MGSC

Cómputo Paralelo y Distribuido.
Ejercicios de OpenMP+MPI
ITESM
Septiembre 2014.

Ejercicios.

1. Probar el primer ejemplo con los hilos y procesos por default de la computadora. (Ejemplo 1). (Calificación 0.4%).
2. Probar el primer ejemplo con cuatro hilos y dos procesos. (Ejemplo 1). (Calificación 0.4%).
3. Probar el primer ejemplo con cuatro hilos y cuatro procesos. (Ejemplo 1). (Calificación 0.4%).
4. Probar el primer ejemplo con dos hilos y cuatro procesos. (Ejemplo 1). (Calificación 0.4%).
5. Realizar la suma de dos vectores utilizando OpenMP y MPI con dos estrategias de solución. Compara el tiempo de ejecución de las dos estrategias. (Calificación 1.7%).
6. Realizar el ordenamiento de un matriz de *N* elementos por el método de la burbuja con OpenMP y MPI. (Calificación 1.7%). Se puede realizar en binas.

Nota: Los ejercicios 5 y 6 los pueden subir al blackboard hasta el domingo 5 de octubre a las 11:59. Los alumnos que hayan terminado los ejercicios el viernes 3 de octubre, lo puedo revisar en la clase.

```
/*
Ejemplo 1 de mpi+openmp
*/
#include <stdio.h>
#include "mpi.h"
#include <omp.h>
int main(int argc, char *argv[]) {
  int numprocs, rank, namelen;
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int soy = 0, np = 1;
MPI_Init(&argc, &argv); //inicia el paralelismo con mpi
  MPI_Comm_size(MPI_COMM_WORLD, &numprocs); //obtiene el número de
proceso
  MPI_Comm_rank(MPI_COMM_WORLD, &rank); // rango. Cantidad de procesos
totales
  MPI_Get_processor_name(processor_name, &namelen); //nombre de la
computadora
  #pragma omp parallel default(shared) private(soy, np)
  {
    np = omp_get_num_threads(); //obtiene el número de hilos totales
    soy = omp_get_thread_num(); //obtiene el número del hilo dentro del total
    printf("Hola... Desde el hilo %d de un total de %d procesadores (hilos) ejecutado
dentro del proceso %d de un total de %d procesos en %s\n",
```

```
        soy, np, rank, numprocs, processor_name);
 }
 MPI_Finalize(); //finaliza paralelismo con mpi
}
```