Esame Software Engineering (AA 2022/23)

07 Luglio 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 (25 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 server consiste di W processi (*workers*) con identificatore i compreso tra 1 ed W.

Il server elabora *jobs* che riceve dal *dispatcher*. Ogni *job* consiste di N *tasks* numerati da 1 ad N e che vengono eseguiti in sequenza.

I valori N e W sono parametri del modello.

Un task consiste un una coppia $(v,\ k)$ dove, v è l'identificatore del *job* e k ($k = 1, \ldots N$) è l'identificatore del task per il *job* v.

Gli oggetti del presente sistema possono essere modellati con delle *Discrete Time Markov Chain* (DTMC) con un buffer di input gestito con strategia FIFO (*First In First Out*). L'inserzione di un elemento in un buffer pieno provoca un errore ed arresta la simulazione. Il tentativo di rimozione in un buffer vuoto provoca un errore ed arresta la simulazione.

Un *worker* può essere modellato con una DTMC con un buffer di input di dimensione B. Il buffer di input contiene tasks $(v,\ k)$. Il *worker* i può essere nello stato 0 (*idle*) ovvero k (*busy*) con $k = 1, \ldots N$.

Se il *worker* i è idle (stato 0) allora rimuove un task dal buffer. Se non c'è alcun task disponibile rimane idle. Se il task rimosso è $(v,\ k)$ allora il *worker* i passa nello stato k. Il *worker* i resta poi nello stato k con probabilità $p(i, k)$ oppure transisce nello stato 0 (task completed) con probabilità $(1 - p(i, k))$.

Quando il *worker* i transisce dallo stato k allo stato 0 si comporta come segue. Se $k \geq N$ il task $(v,\ k)$ viene inserito nel buffer del *delivery manager* poichè il *job* v è completato. Se $k < N$ il task $(v,\ k+1)$ viene inserito nel buffer di input di un *worker* scelto a random.

Il *dispatcher* che invia *jobs* al server può essere modellato con una DTMC con un buffer di input di dimensione B. Il buffer di input del *dispatcher* contiene jobs. Ogni D time steps il *dispatcher* rimuove un job (se disponibile) dal proprio buffer di input e lo assegna ad un *worker* scelto a random. Nello specifico, quando il *dispatcher* rimuove il *job* v dal buffer, assegna ad un *worker* scelto a random il task $(v,\ 1)$.

Il *delivery manager* può essere modellato con una DTMC con un buffer di input di dimensione B. Il buffer di input contiene *jobs*. Ogni D time steps il *delivery manager* rimuove un *job* dal suo buffer di input.

Dai dati storici si vede che la probabilità $p(i, k)$ che il *worker* i continui a lavorare al task (v, k) è:

$$p(i,k) = \frac{1}{2} e^{-(\frac{i}{W} + \frac{k}{N})}$$

Quindi, il tempo atteso (in secondi) affinché il *worker* i completi il task (v, k) è:

$$\theta(i,k) = \frac{1}{1 - p(i,k)}$$

Si noti che in questo modello semplificato il tempo di processamento del task (v, k) non dipende dal *job* v, ma solo dal task k.

Inizialmente tutti i *workers*, il *dispatcher* ed il *delivery buffer* sono nella stato 0 e con il buffer di input vuoto.

1 Modelli Modelica

Si realizzino i seguenti modelli Modelica per il sistema di cui sopra.

1. Modellare ciascun *workers* con una DTMC come descritto sopra.

2. Modellare il *dispatcher* come una DTMC come descritto sopra.

3. Modellare il *job generator* con una DTMC con due stati: 0 ed 1. Il numero atteso di transizioni per lasciare lo stato 0 (*soujourn time*) è L. Questo vuol dire che la probabilità p di transire da 0 ad 1 soddisfa:

   $$L = \frac{1}{1 - p}$$

   Nello stato 1 il *job generator* invia un intero v a random (job id) al *dispatcher*. Cioè in media il server riceve un job ogni L secondi. Dallo stato 1 il *job generator* transisce nello stato 0 con probabilità 1.

4. Il tempo di completamento $G(v)$ per il job v è il tempo che trascorre tra l'istante in cui il job v entra nel buffer del *dispatcher* e quello in cui entra nel buffer del *delivery manager*.

   Si realizzi un monitor che calcola il valore atteso V e la deviazione standard S del tempo di completamento di un job. Il calcolo può essere fatto come media dei tempi di completamento dei *jobs* ricevuti dal *dispatcher*.

2 Valori dei parametri

Si assumano i seguenti valori per i parametri del modello.

1. $W = 5$
2. $N = 10$
3. $B = 20$
4. $L = 10$
5. $D = 1$

3 Output della simulazione

Si usi l'istruzione Modelica `terminate` per terminare la simulazione dopo 10000 unità di tempo (secondi) simulato.

Alla terminazione si stampino nel file `outputs.txt` le seguenti informazioni.

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

Valore dei parametri nell'ordine in cui sono listati nella sezione 2, `ID` = `yyy`, `MyMagicNumber` = `zzz`, `time` = `xxx`

dove:

1. `yyy` è il vostro numero di matricola (nel parametro `ID`)
2. `zzz` è il vostro MagicNumber calcolato nel parametro `MyMagicNumber`
3. `xxx` è il valore della variabile Modelica `time` quando la simulazione viene terminata dal comando `terminate`.

La seconda riga ha il seguente formato:

`V` = <Valore medio del tempo di completamento di un job>, `S` = <Standard deviation tempo di completamento di un job>,

Si avranno quindi in totale 2 righe.

Si usi un orizzonte di simulazione molto grande. 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.