
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13S114MUPS)

Nastavnik: dr Milo Tomašević, red. prof.

Asistent: doc. dr Marko Mišić

Ispitni rok: Prvi kolokvijum (Novembar 2021.)

Datum: 02.11.2021.

Kandidat:* _____

Broj Indeksa:* _____

*Kolokvijum traje 105 minuta, prvih sat vremena nije dozvoljeno napuštanje kolokvijuma.
Upotreba literature nije dozvoljena.*

| | | | |
|------------------|-----------|------------------|-----------|
| <i>Zadatak 1</i> | _____ /20 | <i>Zadatak 5</i> | _____ /15 |
| <i>Zadatak 2</i> | _____ /10 | <i>Zadatak 6</i> | _____ /10 |
| <i>Zadatak 3</i> | _____ /15 | <i>Zadatak 7</i> | _____ /15 |
| <i>Zadatak 4</i> | _____ /15 | | |

Ukupno na kolokvijumu: _____ /100

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**. * popunjava student.

1. **[20]** Opisati pet tipičnih klasa računara. Navesti za koje aplikacije se obično koriste, kao i najbitnije projektne ciljeve.

2. **[10]** Opisati istorijske trendove u arhitekturi računara u pogledu ostvarivanja paralelizma obrade.

3. [15] Nacrtati UMA i NUMA arhitekture. Objasniti sličnosti i razlike između njih.

4. [15] Objasniti programski model *Data parallel*. Nacrtati i objasniti tipičnu arhitekturu koja podržava ovaj model.

5. [15] Korišćenjem OpenMP tehnologije, paralelizovati deo koda u prilogu koji pronalazi parametar alfa sa zadovoljavajućom tačnošću u metodu konjugovanih gradijenata. Obratiti pažnju na efikasnost i korektnost paralelizacije. Paralelizaciju obaviti ručnim raspoređivanjem posla nitima. Smatrati da su sve promenljive ispravno deklarisanе.

```
bool alfaSet = false;

int alfaIter = 0;

int step = 1;

double localAlfa = alfa;

double *testX = (double *) malloc(dim * sizeof(double));

while (!alfaSet) {

    alfaIter += step;

    localAlfa = pow(gamma, alfaIter - 1);

    for (i = 0; i < dim; i++) {

        testX[i] = x[i] + localAlfa * d[i];
    }

    if (func(testX, dim) - delta * localAlfa * b <= oldFunc) {

        if (!alfaSet) {

            alfaSet = true;

            alfaIter = 0;

            alfa = localAlfa;
        }
    }
}

free(testX);
```

6. [10] Na koji način se promenljive podrazumevano prosleđuju u okviru OpenMP *task* direktive? Zbog čega je to neophodno i da li je uvek neophodno? Obrazložiti odgovor.

7. [15] Neka se posmatra jedna numerička simulacija. Aplikacija najpre učitava parametre simulacije, zatim rešava složene sisteme linearnih diferencijalnih jednačina i na kraju upisuje rezultate svoga rada. Nakon merenja performansi sekvencijalne implementacije posmatrane aplikacije pri uobičajenoj upotrebi, dobijeni su sledeći rezultati: aplikacija 5% vremena provodi obavljajući ulazno-izlazne operacije, a 95% vremena provodi u obradi podataka. Tipično vreme obrade u okviru simulacije korišćenjem jednog jezgra je 1000s.

a) [7] Ukoliko se aplikacija paralelizuje za izvršavanje na SMP sistemu sa 4 jezgara na 2GHz sa 32GB memorije, navesti formulu za Amdalov zakon i odrediti maksimalno moguće ubrzanje koje se može postići za zadatu aplikaciju sa datom konfiguracijom.

b) [8] Šta predstavlja osobina skalabilnosti u kontekstu paralelnog računarstva i specifično paralelnog hardvera? Da li bi dodavanje novih procesorskih jezgara u slučaju pod a) doprinelo ubrzavanju opisane paralelne aplikacije i u kojim slučajevima?