

Esame Software Engineering (AA 2022/23)

24 Marzo 2023

Enrico Tronci

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

tronci@di.uniroma1.it

<http://mclab.di.uniroma1.it>

Esercizio 2 (20 punti)

L'unità di tempo per questo esercizio è il secondo.

Si vuole realizzare un sistema di prenotazione ristoranti.

Si assume che, in media, il tempo tra due richieste di prenotazione è di τ secondi.

Una richiesta consiste in una tupla (id, f) dove:

1. id : è l'identificatore del cliente. Si assume che id sia un intero tra 1 e N .
2. f : è l'identificatore del ristorante. Si assume che f sia un intero tra 1 e F .

Ogni ristorante ha al più K posti a disposizione, cioè ci possono essere al più K clienti nello stesso ristorante.

C'è un database DB centralizzato che contiene tutte le prenotazioni, cioè coppie (id, f) . Quando arriva una nuova richiesta si usa il DB per verificare se c'è un posto libero nel ristorante richiesto. In caso affermativo la prenotazione viene aggiunta al DB ed una conferma è mandato all'utente. Se non ci sono posti disponibili viene notificato che la prenotazione non è andata a buon fine.

Le richieste di prenotazione arrivano ad un server che usa il DB di cui sopra per rispondere. Il server impiega ρ secondi per rispondere alla richiesta.

Si vuole ridurre il tempo di risposta. Per questo motivo viene introdotto in dispatcher che:

1. quando arriva una richiesta la manda ad uno degli S server a random;
2. ritorna al richiedente la risposta del server

Vengono quindi introdotti Q server collegati al dispatcher per ridurre i tempi di risposta.

Questo esercizio si focalizza sulla modellazione dell'ambiente **Env** del sistema. A tal fine si sviluppano i seguenti blocchi.

1. Blocco **Env** nel file **env.mo** che modella un blocco che genera a richieste di prenotazione a random.

2. Blocco `monitor1` nel file `monitor1.mo` che prende come input le richieste generate dal blocco `Env` e ritorna come output valor medio e deviazione standard del tempo tra un richiesta e l'altra. Se tutto è realizzato correttamente il valor medio del tempo tra una richiesta e l'altra dovrebbe essere circa τ .

Parametri del Modello

I parametri del modello sono i seguenti:

1. `HORIZON`, contenente l'orizzonte di simulazione, cioè il tempo simulato. Potete usare `HORIZON = 10000`.
2. Parametri: $\tau = 10$, $\rho = 2$, $N = 300$, $F = 10$, $K = 20$, $S = 10$, $Q = 5$,

Output della simulazione

Si usi l'istruzione Modelica `terminate` per terminare la simulazione quando la variabile Modelica `time` ha un valore maggiore del parametro `HORIZON`.

Alla terminazione si stampino nel file `outputs.txt` valori medi e deviazioni standard per ogni componente della tripla e del tempo tra richieste, con il seguente formato.

La prima riga (di *intestazione*) del file `outputs.txt` contiene:

Avg StdDev (ID = aaa, MyMagicNumber = bbb, HORIZON = ccc, time = ddd)

dove:

- `aaa` è il valore del parametro ID (numero di matricola),
- `bbb` è il valore del parametro `MyMagicNumber`,
- `ccc` è il valore del parametro Modelica `HORIZON`,
- `ddd` è il valore della variabile Modelica `time` quando la simulazione viene terminata dal comando `terminate`.

Le altre righe hanno il seguente formato:

<Valor medio del tempo tra una richiesta e l'altra> <Valore della deviazione standard del tempo tra una richiesta e l'altra>

Si avranno quindi, a parte la prima riga di intestazione, 1 sola riga con 2 colonne.

Si usi un orizzonte di simulazione molto grande (maggiore di HORIZON). In particolare si verifichi che l'orizzonte di simulazione sia maggiore del valore del `time` quando la simulazione viene terminata dal comando `terminate`. Se questo non è verificato il modello è sbagliato. Questo valore di `time` è visibile su `stdout`.

NOTA

Si vedano le istruzioni ed in particolare la sezione *NOTA BENE* delle istruzioni.