Programação em Computação Paralela e Distribuída

Laboratório, ERBASE 2009

Programação Híbrida

Domingo Giménez

Universidad de Murcia, Spain

Grupo de Computación Paralela

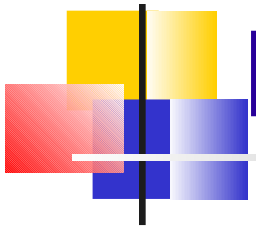
http://dis.um.es/~domingo

Murilo Boratto

UFBa

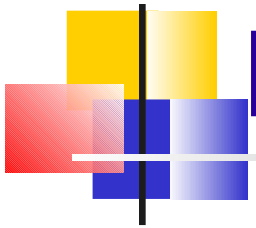
Leandro Coelho

UNEB



Noções básicas

- A maioria dos sistemas actuais permitem programación híbrida
- Os clusters estão formados por nodos multicore (OpenMP) conectados por uma rede (MPI)
- Algumas vantagens da programación híbrida:
  - pode ajudar a aumentar a escalabilidad
  - em aplicações que combinam paralelismo de pequena e grande granularidade
  - quando se mistura paralelismo funcional e de dados
  - quando o número de processos MPI é fixo



Esquema básico

- São possíveis vários esquemas
- O mais simples é MPI + OpenMP:
  - Inicializam-se processos MPI (possivelmente um na cada processador)
  - e a cada processo atua como mestre de grupos de escravos que se criam com OpenMP

Exemplo

hello_mpi+omp.c

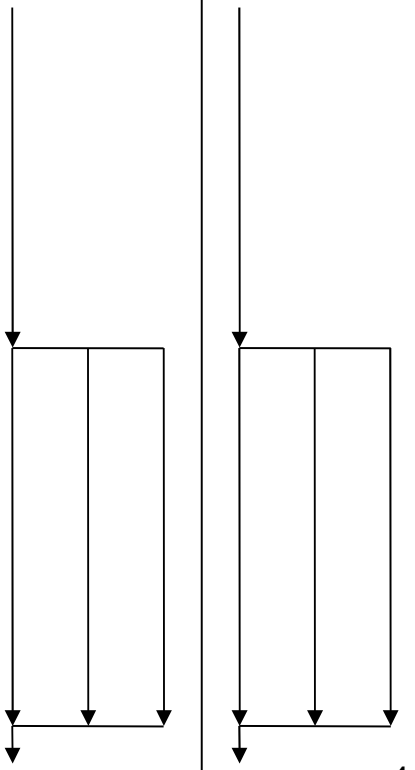
```
#include <mpi.h>
#include <omp.h>
int main( int argc, char *argv[]) {
   int nthreads, nprocs, idpro, idthr, namelen;
   char processor_name[MPI_MAX_PROCESSOR_NAME];
   MPI_Init(&argc,&argv);
   MPI_Comm_size(MPI_COMM_WORLD,&nprocs);
   MPI_Comm_rank(MPI_COMM_WORLD,&idpro);
   MPI_Get_processor_name(processor_name,&namelen);
#pragma omp parallel private(idthr) firstprivate(idpro,processor_name)
   {
      idthr = omp_get_thread_num();
      printf("Hola ...",idthr,idpro,processor_name);
      if (idthr == 0)  {
         nthreads = omp_get_num_threads();
         printf("En el ...",idpro,nthreads,nprocs);
      } }
   MPI_Finalize();
}
```

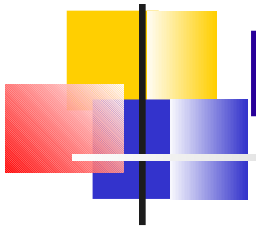
Todos os processos executam o mesmo código

As variables estão replicadas nas memórias de todos os processos

Trabalham com MPI entre Init e Finalize

A cada processo cria threads com parallel





Exemplo

compilação:

mpicc hello_mpi+omp.c ... -fopenmp

execução:

$> mpirun -np 4 hello_mpi+omp

executa 4 processos iguais nos processadores estabelecidos por defeito

a cada processo põe em marcha o número de threads estabelecido por defeito no processador ao que se atribuiu

– outro exemplo: **buscar_mpi+omp.c**

procura um dado dentro de um array

Multiplicação de matrices

- OpenMP: **codigo6-6.c**
- MPI: **codigo6-10.c**
- MPI+OpenMP: **mmhibrido.c**