

Sistemi IV

(Corso di Informatica, 5 anno)

Linguaggi, Strumenti e modelli di Simulazione

Luciano Bononi

bononi@cs.unibo.it

<http://www.cs.unibo.it/~bononi>

Ricevimento: Lun-Ven 9-18 :-)

presso ufficio dottorandi, Mura Anteo Zamboni 7, Bologna

Simulazione (1)

- Una tecnica di valutazione delle prestazioni di sistemi: vediamo i pro e contro
 - **simulazione** vs. misure del sistema vs. analisi
 - se il sistema non esiste ancora non posso fare misure
 - previsioni sulla performance o confronto tra sistemi
 - permette di valutare i sistemi nelle stesse condizioni
 - permette di testare condizioni del sistema che accadono di rado
 - permette di ottenere stime non misurabili su sistemi reali
 - richiede molto tempo (rispetto a analisi)
 - non introduce overhead di misura nel sistema
 - richiede conoscenza di tecniche e linguaggi o tool specifici
 - permette un buon livello di accuratezza

Simulazione (2)

- Errori comuni nel progettare una simulazione
 - livello di dettaglio non appropriato
 - una tecnica di analisi è poco dettagliata: assunzioni e semplificazioni
 - il modello di simulazione è potenzialmente estendibile fino ad includere ogni dettaglio
 - troppo dettaglio richiede troppo “codice” e quindi errori, lentezza del processo di generazione del modello e di simulazione, consumo di risorse, non intuitività.
 - meglio procedere a partire da modelli semplici e via via aggiungere dettagli dove l’analisi di sensitività richiede di investigare

Simulazione (3)

- Errori comuni nel progettare una simulazione
 - Linguaggio (o strumento) non appropriato
 - linguaggi orientati alla simulazione
 - favoriscono la creazione del modello
 - favoriscono la validazione (tracce di esecuzione)
 - favoriscono analisi statistica
 - linguaggi general purpose
 - sono portabili su diverse architetture
 - permettono di creare simulatori distribuiti (+ HLA)
 - strumenti (tool di simulazione)
 - favoriscono in tutto ciò che è possibile favorire
 - richiedono architetture HW o SW specifiche e costose?

Simulazione (4)

- Errori comuni nel progettare una simulazione
 - Errori di modellizzazione
 - modelli complessi (reti di code o programmi) richiedono validazione accurata (alto rischio di bug presenti)
 - assunzioni non valide nel modello rispetto alla caratterizzazione del sistema dato
 - Errori sulla condizione iniziale del sistema
 - la condizione iniziale del sistema può avere effetto sui dati ottenuti, a meno che non si intenda valutare lo stato stazionario (steady state): in tal caso il transiente va eliminato.

Simulazione (5)

- Errori comuni nel progettare una simulazione
 - Simulazioni troppo corte
 - più la simulazione è corta più risente dello stato iniziale
 - simulazioni corte non permettono di assumere le distribuzioni fissate per il workload
 - la durata della simulazione deve essere determinata in modo da garantire stime accurate (piccoli intervalli di confidenza)
 - la correlazione dei risultati di simulazione complica le tecniche di interruzione delle simulazioni basate su stime della varianza

Simulazione (6)

- Errori comuni nel progettare una simulazione
 - Generatori di numeri pseudo-casuali
 - vanno scelti con cura. Mai fidarsi delle offerte...
 - pseudo-casuale \neq casuale (il seme deve essere scelto, non generato a caso)
 - Semi dei generatori pseudo-casuali
 - diverse sequenze pseudocasuali devono avere semi scelti per garantire assenza di correlazione
 - le sequenze pseudocasuali non devono essere condivise tra più processi per garantire assenza di correlazione

Simulazione (7)

- Errori comuni nel progettare una simulazione
 - Sottostima dei tempi richiesti
 - se la simulazione non è convincente la si estende
 - problemi (Murphy's law) implicano dilatazione tempi
 - Obiettivi non raggiungibili
 - non è possibile “modellare tutto il sistema”!
 - Si può solo specificare quali parti o specifiche del sistema modellare
 - Carenza di uno o più tra:
 - coordinazione del gruppo, capacità di modellizzazione e statistica, programmazione corretta, comprensione del modello

Simulazione (8)

- Errori comuni nel progettare una simulazione
 - “Gruppo col pizzaiolo/a”
 - evoluzione del progetto deve essere seguita alla pari
 - Documentazione scarsa
 - non favorisce evoluzione e riutilizzo di modelli o moduli
 - Strumenti inadeguati per il lavoro di gruppo
 - trascurare i “valori misteriosi”...hanno sempre una spiegazione: quale?

Simulazione (9)

- Un po' di terminologia spicciola
 - Variabili di stato
 - variabili che descrivono lo stato del sistema
 - permettono di interrompere e riprendere la simulazione
 - Evento
 - un cambiamento nello stato del sistema
 - Modelli a **tempo** continuo/discreto
 - tempo continuo: lo stato del sistema è definito sempre
 - tempo discreto: lo stato del sistema è definito in $t, t+1, \dots$
 - Modelli a **stato** continuo/discreto
 - continuous-event model: le var. di stato sono continue
 - discrete-event model: le var. di stato sono discrete

Simulazione (10)

- Un po' di terminologia spicciola
 - **modello** deterministico/probabilistico
 - deterministico: $\text{output}(\text{input } k) = x$
 - probabilistico: $\text{output}(\text{input } k) = x_1, x_2, x_3 \dots$
 - Modello statico/dinamico
 - statico: lo stato non dipende dal tempo
 - dinamico: lo stato dipende dal tempo
 - Modello lineare/non lineare
 - lineare: $\text{output}(x) = k * x$
 - non lineare: $\text{output}(x) \neq k * x$

Simulazione (11)

- Un po' di terminologia spicciola
 - Modello aperto/chiuso
 - chiuso: nuovi job non entrano o escono dal sistema
 - aperto: source e sink per i job
 - Modello stabile/instabile
 - stabile: comportamento converge allo stato stazionario
 - instabile: comportamento che cambia continuamente (e se è oscillante?)

Simulazione (12)

- Tipi di simulazione
 - (Emulazione: simulazione usando HW e firmware)
 - **MonteCarlo simulation:**
 - non esiste l'asse del tempo
 - serve a modellare fenomeni che non cambiano nel tempo
 - richiede generazione pseudocasuale
 - es. calcolo di integrali, calcolo di Pigreco, ecc.

Simulazione (13)

- Tipi di simulazione
 - Trace-driven simulation: pro
 - simulazione basata su eventi ordinati per tempo registrati dal sistema reale
 - es. trasmissione traffico video MPEG
 - credibile, in quanto non basata su distribuzioni di input
 - semplice validazione
 - nessuna assunzione sul workload e correlazione
 - buona analisi di sensitività
 - meno effetti di randomizzazione (varianza minore)
 - confronto tra sistemi è fair

Simulazione (14)

- Tipi di simulazione
 - Trace-driven simulation: contro
 - complessità del modello è maggiore (di solito)
 - il trace è riduttivo in termini di rappresentazione dei carichi possibili (soprattutto per sistemi diversi)
 - il trace è lungo e legato alla situazione in cui è registrato
 - occorrono molti trace diversi per motivare e dimostrare i risultati
 - comportamenti transienti devono essere registrati nel trace
 - un trace diverso per ogni workload

Simulazione (15)

- Tipi di simulazione
 - **Discrete-event** simulation
 - simulazione che usa un modello del sistema a **stato discreto**
 - opposto di continuous-event simulation
 - può essere basata su tempo continuo o discreto!

Simulazione (16)

- Tipi di simulazione: **Discrete-event simulation**
 - Richiede una struttura comune a tutte le simulazioni
 - scheduler di eventi: mantiene lista ordinata e linkata di eventi futuri (schedulati, cancellati, rinviati, bloccati). Gestisce il clock. Evento=(time, puntatore al codice della routine di evento)
 - Liste ordinate (SIMULA), liste lineari indicizzate (listaK=dT), tree (heap)
 - Clock di simulazione e meccanismo di avanzamento del tempo
 - » event-driven: $\text{clock} += \text{tempo al prossimo evento il lista}$
 - » unit-time: $\text{clock} += \text{Delta... Sono accaduti eventi? (Non usata)}$
 - variabili globali (stato del sistema), locali (stato dei job)
 - routine di evento: ogni evento è simulato dalla propria procedura, che aggiornano le var. locali, di stato e schedulano nuovi eventi.
 - Routine di input: procedura di generazione dei dati di input a partire dai parametri del modello

Simulazione (17)

- Tipi di simulazione: **Discrete-event** simulation
 - Richiede una struttura comune a tutte le simulazioni
 - report generator: procedure che al termine della simulazione generano i dati di stima
 - routine di inizializzazione: chiamate per prime, definiscono lo stato iniziale del sistema, gli stream pseudo-casuali, ecc.
 - routine di trace: procedure per notificare eventi o stime a run-time
 - gestione dinamica della memoria: il numero di job creati nel corso della simulazione è variabile (source e sink). Occorre un meccanismo di garbage collection.
 - MAIN program: chiama init, input, esegue il corpo della simulazione e genera i risultati.

Simulazione (18)

- Modellare interarrivi e servizi
 - tempo di interarrivo t_a è var. aleatoria con funzione densità $f(t_a)$ e funzione cumulativa $F(t_a)$.
 - $E[t_a]=T_a$ =tempo medio di interarrivo
 - $\text{Lambda}=1/T_a$ = freq. Media degli arrivi
 - stesso dicasi per i tempi di servizio (T_s e Mu)
 - rapporto $\text{sigma}=\text{Lambda}/\text{Mu}$ = fattore di utilizzo=prob che il servizio sia occupato.

Simulazione (19)

- Distribuzioni usate frequentemente
 - Bernoulli distribution (discreta)
 - p = prob di successo, $1-p$ = prob. di insuccesso
 - usata per modellare la probabilità di eventi 1/0 su prove ripetute e indipendenti (tali che prob. 1 = p in ogni prova)
 - Media = p
 - Varianza = $p*(1-p)$
 - Generazione di v.c. soggette a distrib, Bernoulliana
 - genero v.c. u uniforme in $(0,1)$: $(u \leq p)? 1$ o.w. 0
 - es. p =prob. Di bit errato (BER)

Simulazione (20)

- Binomial distribution (discreta)
 - modella il numero di successi in una sequenza di prove bernoulliane (prove i.i.d.)
 - prob. successo in una prova = p
 - numero di prove effettuate = n , numero successi attesi = x
 - $f(x) = \binom{n}{x} p^x (1-p)^{n-x}$
 - Media: np
 - Varianza: $np(1-p)$
 - genero n $U(0,1)$: quanti di n sono $<p$?
 - es. $f(0)$ su 8 bit è la prob. di un Byte OK con $BER=p$.
 - es. $f(1)$ su 8 bit è la prob. di un bit errato su 8 con $BER=p$.

Simulazione (21)

- Exponential distribution (distr. continua)
 - usatissima in QNM per tempi di interarrivo e di servizio.
 - Parametro $a = \text{media} = 1/\text{Lambda}$ ($a > 0$)
 - range: 0..infinito
 - $f(x) = 1/a * e^{(-x/a)}$
 - $\text{CDF} = F(x) = 1 - e^{(-x/a)} \iff F(T) = P(T < t) = 1 - e^{(-\text{Lambda} * t)}$
 - $\text{Varianza} = a^2$
 - memoryless property (semplifica analisi).
 - Modella i tempi di eventi con cause molteplici e indep.
 - Es. tempi tra rotture, tempi di interarrivo in servizio
 - $\text{exp}(a) = \text{genero } u \text{ U}(0,1), \text{ poi ritorno } -a * \ln(u)$

Simulazione (22)

- Erlang distribution
 - estensione della distribuzione esponenziale, se il coeff. Di variazione è minore di 1
 - modella i tempi di servizio in QNM t.c. un server erlangiano(a,m) ha tempi di servizio distribuiti come quelli di una serie sequenziale di m server esponenziali.
 - Genero m valori $U_i = U(0,1)$,
poi $\text{Erlang}(a,m) = -a \ln(\prod_{i=1,m} U_i)$

Simulazione (23)

- Geometric Distribution (equivale a exp discreta)
 - modella la distribuzione del numero di prove bernoulliane fino ad avere il primo successo.
 - p =prob. di successo. Range = 1,2,...infinito
 - pmf= $f(x)=(1-p)^{(x-1)} * p$
 - CDF= $F(x)=1-(1-p)^x$
 - Media= $1/p$
 - Varianza= $(1-p)/p^2$
 - memoryless property.
 - Genero $u \sim U(0,1)$ e calcolo $G(p)=\text{ceiling}(\ln(u)/\ln(1-p))$

Simulazione (24)

- Normal (Gaussian) Distribution (continua)
 - $N(0,1)$ con $\mu=0$ e $\sigma=1$ è Distr. Normale Std.
 - μ =media, σ =dev std > 0 , $\text{varianza}=\sigma^2$
 - Range $-\infty \dots \infty$
 - modella le somme di sorgenti indipendenti
 - equivale alla poissoniana per le distr. continue.

Simulazione (25)

- Pascal Distribution
 - estende la geometrica.
 - Modella il numero di prove bernoulliane fino a includere il successo numero m
 - p =prob di successo, m = numero di successi
 - range: $x=m, m+1, \dots, \text{inf}$
 - $f(x) = \binom{x-1}{m-1} p^m (1-p)^{x-m}$
 - Media= m/p
 - Varianza= $m(1-p)/p^2$
 - genero $m G(p)$ e ritorno la loro somma.

Simulazione (26)

- Poisson distribution (discreta)
 - forma limitata della binomiale
 - $\text{Lambda} = \text{media} > 0$, $\text{Range} = 0, 1, 2, \dots, \text{inf}$
 - $f(x) = P(X=x) = \frac{\text{Lambda}^x * (e^{-\text{Lambda}})}{x!}$
 - $\text{Media} = \text{Lambda} t$, $\text{Varianza} = \text{Lambda} t$
 - modella il numero di arrivi in un dato intervallo t
 - ottimo se gli arrivi sono da molte sorgenti indep. Ed equiprobabili.
 - Il tempo di interarrivo t_a compreso tra arrivi poissoniani è esponenziale.
 - Vedere dispense per generare v.c. poissoniane.

Simulazione (27)

- Uniform distribution (continua e discreta)
 - range : a..b
 - $f(x) = 1/(b-a)$
 - CDF = $F(x) = 0$ se $x > b$, $(x-a)/(b-a)$ se $a < x < b$, 0 o.w.
 - Media: $(a+b)/2$
 - Varianza: $(b-a)^2/12$
 - genero u secondo $U(0,1)$ e ritorno $a+(b-a)*u$