

# Esame Software Engineering (AA 2024/25)

08 Settembre 2025 Lab. Colossus - Via salaria 113

*Enrico Tronci*

*Computer Science Department, Sapienza University of Rome  
Via Salaria 113 - 00198 Roma - Italy*

tronci@di.uniroma1.it

<https://raise.uniroma1.it>

## Esercizio 3 (20 punti)

Tutti i sistemi hanno, per ciascun canale di input, una coda FIFO infinita. Ciascuna coda può essere realizzata con una FIFO molto grande, con una resize in caso di overflow ovvero, più semplicemente, usando il tipo `queue` del C++.

Un sistema di *e-commerce* ha  $C$  customers. Ciascun *customers* può essere modellato come una catena di Markov con un solo stato (0). Ciascun customer resta nello stato 0 per un tempo  $\tau$  scelto uniformemente a random nell'intervallo  $[A, B]$ .

Se il customer decide di fare una transizione (che lo riporta nello stato 0) allora:

1. sceglie un server  $s$  a random tra  $S$  server (numerati  $1, 2, \dots S$ );
2. invia al server scelto  $s$  un valore  $i > 0$ .

Questo vuol dire che il customer ha richiesto il prodotto  $i$  al server  $s$ .

Si noti che quando il customer permane nello stato 0 nulla viene inviato ai servers.

Ci sono  $P$  prodotti, quindi l'output del customer è  $i \in \{1 \dots P\}$ .

I servers sono tutti uguali. Ciascun server ha un DB che, per ogni prodotto, contiene il numero di items disponibili.

Il sistema ha  $F$  fornitori. Ciascun *fornitore* può essere modellato come una catena di Markov con un solo stato (0). Ciascun fornitore resta nello stato 0 per un tempo  $\gamma$  scelto uniformemente a random nell'intervallo  $[V, Q]$ .

Se il fornitore decide di fare una transizione (che lo riporta nello stato 0) allora:

1. sceglie un server  $s$  a random tra  $S$  server (numerati  $1, 2, \dots S$ );
2. invia al server scelto  $s$  un valore  $i > 0$ .

Quando un server riceve una richiesta dal customer per il prodotto  $j$  dalla FIFO di input si comporta come segue. Se nel DB ci sono items per il prodotto  $j$  richiesto il server aggiorna il DB e invia al customer il messaggio  $j$ . Se nel DB non ci sono items disponibili per il prodotto  $j$ , allora manda al customer il messaggio  $-j$ , che denota *mancata vendita* per il prodotto  $j$ .

Quando un server riceve il messaggio  $j$  dal fornitore dalla FIFO di input incrementa di 1 il numero di items disponibili per il prodotto  $j$ .

Per ognuno dei  $P$  prodotti, il DB di ciascun server è inizializzato a random con valori compresi tra 0 e  $K$ .

## 1 Obiettivo

Il numero di mancate vendite al tempo  $t$  è il numero totale di mancate vendite entro il tempo  $t$ , cioè il numero di messaggi del tipo  $-j$  ricevuti dal customer entro il tempo  $t$ .

Il rate di mancate vendite al tempo  $t$  è il numero di mancate vendite al tempo  $t$  diviso  $t$ .

Il rate di mancate vendite è rate di mancate vendite alla fine della simulazione.

Si vuole riportare in uscita il valore atteso del rate di mancate vendite. Cioè il valore medio del rate di mancate vendite usando  $M$  simulazioni Montecarlo.

## 2 Formato dei parametri di input

I parametri della simulazione sono forniti nel file `parameters.txt`. Le righe del file `parameters.txt` sono formattate come segue.

- T <valore reale positivo>  
definisce il valore del parametro  $T$ .
- H <valore real positivo>  
definisce il valore del parametro  $H$ , orizzonte di simulazione (in secondi).
- M <valore intero positivo>  
definisce il valore del parametro  $M$ .
- C <numero intero positivo>  
definisce il valore del parametro  $C$ .
- A <valore reale positivo>  
definisce il del parametro  $A$ .
- B <valore reale positivo>  
definisce il del parametro  $B$ .
- F <numero intero positivo>  
definisce il valore del parametro  $G$ .
- V <valore reale positivo>  
definisce il valore del parametro  $V$ .

- `W <valore reale positivo>`  
definisce il valore del parametro  $W$ .
- `P <valore intero positivo>`  
definisce il numero di prodotti.
- `S <valore intero positivo>`  
definisce il valore del parametro  $S$ .

Un esempio di file `parameters.txt` è:

```
T 0.5
H 234.6
M 100
C 10
A 1.0
B 2.0
F 3
V 3.0
Q 5.0
P 10
S 4
```

### 3 Formato di output

L'output dell'esercizio è memorizzato nel file `results.txt` la cui prima riga è formattata come indicato nelle istruzioni generali.

Le rimanenti righe del file `results.txt` hanno il formato

$R$  < rate mancate vendite >

Un esempio di file `results.txt` per il file `parameters.txt` dato sopra è:

```
2025-01-09-Mario-Rossi-1234567
R 12.3
```