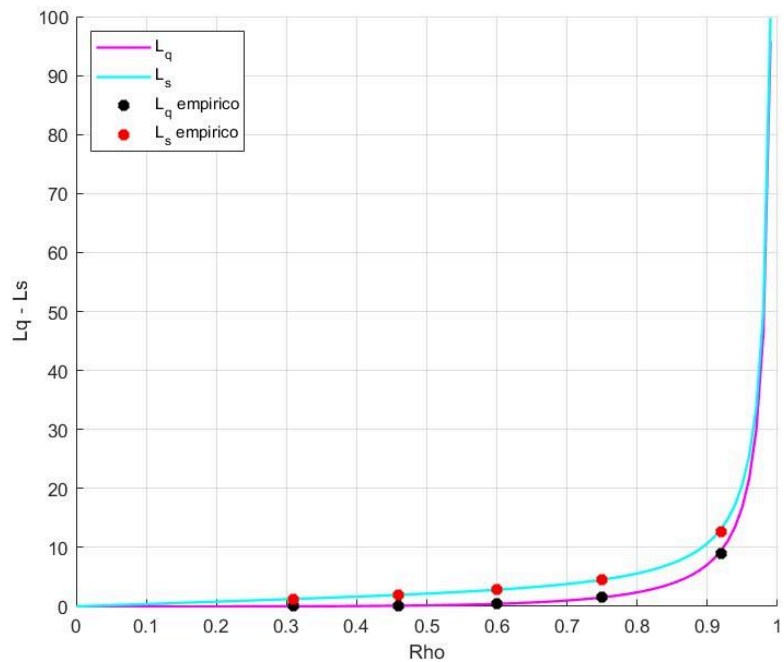
## Lunghezza teorica media della coda e del sistema

Le curve  $L_s$  e  $L_q$  rappresentano l'andamento teorico degli omonimi valori. I punti presenti sulle curve indicano i valori empirici trovati tramite 5 diverse simulazioni al variare di  $\lambda$  e  $\mu$ .

Per trovare le curve teoriche è necessario utilizzare le seguenti formule:

$$Lq = C(c, c\rho) \cdot \frac{\rho}{1-\rho}$$

$$Ls = Lq + c \cdot \rho$$



### Confronto dati di un sistema a Coda M/M/7, con $\lambda = 4.9$ e $\mu = 2.3$ Tempo di Simulazione: 8 ore

Valori	Valori Teorici Previsti	Valori Empirici
$W_q$	<b>0.001</b>	<b>0.0007</b>
$W_s$	<b>0.44</b>	<b>0.4328</b>
$L_q$	<b>0</b>	<b>0.0029</b>
$L_s$	<b>2.13</b>	<b>2.1703</b>
$P_0$	<b>0.1187</b>	<b>0.1141</b>

### Conclusioni:

A seguito della visione e analisi dei dati e dei grafici si conclude che il simulatore si comporta come previsto.