

Tekst zadatka:

Potrebno je razviti rešenje za distribuirano računanje determinante matrice. Sistem se sastoji iz tri osnovne komponente: klijent, server i jednog ili više izvršioaca.

Klijent treba da omogući generisanje matrica nasumičnih vrednosti i dimenzija i njihov prenos do servera. Pored ovoga, korisnik treba da omogući unos matrice  $A \times B$ . Nakon unosa matrica se šalje serveru na obradu, odnosno računanje determinante.

Server treba da omogući prihvatanje matrice za računanje. Po prihvatanju matrice, po potrebi (dimenzija veća od 3) organizuje razbijanje determinante matrice na  $N$  izvršioaca i računa je od dobijenih rezultata. Kada se računanje završi rezultat se čuva na serveru i prosleđuje klijentu.

Izvršilac - obavlja zadatak koji dobije od servera potencijalno startujuću dodatne izvršioce u zavisnosti od veličine matrice.

Klijent dobija rezultat operacije od servera.

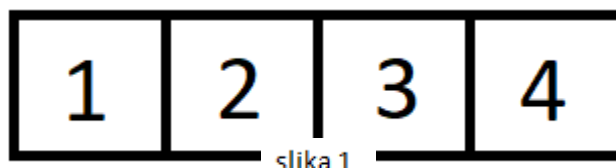
Rešenje:

**Klijent** se sastoji iz dve niti: main-a u kome se za svaki pritisnut taster generiše nasumična matrica i zatim šalje na obradu serveru, i recvOdgovor niti u kojoj se odgovori očekuju i ispisuju. Ovo je jedan od načina da funkcije send i recv ne moraju čekati jedna na drugu da bi se izvršile

1. sizeof(char), potpis entiteta koji šalje poruku.

U ovom slučaju  $1 ==$  potpis klijenta

2. sizeof(int), redni broj matrice koju šalje na obradu.



3. sizeof(char), dimenzija matrice.

4. sizeof(char)\*dimenzija\*dimenzija. Elementi matrice

Izgled poruke koju klijent dobija od servera:

1. sizeof(double), rešenje obrade: determinanta.

2. sizeof(char), ispravnost obrade: ima vrednost razlicitu od 0 ako je obrada uspeła.

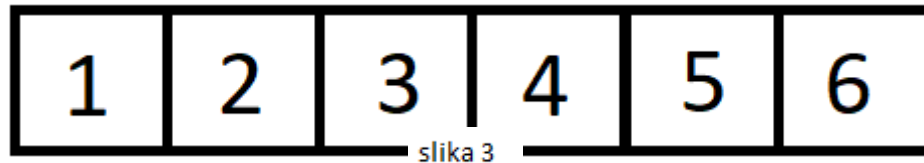


3 sizeof(int), redni broj matrice koja je obradjena.

**Server** tokodje koristi blokirajući soket, u to da se svaki uspesan accept posalje u posebnu nit na obradu. Obrada se sastoji u scenarijima.

Scenario 1: uslov: prvi char poruke je  $1 ==$  potpis klijenta. Klijent zahteva da se obradi matrica. Server stavi zahtev u listu, pamteci SOKET korisnika (kome se vraća rešenja), broj zahteva, dimenziju matrice i matricu. Nakon toga pravi novog workera.

Scenario 2: uslov: prvi char poruke je 2==potpis workera I drugi char poruke je 1==worker trazi da mu server prosledi matricu na obradu. Ovde se pristupa prethodnoj listi i salje najsveziji zahtev. Izgled poruke:



1 sizeof(char), potpis servera

2 sizeof(SOCKET), soker klienta koji je poslao matricu

3 sizeof(int) br zahteva

4 sizeof(double) mnozilac matrice, u ovom slucaju 1

5 sizeof(char) dimenzija matrice

6 sizeof(char)\*dim\*dim, matrica

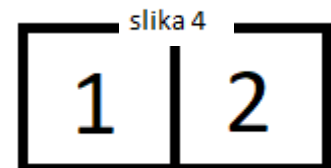
Scenario 3: uslov: prvi char poruke je 2==potpis workera I drugi char poruke je 2==worker zavrrio obradu. Otpakuje se poruka I salje resenje odgovarajucem klientu. Izgled ostatka poruke: slika 6

**Worker** tokom obrade matrice pravi strukturu stable. Njegovom posmatranju pristupamo iz dva ugla. Worker list I worker cvor. Worker list su oni workere kojima se prosledjuje matrica dimenzije <= 3.

Boss pokrenutog workera moze biti server ili worker cvor koji ga je pokrenuo da bi obradio matricu manje dimenzije. Nakon povezivanja na boss-a worker mu salje prvu poruku u kojoj se predstavlja:

1 sizeof(char) vrednost 2==potpis workera

2 sizeof(char) vrednost 1==prva poruka



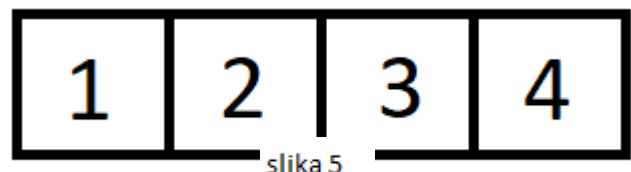
Ovim predstavljanjem boss zna da mu se javlja worker I zatim mu salje matricu na obradu. Slika 3 je izgled poruke ako je salje server. Izgled poruke ako je salje nadredjeni worker:

1 sizeof(char) potpis workera == 2

2 sizeof(double) mnozilac podmatirce

3 sizeof(char) dimenzija matrice

4 sizeof(char)\*dim\*dim, matrica



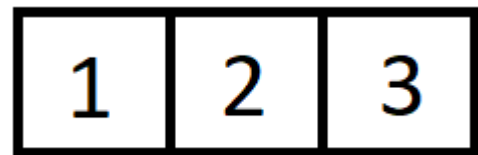
Ove dve poruke se na prvi pogled razlikuju međjutim zadnja 3 polja su im ista. Razlika je u tome što resenja podmatrica služe samo kao zbir boss-evima koji te zbireve dalje šalju vrhovnom worker bossu. Konacna determinanta jeste resenje zahteva, međjutim potrebno je znati kome to resenje pripada. Dakle ako je poruka stigla od server(prvi char poruke) postavlja se offset kojim se preskacu identifikatori klienta za kasniju upotrebu i kojim se istovremeno poruka 3 transformise u poruku 5 cime se obe poruke uniformno tretiraju.

U nastavku se iz poruke vade informacije i ako se ustanovi da je matrica dimenzije  $\leq 3$  determinanta se odmah racuna pomocu odgovarajuce formule i prosledjuje bossu. Izgled poruke zavisi od vrste bossa koji se zna jos uvek unutar sacuvane poruke koju je dobio od njega prilikom javljanja, jer nije menjana. Ako je u prvom char-u poruke potpisan worker izgled poruke koju salje:

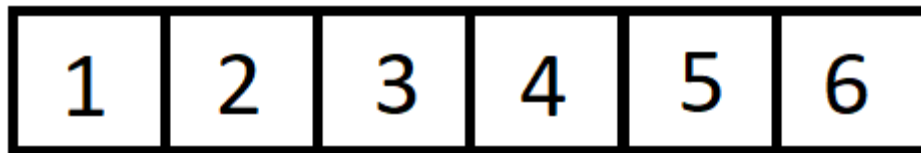
1 sizeof(char) potpis workera

2 sizeof(char) vrednost 2 == resenje zahteva

3 sizeof(double) dererminanta



U suprotnom poruka koju vrhovni worker salje server:



Prva 3 polja su sita kao u prethodnom.

4 sizeof(char) ispravnost obrade

5 sizeof(SOCKET) soket klient koje se salje resenje

6 sizeof(int) redni broj zahteva

Iz prethodnog se moze zakljuciti da funkcije vrše obradu uniformno u odnosu na izvor poruke. To je vazno zato što worker cvor simulira server workerima koji su zaposleni za njega. Prvi char poruka potpisuje sa 2.

Worker cvor je zaduzen za obradu matrica dimenzija  $n > 3$ . To radi tako što pravi  $n$  workera i šalje im podmatrice dimenzija  $n-1$ . Obrada se nastavlja rekurzivno.

Worker cvor koristi neblokirajuci soket umesto kombinacije blokirajuceg soketa sa nitima jer sve workere koje je napravio radi na istom cilju, da izdacunaju njegovu matricu. Pomocu SELECT funkcije je mnogo laksa kontrola nad desavanjima njegovih workera i skupljanja njihovih resenja u lokalnu

determinantu. Kombinacija blokirajućeg soketa sa nitima više odgovara pokretanju nezavisnih zadataka, kao što je to zahtev dva različita klijenta za obradu njihovih matrica.

**Pokretanje.** Prvo se pokrene server pa klijent. Workeri se pokreću automatski tokom rada.

Milutin Vuckovic PR7-2017.