Esame Software Engineering (AA 2021/22)

4 Novembre 2022

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 3 (25 punti)

Questo esercizio si focalizza sulla modellazione della componente *dispatcher* per il sistem descritto nell'esercizio 2. Se sviluppi quindi un modello Modelica per il sistema *dispatcher* di cui sopra. Il modello Modelica includerà i blocchi seguenti.

1. Il blocco `Dispatcher` nel file `dispatcher.mo`. `Dispatcher` ha N inputs e K outputs. Ciascuno degli inputs è connesso ad una FIFO di richieste. Nel nostro caso ciascun input di `Dispatcher` sarà connesso ad una istanza di `DQueue` a sua volta alimentata da una istanza di `Env`.

2. I blocchi dall'esercizio 2 dove però l'output della FIFO `DQueue` sarà connesso ad un input del blocco che modella il dispatcher (invece che a `QueueTerminator` come nell'esercizio 2.

3. Blocco `MQueue` nel file `mqueue.mo` che modella la FIFO delle richieste all'istanza di microservizio. Il blocco `Dispatcher` inserisce le richieste in questa FIFO.

4. Blocco `MQueueTerminator` nel file `mqueueterminator.mo` che modella un processo che ogni unita di tempo estrae un dato (se presente) dalla FIFO `MQueue`. Questo blocco ha la stessa funzione per `MQueue` di quella che `DQueueTerminator` nell'esercizio 2 ha per `DQueue` e può ovviamente essere realizzato in modo analogo.

5. Il block `Monitor2` nel file `monitor2.mo`.

Il blocco `Dispatcher` ha N inputs ognuno connesso ad una FIFO. Ogni input è una tripla come descritto nell'esercizio 2. Ogni 5 unità di tempo il dispatcher esegue le seguenti operazioni.

Per ognuno dei suoi K outputs, seleziona a randonm una FIFO di input q. Se la FIFO q non è vuota il valore letto dalla FIFO di input viene trasferito sulla FIFO di output (cioè una istanza di `MQueue`). L'obiettivo è svuotare in modo uniforme le FIFO di input e bilanciare il carico sulle istanze dei microservizi.

Il block `Monitor2` prende come input i K outputs del dispatcher e produce l'output nel file `outputs.txt` descritto nel seguito.

Parametri del Modello

Il vostro modello, oltre ai parametri dell'esercizio 2, conterrà i seguenti parametri:

1. `N`, numero di code di inputs al blocco dispatcher (potete assumere `N` = 5),
2. `K`, numero di outputs del blocco dispatcher (potete assumere = $K3$),

Output della simulazione

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

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

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

`OutputIndex AvgWait StdWait` (`ID` = `aaa`, `MyMagicNumber` = `bbb`, `HORIZON` = `ccc`, `time` = `ddd`)

dove:

- `aaa` è il valore del parametro `ID`,
- `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:

<Indice i dell'output del dispatcher (i compreso tra 1 e K)> <Valor medio del tempo tra una richiesta e l'altra sull'output i> <Valore della deviazione standard del tempo tra una richiesta e l'altra sull'output i >

Si avranno quindi, a parte la prima riga di intestazione, K righe e 3 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.