

# Esame Software Engineering (AA 2024/25)

12 Giugno 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 4 (25 punti)

Si consideri di nuovo il problema nell'esercizio 3.

Si vuole costruire il sito di *e-commerce* per servire i customers modellia come nell'esercizio 3.

Tutti i componenti hanno una coda FIFO infinita per ciascuno dei canali di input. Queste code possono essere realizzate con una FIFO molto grande, con una resize in caso di overflow ovvero, più semplicemente, usando il tipo `queue` del C++.

Il sito ha un dispatcher che riceve le richieste dal customer e le manda ad uno degli  $S$  server selezionati uniformemente a random. Solo le richieste di prodotto (output del customer maggiore di 0) vengono inviate ai servers. Per produrre il suo output il dispatcher impiega un tempo  $D$ .

Ogni server ha un DB local che, per ogni prodotto, contiene il numero di items disponibili. Quando il server  $i$  prende in carico una richiesta per il prodotto  $j$  dalla FIFO di input si comporta come segue. Se nel DB ci sono item per il prodotto  $j$  richiesto il server aggiorna il DB e manda al dispatcher il messaggio  $(i, j, 1)$ . Se nel DB non ci sono items disponibili per il prodotto  $j$  manda al dispatcher il messaggio  $(i, j, 0)$  che denota *mancata vendita*. Per produrre il suo output il server impiega un tempo  $F$ .

Per ognugno degli  $N$  prodotti, i DB dei servers sono inizializzati a random con valori compresi tra 1 e  $K$ .

Quando il dispatcher riceve un messaggio del tipo  $(i, j, 0)$  si comporta come segue:

- con probabilità  $p$  richiede di rifornire il server  $i$  con 10 items del prodotto  $j$
- con probabilità  $1 - p$  sceglie un server  $i'$  a random ed un item  $j'$  a random e rifornisce il server  $i'$  con 10 items del prodotto  $j'$ .

## 1 Obiettivo

L'obiettivo di questo esercizio è stimare la probabilità  $q$  di *mancata vendita*, cioè il numero  $\alpha$  di volte che il dispacher riceve un messaggio del tipo  $(i, j, 0)$  diviso il numero totale  $\beta$  di richieste che il dispatcher invia ai server. Cioè:

$$q = \frac{\alpha}{\beta}$$

Per ogni simulazione, si considera solo il valore di  $q$  alla fine della simulazione. Infine il valore di  $q$  viene stimato con metodo Montecarlo prendendo il valore medio ottenuto da  $M$  simulazioni.

## 2 Formato dei parametri

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

- `M <numero di simulazioni>`  
definisce il valore del parametro  $M$ .
- `H <orizzonte di simulazione>`  
definisce l'orizzonte di simulazione (in secondi).
- `N <numero prodotti>`  
definisce il numero di veicoli dispiegati.
- `A <valore minimo tempo attesa customer>`  
definisce il valore minimo per il tempo  $\tau$  del customer.
- `B <valore massimo tempo attesa custgomer>`  
definisce il valore massimo per il tempo  $\tau$  del customer.
- `T <valore in secondi del time step>`  
definisce il valore del parametro  $T$ .
- `S <numero dei servers>`  
definisce il valore del parametro  $S$ .
- `K <numero items>`  
definisce il valore del parametro  $K$ .
- `p <probabilità>`  
definisce il valore del parametro  $p$ .
- `D <tempo calcolo dispatcher>`  
definisce il valore del parametro  $D$ .
- `F <tempo calcolo server>`  
definisce il valore del parametro  $F$ .

Un esempio di file `parameters.txt` è:

```
M 1000
H 100
N 3
A 1.0
B 2.0
T 0.5
S 3
K 20
p 0.5
D 1
F 5
```

Il file sopra indica che: il numero di simulazionei da effettuare è  $M = 1000$ , l'orizzonte  $H$  di simulazione è 100 secondi, ci sono  $N = 3$  prodotti, il tempo minimo tra un richiesta e l'altra dal customer è  $A = 1.0$  secondi, il tempo massimo tra un richiesta e l'altra dal customer è  $B = 2.0$  secondi, il time step è  $T = 0.5$  secondi, il numero di server è 3,  $K = 20$ ,  $p = 0.5$ , il tempo di calcolo per il dispatcher è  $D = 1$  secondo e il tempo di calcolo per il server è  $F = 5$  secondi.

### 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:

```
Q <valore di q>
```

Un esempio di file `results.txt` è:

```
2025-01-09-Mario-Rossi-1234567
Q 0.7
```