

Paralelismo

OpenMP

Francisco García Sánchez

Departamento de Informática y Sistemas

UNIVERSIDAD DE
MURCIA



Contenido

- Introducción
- 1) Concurso de programación paralela
- 2) Acceso al sistema
- 3) Multiplicación de matrices
- 4) Ordenación por mezcla

Introducción

- **OpenMP**: API para programación multiproceso de **memoria compartida**
 - Añadir concurrencia a programas en C, C++, Fortran
 - Modelo **fork-join**: *el hilo principal crea varios hilos para que realicen una tarea de forma paralela, y él se queda esperando que todos los hilos creados terminen antes de continuar*
 - Disponible para Unix y Microsoft Windows

Introducción

- Cluster de 4 nodos, cada uno con 8 cores
 - **OpenMP (¡¡memoria compartida!!): usar un único nodo**

- Usuarios

• AED1 → AED1234
• AED2 → AED2345

• AED3 → AED3456
• AED4 → AED4567

- Puntuación adicional de hasta **0.3**
- Mandar fichero txt con respuestas a

frgarcia@um.es

1) Concurso de programación paralela

- Consultar instrucciones de la prueba de calentamiento de la edición de 2011 en La Laguna

➤ <http://cpp.fpcmur.es/2011/>

➤ <http://cpp.fpcmur.es/2011/data/cpp11-calentamiento.tar.gz>

❖ **Cuestión:** *Identificar el sistema con el que se realizan las pruebas e indicar como lanzar secuencial y OpenMP*

2) Acceso al sistema

- Dos problemas:
 - Multiplicación de matrices
 - Ordenación por mergesort

- ❖ Recoger información de los problemas
- ❖ **Cuestión:** *Ejecutar los problemas secuenciales para multiplicación de matrices y ordenación por mergesort; ¿qué tiempo de ejecución se tiene en ambos casos?*

3) Multiplicación de matrices

- Cuatro programas para multiplicación de matrices

➤ Examinar «**esquema.c**» y «**mmsec.c**»

❖ **Cuestión 1:** *Modificar programa secuencial para mejorar forma de acceder a los datos. Obtener el tiempo de ejecución y compararlo con la versión secuencial anterior*

3) Multiplicación de matrices

- Cuatro programas para multiplicación de matrices

➤ Examinar «[mmomp.c](#)»

❖ **Cuestión2:** *Ejecutar `mmomp` cambiando el número de threads de 1 a 8. Obtener tiempos y comparar. ¿Con cuál se obtiene el menor tiempo? ¿Cuál es el máximo valor alcanzado del speed-up?*

3) Multiplicación de matrices

- Cuatro programas para multiplicación de matrices

➤ Modificar «**mmomp.c**»

❖ **Cuestión3:** Programar versión OpenMP mejorando acceso a memoria. Ejecutar cambiando número de threads de 1 a 8. Obtener tiempos y comparar. ¿Con cuál se obtiene el menor tiempo? ¿Cuál es el máximo valor alcanzado del speed-up?

4) Ordenación por mezcla

- Se proporciona únicamente programa secuencial

➤ Examinar «**esquema.c**» y «**mssec.c**»

❖ **Cuestión:** *Realizar versión OpenMP (utilizar ejemplos vistos en teoría). Ejecutar cambiando número de threads de 1 a 8. Obtener tiempos y comparar. ¿Con cuál se obtiene el menor tiempo? ¿Cuál es el máximo valor alcanzado del speed-up?*