

SIMULACIJA STREŽNIH SISTEMOV Z GPSS

N. GUID

UDK: 681.3 : 519.682.6

VISOKA TEMNIŠKA ŠOLA, MARIBOR

Članek obravnava reševanje najrazličnejših osnovnih tipov strežnih sistemov s simulacijo z jezikom GPSS. Podani so problemi pri vrednotenju rezultatov in priporočila za izboljšanje točnosti simulacije.

QUEUEING SYSTEMS SIMULATION WITH GPSS. This paper deals resolution of the most different fundamental types of queueing systems by the simulation with GPSS. The problems in validating simulation results and recommendations in improvement of simulation precision are considered.

I. UVOD

Problemi čakanja, ki jih uspešno rešujemo z metodami množične strežbe, pogosto obstajajo tudi v računalniških sistemih, kjer se pojavljata neregularnost dolžine strežbe in neregularnost prihoda zahtev ali kjer mnogo zahtev uporablja eno ali več računalniških strežnih naprav. Pomen strežnih sistemov in njihove analize je zlasti narasel s povečujočo uporabo računalniških sistemov z delovanjem v realnem času (procesni računalniki, računalniki s časovnim dodeljevanjem). Z metodami množične strežbe je možno oceniti zakasnitve pri obdelavi vseh vrst (programska ali elektronska oprema), velikost potrebnega pomnilnika, izkoriščenost pomnilnika in procesorja, dalje nam pomaga določiti najučinkovitejše strežno pravilo itd.

Teorija množične strežbe je razvila eksaktne analitične metode, s katerimi pa je mogoče rešiti le nekatere idealizirane strežne sisteme. Tako smo prisiljeni razvijati aproksimativno analizo iz kombinacij nekaj idealiziranih sistemov. Omenjene direktne analitične metode lahko nadomestijo drage simulacijske postopke, toda samo do neke meje, ko je potrebno zaradi večjega obsega in natančnosti uporabiti simulacijo, ki jo izvedemo z računalniškimi jeziki za simulacijo diskretnih sistemov.

Prednost simulacije pred analitičnimi metodami je v tem, da lahko obravnava nek zapleten detajl v logiki delovanja sistema in izračuna učinek sprememb tega detajla. Dalje lahko s simulacijo ocenimo medsebojni vpliv med več podsistemi.

Simulacija tudi večinoma ni direktno orodje za sintezo. V splošnem je simulacija omejena na določitev, kako posebna konfiguracija reagira na posebno okolje. Še vedno ostane načrtovalcu funkcija analize rezultatov in odločitve, kje in kako izboljšati sistem, ali z drugimi besedami, načrtovalec ostaja povratni element v načrtovalni zanki.

II. SIMULACIJSKI JEZIKI

Simulacijske jezike (simulatorje) delimo v [1]:

1. uporabnikove simulatorje,
2. posebno namenske simulatorje in

3. splošno namenske simulatorje.

Kdorkoli hoče napisati svoj lastni simulacijski program (t.i. uporabnikov simulator), mora imeti za to naslednje razloge:

- želene karakteristike modela niso niti primerne, niti možne v obstoječih simulacijskih jezikih,
- zahtevane so posebne vhodne in izhodne lastnosti ter
- program se bo uporabljal dostikrat.

Glavna prednost uporabnikovih simulatorjev je učinkovitost. Cena za njihov razvoj mora biti poplačana v večkratni uporabi. Njihova slabost je nefleksibilnost. Pisani so seveda v višjih programskih jezikih.

Posebno namenski simulator ima vgrajeno logiko posebnega strežnega sistema, tako da uporabniku ostane samo vstavljanje nekaterih parametrov. Število posebno namenskih simulatorjev je ogromno. Dobimo jih lahko pri prodajalcih računalnikov in konzultantskih organizacijah. Dokumentacija javnosti žal ni dostopna. Omenil bi samo tri jezike:

1. GERTS III QR je razširjega verzija GERT (Graphical Evaluation and Review Technique, [2]). Razvili so ga za analizo sistemov s strežnimi napravami, ki zahtevajo večkratne izvore za izvršitev strežne aktivnosti.
2. QAL (Queuing Analysis Language, [3]) je visoko nivojski jezik za poenostavitev reševanja tako enostavnih kot kompleksnih modelov množične strežbe. Z njim zelo učinkovito simuliramo sodobne računalniške sisteme, ki imajo vedno več paralelnosti v delovanju.
3. CSS (Computer System Simulator, [4]) je manj splošen od GPSS. Uporablja precej tehnik iz GPSS, namenjen je za simulacijo računalniškega programskega sistema, predvsem operacijskega sistema.

Splošno namenski simulator je namenjen za simulacijo najrazličnejših strežnih sistemov. Najbolj razširjeni jeziki so GPSS, SIMSCRIPT in SIMULA.

Sam sem izbral GPSS, saj je bil to edini jezik, ki mi je bil dostopen. GPSS (General Purpose Simulation System) so razvili pri IBM. Obstojata več verzij in je instaliran na večini računalniških sistemov. Ima naslednje pomembne

lastnosti:

- je lahek za učenje,
- zahteva velik hitri pomnilnik,
- zahteva daljši čas simulacije in
- je primeren za simulacijo strežnih sistemov vseh vrst.

III. VREDNOTENJE REZULTATOV SIMULACIJE

Vprašanje, ali je simulacijski rezultat pravilen, zahteva obravnavo dveh neodvisnih zahtev [1]:

1. prva se nanaša na točnost modela
2. druga pa na preciznost statičnih rezultatov.

Ad. 1: Visoka stopnja natančnosti zahteva, da mnogo poenostavljenih ali izpuščenih detajlov bistveno ne vpliva na sistemsko učinkovitost. S primerjavo simulacijskih tekov z ali brez nekega detajla lahko izmerimo relativni efekt spremembe. Zmeraj spreminjamo samo en parameter ali spremenljivko, da lahko ugotovimo njen vpliv. To je zlasti pomembno pri kompleksnih modelih, saj se le tako izognemo logičnim, t.i. sistematskim napakam.

Ad. 2: Ta problem je posledica narave procesa v tehniki simulacije, ki temelji na odbirkih iz neke porazdelitve (statistike). Problem je določiti točko ali trenutek, ko je dosežena iskana preciznost. To dosežemo z nivojem zaupanja, kar pa je težko izvesti.

Simulacijski rezultati navadno niso neodvisni, temveč imajo celo visoko korelacijo, kar ima za posledico težko določitev variance, ki je potrebna pri izračunu intervala zaupanja [5]. V praksi opazujemo nekaj zaporednih kumulativnih rezultatov in jih primerjamo med seboj. Če ti rezultati fluktuirajo v ozkem pasu, je preciznost dosežena, če opazimo njihov trend, povečamo število simulacijskih tekov.

Potem je tu še nestacionarnost porazdelitve odbirkov oziroma odvisnost od začetnih pogojev. Vpliv prehodnega pojava odpravimo na več načinov:

- a) Postavimo tipične začetne pogoje, kar pomeni, da moramo sistem že delno poznati, preden sploh začnemo s simulacijo.
- b) Simuliramo daljši čas.
- c) Z metodo paketne srednje vrednosti simuliramo krajši čas.

N tekov razdelimo v p paketov dolžine n tekov ($N=np$). Učinek je enak učinku eksperimenta dolžine n tekov, ki ga ponovimo p -krat, tako da končno stanje enega teka postane začetno stanje naslednjega simulacijskega teka. Prednost te metode je v tem, da ni potrebno odstraniti začetno periodo pri vsakem paketu in da zahteva majhen pomnilnik, saj shranjuje samo vsoto paketnih srednjih vrednosti in vsoto njihovih kvadratov ter števila za tvorbo tekoče paketne srednje vrednosti.

- d) Simuliramo nekaj tekov z netipičnimi začetnimi pogoji, nato samo zbrisemo statistiko, ne da bi spremenili stanje sistema. Nato nadaljujemo s simulacijskimi teki ter izračunamo novo statistiko. Toda žal ne eksistira preprosto pravilo, ki bi povedalo, koliko tekov moramo zbrisati. Metoda za določitev števila izbranih tekov temelji na poznavanju funkcije standardne deviacije v odvisnosti od števila

tekov pri simulaciji brez brisanja začetnih tekov.

Pri GPSS je najlažje uporabiti zadnji način (stavke RESET za brisanje statistike).

Rezultat je nadalje odvisen od zaporedja naključnih števil [6]. Izkaže se, da je fluktuacija še zmeraj opazna tudi pri tisočem teku (cca 5%), no vseeno akumulirana povprečna vrednost teži k stacionarni povprečni vrednosti. Temu bi se izognili s p -kratno ponovitvijo serije n simulacijskih tekov z različnimi zaporedji naključnih števil, kar pa s standardnimi programskimi instrukcijami pri GPSS ni mogoče.

IV. SIMULACIJA Z JEZIKOM GPSS

Narava in osnovni koncepti GPSS so taki, da omogočajo enostavno simulacijo vseh vrst strežnih sistemov in mrež. GPSS ima vgrajen program za izpis določenih statističnih izhodnih rezultatov (izkoriščenost strežnika, intenzivnost prometa, povprečno in maksimalno število čakajočih enot v vrsti, delež enot, ki jim ni treba čakati v vrsti, povprečna dolžina strežbe, povprečna dolžina čakanja v vrsti itd.). To so merila, ki nastopajo tudi pri analizi strežnih sistemov in jih lahko dobimo zato, ker GPSS vsebuje generatorje psevdonaključnih števil. S pomočjo le teh in najrazličnejših funkcijskih odvisnosti je možno predstaviti še tako zapleteno verjetnostno porazdelitev presledkov med zaporednima prihodoma enot in verjetnostno porazdelitev dolžine strežbe [7].

Moje izkušnje pri delu z GPSS

Edini večji problem predstavljajo generatorji naključnih števil, saj nobeden od številnih učbenikov ne zajema kompleksno analizo njihovega delovanja.

Zaporedje, ki ga generira eden izmed 8 enakih generatorjev naključnih števil (RN1 do RN8), spremenimo le s spremembo začetne vrednosti v algoritmu za generiranje naključnih števil (imenovano "seed").

1. Če imamo v nekem sistemu dva ali več naključnih procesov, tedaj ne smemo uporabiti različne generatorje z isto začetno vrednostjo, saj pride do neke vrste "resonance". Posledica tega je, da rezultati tudi pri mnogo simulacijskih tekov (cca 10.000) odstopajo od povprečnih vrednosti za cca 50%.
2. Sprememba enega ali več začetnih vrednosti naključnih generatorjev spremeni rezultate pri cca 10.000 simulacijskih tekov do 5%, pri cca 20 simulacijskih tekov pa do 40%. To pomeni, da je simulacija z malo teki nenatančna.
3. Bloki z enakomerno porazdelitvijo (n.pr. ADVANCE 80, 40; GENERATE 3,1 itd.) uporabljajo vrednosti samo enega generatorja naključnih števil, t.j. RN1.
4. Če več naključnih procesov v strežnem sistemu uporablja isti generator naključnih števil, ne moremo študirati vpliva prioritete, saj tedaj nastopi drugi vrstni red zaseganja števil iz generatorja naključnih števil kot pri sistemu brez prioritete [7]. To pomeni, da se hkrati z vplivom prioritete vrine še drug vpliv, ki pokvari rezultate. Temu se izognemo tako, da vsakemu naključnemu procesu priredimo en generator naključnih števil z različno začetno vrednostjo. Če imamo samo eno enakomerno porazdelitev, tedaj ji pustimo generator RN1, kar pa seveda nikjer v programu eksplicitno ne zapišemo (za druge na-

ključne procese tedaj ne smemo uporabiti RN1).

5. Primerjava analitično izračunanih vrednosti ter rezultatov simulacije pri različnih številih tekov je pokazala, da je potrebno simulirati 10.000 tekov oz. enot, ki gredo skozi strežni sistem. Večina primerjav je dala odstopanje manjše od 5 %, edino v nekaj primerih sem dobil $8 + 12$ % odstopanja od analitično dobljene vrednosti. Primerjavo sem izvršil tudi z različnimi sekvencami naključnih števil (različne začetne vrednosti generatorjev) in vsa odstopanja so bila v gornjih mejah.
6. Primerjal sem tudi vrednosti, dobljene pri različno dolgih eliminiranih začetnih periodah (odprava prehodnega pojava) in nisem opazil hitrejšje konvergenke k analitičnim povprečnim vrednostim.

Moje priporočilo je sledeče:

Upoštevati je treba podana priporočila v zvezi z uporabo generatorjev naključnih števil. Simulirati moramo eno tako, da bo šlo skozi strežni sistem cca 10.000 enot enega tipa, ne da bi upoštevali prehodni pojav. Rezultat simulacije bo z veliko verjetnostjo odstopal od analitično dobljene vrednosti največ za 5 %.

V. SIMULACIJA RAZLIČNIH TIPOV STREŽNIH SISTEMOV Z GPSS

GPSS je primeren za simulacijo strežnih sistemov vseh vrst. Z njim lahko enostavno simuliramo najbolj splošno naključno porazdelitev. Izvor zahtev in porazdelitev vhodnega toka omogoči blok GENERATE, porazdelitev dolžine strežbe blok ADVANCE. Enega strežnika predstavimo z blokoma SEIZE in RELEASE, več paralelnih strežnikov pa z blokoma ENTER in LEAVE. Vso statistiko čakajoče vrste dobimo s pomočjo blokov QUEUE in DEPART, medtem ko se statistika strežnih naprav izvrši avtomatsko.

Pri simulaciji nehomogenih strežnih sistemov uporabimo več enakih segmentov modela s samo različnimi bloki GENERATE.

Pri cikličnih strežnih sistemih koristimo blok TRANSFER.

Ne da bi bilo potrebno posebej specificirati, GPSS simulira strežni sistem z neskončno kapaciteto in pravilom FIFO.

Sisteme z omejeno kapaciteto simuliramo s pomočjo blokov ENTER, ADVANCE \emptyset (s časovno zakasnitvijo \emptyset) in LEAVE (poslednji blok LEAVE moramo postaviti za blok, ki označuje vstop enote v proces strežbe).

Strežne sisteme s poljubnim neprioritetnim strežnim pravilom (n.pr. LIFO, RSS) simuliramo s pomočjo blokov LINK in UNLINK.

Pri sistemih z neprekinjalnim prioritetenim strežnim pravilom je potrebno samo prirediti ustrezeni prioriteten indeks določenemu tipu enot (uporabimo samo operand E v bloku GENERATE).

Za simulacijo sistemov s prekinjalnim prioritetenim strežnim pravilom z nadaljevanjem uporabimo poleg operanda E v blokih GENERATE še PREEMPT in RETURN namesto SEIZE in RELEASE v segmentih modela s prekinjalnimi enotami.

Pri simulaciji sistemov z drugimi prioritetenimi pravili moramo uporabiti še logične, testne in selektivne operacije.

Sisteme z grupno strežbo simuliramo s pomočjo operanda B v blokih SEIZE in RELEASE oziroma ENTER in LEAVE.

Pri simulaciji strežnih sistemov z grupnim vhodnim tokom uporabimo blok SPLIT.

Simulacija nreže strežnih sistemov je preprosta in ne zahteva kakih posebnih navodil, saj je glavna značilnost GPSS, da je bločni diagram nekega sistema ekvivalenten tako diagramu poteka kot glavnemu programu.

VI. PRIMER SIMULACIJE Z GPSS

V center za obdelavo sporočil prihajata dva tipa (razreda) sporočil s poissonsko porazdelitvijo ($\lambda_1=0,5 s^{-1}$, $\lambda_2=0,1 s^{-1}$). Prvi tip zahteva krajšo in konstantno dolžino obdelave ($b_{11}=0,2 s$), drugi tip pa daljšo dolžino obdelave z eksponentno porazdelitvijo. Primerjal sem delovanje sistema:

1. brez prioritete
2. če uvedemo neprekinjalno prioriteto sporočilu tipa 1 in
3. če uvedemo prekinjalno prioriteto sporočilu tipa 1.

Problem sem rešil analitično in s pomočjo simulacije. Naš primer predstavlja strežni sistem, ki ga označimo z razširjeno Kendallovo oznako $M_1, M_2/D_1, M_2/1/\infty, \infty/\infty, \infty/FIFO (PRI)$.

Za vse tri variante sem izračunal analitično povprečno dolžino zadrževanja za posamezen tip sporočila (W_1, W_2) in za oba tipa skupaj (W). Analitično žal ni mogoče določiti povprečnih dolžin čakanj za posamezen tip sporočila pri sistemu brez prioritete.

Da se izognemo vplivu zaradi različnega zaseganja po zaporedju naključnih števil, potem ko uvedemo prioriteto (glej poglavje IV), priredimo vsaki porazdelitvi samostojni generator naključnih števil, ki pa ne sme startati z isto začetno vrednostjo.

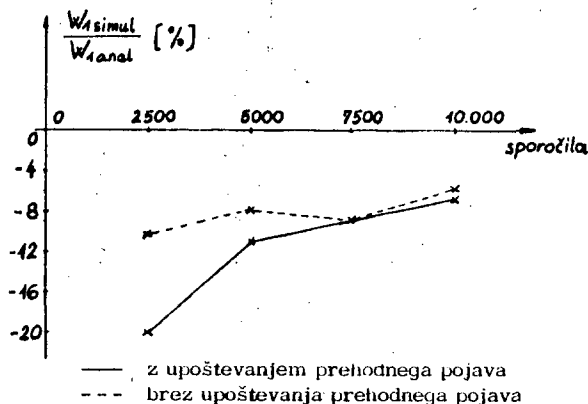
Rezultati simulacije so podani v oklepaju zraven ustreznih analitičnih rešitev (tabela 1) in ustrezajo cca 10.000 sporočilom 1. razreda in cca 2.000 sporočilom 2. razreda, potem ko v statistiki nisem upošteval prehodni pojav s 500 sporočili 1. razreda in 100 sporočili 2. razreda.

Hitro opazimo, da s prioritetenimi strežnimi pravili pri enaki izkoriščenosti procesorja dosežemo krajše povprečne dolžine zadrževanja in dolžine čakanja za oba razreda skupaj (W in W_Q). Prekinjalno strežno pravilo je v tem primeru več kot 3-krat učinkovitejše od navadnega pravila FIFO.

	brez prioritet	neprek. str. prav.	prek. str. prav.
W	7,25 (7,03)	4,5 (4,30)	2,26 (2,25)
W_1	- (6,20)	3,0 (2,78)	0,211 (0,212)
W_2	- (11,18)	12,0 (11,91)	12,55 (12,45)

Tabela 1

Študiral sem še relativna odstopanja od analitične vrednosti povprečnih dolžin zadrževanj v odvisnosti od števila simulacijskih tekov (prispelih sporočil v proces streženja) pri upoštevanju in neupoštevanju prehodnega pojava (slika 1).



Slika 1

VII. POVZETEK

Današnja razvojna stopnja teorije množične strežbe ne dopušča detaljne analize kompletnega sistema, ampak le analizo podsistemov. Pri slednjih je teorija razvila vrsto kriterijev, ki jih lahko koristno uporabljamo pri aproksimativni analizi in tudi pri simulaciji kompleksnejših sistemov množične strežbe.

Bilo bi napak, če bi trdil, da je najprimernejši jezik GPSS, saj je bil to edini jezik, ki mi je bil dostopen. Res pa je v ZDA to najbolj razširjen jezik za simulacijo diskretnih sistemov. Z njim je mogoče z lahkoto si-

mulirati še tako kompleksen strežni sistem ali mrežo strežnih sistemov. Njegovo moč veča tudi možnost klicanja subrutin v jeziku FORTRAN. Problematična je hitrost simulacije, ki bi jo lahko izboljšali z metodo paketne srednje vrednosti z različnimi neodvisnimi zaporedji naključnih števil.

Preseneča tudi relativno veliko odstopanje od analitično dobljenih vrednosti, za kar upravičeno sumim generatorje naključnih števil, ki ne uporabljajo ravno najboljšega algoritma za generacijo naključnih števil.

LITERATURA

1. P.H. Seaman: "On teleprocessing system design. Part VI: The role of digital simulation", IBM Systems Journal, Vol. 5, No. 3, 1966
2. M.J. Maggard and others: "GERTS III QR: A multiple resource constrained network simulation model", Management Datamatics, Vol. 5, No. 1, 1976.
3. D.V. Foster and others: "A language for analysis of queuing models", Modeling & Simulation, Vol. 5, Pittsburgh (USA), 24. + 26. april 1974.
4. P.H. Seaman, R.C. Soney: "Simulating operating systems" IBM Systems Journal, Vol. 8, No.4, 1969.
5. G. Gordon: "System Simulation", Prentice-Hall, Inc., New Jersey, 1969.
6. N. Guid: "Uporaba metod množične strežbe pri analizi in načrtovanju računalniških in telekomunikacijskih mrež in sistemov", magistrsko delo, Fakulteta za elektrotehniko, Ljubljana, 1977.
7. T.J. Schriber: "Simulation using GPSS", John Wiley & Sons, New York, 1974.



"HOBBY"
WIRE WRAPPING TOOLS

ok wire wrapping center ok

<p>ANOTHER UNIQUE PRODUCT DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE BY OK MACHINE & TOOL CORPORATION</p>	<p>1</p> 
<p>SIRIP / WRAP / UNWRAP TOOL MODEL WSU-30</p> 	<p>2</p>
<p>ANOTHER UNIQUE PRODUCT DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE BY OK MACHINE & TOOL CORPORATION</p>	<p>3</p> 
<p>DIP IC INSERTION TOOL WITH PIN STRAIGHTENER MODEL INS-1416</p> 	<p>4</p>
<p>ANOTHER UNIQUE PRODUCT DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE BY OK MACHINE & TOOL CORPORATION</p>	<p>5 WHAT'S? NEXT</p>

OK MACHINE & TOOL CORPORATION
3450 Corner St. - Brea, CA - 92623 • Tel: 714/991-1111 • Telex: 9527



"HOBBY"
WIRE WRAPPING TOOLS