

Tema 4 – Simularea unei rețele in MPI

Responsabil: Valeriu Stanciu si Alecsandru Patrascu

Data ultimei modificari: 11.11.2013

Termenul de predare: 14.01.2014, ora 23:55

In secolul 23, in urma unui razboi nuclear singurii supravietuitori se gasesc in buncare bine protejate. Oamenii nu pot iesi la suprafata iar singurul lor mijloc de comunicatie il reprezinta cateva linii de telegraf ingropate intre cateva buncare. Supravietuitorii incearca in aceasta situatie sa reconstruiasca o parte din civilizatia pierduta si sa incerce sa comunice folosind aceste mijloace rudimentare.

Se cere simularea rețelei de comunicatie in MPI. Reteaua va reprezenta un graf, care poate contine si cicluri, ale carui noduri reprezinta buncarele descrise mai sus iar muchiile grafului reprezinta legaturile de comunicatie. Fiecare nod are aceeași functionalitate, fiind simulat folosind un process MPI. Nici un buncar nu poate trimite mesaje decat unui alt buncar cu care este conectat direct.

Se considera topologia stabila, nu vor aparea modificari ale sale in timpul rularii programului. Atat numele fisierului ce va contine topologia cat si cel pentru mesaje trimise vor fi transmise ca parametru la rulare. Exemplu de comanda de rulare

```
mpiexec -np 10 ./tema4 fisier_topologie fisier_mesaje
```

Pentru refacerea civilizatiei supravietuitorii au stabilit urmatoarele etape:

1. Sa descopere modul in care sunt conectate toate buncarele pentru a putea incepe sa se organizeze.

La pornire fiecare proces MPI va citi dintr-un fisier lista sa de adiacenta. In acest mod fiecare nod cunoaste doar nodurile cu care este conectat direct si poate comunica doar cu acestea.

In acest moment fiecare nod al grafului va rula algoritmul de stabilire a topologiei folosind mesaje de sondaj cu ecou (gasiti o descriere a algoritmului in laboratorul de stabilire a topologiei) pentru a genera o tabela de routare. Fiecare proces va afisa tabela creata.

Output-ul fiecarui proces va contine pe prima linie id-ul sau, iar apoi pe urmatoarele linii intrarile din tabela proprie de routare de la un nod sursa catre un nod destinatie. De asemenea, procesul cu id-ul 0 va afisa intreaga matrice de adiacenta.

Exemplu de fisiere de intrare:

```

0 - 1 2
1 - 0 3 4 5 6
2 - 0 7 8
3 - 1
4 - 1 9 10
5 - 1
6 - 1
7 - 2
8 - 2 11
9 - 4
10 - 4
11 - 8

```

Exemplu de output:

```

id_proces
nod_sursa1 nod_destinatie1
nod_sursa2 nod_destinatie2
...
nod_sursaN nod_destinatieN

```

În cazul de mai sus se considera un singur fișier din care fiecare nod va citi linia cu lista de adiacență care îi aparține.

2. Stabilirea unui sistem de poșta prin care supraviețuitorii să-și poată trimite mesaje

După ce fiecare nod și-a creat propria tabelă de rutare se pot trimite mesaje între bunecare. Va exista un fișier la care va avea acces fiecare proces MPI care va avea pe prima linie numărul n de mesaje și pe următoarele n linii câte o comandă de formă:

```
<nod_sursa> <nod_destinatie> mesaj
```

După citirea fișierului, fiecare nod va anunța printr-un mesaj de broadcast toate bunecarele ca va urma o comunicare. Mesajul de broadcast este simbolizat prin apariția literei B în cazul destinației. În acest mod puteți determina când începe etapa 2.

După mesajul de broadcast inițial, procesul cu rangul *nod_sursa* va trimite mesajul cu destinația specificată. Toate procesele intermediare și destinația vor afișa informații legate de mesaj: nodul de la care a fost primit, următorul nod caruia îi este transmis, destinația, etc.

Exemplu de fișier care conține mesaje ce vor fi trimise după determinarea tabelelor de rutare:

```
9 1 Mesaj trimis
7 9 Buna ziua!
5 8 Alt mesaj
3 B Mesaj de broadcast
4 2 Salut...
```

Fiecare nod va anunta toate buncarele cand termina de trimis toate mesajele. In acest mod puteti determina cand se incheie etapa 2.

3. Stabilirea unor structuri de conducere

Se doreste desemnarea unui buncar care sa aiba rol de lider si a unui buncar cu rol de lider adjunct. In acest sens, se va implementa un algoritm de tip unda (mai multe informatii in cursul "Alegerea liderului").

Alegerea adjunctului se va efectua dupa aceeaasi regula ca si alegerea liderului.

Aceasta etapa se termina cand s-au stabilit cele doua pozitii si toate buncarele cunosc rezultatele. In final se va afisa de catre fiecare proces buncarul cu rol de lider si buncarul cu rol de lider adjunct, pe aceasi linie si separati cu spatiu, de exemplu:

```
id_proces id_nod_lider id_nod_lider_adjunct
```

Predare tema:

Este obligatoriu sa existe in arhiva trimisa:

sursele temei

un fisier makefile

un fisier readme

cel putin 1 fisier de intrare