AULA 11

Programação de Sistemas Paralelos e Distribuídos

Criação dinâmica de processos

- Criação dinâmica de processos:
- MPI_Spawn(char program[],char *argv[],int maxprocs,MPI_Info info, int root,MPI_Comm *parents,MPI_Comm *children,int errs[]);
- MPI_Comm *parents: comunicador do processo pai
- MPI_Comm *children: comunicador com o qual o processo se comunica com os processos criados (spawned)

```
MPI_Comm_spawn("worker_teste1_a", MPI_ARGV_NULL, 1,
        MPI_INFO_NULL, 0, MPI_COMM_SELF, &everyone1,
        MPI_ERRCODES_IGNORE);
  MPI_Comm_spawn("worker_teste1_b", MPI_ARGV_NULL, 1,
        MPI_INFO_NULL, 0, MPI_COMM_SELF, &everyone2,
        MPI_ERRCODES_IGNORE);
X=3;
  MPI_Send(&X,1,MPI_INT,0,11,everyone1);
X=7;
  MPI_Send(&X,1,MPI_INT,0,11,everyone2);
MPI_Recv(&X,1,MPI_INT,0,12,everyone1,&status);
MPI_Recv(&X,1,MPI_INT,0,12,everyone1,&status);
```

Arquivos: worker_teste1_a.c worker_teste1_b.c teste-spawn1.c (compilar todos com mpicc)

mpirun –np 1 teste-spawn1

Outros exemplos

- teste-spawn2.c

  worker_teste2_a.c worker_teste2_b.c
- teste-spawn3.c

  worker_teste3_a.c worker_teste3_b.c
- teste-spawn3.c

  worker_teste3_a.c worker_teste3_b.c

soma_vetor

- Mensagens entre processos do mesmo grupo
- Mensagens entre o processo pai e os processos dos 2 grupos

teste-soma-vets.c : soma dos elementos dos vetores A e B

worker-soma-a.c

worker-soma-b.c

Exercicio

- Processo principal cria 2 grupos
- Grupo1: executa X= AXB
- Grupo2: executa Y=CXD

A,B,C,D matrizes de double 1000X1000

Enviar para processo principal: X e Y

MPI + OpenMP

- soma2-mpi-openmp.c : soma dos elementos de uma matriz (n/n_nos linhas para cada nó e cada nó executa as linhas a ele atribuído para lelamente utilizando OpenMP)

Multiplicação de matrizes

- Processo 0: X=AXB (4 cores - OpenMP)
- Processo1: Y=CXD (4 cores - OpenMP)
- Processo0: Z=X+Y (4 cores - OpenMP)

- Matrizes 1000 X 1000 double