

UNIVERSITÀ DI FERRARA

CORSO DI INFORMATICA

Studio e simulazione di un sistema a coda M/M/c

Autore:

Alessio CELENTANO
Nicola RICCI MACCARINI
Tommaso SOLDANI



**Università
degli Studi
di Ferrara**

1 Obiettivo

L'obiettivo è quello di creare una simulazione di un sistema a coda M/M/c, caratterizzando il numero di utenti nel sistema (utenti in servizio e utenti in coda) nel tempo e gli elementi caratteristici del sistema a coda:

- Probabilità di 0 utenti nel sistema (P_0)
- Probabilità di k utenti nel sistema (P_k)
- Probabilità di coda ($\mathbb{P}\{\text{queue}\}$)
- Numeri medio di utenti nel sistema (L_s)
- Numero medio di utenti in coda (L_q)
- Tempo medio di attesa nel sistema (W_s)
- Tempo medio di attesa in coda (W_q)

2 Presentazione del programma

Per creare il simulatore ad eventi abbiamo utilizzato il linguaggio di programmazione Python con l'utilizzo delle seguenti librerie:

- **Simpy**: Libreria utilizzata per il funzionamento della simulazione ad eventi
- **Plotly**: Libreria utilizzata per la creazione dei grafici
- **Tkinter**: Libreria utilizzata per la creazione della GUI con la quale si inseriscono i dati in input
- Altre librerie per funzioni minori:
 - **re** (RegEx): controllo degli input numerici
 - **statistics**: calcolo della media statistica di un insieme di valori
 - **random**: generazione pseudocasuale dei valori dei tempi di interarrivo e di servizio
 - **math**: calcolo del fattoriale

Dunque, per poter eseguire il programma è necessario installare queste librerie (elencate in un file `requirements.txt`) con il comando:

```
pip install -r requirements.txt
```

Il programma è diviso in due file:

- **main.py**: file principale che crea l'istanza della classe `MMcSystem` e lancia le funzioni che creano i grafici
- **MMcSystem**: classe che si occupa di avviare la simulazione, alla fine della quale renderà disponibile al chiamante i dati necessari per la creazione dei grafici. Inoltre, calcola i principali elementi caratteristici teorici per poterli comparare a quelli della simulazione

Spostandoci nella cartella sorgente ed eseguendo da terminale il seguente comando:

```
python main.py
```

Viene avviato il programma mostrando dapprima l'interfaccia di input, dove viene richiesta la compilazione dei seguenti campi al fine di simulare il sistema a coda e creare i relativi grafici:

- Numero di servitori (c)
- Ritmo delle nascite (λ)
- Ritmo delle morti (μ)
- Numero di utenti da generare¹

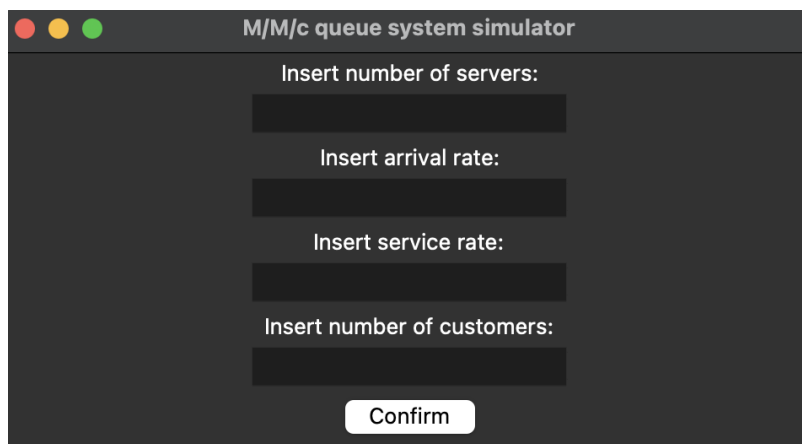


FIGURE 1: Interfaccia grafica per gli input

Devono essere create le condizioni affinché la coda non cresca all'infinito in modo esponenziale:

$$\rho = \frac{\lambda}{c\mu} < 1 \iff \lambda < c\mu \quad (1)$$

In caso contrario, il programma mostrerà un errore ed inviterà l'utente a modificare i dati. Cliccando su "Confirm" vengono generati e mostrati 4 grafici.

3 Grafici e osservazioni

3.1 Grafico della simulazione

Viene mostrato l'andamento del numero di utenti nel sistema (in blu gli utenti in servizio, in rosso gli utenti in coda). Sulla barra laterale abbiamo tre sezioni:

- I parametri caratteristici dati in input
- Gli elementi caratteristici teorici calcolati dalle formule
- Gli elementi caratteristici estrapolati dalla simulazione

¹Si suppone un sistema inizialmente vuoto.

Una prima osservazione che possiamo fare é che il numero degli utenti nel sistema varia continuamente dando al grafico un andamento oscillatorio. Ciò perché μ_k stesso varia continuamente:

$$\mu_k = \min\{k\mu, c\mu\}$$

Dunque il ritmo delle morti é proporzionale al numero degli utenti in servizio. Nei momenti in cui non tutti i servitori sono pieni, il ritmo delle nascite può essere superiore a quello delle morti mandando numerosi utenti in coda, che vengono successivamente smaltiti dato che con tutti i servitori pieni otteniamo il massimo ritmo delle morti $c\mu$, che, per quanto detto in fase di input, é certamente superiore al ritmo delle nascite.

Nel grafico 2 abbiamo dato in input una λ molto vicina $c\mu$ per visualizzare meglio ciò.

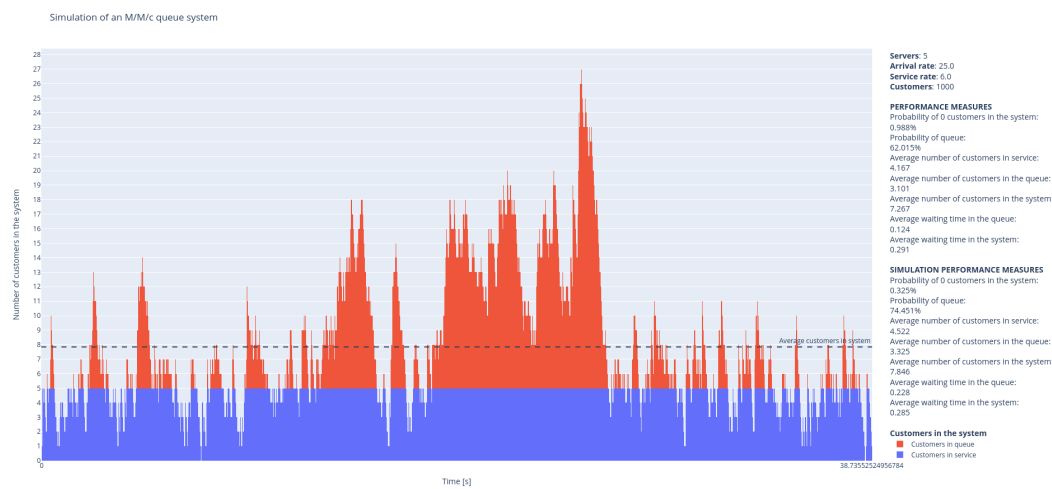


FIGURE 2: Numero degli utenti nel sistema nel tempo

3.2 Probabilità di avere k utenti nel sistema

Viene mostrato l'andamento della P_k per questo sistema a coda. Notiamo che, generalmente, abbiamo probabilità alte di avere k utenti laddove k é minore del numero dei servitori e diminuisce progressivamente all'aumentare di k .

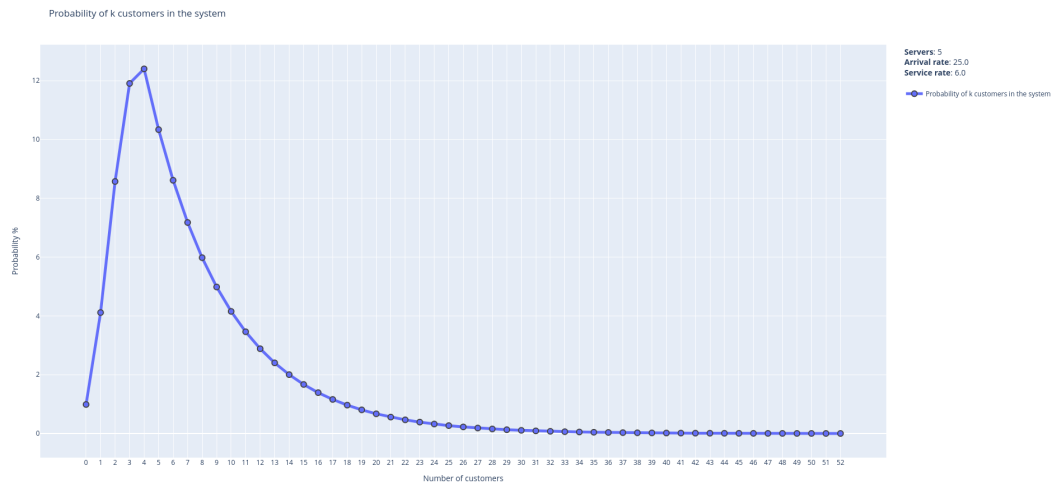


FIGURE 3: Andamento della P_k

3.3 Altri elementi caratteristici

Vengono mostrati gli andamenti degli altri elementi caratteristici al variare del numero di servitori. Il primo valore sull'asse x è il più piccolo numero di servitori affinché venga rispettata l'equazione (1). Come ci si potrebbe aspettare, all'aumentare del numero dei servitori si ottiene un miglioramento di ogni elemento caratteristico:

- Aumenta la probabilità di avere zero utenti nel sistema
- Tutti gli altri elementi caratteristici diminuiscono

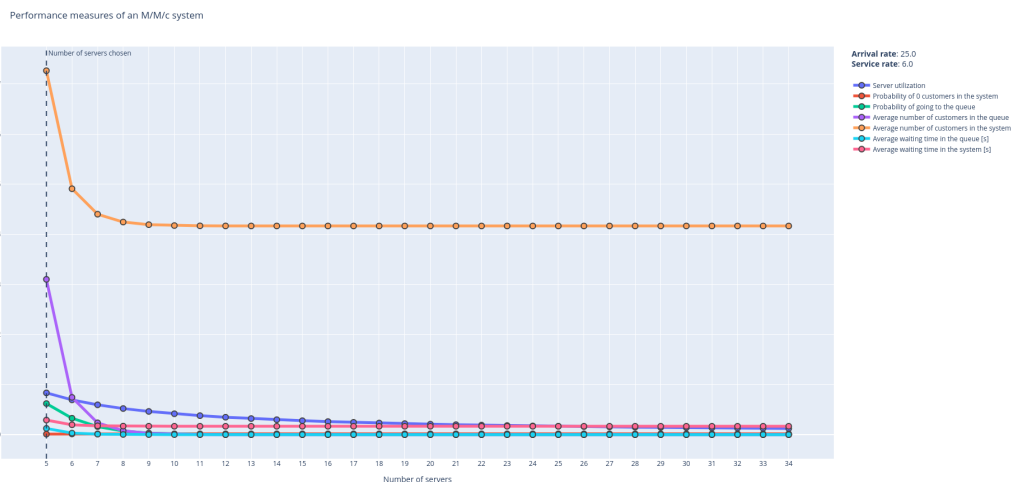


FIGURE 4: Curve degli elementi caratteristici

3.4 Tempi media d'attesa con ρ costante

Riprendiamo ciò che abbiamo notato nel grafico della simulazione: esiste un modo per ottimizzare un sistema a coda M/M/c mantenendo invariato il tasso di utilizzo ρ (cioè variando il numero dei servitori c e il ritmo delle morti μ , ma non il loro prodotto)?

Diminuendo il numero dei servitori, la probabilità di andare in coda cresce e così la probabilità di avere il massimo ritmo delle morti $c\mu$. Ad esempio, nel caso estremo di un sistema con un solo servitore, si ha sempre il massimo ritmo delle morti $c\mu$, a meno che non ci siano utenti nel sistema. Ciò comporta il miglioramento di tutti gli elementi caratteristici, in particolare del tempo medio di attesa nel sistema. A conferma di questo, il grafico generato dal programma, che mostra gli elementi caratteristici al variare di c e μ (ma con ρ costante) mostra andamenti coerenti con quanto detto.

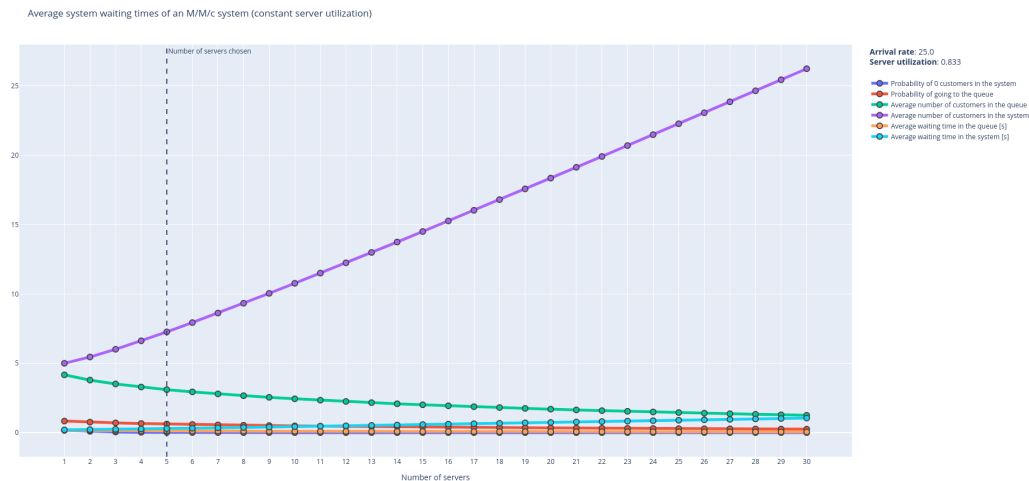


FIGURE 5: Curve degli elementi caratteristici al variare di c e μ con ρ costante

4 Considerazioni finali

Oltre a ciò che riguarda l'andamento degli elementi caratteristici al variare di vari parametri, la principale considerazione che traiamo dallo studio di una simulazione di un sistema a coda M/M/c é:

É più efficiente un sistema a coda M/M/c con pochi servitori veloci che uno con tanti servitori lenti.