

Multiprocesorski sistemi

Laboratorijska vežba 3

MPI – korisnički tipovi podataka, grupe i komunikatori, virtuelne topologije

Uvod

Cilj vežbe je da studente obučiti da samostalno mogu podesiti MPI okruženje i razvijati MPI programe. Vežbom su obuhvaćeni jednostavni primeri.

Podešavanje okruženja

Preuzeti http://mups.etf.rs/vezbe/mpi/code/mpi_win_vs2005.zip (odnosno, ..._vs2008.zip, zavisno od korišćene verzije VS). Prema uputstvima u priloženom `readme.txt` fajlu podesiti okruženje za razvoj i kontrolisano izvršavanje (engl. debugging) MPI programa na lokalnom računaru.

Zadaci

Svaki od programa napisati tako da može biti izvršen sa bilo kojim od broja procesa određenih vrednostima navedenih u uglastim zagradama na kraju postavke zadatka. Broj N označava maksimalan mogući broj procesa u trenutno dostupnom MPI klasteru. Za programe koji će biti izvršavani samo na lokalnom računaru, pretpostaviti da važi $N=5$. Poslednja tri programa treba da vrše proveru da li je broj procesa odgovarajući postavci zadatka. U slučaju da broj procesa nije odgovarajući, prekinuti izvršavanje korišćenjem MPI poziva `Abort`.

Svaki program treba da ima ispis formatiran na sledeći način: rang procesa unutar `COMM_WORLD` komunikatora (2 cifre, sa vodećim nulama), znak `:`, redni broj poruke koju posmatrani program ispisuje (3 cifre, sa vodećim nulama), znak `:`, tekst konkretne poruke (na primer, u prvom zadatku proces sa rangom 0 će ispisati `00:001:Hello World!`, proces sa rangom 3 će ispisati `03:001:Hello World!` itd.).

Korisnički tipovi podataka

1. Sastaviti program koji učitava i ispisuje elemente matrice bez obrade elemenata. Broj vrsta odrediti na osnovu broja procesa. Proces sa rangom 0 treba da učita broj kolona (najviše 100) i elemente matrice, a zatim da pošalje svakom od ostalih procesa po jednu vrstu matrice. Za prenos jedne vrste koristiti samo jedan MPI poziv `send` i njemu odgovarajući `receive`. [2..32]
2. Proširiti prethodni program tako da radi sa bilo kojim brojem redova i kolona. Proces sa rangom 0 treba da učita dimenzije i elemente matrice, a zatim da pošalje svakom od ostalih procesa što je moguće sličniji broj vrsta matrice. [2..N]
3. Sastaviti program koji učitava i ispisuje elemente matrice bez obrade elemenata. Broj kolona odrediti na osnovu broja procesa. Proces sa rangom 0 treba da učita broj vrsta (najviše 100) i elemente matrice, a zatim da pošalje svakom od ostalih procesa po jednu kolonu matrice. Za prenos jedne kolone koristiti samo jedan MPI poziv `send` i njemu odgovarajući `receive`. [2..32]
4. Sastaviti program koji za diplomirane studente na istom smeru određuje prosečne ocene. Podaci o ocenama studenata su za svakog studenta smešteni u jednodimenzionalnom nizu. Pošto student u toku jedne školske godine može imati maksimalno 20 ocena, niz ima najviše 80 elemenata. Za svakog od studenata na istom smeru je broj ocena po godini isti. Proces sa rangom 0 treba da učita potrebne podatke o ocenama studenata i podatke o broju ocena na svakoj od godina studija, raspodeli posao ostalim procesima i ispiše prosečne ocene. Proces sa rangom 0 treba da šalje ostalim procesima samo one elemente niza ocena koji predstavljaju dobijene ocene. Za prenos ocena jednog studenta koristiti samo jedan MPI poziv `send` i njemu odgovarajući `receive`. [2..N]
5. Sastaviti program koji u nizu kompleksnih brojeva pronalazi broj sa najvećom apsolutnom vrednošću. Kompleksni broj je predstavljen sa dva realna broja i jednim celim brojem. Ako je celi broj jednak 0, realni brojevi predstavljaju realni i imaginarni deo kompleksnog broja, inače predstavljaju moduo i argument. Proces sa rangom 0 treba da učita broj elemenata niza i elemente niza, a zatim da pošalje ostalim procesima odgovarajući deo niza na obradu. Za prenos podniza koristiti samo jedan MPI poziv `send` i njemu odgovarajući `receive`. [2..N]

Grupe i komunikatori

1. Sastaviti program koji raspoložive procese deli u dve grupe, prema parnosti ranga [2..N]. Za podelu koristiti dva poziva `MPI_Group_incl`. Svaki proces treba da ispiše svoj rang unutar MPI sveta, svoj rang unutar novog komunikatora i zbir kvadrata rangova unutar novog komunikatora svih procesa obuhvaćenih novim komunikatorom.
2. Izmeniti prethodni program tako da koristi jedan poziv `MPI_Group_incl` i jedan poziv `MPI_Group_excl` [2..N].
3. Izmeniti prethodni program tako da koristi `MPI_Comm_split` [2..N].
4. Sastaviti program koji računa vrednost broja π po grupama od po približno K članova, gde je K manje od broja procesa [2..N]. Proces treba da budu što ravnomernije raspoređeni po novim grupama i komunikatorima. Svaka grupa treba da izračuna vrednost broja π . Proces sa rangom 0 u svakoj novoj grupi treba da ispiše dobijenu vrednost i pošalje je procesu sa rangom 0 u MPI svetu, koji treba da usrednji dobijene vrednosti i ispiše konačni rezultat.
5. Sastaviti program koji ispisuje najveći element u svakoj vrsti matrice i najmanji element u svakoj koloni matrice [2..N]. Svaku pretragu treba da obavlja posebna grupa procesa. Posao podeliti tako da svaki proces dobije da obradi po jednu vrstu ili kolonu. Pretpostaviti da je broj procesa istovetan tačno potrebnom.

Virtuelne topologije

1. Sastaviti program koji učitava niz realnih brojeva (broj elemenata je određen brojem raspoloživih procesa), uvezuje procese u `Cart` komunikator i ispisuje rangove levog i desnog procesa-suseda u `Cart` komunikatoru kao i vrednosti susednih elemenata niza [2..N].
2. Isto kao i prethodni program, samo što je virtuelna topologija prsten umesto vektora [2..N].
3. Sastaviti program koji učitava dimenzije i elemente matrice, uvezuje raspoložive procese u `Cart` komunikator i ispisuje rangove gornjeg, donjeg, levog i desnog suseda u `Cart` komunikatoru, kao i vrednosti susednih elemenata matrice [2..N].
4. Isto kao i prethodni program, samo što je virtuelna topologija cilindar umesto matrice [2..N].
5. Isto kao i prethodni program, samo što je virtuelna topologija torus umesto cilindra [2..N].
6. Sastaviti program koji stvara ciklični graf (prsten) od prisutnih procesa [2..N]. Svaki proces treba da svoj rang doda na broj primljen od „levog“ suseda i rezultat prosledi „desnom“ susedu. Proces sa rangom 0 treba da prvi pošalje svoj rang i ispiše primljeni rezultat nakon što zbir rangova zatvori krug.

VAŽNO: Ukoliko u bilo kom zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku i da nastavi da izgrađuje svoje rešenje temeljima uvedene pretpostavke.