HP-SEE

Uvod u MPI programiranje primjer paralelizacije

www.hp-see.eu

Mihajlo Savic

Elektrotehnicki fakultet Banja Luka

HP-SEE

High-Performance Computing Infrastructure
for South East Europe's Research Communities

Izvori informacija

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

Prezentacija zasnovana na Gregory-Leibniz računanju vrijednosti pi

$$\pi = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1} = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \cdots.$$

Algoritmu sa lokacije http://www.mcs.anl.gov/research/projects/mpi/usingmpi/examples/simplempi/main.htm

Serijski pristup (1)

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main (argc, argv)
  int argc; char *argv[]; {
  double pi_30_decimala = 3.14159265358979323846264338327 9;
  int i, intervala; // pomocne varijable
  double pi, h, suma, x; // pomocne varijable
  if (argc!=2) { // treba nam 1 parametar u komandnoj liniji
    printf("Uputstvo: %s <intervala>\n", argv[0]);
    return 1;
  }
```

Serijski pristup (2)

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

```
intervala=atoi(argv[1]); // koliko intervala za racunanje
  printf("Racunamo pi na %d intervala...\n",intervala);
  if (intervala > 0) { // ako je vise od 0 racunamo
      suma = 0.0;
      h = 1.0 / (double) intervala; // racunamo pi jednostavno
      for (i = 1; i <= intervala; i ++)
         suma += (4.0 / (1.0 + pow(h * ((double) i - 0.5),2)));
      pi = h * suma;
      printf ("pi je priblizno %.30f, greska je %.30f\n", pi,
               fabs (pi – pi_30_decimala)); // rezultat
  }
  return 0;
}
```

Pokretanje programa

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Fajl snimimo kao serial_pi.c
- Kompajliranje
```
gcc -o serial_pi serial_pi.c
```
- Pokretanje
```
./serial_pi 1000000000
```
- Mjerimo vrijeme
```
time ./serial_pi 1000000000
...
real    0m16.145s
user    0m16.130s
sys     0m0.000s
```

Paralelizacija

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Koristimo MPI
- Moramo podijeliti posao čvorovima
- *Srećom*, odabrani algoritam se lako paralelizuje
- Svaki čvor računa svoju parcijalnu sumu nezavisno (***i*** u ptelji kreće od ***broj_čvora+1*** i uvećava se za ***ukupan broj čvorova***)
- Na kraju sumiramo pomoću MPI_Reduce sa MPI_SUM transformacijom (vidjeti prethodne vježbe za detalje)
- Voditi računa o opštim stvarima (samo jedan čvor komunicira sa korisnikom, itd)

MPI pristup (1)

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

```
...
#include "mpi.h"

int main (argc, argv) int argc; char *argv[]; {
  double pi_30_decimala = 3.14159265358979323846264338327 9;
  int i, intervala, ukupno, cvor; // MPI varijable
  double pi, h, suma, x, globalni_pi;
  MPI_Init (&argc, &argv); // inicijalizacija MPI
  MPI_Comm_size (MPI_COMM_WORLD, &ukupno); // ukupno procesa
  MPI_Comm_rank (MPI_COMM_WORLD, &cvor); // moj broj
  if (argc!=2) { // samo cvor 0 komunicira sa korisnikom
    if (cvor==0) printf("Uputstvo: %s <intervala>\n", argv[0]);
    MPI_Finalize (); // gasimo MPI
    return 1;
  }
```

MPI pristup (2)

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

```
intervala=atoi(argv[1]);
  if (cvor==0)
    printf("pi na %d cvorova i %d intervala\n", ukupno,intervala);
  if (intervala > 0) {
      suma = 0.0; h = 1.0 / (double) intervala;
      for (i = cvor + 1; i <= intervala; i += ukupno)
         suma += (4.0 / (1.0 + pow(h * ((double) i - 0.5),2)));
      pi = h * suma;
      MPI_Reduce (&pi, &globalni_pi, 1, MPI_DOUBLE, MPI_SUM, 0,
                  MPI_COMM_WORLD); //skupljamo i sumiramo sve
      if (cvor==0)
         printf ("pi je priblizno %.30f, greska je %.30f\n", globalni_pi,
               fabs (globalni_pi – pi_30_decimala)); // rezultat
  }
  MPI_Finalize (); // gasimo MPI
  return 0;}
```

Pokretanje programa

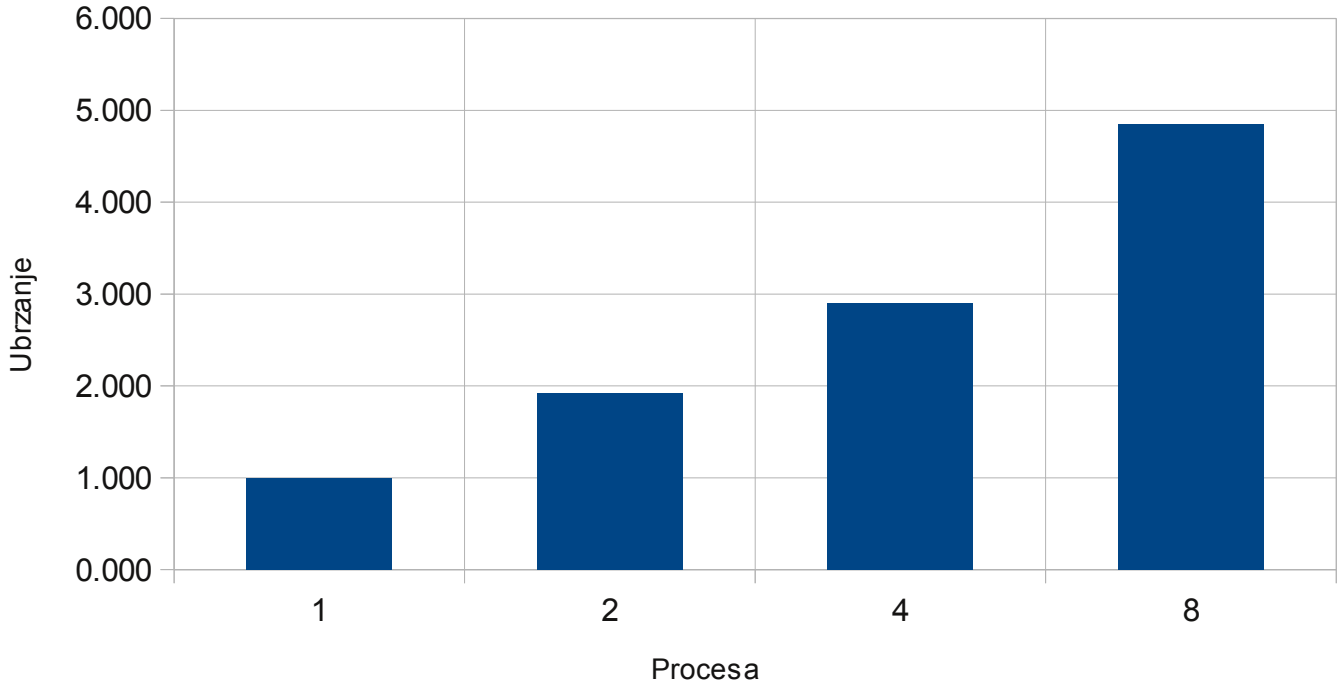
HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Fajl snimimo kao mpi_pi.c
- Kompajliranje
```
mpicc -o mpi_pi mpi_pi.c
```
- Pokretanje
```
mpirun -n 4 ./mpi_pi 100000000
```
- Mjerimo vrijeme
```
time mpirun -n 4 ./mpi_pi 100000000
real    0m5.607s
user    0m17.820s
sys     0m0.150s
```

Mjerenje performansi

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

NP	Real [s]	User [s]	Ubrzanje
1	16.295	16.260	1.000
2	8.477	16.860	1.922
4	5.607	17.840	2.906
8	3.362	17.890	4.847



Radni zadatak

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Prepraviti program tako da:
  - Provjerava da li je MPI inicijaliziran pravilno (vidjeti prethodne vježbe)
  - Ukoliko broj intervala nije definisan u komandnoj liniji traži ga od korisnika
  - Tako dobijen broj intervala šalje svim čvorovima putem MPI_Broadcast (vidjeti prethodne vježbe)
  - Sam računa vrijeme provedeno za izvršavanje cijelog programa i samo dijela računanja pi u ms (od poslije ispisa poruke o računanju do prije ispisa poruke o rezultatu) i ispisuje ga na kraju računanja – možete koristiti MPI_Wtime() za trenutno vrijeme
  - Provjeriti mogu li se poboljšati performanse već paralelizovanog programa...