

Progetto reti

sistema a coda M/M/c

2 novembre 2023



UNIVERSITÀ
DEGLI STUDI
DI FERRARA
- EX LABORE FRUCTUS -

Simone Acuti - 183843 - *simone.acuti@edu.unife.it*

Leonardo Lodi - 183845 - *leonardo03.lodi@edu.unife.it*

Marius Ceban - 187612 - *marius.cebann@edu.unife.it*

Indice

1	Introduzione	3
1.1	Divisione del lavoro	3
2	realizzazione del simulatore	3
2.1	Linguaggi utilizzati per l'implementazione:	3
3	Trattazione dell'argomento	3
4	Calcoli e cenni teorici	4
5	Conclusioni	6

1 Introduzione

Questa relazione è stata scritta per completare l'insieme di elementi che costituiscono il progetto di reti di telecomunicazioni.

L'obiettivo formativo legato allo svolgimento di questo compito è quello di comprendere il comportamento del sistema a coda $M/M/c$.

La lettera c in questo sistema descrive il numero di servitori. Il tasso di servizio viene rappresentato da μ e descrive le utenze per ogni secondo [pkt/s]. L'ultimo dato noto viene descritto dalla lettera λ e rappresenta il tasso di arrivi per ogni secondo [pkt/s].

1.1 Divisione del lavoro

Il lavoro è stato suddiviso equamente fra tutti i partecipanti, in particolare una persona si è occupata del codice, un'altra della relazione e infine l'ultimo non per importanza si è occupato della realizzazione grafica.

2 realizzazione del simulatore

2.1 Linguaggi utilizzati per l'implementazione:

- Django: un framework di Python per costruire pagine web.
- Javascript.
- HTML e CSS.

3 Trattazione dell'argomento

4 Calcoli e cenni teorici

Vediamo un caso generale relativo alla risoluzione di un sistema a coda M/M/c. Come è stato definito in precedenza il numero dei servitori è c , questo implica che il tasso di servizio può arrivare fino a $c\mu$.

$$\begin{aligned}\lambda k &= \lambda \\ \mu_k &= \min\{k\mu, c\mu\} \\ \rho &= \frac{\lambda}{\mu_{tot}}\end{aligned}$$

Da cui:

$$P_k = \begin{cases} P_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu}; & k \leq c \\ P_0 \prod_{i=0}^{c-1} \frac{\lambda}{(i+1)\mu} \prod_{j=c}^{k-1} \frac{\lambda}{c\mu}; & \mu > c \end{cases} \quad P_k = \begin{cases} P_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}; & k \leq c \\ P_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{c!c^{k-c}}; & k > c \end{cases}$$

$E\{\lambda\} E\{T_x\} = \frac{\lambda}{\mu}$, con massimo tasso di smaltimento c , quindi:

$$\rho = \frac{\lambda}{c\mu} \text{ dove } c\mu = \mu_{tot}$$

$$P_k = \begin{cases} P_0 \frac{(c\rho)^k}{k!}; & k \leq c \\ P_0 \frac{(c\rho)^k}{c! c^{k-c}}; & k > c \end{cases} = \begin{cases} P_0 \frac{(c\rho)^k}{k!}; & k \leq c \\ P_0 \frac{\rho^k c^c}{c!}; & k > c \end{cases}$$

$$P_0 = \left[\sum_{k=0}^{c-1} \left(\frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right) \right]^{-1} \quad (P_0 + \sum_{k=1}^{\infty} P_k) = 1$$

Un pkt/ut va in coda se ce ne sono già c nel sistema che stanno occupando tutti i server.

$$P\{\text{queue}\} = \sum_{k=c}^{\infty} P_k = P_0 \frac{(c\rho)^c}{c! (1-\rho)} = \frac{\frac{(c\rho)^c}{c!} \frac{1}{1-\rho}}{\sum_{k=0}^{c-1} \left(\frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right)}$$

$$\text{Formula di C-Erlang} \rightarrow C(c, A) = \frac{\frac{A^c}{c!} \frac{1}{1-\frac{A}{c}}}{\sum_{k=0}^{c-1} \left(\frac{A^k}{k!} + \frac{A^c}{c!} \frac{1}{1-\frac{A}{c}} \right)}$$

$P\{\text{queue}\} = C(c, c\rho) [c\rho = \frac{1}{\mu}]$ dove c è il numero di server;

$$A = c\rho; \quad C(c, A) = 0 \text{ se } \rho = 0.$$

$$L_q = E\{q\} = \sum_{k=c}^{\infty} (k-c) P_k = C(c, c\rho) \frac{\rho}{1-\rho}$$

L_q rappresenta il numero medio di pkt/ut in coda.

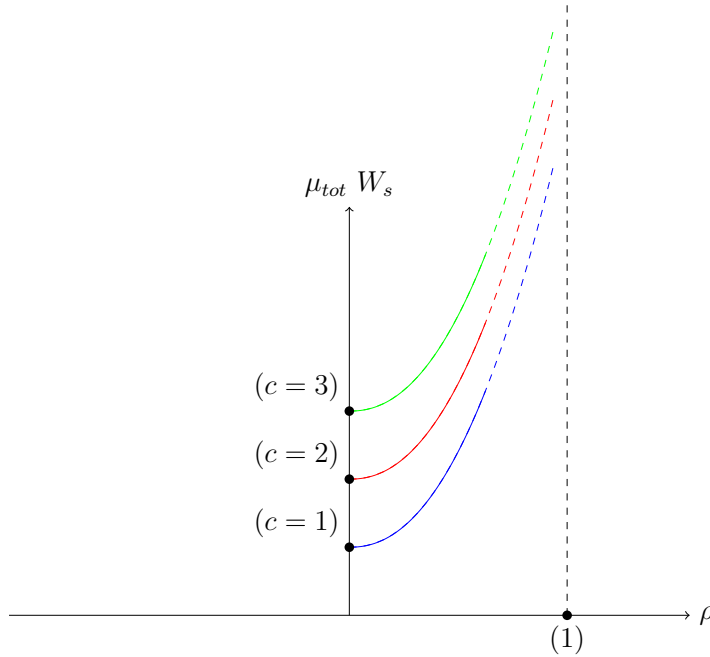
$(k-c)$ rappresenta il numero di ut. in coda nello stato k .

$$L_x = E\{x\} = c\rho \rightarrow \text{numero medio di pky/ut. in servizio } (x = k - q).$$

$$L_s = L_q + L_x = \left[\frac{C(c, c\rho)}{1-\rho} + c \right] \rho \rightarrow \text{numero medio di pkt/ut. nel sistema.}$$

$$W_q = \frac{L_q}{E\{\lambda\}} = \frac{C(c, c\rho)}{\mu_{tot}(1-\rho)}$$

$$W_s = \frac{L_s}{E\{\lambda\}} = \frac{C(c, c\rho) + c(1-\rho)}{\mu_{tot}(1-\rho)}$$



Osservazione: conviene un sistema con pochi server veloci rispetto ad uno con numerosi server lenti.

5 Conclusioni