Avaluació PAC2 i una possible solució

dt. nov. 10, 2015 7:16 pm by Ivan Rodero Castro

Hola a tothom,

He penjat avuí les qualificacions de la PAC2. Discupleu la demora, porto dos dies de retràs ja que vaig estar fotut durant el cap de setmana. Propagaré aquesta demora en la PEC3 i espero que dos dies no us alteri massa el calendari.

A nivell de solució podem utilitzar la gravació de la teleconferència pels primers apartat is us comento els darrers 2 exercicis a continuació. Veureu que he fet una correcció força relaxada i no he penalitzat els problemes de sincronització en l'apartat però veureu pels comentaris a l'AC que en l'exercici 4 no he deixat correr. M'interessa que us quedin els conceptes de crides síncrones clares i us proposo que ho discutim al fòrum on hi podeu penjar les versions que us semblin oportunes amb codi si és necessari.

La majoria de vosaltres ha entregat l'exercici 3 sense tenir en compte el possible deadlock al fer que tots els processos fessin un MPI_send alhora. A la pràctica l'execució us haurà funcionat perquè els buffers del sistema pot absorvir un byte per procés. En el cas d'un codi amb missatges més grans aquesta situació podria produir deadlock.

El que heu entregat principalment és el següent:

```
#include "mpi.h"

#include


int main(int argc, char** argv)

{

  int MyProc, size, tag = 0;

  int send_proc = 0, recv_proc = 0;

  char msg = 'A', msg_recpt;

  MPI_Status status;



  MPI_Init(&argc, &argv);

  MPI_Comm_rank(MPI_COMM_WORLD, &MyProc);

  MPI_Comm_size(MPI_COMM_WORLD, &size);


  printf("Process # %d started \n", MyProc);

  MPI_Barrier(MPI_COMM_WORLD);



  for (send_proc = 0; send_proc < size; send_proc++)
```

```
 {
  if (send_proc != MyProc)
  {
   printf("Proc #%d sending message to Proc #%d \n", MyProc, send_proc);
   MPI_Send(&msg, 1, MPI_CHAR, send_proc, tag, MPI_COMM_WORLD);
  }
 }

 for (recv_proc = 0; recv_proc < size; recv_proc++)
 {
  if (recv_proc != MyProc)
  {
   MPI_Recv(&msg_recpt, 1, MPI_CHAR, recv_proc, tag, MPI_COMM_WORLD, &status);
   printf("Proc #%d received message from Proc #%d \n", MyProc, recv_proc);
  }
 }


 sleep(10);


 MPI_Barrier(MPI_COMM_WORLD);
 MPI_Finalize();
}
```

El que caldria fer en aquest cas és o bé utilitzar crides asíncrones com es mostra a continuaicó o bé assegurar-se que uns processos fan MPI_send i els altres MPI_recv. Només hi ha hagut uns pocs estudiants que ho han fet d'aquesta manera i ho podem discutir al fòrum.

(verisó asíncrona, resumida - falta una mica de codi...)

(...)

```
 for (send_proc = 0; send_proc < size; send_proc++)
```

```
 {
  if (send_proc != MyProc)
  {
    res =  MPI_Irecv(&buf, 1, MPI_CHAR, src, tag, MPI_COMM_WORLD,    &req);


  }
 }

 for (recv_proc = 0; recv_proc < size; recv_proc++)
 {
  if (recv_proc != MyProc)
  {
    MPI_Send(&msg, 1, MPI_CHAR, send_proc, tag, MPI_COMM_WORLD);
    printf("Proc #%d sending message from Proc #%d \n", MyProc, send_proc);
  }
 }

  res =  MPI_Waitall(size, req, stat);
(...)
```

Per una altra banda, en l'exercici 4 molts de vosaltres heu sincronitzat correctament els processos que fan MPI_send i els que fan MPI_recv com es mostra a continuació:

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

int main(argc,argv)
int argc;
char **argv;
{
```

```
int MyProc, tag=1, size;
char msg='A', msg_recpt ;
MPI_Status *status ;
int left, right ;



MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &MyProc);
MPI_Comm_size(MPI_COMM_WORLD, &size);



left = (MyProc + size - 1) % size;
right = (MyProc + 1) % size;

if (MyProc == 0)
{
  printf("Proc %d sending message to proc %d \n", MyProc, right);


  MPI_Send(&msg, 1, MPI_CHAR, right, 1, MPI_COMM_WORLD);
  MPI_Recv(&msg_recpt, 1, MPI_CHAR, left, 1, MPI_COMM_WORLD, status);
}
else
{
  MPI_Recv(&msg_recpt, 1, MPI_CHAR, left, 1, MPI_COMM_WORLD, status);
  MPI_Send(&msg, 1, MPI_CHAR, right, 1, MPI_COMM_WORLD);
  printf("Proc %d : received message from proc %d and sending message to %d\n", \
  MyProc, left, right);
}
```

Salutacions,
  MPI_Barrier(MPI_COMM_WORLD);
Ivan
  MPI_Finalize();

}

Agraeixo l'esforç a tots vosaltres que heu fet versions dels programes no obvies més enllà del que pot estar més a mà. Aprofito per comentar-vos que no us enrefieu massa del que podeu trobar online - algunes vegades poden tenir "trampa"