

Multiprocesorski sistemi

Domaći zadatak 4

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

Uvod

Cilj zadatka je studente upozna sa značajnim MPI konceptima. Zadatak obuhvata jednostavne primere.

Podešavanje okruženja

Preuzeti http://mups.etf.rs/vezbe/mpi/code/mpi_win_vs2008.zip (ili [..._vs2005.zip](http://mups.etf.rs/vezbe/mpi/code/mpi_win_vs2005.zip)). Prema uputstvima u priloženom `readme.txt` fajlu podesiti okruženje za razvoj i kontrolisano izvršavanje (engl. debugging) MPI programa na lokalnom računaru. Alternativno, koristiti OpenMPI na računaru `rtidev4.etf.rs`.

Zadaci

Svaki od programa treba napisati tako da može biti izvršen sa bilo kojim od broja procesa određenih vrednostima navedenih u uglastim zagradama iza postavke zadatka. N označava maksimalan mogući broj procesa u trenutno dostupnom MPI okruženju. Za programe koji će biti izvršavani na samo jednom računaru, pretpostaviti da važi $N=6$. Svaki program treba da vrši proveru da li je broj procesa odgovarajući postavci zadatka, pod uslovom da nije predviđen za broj procesa u intervalu $[1..N]$. 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 proces 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 1 će ispisati `01:001: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 učitava 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 učitava 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 učitava 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 učitava 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 učitava 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. 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..N]
2. Izmeniti prethodni program tako da koristi po jedan poziv `MPI_Group_incl` i `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. Procesima 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. [2..N]
5. Sastaviti program koji ispisuje najveći element u svakoj vrsti matrice i najmanji element u svakoj koloni matrice. Svaku pretragu treba da obavlja posebna grupa procesa. Posao podeliti tako da svaki proces dobije da obradi tačno jednu vrstu ili tačno jednu kolonu. [2..N]

Virtuelne topologije

1. Sastaviti program koji učitava niz realnih brojeva, uvezuje procese u `Cart` komunikator i ispisuje rangove levog i desnog procesa-suseda u `Cart` komunikatoru kao i vrednosti susednih elementa niza. Broj elemenata niza jednak je broju procesa u MPI svetu. [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. Broj elemenata niza jednak je broju procesa u MPI svetu. [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. 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. [2..N]

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