UDC: 004.4 INFO M: str. 46-50

UTICAJ UPOTREBE VIŠESTRUKIH MREŽNIH VEZA MPI KLASTERA NA BRZINU IZVRŠAVANJA PROGRAMA ZA ANALIZU KONSTRUKCIJA METODOM KONAČNIH TRAKA
INFLUENCE OF USING MULTIPLE MPI CLUSTER NETWORK LINKS TO THE EXECUTION SPEED OF THE FINITE STRIP METHOD CONSTRUCTION ANALYSIS PROGRAM

Lazar Stričević, Predrag Rakić, Miroslav Hajduković

REZIME: Komunikacione veze među nodovima MPI klastera ponekad predstavljaju usko grlo u njegovom radu. Ovaj problem se manje ili više ublažava ubrzavanjem komunikacionih kanala, a jedan od načina da se to postigne je upotreba višestrukih mrežnih linkova. U ovom radu se porede vremena izvršavanja programa za analizu građavinskih konstrukcija metodom konačnih traka na MPI klasteru uz upotrebu jednog, dva zasebna i dva agregirana mrežna interfejsa po računaru, pri čemu se variraju parametri kao što su broj nodova, vrsta ulaza i vrsta algoritma. Analiza, koja je data na kraju, pokazuje prednosti i mane upotrebe višestrukih mrežnih veza, kada je opravdano njihovo korišćenje i kada i pod kojim uslovima to daje najbolje rezultate.

KLJUČNE REČI: MPI klaster, agregacija mrežnih linkova, HPC, Open MPI, Metod konačnih traka, FSM

ABSTRACT: Communication links between the nodes of the MPI cluster sometimes present a bottleneck in its functioning. This problem is more or less alleviated by speeding up the network links and one of the ways to achieve this is the use of multiple network links. This paper compares finite strip method construction analysis program execution times on MPI cluster when nodes use single network interface, two network interfaces and two aggregated network interfaces, varying the number of nodes types of input and the types of algorithms. Analysis, which is provided at the end, shows advantages and faults of using multiple network links, when is it suitable to use them and when and under which conditions it gives the best results.

KEY WORDS: MPI cluster, link aggregation, HPC, Open MPI, Finite Strip Method, FSM

1 UVOD

Savremeni računari nam omogućavaju da rešavamo veće probleme i da nalazimo rešenja sve brže, sa većom preciznošću i po nižoj ceni. Kada brzina računara nije dovoljna za određenu namenu, pristupa se jednom od tri moguća rešenja: optimizacija algoritma, kupovina bržeg računara i podela posla na više računara (paralelizacija programa)[1]. Ovaj rad se bavi trećim slučajem, tačnije načinima za poboljšanje performansi računskog klastera (*computing cluster*) korišćenog za izvršavanje programa baziranih na metodi konačnih traka. Klaster je vrsta paralelnog ili distribuiranog sistema koji se sastoji od skupa međusobno povezanih računara i koji se koristi kao jedinstveni računarski resurs. [2]

Tema ovog rada je ispitivanje uticaja povećanja brzine mrežne komunikacije među računarima koji čine računski klaster, odnosno među njegovim *nodovima*. Kada se govori o vremenu potrebnom da se prenese informacija između nodova, razlikujemo vreme koje informacija provede u mrežnom hardveru, koje se zove mrežna latentnost (*latency*), i vreme koje procesor troši da informaciju pripremi za slanje, pošalje i primi, koje se zove overhed (*overhead*) [3]. Mrežna latentnost može da se potencijalno preklopi sa računanjem, za razliku od overheda.

Vreme mrežne latentnosti može da se smanji povećanjem propusnog opsega, tako što se pređe na upotrebu bržeg mrežnog interfejsa ili se za komunikaciju koristi više interfejsa.

Ovaj rad se bavi načinima za skraćenje vremena mrežne latentnosti korišćenjem više interfejsa. Cilj je da se ispita da li je moguće na ovaj način skratiti vreme izvršavanja numeričkog postupka harmonijski spregnute metode konačnih traka (*Harmonic Coupled Finite Strip Method - HCFSM*) koja se upotrebljava za rešavanje brojnih problema u mehanici, uglavnom vezanih za pločaste konstrukcije [4].

2 HARMONIJSKI SPREGNUT METOD KONAČNIH TRAKA

Metod konačnih traka (FSM) je izveden iz metoda konačnih elemenata (*Finite Element Method* - FEM). FSM je razvijen kao specijalizacija FEM i široko se koristi za analizu čeličnih konstrukcija. Originalna (nespregnuta) formulacija data u [5tse koristi u linearnoj analizi konstrukcija i oslanja se na ortogonalnost harmonijskih funkcija u formulaciji matrice krutosti. Nasuprot ovoj formulaciji, kada se matrice krutosti formiraju korišćenjem triginometrijskih funkcija sa eksponentima višeg reda, govorimo o harmonijski spregnutom metodu konačnih traka (HCFSM). Ovaj metod je pogodan za geometrijsku, nelinearnu analizu pločastih konstrukcija.

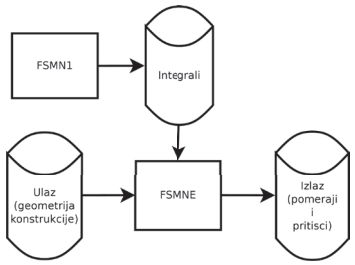
U ovom radu korišćen programski sistem koji realizuje HCFSM je projektovan i implementiran kroz dva izvršna programa [6], nazvana FSMN1 i FSMNE.

Program FSMNE nalazi rešenja jednačine stabilnosti. Kao ulazne parametre, ovaj program prihvata opis razmatrane konstrukcije zajedno sa planiranom raspodelom opterećenja, a kao rezultat daje vektore pomeraja i vektore unutrašnjih sila u zadatim tačkama konstrukcije.

Konstrukcija se tereti inkrementalno, a rešenje za svaki inkrement opterećenja se dobija korišćenjem iterativnog postupka. Svaka iteracija zahteva rešavanje sistema jednačina stabilnosti, čije rešenje zavisi od rešenja izračunatih u prethodnoj iteraciji. Za rad ovog programa su neophodne vrednosti integrala, prethodno izračunate, programom FSMN1. Arhitektura ovog programskog sistema je prikazana na slici 1.

Vrednosti integrala se koriste za računanje podužnih (duž traka) karakteristika konstrukcije. Za svaki harmonik, čiji se doprinos uzima u razmatranje, računa se odgovarajući broj vrednosti. Ove vrednosti predstavljaju uticaj tog harmonika na naponsko stanje trake i koriste se pri računanju matrice krutosti svake trake. Pri izračunavanje karakteristika konstrukcije se uzima u razmatranje uticaj jednog ili više harmonika. Ako se koristi više harmonika, koriste se svi harmonici od prvog, pa do nekog zadatog hamonika koji je predstavljen promenljivom NTERM. Vrednosti integrala izračunate programom FSMN1 su nezavisne od karakteristika konstrukcije za koju se koriste. Proces računanja je normalizovan - vrednosti se nalaze u granicama od 0 do 1, tako da su nezavisni od dužine (traka)

konstrukcije. Broj potrebnih vrednosti integrala je u opštem slučaju velik i zavisi od preciznosti računanja, tj. od broja harmonika koje je potrebno uključiti u analizu [4]. Broj potrebnih vrednosti integrala raste proporcionalno broju harmonika na četvrti stepen [7].



Slika 1. – *Arhitektura HCFSM programskog sistema*

Tokom pripreme konstrukcije za obradu FSMNE programom, identifikuju se karakteristične geometrijske tačke koje se nazivaju čvornim. Kroz ove tačke se provlače (zamišljene) čvorne linije, a zatim se konstrukcija dekomponuje na "trake" oivičene čvornim linijama. Ove trake su osnovni elementi metoda. Sve informacije o ponašanju konstrukcije, dobijaju se kombinacijom uticaja izračunatih za pojedine trake. Tokom izvršavanja programa se računaju matrice krutosti za svaku od traka, da bi se od njih "sklopila" matrica krutosti cele konstrukcije.

FSMNE podržava 3 numeričke metode za dobijanje rezultata: linearna metoda, metoda Fon Karmana i Lagranžova metoda. Linearna metoda je algoritamski najmanje kompleksna, dok je Lagranžova najkompleksnija. Sve tri metode prihvataju iste ulazne podatke i daju izlaze istog formata (imaju jednak broj podataka na izlazu), ali se broj računskih operacija koje je potrebno izvršiti za svaki od ovih metoda drastično razlikuje. Ova mogućnost jednostavne promene broja izvršenih numeričkih operacija je u ovom radu iskorišćena za analizu uticaja povećanja broja procesorskih operacija, bez ikakve promene količine prenetih podataka između nodova klastera.

Kao što je rečeno, određivanje karakteristika konstrukcije pod opterećenjem zahteva računanje matrice krutosti za svaku od traka. Pošto su računanja matrica krutosti pojedinih traka nezavisna, moguće ih je podeliti po različitim računarima, koji predstavljaju različite nodove klastera. Cilj paralelizacije je smanjenje vremena potrebnog za dobijanje rešenja odnosno ubrzavanje postupka rešavanja. Pošto se najveći deo vremena izvršavanja programa troši baš na računanje matrica krutosti pojedinih traka, paralelizacija ovog dela program daje najbolje rezultate.

Za računanje matrice krutosti svake pojedinačne trake se koriste sve vrednosti integrala izračunate za zadati broj harmonika. Veliki broj vrednosti integrala potrebnih za ovo računanje je jedan od razloga dugotrajnosti ovog dela programa. Pošto se matrice krutosti traka računaju na svim nodovima klastera, poželjno je vrednosti integrala (prethodno izračunate programom FSMN1) prebaciti na lokalni disk svakog noda, jer se na taj način drastično smanjuje komunikacija (količina prenesenih podataka kroz mrežu) između nodova tokom rada FSMNE programa.

3 MPI KLASTER

MPI je akronim engleskih reči *Message Passing Interface* [8]. To je standardizovan i prenosiv (između platformi) interfejs za razmenu poruka. MPI je razvijen prvenstveno imajući u vidu paralelni programski model baziran na razmeni poruka. U ovom modelu se podaci intenzivno prenose iz adresnog prostora jednog procesa u adresni prostor drugih procesa.

Standard [9] definiše sintaksu i semantiku jezgra komunikacione biblioteke korisne različitim kategorijama programera koji pišu prenosive paralelne programe bazirane na razmeni poruka. MPI interfejs je definisan za programske jezike Fortran, C i C++ mada sada postoje biblioteke i za druge programske jezike.

Privlačnost razmene poruka kao paradigme za komunikaciju između procesa, dobrim delom, leži u lakoj portabilnosti. Paralelni program napisan na ovaj način može, bez (značajnih) izmena, biti izvršavana na multiprocesorima sa distribuiranom memorijom (npr. superračunari), na mrežama radnih stanica (tj. na klasterima) ili na kombinaciji oba [9].

Računski klaster (*computing cluster*) je hardverska platforma upotrebljena za dobijanje rezultata opisanih u ovom radu. Naš klaster se sastoji od 5 PC računara, povezanih *Ethernet* 100BASET mrežom [10]. Za komunikaciju između nodova navedenog klastera se koristi implementacija MPI standarda pod nazivom *Open MPI* [11]. Ova implementacija trenutno podržava veliki broj komunikacionih protokola, kao što su deljena memorija, *Infiniband*, *Myranet*, ali i *Ethernet* u kombinaciji sa TCP/IP protokolom [12, 13]. *Ethernet* i TCP/IP su u praksi najviše pristupačni i najčešće upotrebljavani protokoli [14], tako da će biti korišćeni i u ovom eksperimentu. To znači da će svaki od interfejsa imati svoju fizičku, odnosno MAC adresu, a mrežna (IP) adresa će biti dodeljena pojedinačnom ili grupi interfejsa, u zavisnosti od konfiguracije.

Za arhitekturu našeg MPI klastera je odabrana master-slejv arhitektura, koja podrazumeva da jedan od nodova preuzme ulogu mastera (vođe), koji svim ostalim nodovima (slejvovima) šalje zadatke, a na kraju preuzme rezultate. Zadatak je u ovom slučaju računanje matrice krutosti jedne trake. Svaki nod računa matricu krutosti jedne (ili više, obično ukupan_broj_traka/broj_nodova) trake. Na osnovu matrica krutosti traka, formira se matrica krutosti sistema. Zato master nod mora da prikupi matrice krutosti svih traka (tj. sve delove rezultujuće matrice) izračunate u tekućoj iteraciji, pre nego što pristupi sledećoj iteraciji.

4 AGREGACIJA LINKOVA (MREŽNIH VEZA)

Agregacija linkova predstavlja kombinovanje dva ili više fizičkih mrežnih interfejsa u jedan logički, sa ciljem povećanja raspoloživog propusnog opsega (*load balancing*), kao i da se obezbedi nastavak komunikacije u slučaju da jedna od mrežnih veza otkaže (*fault tolerance*). U zavisnosti koji od ovih zahteva je bitniji, bira se konkretan način agregacije, od kojih su neki standardizovani (na primer standard 802.1AX-2008 koji je poznatiji pod svojim starim imenom 802.3ad [15]), a neki ne [16]. Budući da nodovi klastera koriste 100BASET *Ethernet* mrežnu tehnologiju i *GNU/Linux* operativni sistem, na raspolaganju nam stoje režimi agregacije koje ovaj sistem podržava [17]:

1. "*balance-rr*": (*round robin*) Ovaj način agregacije podrazumeva da se prilikom slanja paketi podataka sekvencijalno (kružno) raspoređuju od prvog do poslednjeg raspoloživog mrežnog interfejsa. Na primer, ako treba preneti *n* paketa preko dva na ovaj način agregirana interfejsa, onda će prvi paket ići preko prvog, drugi preko drugog interfejsa, treći opet preko prvog, četvrti preko drugog, i tako do poslednjeg paketa.
2. "*active-backup*": Samo jedan od agregiranih interfejsa je aktivan, a u slučaju njegovog otkaza aktivira se neki od rezervnih.

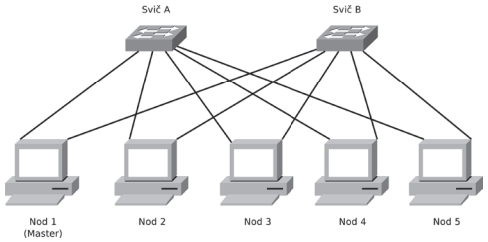
3. "*balance-xor*": Paket se šalje na link određen C formulom [(izvorni_MAC ^ odredisni_MAC) % broj_slejvova]
4. "*broadcast*": Svaki paket se šalje na sve raspoložive interfejse.
5. "*802.3ad*": Dinamička agregacija u skladu sa standardom 802.3ad [15]
6. "*balance-tlb*": (*transmit load balancing*) Odlazni saobraćaj se raspoređuje na interfejse shodno njihovom opterećenju. Kada se pojavi novi komunikacioni partner (*peer*), sav njemu upućen saobraćaj ide na interfejs koji je do tada bio najmanje iskorišćen.
7. "*balance-alb*": (*adaptive load balancing*) Obuhvata sve osobine prethodnog režima, s tim da se radi i raspoređivanje dolaznog saobraćaja pomoću ARP [18] poruka.

Agregacija podrazumeva kreiranje jednog logičkog interfejsa kome se dodeljuje mrežna adresa (IP) i fizička (MAC) adresa. Svi fizički interfejsi koji su agregirani, dobijaju fizičku adresu logičkog interfejsa.

Budući da je tema ovog rada je ubrzavanje mrežne komunikacije među nodovima klastera, u obzir za izbor načina agregacije su uzeti režimi koji omogućavaju iskorišćavanje propusnog opsega više interfejsa ("*load balance*"). Većina ovakvih režima raspoređuje saobraćaj na osnovu fizičke (MAC) adrese odredišta, što za veliki broj komunikacionih nodova daje dobre prosečne rezultate. Nažalost, ovo znači da bi komunikacija između dve mašine bila ograničena brzinom jednog komunikacionog kanala. Samo jedan od režima omogućava da raspoloživi komunikacioni opseg između dve mašine pređe ovu granicu, a to je "*balance-rr*" režim [19]. Ovo je razlog što je pomenuti režim agregacije izabran za upotrebu tokom eksperimenta.

5 EKSPERIMENT

Vršena su merenja vremena izvršavanja paralelizovanog FSMNE dela aplikacije na *Open MPI* klasteru. Pošto konfiguracija klastera obuhvata 5 nodova, povezanih 100BASET mrežom, za eksperiment je izabran primer računanja rešenja jednačina stablinosti za strukturu krova, opisanu u [7], koja je za ovu priliku izdeljena na 5 traka. U idealnom slučaju, kada se koriste svi raspoloživi nodovi klastera, svaki od nodova bi trebao da računa matricu jedne trake. U slučaju da je struktura podeljena na više traka nego što u klasteru postoji nodova, trake se u što je moguće ravnomernijem odnosu raspoređuju na nodove klastera koji su u upotrebi.



Slika 2. – *Šema korišćenog klastera.*

5.1 *Nodovi klastera*

Nodovi korišćeni u eksperimentu rade pod operativnim sistemom GNU/*Linux* (korišćeni su programski paketi GNU/*Linux* distribucije *Debian* 5.03). Korišćeno je ukupno 5 računara, koji su svi hardverski identični. Svaki od njih ima AMD *Sempron* 2500+ procesor (1400 MHz i 256k L2 keš memorije), matičnu ploču *Gigabyte* GA-K8VM800M sa integrisanim video, zvučnim i mrežnim podsistemom (*Realtek* 8100C), 512MB RAM-a (DDR-400), 80GB HDD (*Maxtor* 6Y080L0). Na ovu konfiguraciju je dodata PCI kartica takođe sa *Realtek* mrežnim interfejsom. Za njihovo povezivanje su upotrebljena dva identična sviča (8-portni *TP-Link* TL-SF1008D).

Ovo znači da svaki od nodova ima po dva 100BASET mrežna interfejsa. Prvi je povezan na mrežni svič A, a drugi na mrežni svič B. Na ovaj način su dobijene dve fizičke mreže. (slika 2).

Eksperiment treba da pokaže u kojoj meri brzina računarske mreže tj. komunikacionog kanala koji povezuje nodove klastera, utiče na vreme potrebno da se dobije rezultat. Drugim rečima, kako se menja vreme izvršavanja programa, ako povećamo brzinu mreže. Brzina mreže klastera se menja tako što se menja broj mrežnih interfejsa koje klaster upotrebljava. Kao što je već rečeno, u sve nodove ugrađen dodatni 100BASET mrežni interfejs, koja radi istom brzinom kao i postojeći interfejs na matičnoj ploči. Poređeno je vreme izvršavanja programa bez korišćenja dodatnog interfejsa sa vremenima potrebnim da se program izvrši na klaster konfiguracijama sa 2 mrežna interfejsa.

Postoji više načina da se 2 ili više interfejsa upotrebi u realizaciji klastera. *Open MPI* biblioteka ima mogućnost da se konfiguriše tako da samostalno za komunikaciju upotrebljava sve raspoložive mrežne interfejse. U ovoj konfiguraciji svaki od interfejsa ima svoju fizičku i svoju mrežnu adresu. Aplikacija je pokrenuta u ovoj konfiguraciji i mereno je vreme njenog izvršavanja.

Kao što je navedeno u poglavlju o agregaciji, GNU/*Linux* operativni sistem može da se konfiguriše tako da agregira više interfejsa i predstavi ga aplikaciji kao jedan. U tom slučaju aplikacija ima utisak da koristi jedan interfejs većeg propusnog opsega. Da bi se dobio veći propusni opseg, konfigurisana je agregacija po "*balance-rr*" algoritmu i mereno je vreme izvršavanja aplikacije.

To znači da je svako merenje vršeno na sledećim konfiguracijama:

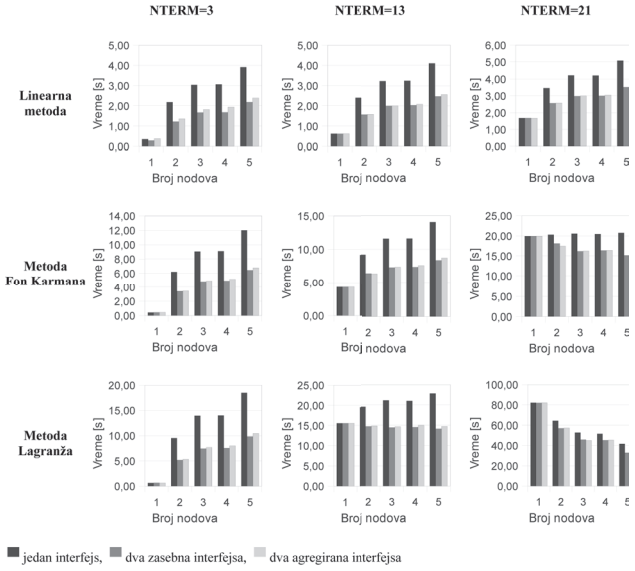
1. *Open MPI* biblioteka je konfigurisana da koristi samo jedan mrežni interfejs
2. *Open MPI* biblioteka je konfigurisana da koristi oba interfejsa
3. dva interfejsa su agregirana po "*balance-rr*" algoritmu (a *Open MPI* biblioteka koristi novonastali agregirani interfejs)

Prva mrežna konfiguracija koja je korišćena zapravo predstavlja uobičajen način povezivanja nodova klastera pomoću računarske mreže. Svaki od nodova ima po jedan 100BASET mrežni interfejs koji je spojen na mrežni svič. Time je omogućena komunikacija maksimalnom brzinom između bilo koja dva pojedinačna noda.

Pre početka eksperimenta izmerena je maksimalna brzina komunikacije na svakom od interfejsa. Ova merenja su vršena test programom *iperf* [20]. Utvrđeno je da je maksimalna brzina komunikacije u jednom smeru na svim interfejsima identična i da iznosi 94.2MBit/s.U drugoj konfiguraciji mreže svaki od nodova ima po dva 100BASET mrežna interfejsa. Funkcije *Open MPI* biblioteke su podešene tako da za komunikaciju među nodovima koriste oba mrežna interfejsa. Ovo se postiže time što se TCP konekcije, koje openMPI biblioteka uspostavlja, raspoređuju po interfejsima, tako da je ukupan raspoloživi propusni opseg što bolje iskorišćen.

U trećem slučaju nodovi takođe imaju po dva 100BASET mrežna interfejsa, ali su oni agregirani na nivou operativnog sistema, pa korisnički programi vide samo jedan agregirani interfejs. Svi fizički

interfejsi dobijaju istu fizičku (MAC), a samo agregirani interfejs dobija mrežnu (IP) adresu. Postoji više načina kako se saobraćaj koji ide na agregirani interfejs može rasporediti po fizičkim interfejsima. Za ovu priliku je izabran tzv. "*balance-rr*" algoritam, koji podrazumeva da se prilikom slanja paketi podataka sekvencijalno raspoređuju na sve raspoložive mrežne interfejse. Na ovaj način se ravnomerno koristi sav na ovaj način dostupan propusni opseg. Kao što je ranije rečeno, razlog za izbor ovog algoritma je to što on omogućava da i pojedinačna TCP konekcija iskoristi propusni opseg svih raspoloživih interfejsa i ostvari ubrzanje [21].



Slika 3. – *Grafikoni sa rezultatima merenja vremena izvršavanja programa*

Pre izvođenja eksperimenta izvršeno je merenje maksimalne brzine komunikacije agregiranih interfejsa programom *iperf*, kao i u prethodnom slučaju. Dobijena je brziina od 187 Mbit/s, čime je pokazano da je maksimalna brzina komunikacije među nodovima gotovo udvostručena.

Da bi se dobio utisak o uticaju promene mrežne konfiguracije na klaster, poređena su merenja vremena izvršavanja FSMNE programa na sve tri konfiguracije u pri čemu je variran broj nodova (korišćen je klaster sastavljen od 1, 2, 3, 4 i 5 nodova) i kompleksnost algorima (broj procesorskih instrukcija koje treba obaviti za istu količinu komunikacije). Kompleksnost algoritma je menjana izmenom vrste numeričke metode kojom FSMNE dobija rezultat (linearna metoda, von Karman i Lagranž), kao i izmenom preciznosti računanja, odnosno broja NTERM-ova (rađena su merenja za NTERM vrednosti 3, 13, 21).

Eksperiment je potom pokretan na 1, 2, 3, 4 i 5 nodova na novodobijenoj konfiguraciji. Svako merenje je ponovljeno 3 puta, a aritmetička sredina dobijenih rezultata je je uzeta kao konačan rezultat. Svi konačni rezultati su prikazani na slici 3.

6 ANALIZA REZULTATA

Rezultati merenja vremena su prikazani grafikonima na slici 3. Svaki od grafikona prikazuje trajanje eksperimenta za dati metod rešavanja (linearni, fon Karman ili Lagranž) i datu kompleksnost ulaza (NTERM je jednak 3, 13 ili 21). Na svakom grafikonu je prikazano trajanje eksperimenta za sve tri mrežne konfiguracije (jedan interfejs, dva zasebna, dva agregirana) u funkciji broja korišćenih nodova u klasteru.

Utvrđeno je da se ubacivanjem dodatnog interfejsa u upotrebu vreme računanja rešenja skraćuje u skoro svim slučajevima. Vremenskog skraćenja nema jedino u slučaju kada se koristi samo jedan računar, što je logično jer se u tom slučaju mrežni interfejsi ne koriste, pošto nema potrebe za komunikacijom sa ostalim nodovima klastera.

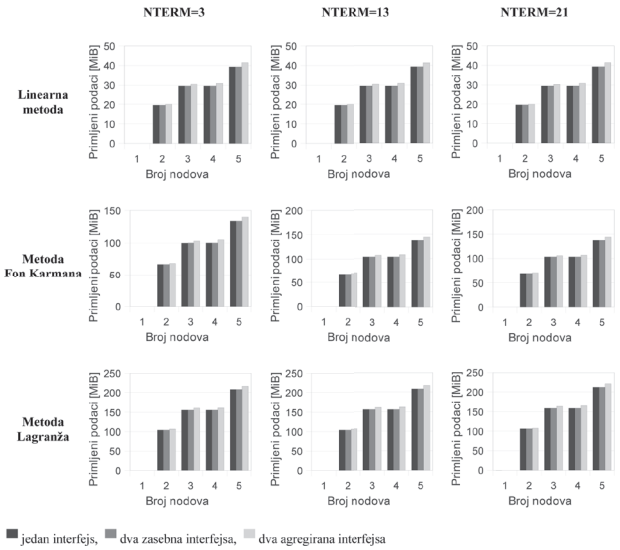
Dobitak u vremenu je procentualno najveći u slučaju kada je kompleksnost računa najmanja - linearna metoda za NTERM=3. Ovo se objašnjava time što je u ovom slučaju vremenski udeo komunikacije u ukupnom trajanju računanja najveći, pa je samim tim najveći i dobitak koji se postiže ubrzavanjem komunikacije.

Primećuje se da do ubrzavanja procesa računanja sa povećanjem broja uključenih nodova dolazi tek kada se radi o kompleksnijem računu na primer metoda Lagranža za NTERM=21. Skraćenje vremenskog perioda pomoću dodavanja dodatnog mrežnog interfejsa je ovde bilo najmanje, što se objašnjava činjenicom da je ovde udeo komunikacije u vremenskom periodu računanja najmanji.

Moguće je uočiti da u pojedinim slučajevima, kao što su računanje po Lagranžovom metodu za 13 harmonika i po fon Karmanu za 21 harmonik ubrzanje mreže dovodi do toga da dođe do pojave ubrzanja sa povećanjem broja nodova.

6.1 *Razlika u rezultatima između konfiguracija sa agregiranim i zasebnim interfejsima*

Primećeno je da je u većini slučajeva vreme potrebno za izvođenje eksperimenta sa agregiranim interfejsom nešto duže od vremena kada se eksperiment izvršava sa dva samostalna interfejsa. Razlog za ovu pojavu je mogućnost pojave "isporuke preko reda" (*out of order delivery*) paketa TCP konekcije, koja može da se javi kod *balance-rr* načina agregacije. [17] Kada se to desi, TCP protokol to može da vidi kao neisporučen paket i da inicira njegovo ponovno slanje. Na ovaj način se pojedini paketi prenose dva ili više puta, što za posledicu ima neznatno usporenu komunikaciju.



Slika 4. – *Grafikoni sa ukupnom količinom podataka prenetih sa i do master noda*

Ova tvrdnja je potvrđena merenjem količine prenetih podataka tokom trajanja eksperimenta, čiji su rezultati dati na slici 4. Utvrđeno je da je količina podataka agregiranog interfejsa nešto veća od količine podataka prenetih preko dva samostalna interfej-

sa, kao i da su za vezu između master noda i svakog pojedinačnog slejva korišćene po 4 TCP konekcije.

Vremenska razlika je u najgorem slučaju (linearno, NTERM=3) bila oko 15% (0,25 s), ali u većini ostalih eksperimenata je bila manja od 5%, dok je za eksperimente koji su trajali duže od 100 sekundi razlika bila manja od 1%. To znači da je ova razlika u većini razmatranja nema veliki značaj, ali i da postoje slučajevi kada bi navedeno trebalo imati na umu.

7 ZAKLJUČAK

Eksperimentom je pokazano da u svakom od slučajeva upotrebe klastera, ubrzanje mreže donosi ubrzanje računanja rešenja FSMNE. Rezultati eksperimenta su analizirani i zaključeno je da pošto se ubrzava samo mreža, stepen ubrzanja programa koji se time postiže, zavisi pre svega od udela u kom se vreme troši na komunikaciju među nodovima, u ukupnom vremenu potrebnom da se dođe do rešenja. Dobra strana upotrebe više interfejsa u komunikaciji je skalabilnost, odnosno propusni opseg se može dozirati u meri u kojoj je potreban. Ukoliko je potrebno više propusnog opsega, u konfiguraciju se doda potreban broj interfejsa koji se povežu u mrežu. Mana ovakvog pristupa je da se tada mora voditi računa o raspoređivanju saobraćaja po raspoloživim interfejsima, što unosi dodatni overhed, bilo da se time bavi aplikacija (MPI biblioteka) ili operativni sistem (agregacija). Pokazalo se da je aplikacija nešto bolja u raspoređivanju od operativnog sistema, iako je ta prednost vrlo mala. Ipak, kada je vreme utrošeno na komunikaciju u odnosu na ukupno vreme bilo značajno i broj nodova veliki, uvođenje dva umesto jednog linka je donelo skraćenje ukupnog vremena za do 45%.

Dalje istraživanje bi moglo da ispita mogućnosti smanjenja overheda u komunikaciji, čime bi se rezultati verovatno još poboljšali.

8 ZAHVALNICA

Zahvaljujemo se Ministarstvu prosvete i nauke Republike Srbije, koje je podržalo ovo istraživanje u okviru istraživačkog projekta ON 174027 "*Computational Mechanics in Structural Engineering*".

9 LITERATURA

[1] Sloan, Joseph D.; *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI*, O'Reilly, 2004
[2] Pfister, Greg ; In Search of Clusters,; Prentice Hall, 1997
[3] Anderson, T.E. et al.; A Case for NOW (Networks of Workstations). IEEE Micro 15, 1 (February 1995), 54-64. DOI=10.1109/40.342018 http://dx.doi.org/10.1109/40.342018
[4] Milašinović D.D.; The finite strip method in computational mechanics. Faculties of Civil Engineering: University of Novi Sad, Technical University of Budapest and University of Belgrade: Subotica, Budapest, Belgrade; 1997.
[5] Cheung YK, Tham LG. Finite strip method. CRC Press; 1998.
[6] Rakić P.S. et al.; MPI-CUDA parallelization of a finite-strip program for geometric nonlinear analysis: A hybrid approach. Adv. Eng. Softw. 42, 5 (May 2011), 273-285. DOI=10.1016/j.advengsoft.2010.10.008
[7] Rakić, P.S. et al.: MPI-CUDA Parallelisation of the Finite Strip Method for Geometrically Nonlinear Analysis. in "Proceedings of the First International Conference on Parallel, Distributed and Grid Computing for Engineering", Civil-Comp Press, Stirlingshire, UK, Paper 33. doi:10.4203/ccp.90.33. (2009)
[8] Message Passing Interface; 2011; Wikipedia, Wikimedia Foundation Inc. , retrieved 09.09.2011. http://en.wikipedia.org/wiki/Message_Passing_Interface
[9] Message Passing Interface Forum; MPI: A Message-Passing Interface Standard, Version 2.2, 2009http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf
[10] 802.3u-1995 IEEE Standards for Local and Metropolitan Area Networks: Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T (Clauses 21-30), 1995 E-ISBN: 0-7381-0276-8
[11] Gabriel, E. et al.; Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In Proceedings, 11th European PVM/MPI Users' Group Meeting, 2004.
[12] Internet Protocol; IETF; RFC 791:, 1981. http://www.ietf.org/rfc/rfc791.txt
[13] Transmission Control Protocol; IETF; RFC 793:, 1981. http://www.ietf.org/rfc/rfc793.txt
[14] Hoefler, T. et al.: Low Overhead Ethernet Communication for Open MPI on Linux Clusters. CSR-06-06, TU Chemnitz, 2006, ISSN: 0947-5125
[15] IEEE Std 802.1AX-2008 IEEE Standard for Local and Metropolitan Area Networks — Link Aggregation. IEEE Standards Association. 2008-11-03. p. 30. doi:10.1109/IEEESTD.2008.4668665
[16] Linux Aggregation;2011; Wikipedia, Wikimedia Foundation Inc., retrieved 09.09.2011. http://en.wikipedia.org/wiki/Link_aggregation
[17] Davis, T.; Linux Ethernet Bonding Driver HOWTO, 2011, retrieved 09.09.2011. http://www.kernel.org/doc/Documentation/networking/bonding.txt
[18] An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware; IETF; RFC 793:, 1981. http://www.ietf.org/rfc/rfc826.txt
[19] Mohamed, N. et al.; High-performance message striping over reliable transport protocols; The Journal of Supercomputing, Volume 38, Number 3, 261-278, DOI: 10.1007/s11227-006-8443-6
[20] Yildirim, E. et al.; Which network measurement tool is right for you? a multidimensional comparison study; 9th IEEE/ACM International Conference on Grid Computing, 2008
[21] Woodall, T. et al.; Open MPI's TEG Point-to-Point Communications Methodology: Comparison to Existing Implementations; In Proceedings, 11th European PVM/MPI Users' Group Meeting, 2004

Lazar Stričević, Katedra za primenjene računarske nauke Fakulteta tehničkih nauka Univerziteta u Novom Sadu.
Oblast interesovanja: računarske nauke

Predrag Rakić, Katedra za primenjene računarske nauke Fakulteta tehničkih nauka Univerziteta u Novom Sadu.
Oblast interesovanja: računarske nauke

Miroslav Hajduković, Katedra za primenjene računarske nauke Fakulteta tehničkih nauka Univerziteta u Novom Sadu.
Oblast interesovanja: računarske nauke

CIP - Katalogizacija u publikaciji Народна библиотека Србије, Београд 659.25
INFO M: časopis za informacionu tehnologiju i multimedijalne sisteme = journal of information technology and multimedia systems
glavni i odgovorni urednik Dragana Bečejski Vujaklija. . - . –
Beograd : Fakultet organizacionih nauka,
2002- (Stara Pazova : SAVPO). - 30 cm
- Tromesečno ISSN 1451-4397 COBISS.SR-ID 105690636