

# Esame Software Engineering (AA 2024/25)

05 Febbraio 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)

L'unità di tempo per il presente esercizio è il secondo e  $T = 1$  (secondi) è il time step per tutti i sistemi nel presente esercizio.

Un sito di *e-commerce* deve fornire il servizio ad  $n$  customers: 1, 2, ...  $n$ . Questo viene fatto usando  $n$  servers. Il customer  $i$  è connesso al server  $i$  ( $i = 1, \dots, n$ ).

Tutti i customer sono modellati con la stessa Markov chain con un solo stato: 0. Quando è nello stato 0, con probabilità  $p_0$  il customer resta nello stato 0 e non invia nessuna richiesta al server mentre con probabilità  $p_i$  ( $i = 1, \dots, k$ ) invia al server la richiesta del prodotto  $i$ . Ovviamente  $\sum_{i=0}^k p_i = 1$ .

Il customer invia messaggi al server attraverso una FIFO di dimensione 1000. Il server invia direttamente le sue risposte al customer (cioè senza FIFO).

Il sistema ha un database (DB) che memorizza il numero di items disponibili per ogni prodotto. Ovviamente il DB può essere rappresentato con un array di  $k$  elementi.

Al DB è connesso un fornitore. Il fornitore ha un solo stato: 0. Quando è nello stato 0, con probabilità  $f_0$  il fornitore resta nello stato 0 e non invia nessuna richiesta di aggiornamento al DB mentre con probabilità  $f_i$  ( $i = 1, \dots, k$ ) invia al DB la richiesta di incrementare di 1 il numero di items presenti per il prodotto  $i$ . Ovviamente  $\sum_{i=0}^k f_i = 1$ . Il fornitore invia messaggi al DB attraverso una FIFO.

Per migliorare l'efficienza del sistema ogni server ha una cache dove memorizza il numero di items disponibili per ciascun prodotto. Il server ha anche un DB (array) locale dove memorizza il numero di items venduti per il prodotto  $i$ .

Inizialmente le cache di tutti i server sono inizializzate con i valori dal DB ed il DB degli items venduti è inizializzato a 0. Quando il server riceve una richiesta dal customer per il prodotto  $i$  risponde con 1 se per il prodotto  $i$  la differenza tra il numero di items disponibili ed items venduti è maggiore di 0, risponde con 0 altrimenti. Inoltre, in questo caso, decrementando di 1 il numero di items disponibili per il prodotto  $i$ . Quando la differenza tra il numero di items disponibili ed items venduti è minore di 0 risponde con 0 al customer.

Per efficienza il server non aggiorna sempre il DB a seguito di una vendita. Il server aggiorna il DB con probabilità  $\alpha$ . L'aggiornamento del DB, quando

viene seguito, richiede al server  $10T$  (cioè 10 secondi). Durante questo tempo il server non risponde al customer e le richieste si accumulano nella coda di input.

Il server invia messaggi al DB attraverso un FIFO.

Quando il server aggiorna il DB decrementa la disponibilità degli items nel DB in accordo alle vendite registrate nel suo DB locale. Se un elemento del DB diventa negativo a seguito dell'aggiornamento vuol dire che viene saturato a 0 e si registra che si è avuto un *overselling* dovuto al disallineamento tra il DB locale del server e la disponibilità effettiva dell'item.

## 1 Formato dei parametri

Il file `parameters.txt` dei parametri fornisce i parametri del sistema. Il file `parameters.txt` è formattato come segue.

```
H
n
k
α
p0 p1 ... pk
f0 f1 ... fk
```

Dove:  $H$  è l'orizzonte di simulazione,  $p_i$  ( $i = 1, \dots, k$ ) sono le probabilità relative customer ed  $f_i$  ( $i = 1, \dots, k$ ) sono le probabilità relative al fornitore.

Un esempio di file `parameters.txt` è:

```
1000
3
2
0.5
0.8 0.1 0.1
0.6 0.2 0.2
```

## 2 Obiettivo

L'obiettivo è stimare il numero di *oversellings* facendo 1000 simulazioni Monte-carlo.

## 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:

$S$  numero di oversellings  $R \frac{S}{n \cdot H}$   
dove:  $S$  è il numero di oversellings ed  $R$  è il rapporto tra  $S$  ed il prodotto tra  $n$  (numero di servers) e l'orizzonte di simulazione.

Un esempio di file `results.txt` è:

```
2025-01-09-Mario-Rossi-1234567  
S 1230  
R 0.6
```