

Jedno rješenje distribuirane obrade zasnovane na rasplinutoj logici

Milen Miodanić, mentori: Miroslav Popović, Ilija Bašičević, Branislav Atlagić

Sadržaj — Potreba za distribuiranim sistemima je veoma rasprostranjena, a jedna od primjena je upravljanje elektrodistributivnom mrežom. U ovom radu je razmotren jedan način distribuiranja korišćenjem rasplinite logike, u cilju smanjenja vremena obrade zadataka u odnosu na postojeće rješenje obrade aplikativnih proračuna. Poređenje rješenja se obavlja upoređivanjem vremena obrade zahtjeva, za različit broj računara u distribuiranom sistemu. Rješenje distribuirane obrade zasnovane na rasplinutoj logici ima manje vrijeme obrade u odnosu na postojeće rješenje, kao i na distribuiranje zadataka strategijom redom u krug. Može se koristiti za realizaciju distribuiranja u sistemima čija obrada mnogo zavisi od performansi računara.

Ključne riječi — Distribuirani sistem, Kontroler rasplinite logike, WCF.

I. UVOD

RASPOREĐIVANJE je ključni koncept u računarstvu, u sistemima sa više zadataka koji treba da budu obrađeni, u višeprocorskim operativnim sistemima, kao i operativnim sistemima u realnom vremenu. Pojava višeprocorskih sistema pružila je nove mogućnosti za povećanje brzine izvršavanja programa. Otvorilo se novo polje u istraživačkoj djelatnosti, čiji je cilj efikasna paralelizacija programa kao i distribuiranje i obrada zadataka na udaljenim računarima.

Danas se sreće potreba za distribuiranim obradama u velikom broju primjena. Jedna od njih je sistem za upravljanje elektrodistributivnom mrežom - DMS (*Distribution Management System*) i ovaj rad predstavlja doprinos u tom smjeru.

Sistemi za upravljanje elektrodistributivnim mrežama su zasnovani na složenim matematičkim proračunima. Srce DMS-a je računanje funkcije *LoadFlow* nad određenom elektrodistributivnom mrežom [1].

WCF (*Windows Communication Foundation*) je aplikativna programska sprega u okviru .NET razvojnog okruženja za formiranje i distribuiranje povezanih aplikacija [2].

Ovaj rad je djelimično finansiran od Ministarstva za nauku Republike Srbije, projekat 12004, od 2008. god.

Milen Miodanić, Fakultet tehničkih nauka u Novom Sadu, Srbija, (email: milen.miodanic@dmsgroup.rs)

Mentori:

Dr Miroslav Popović, Fakultet tehničkih nauka u Novom Sadu, Srbija, (email: miroslav.popovic@rt-rk.com)

Dr Ilija Bašičević, Fakultet tehničkih nauka u Novom Sadu, Srbija, (email: ilija.basicevic@rt-rk.com)

Dr Branislav Atlagić, Fakultet tehničkih nauka u Novom Sadu, Srbija, (email: branislav.atlagic@rt-rk.com)

U računarskim mrežama, balansiranje opterećenja predstavlja tehniku dijeljenja posla na dva ili više računara, procesorskih jedinica ili drugih resursa, u cilju dobijanja najbolje iskorišćenosti resursa [3]. Korišćenjem višestrukih komponenti, umjesto jedne komponente, u postupku balansiranja opterećenja, može se povećati pouzdanost sistema.

Rasplinuta (*fuzzy*) logika predstavlja metodologiju rešavanja problema kontrolisanja sistema. Obezbeđuje jednostavan način da se dođe do određenog zaključka na osnovu nejasnih, dvosmislenih, nepreciznih, ili djelimičnih ulaznih podataka [5].

Sistem CE (*Calculation Engine*) predstavlja referentni sistem obrade aplikativnih proračuna [6]. Predstavlja dio DMS softvera, razvijenog u kompaniji TelventDMS u Novom Sadu.

II. WCF

WCF predstavlja spoj aktuelnih tehnologija za distribuirane sisteme, razvijenih s ciljem što većeg približavanja idealnim uslužno orjentisanim arhitekturama (*Service Oriented Architecture*).

WCF poruka je skup informacija koje se prenose između procesa. Postoje tri tipa razmjene poruka: *simplex* – jednosmjerna komunikacija, *duplex* – otvorena i dvosmjerna komunikacija i *request-reply* – dvosmjerna komunikacija, zahtjev - odgovor.

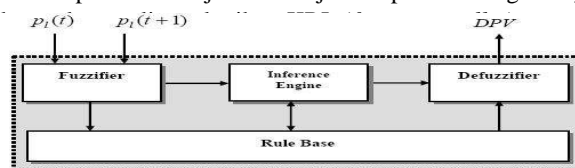
WCF usluživač se sastoji od opisa, krajnjih tačaka, tipova povezivanja, ugovora i realizacije ugovora (sl. 1).



Sl. 1. Struktura WCF usluživača

III. RASPLINUTA LOGIKA

Komponenta koja realizuje rasplinutu logiku je



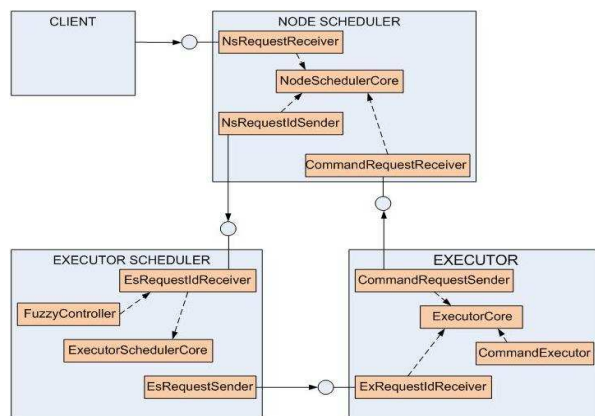
Sl. 2. Kontroler rasplinite logike
Četiri komponente su ugrađene u kontroler (sl. 2):

- rasplinjavanje (*fuzzification*): predstavlja postupak pretvaranja ulaznih podataka u vrijednosti lingvističkih promjenljivih. Funkcija na kojoj se predstavlja preslikavanje ulaznih u lingvističke vrijednosti se zove funkcija pripadnosti (*membership function*).
- skup pravila rasplinite logike (*Rule Base*): predstavlja način upravljanja sistema, odnosno pokazuje odnos ulaznih i izlaznih lingvističkih promjenljivih. Jednostavan način za predstavljanje pravila rasplinite logike je u vidu tabele.
- mehanizam zaključivanja (*Inference Engine*): postupak dobijanja vrijednosti izlaznih lingvističkih promjenljivih koristeći skup pravila.
- pretvaranje rezultata u izlazne vrijednosti (*defuzzification*): postupak dobijanja izlaznih vrijednosti kontrolera se obavlja nad obuhvaćenim rasplinitim skupom izlaznih lingvističkih vrijednosti.

IV. OPIS RJEŠENJA

A. Sistem za distribuiranu obradu

Sistem za distribuiranu obradu aplikativnih proračuna se sastoji od klijentske aplikacije, raspoređivača zadataka i N izvršilaca, prikazan na sl. 3. Komponente međusobno komuniciraju posredstvom WCF mehanizma. Koristi se *duplex* tip razmjene poruka. Klijentska aplikacija je realizovana kao internet aplikacija. Ostale komponente su konzolnog tipa. Razmjena podataka između klijentske aplikacije i raspoređivača se obavlja koristeći HTTP protokol, dok je u ostalim komunikacijama zastupljen TCP protokol.



Sl. 3. Struktura sistema distribuirane obrade aplikativnih proračuna

Klijentska aplikacija sadrži podatke za obradu, koji su u obliku XML datoteka i predstavljaju podatke korijena određene elektrodistributivne mreže.

Raspoređivač zadataka (*NodeScheduler*) komunicira sa svim komponentama i predstavlja centralni dio sistema. Njegova uloga je prihvatanje zadataka od klijentske aplikacije i raspoređivanje nekom od izvršilaca određenom strategijom.

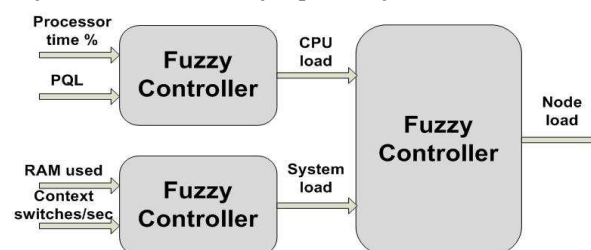
Izvršioči čine komponente *ExecutorScheduler* i *Executor*. Komponenta *ExecutorScheduler* sadrži kontroler rasplinite logike čijim se izvršavanjem dobija opterećenje

izvršioča. Vrijednost opterećenja se šalje raspoređivaču, na osnovu koga se obavlja raspoređivanje zadataka. Takođe, komponenta prosljeđuje i identifikatore zadataka od raspoređivača do izvršioča.

Komponenta *Executor* je zadužena za obradu aplikativnih proračuna. Koristeći dobijene identifikatore zadataka, dobija od raspoređivača potrebne podatke koje obrađuje određena funkcija predstavljena kao DLL (*Dynamic-Link Library*). Nakon obrade rezultati se šalju raspoređivaču.

B. Kontroler rasplinite logike

Na Sl. 4. je prikazan sistem kontrolera rasplinite logike koji se koristi za računanje opterećenja izvršioča.



Sl. 4. Sistem kontrolera rasplinite logike

Sistem je realizovan u dva nivoa. Ulaz predstavljaju četiri systemska brojača, a izlaz konačno opterećenje računara izvršioča.

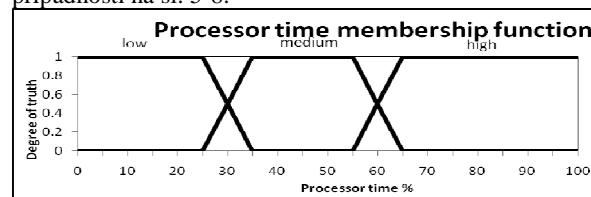
Processor time % predstavlja procenat zauzetosti procesora u jedinici vremena. Najjasnije predstavlja rad procesora. Granična vrijednost opterećenja je 85%.

Processor Queue Length je broj programskih niti koje čekaju u redu na procesorsku obradu. Konstantna srednja vrijednost veća od dva predstavlja povećano opterećenje.

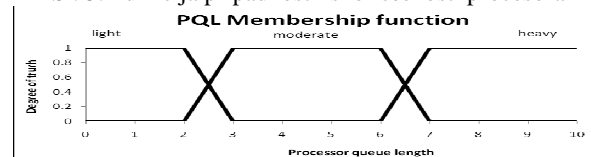
RAM used predstavlja procenat iskorišćene radne memorije. Konstantna vrijednost veća od 75% predstavlja stanje nedovoljno raspoložive memorije.

Context switches/sec je broj preključivanja konteksta programskih niti u jedinici vremena. Vrijednosti manje od 5000 nisu od značaja, dok one preko 15000 predstavljaju kritičnu granicu, odnosno situaciju da veći broj niti sa istim prioritetom čeka na izvršenje.

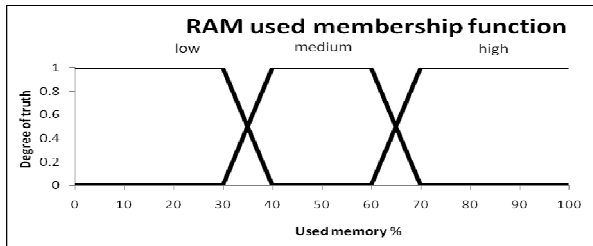
Postupak rasplinjavanja prvog nivoa koristi funkcije pripadnosti na sl. 5-8.



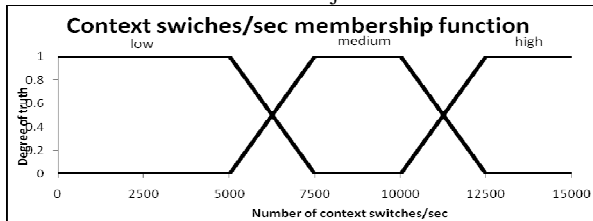
Sl. 5. Funkcija pripadnosti iskorišćenosti procesora



Sl. 6. Funkcija pripadnosti reda čekanja niti na procesorsko vrijeme



Sl. 7. Funkcija pripadnosti procenta upotrebljene RAM memorije



Sl. 8. Funkcija pripadnosti broja preključivanja konteksta niti u procesoru u jedinici vremena

U postupku rasplinjavanja se formiraju vrijednosti lingvističkih promjenljivih u zavisnosti od ulaznih promjenljivih. Vrijednosti lingvističkih promjenljivih predstavljaju stepene istinitosti određenih stanja lingvističke promjenljive. Npr. za procenat iskorišćenja procesorskog vremena 27% - vrijednost *low* odgovarajuće lingvističke promjenljive je 0.8 (stepen istinitosti), a vrijednost *medium* je 0.2. Drugim riječima, procesor je nisko iskorišćen 27*0.8=21.6%, a 27*0.2=5.4% srednje iskorišćen.

Skup pravila se formira za svaki kontroler. Tabele 1. i 2. pokazuju skup pravila kontrolera prvog nivoa.

TABELA 1: Skup pravila kontrolera za dobijanje opterećenja procesora

ProcessorTime %\PQL	L	M	H
L	VL	L	M
M	L	M	H
H	M	H	VH

TABELA 2: Skup pravila kontrolera za dobijanje opterećenja sistema

RAM \Swiches/sec	L	M	H
L	VL	L	M
M	L	M	H
H	M	H	VH

Pravilo je npr. u sledećem obliku:

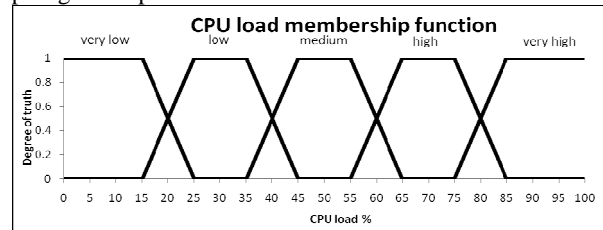
Ako *processor_time* = *M* **i** *pql* = *L* **onda** *cpu_load* = *L*.

Samo ona pravila koja imaju vrijednosti čiji je stepen istinitosti veći od nule učestvuju u daljem postupku.

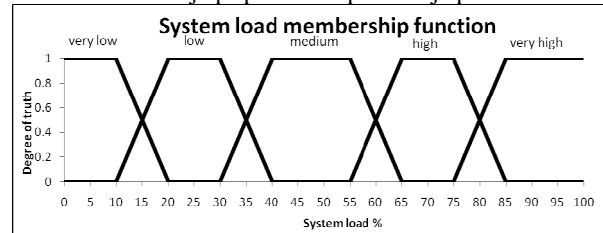
Mehanizam zaključivanja koristi odabrana pravila da formira skup vrijednosti izlaznih lingvističkih promjenljivih. Koristi mehanizam minimuma pri formiranju izlaznih vrijednosti koje predstavljaju stepene istinitosti vrijednosti izlaznih lingvističkih promjenljivih.

Funkcije pripadnosti izlaznih promjenljivih su potrebne za postupak pretvaranja izlaznih lingvističkih promjenljivih u izlaznu promjenljivu kontrolera. Funkcije

pripadnosti izlaznih promjenljivih sistema kontrolera prvog nivoa prikazane su na sl. 9-10.



Sl. 9. Funkcija pripadnosti opterećenja procesora



Sl. 10. Funkcija pripadnosti opterećenja sistema

U sistemu kontrolera se koristi metoda pretvaranja rezultata „centar gravitacije“, koja koristi formulu

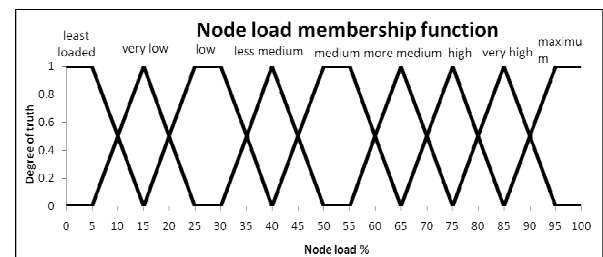
$$u = \frac{\sum_i b_i \int \mu_i(x)}{\sum_i \int \mu_i(x)}$$

da dobije konačan izlaz iz kontrolera. Pri tome, b_i određuje centar izlazne funkcije pripadnosti pravila i , a $\int \mu_i(x)$ predstavlja površinu koju obrazuju funkcija pripadnosti μ_i određenog pravila i i prava paralelna sa apscisom, koja ima vrijednost jednaku stepenu istinitosti funkcije pripadnosti za dato pravilo i , dok u predstavlja izlaz iz kontrolera.

Postupak rasplinjavanja drugog nivoa koristi funkcije pripadnosti sa sl. 9. i 10, skup pravila iz tabele 3. i funkciju pripadnosti izlazne promjenljive sa sl. 11. za dobijanje izlaza iz sistema kontrolera.

TABELA 3: Skup pravila kontrolera za dobijanje opterećenja izvršioca

CPU\System	VL	L	M	H	VH
VL	LL	VL	L	LM	M
L	VL	L	LM	M	MM
M	L	LM	M	MM	H
H	LM	M	MM	H	VH
VH	M	MM	H	VH	ML

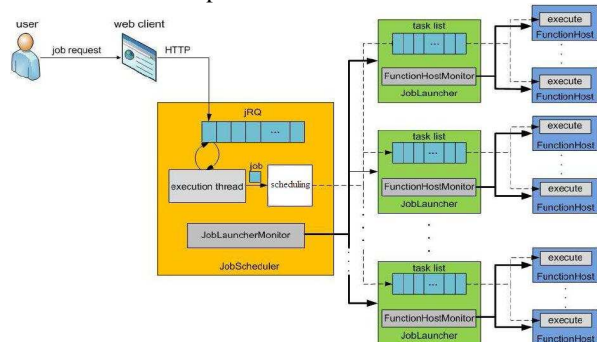


Sl. 11. Funkcija pripadnosti opterećenja izvršioca

Pošto se sistem kontrolera rasplinite logike nalazi na svakom izvršiocu, potrebno je na strani raspoređivača osvježavati vrijednosti opterećenja svih izvršilaca. Da bi se izbjegla nepotrebna komunikacija, izvršilac šalje raspoređivaču svoje opterećenje samo u trenucima promjene vrijednosti.

V. SISTEM CE

Sistem CE predstavlja referentni sistem za obradu aplikativnih proračuna koji se koristi u DMS-u (*Distribution Management System*). Sastoji se od komponenti: *JobScheduler* – prima zahtjeve od korisnika i raspoređuje ih jednom od pokretača; *JobLauncher* – prima zahtjeve od raspoređivača i prosljeđuje ih nekom od slobodnih izvršilaca; *FunctionHost* – obrađuje zahtjeve i vraća rezultate komponenti *JobLauncher*.

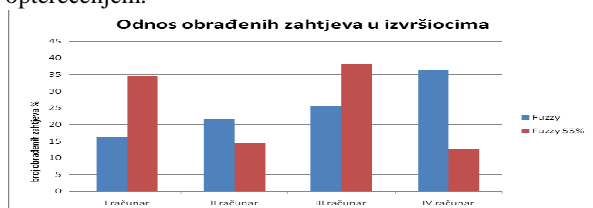


Sl. 12. Princip sistema CE sa distribuiranom obradom zadataka

VI. ZAKLJUČAK

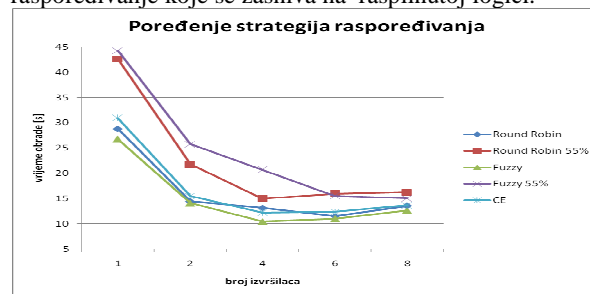
Funkcionalnost rešenja je pokazana na distribuiranom sistemu od deset računara. Jedan predstavlja klijentsku aplikaciju, jedan raspoređivač i 1, 2, 4, 6 ili 8 izvršilaca. Vrijeme kašnjenja između slanja zahtjeva je promjenljivo: 0, 100, 200 i 300 milisekundi. Podaci koji se obrađuju sastoje se od 55 korijena elektroenergetske šeme Beograda. Raspoređivanje zadataka izvršiocima se obavlja na dva načina: redom u krug i prema najmanjem opterećenju, dobijenom na osnovu rasplinite logike. Radi jasnijeg uočavanja principa raspoređivanja, koristeći rasplintu logiku, u pojedinim izvršiocima se formira dodatno opterećenje u vidu izvršenja programa koji uzima 90% procesorskog vremena, tako da je opterećenje izvršioaca 55-60%. Izvršiocci sa parnim brojem su podvrgnuti dodatnom opterećenju. Pamti se vrijeme obrade svih zadataka, od slanja prvog, do primanja rezultata posljednjeg i izraženo je u sekundama.

Na sl. 13. je prikazan odnos broja obrađenih zadataka, koristeći četiri izvršioaca, bez kašnjenja između slanja. U slučaju bez dodatnog opterećenja (svi imaju približno isto opterećenje) više zadataka obrađuju računari sa boljiim karakteristikama, odnosno oni sa većom učestanošću rada procesora. Sa dodatnim opterećenjem, se jasno vidi princip raspoređivanja zadataka ka onima sa manjim opterećenjem.



Sl. 13. Odnos broja obrađenih zadataka u slučaju korišćenja četiri izvršioaca i rasplinite logike

Međusobno su upoređena vremena obrade za različite strategije, i sa sistemom CE. (sl. 14) Najbolje rezultate ima raspoređivanje koje se zasniva na rasplinitoj logici.



Sl. 14. Poređenje strategija raspoređivanja u zavisnosti od broja izvršilaca

Buduća istraživanja rada su u smjeru biranja dodatnih elemenata na osnovu kojih se obavlja raspoređivanje (informacije o stanju računarske mreže, propusnosti i opterećenja ka određenim čvorovima), u cilju što brže obrade.

LITERATURA

- [1] M. Popović, I. Bašičević, V. Vrtunski, „A Task Tree Executor: New Runtime for Parallelized Legacy Software“, Engineering of Computer Based System Conference (ECBS), San Francisco, USA, April 2009.
- [2] Scott Klein, „Professional WCF programming: .NET development with the Windows communication foundation“, Indianapolis, 2007.
- [3] Abbas Karimi, Faraneh Zarafshan, Adzman Jantan, A. R. Ramli, M. Iqbal Saripan, „A New Fuzzy Approach for Dynamic Load Balancing Algorithm“, Department of Computer Systems Engineering, Faculty of Engineering, UPM, Malaysia, Computer Department, IAU, Arak, Iran, 2009.
- [4] Domenico Ferrari, Songnian Zhou, „An Empirical Investigation of Load Indices for Load Balancing Algorithm“, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.
- [5] Kevin M. Passino, Stephen Yurkovich, „Fuzzy Control“, Department of Electrical Engineering, The Ohio State University, 1998.
- [6] B. Trivunović, M. Popović, V. Vrtunski, „An Application Level Parallelization of Complex Real-Time Software“, 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS), UK, 2010.
- [7] I. Bašičević, S. Jovanović, B. Drapšin, M. Popović, V. Vrtunski, „An Approach to Parallelization of Legacy Software“, IEEE ECBS-EERC, Novi Sad, Serbia, Sept. 2009.

ABSTRACT

Need for distributed systems is wide-spread in information technology. It is implemented in power distribution networks. This paper presents distributed processing solution based on fuzzy logic. Main goal is to reduce task execution time with respect to existing DMS solution. Task execution times are compared for various number of computers in distributed system. Solution from this paper shows better performances than existing DMS solution and solution with Round Robin scheduling strategy. It can be used in distributed systems, where data processing significantly depends on computer performances.

DISTRIBUTED PROCESSING SOLUTION BASED ON FUZZY LOGIC

Milen Miodanić

ć