

Practicas de Multiprocesadores

Primavera 2008

Paralelización con OpenMP

Objetivo de la practica

Paralelizar con OpenMP el programa LUBb, con el objetivo de reducir su tiempo de ejecución y obtener una buena escalabilidad al aumentar el número de procesadores. Principalmente se explorarán dos estrategias de paralelización distintas:

- Paralelización de grano grueso de los bucles que recorren bloques de BxB elementos
- Paralelización de grano fino de los bucles que recorren cada uno de los bloques

Al final de la práctica, deberéis entregar a través del “Racó de la FIB” las versiones paralelas del código junto con gráficas de escalabilidad y las observaciones que consideréis necesarias. Las respuestas a preguntas de explicación del rendimiento observado deben apoyarse en vistas o mediciones realizadas con Paraver.

Material para la practica

Copiar ../acmp/paral_OpenMP.tar. Mirar Makefile y run.sh. Al igual que la práctica anterior, se compila con el flag -openmp. La variable de entorno OMP_NUM_THREADS indica con cuantos threads (procesadores) se abrirá paralelismo.

Profiling de la ejecución secuencial con gprof

Compilar con el flag -p activado de manera que al ejecutar se genere un profile del tiempo empleado en las rutinas que componen el programa LUB.c.

PREGUNTA:

- P1. ¿Cuál es el tiempo dedicado y el número de llamadas realizado a cada una de las rutinas que componen el programa LUB.c.

Paralelización de grano grueso

En primera opción, paralelizaremos los bucles en el programa principal que recorren los bloques. Para ello debereis analizar las dependencias que existen en la ejecución de estos bucles.

PREGUNTAS:

- P2. Modificar el programa principal para explotar el paralelismo de grano grueso.
- P3. Dibujar la gráfica de escalabilidad para la ejecución paralela utilizando entre 2 y 32 procesadores.
- P4. ¿A que se debe la pérdida de escalabilidad obtenida al aumentar el número de procesadores?

Paralelización de grano fino

En esta segunda opción, exploraremos la posibilidad de paralelizar los bucles internos en las funciones que más tiempo consumen. Para ello deberías analizar las dependencias existentes en los bucles y realizar transformaciones en los mismos (intercambio de bucles) con el objetivo de maximizar el grano de la paralelización.

PREGUNTAS:

- P5. Modificar el código de las funciones para explotar de forma eficiente el paralelismo de grano fino.
- P6. Completar la gráfica de escalabilidad para la ejecución paralela de esta versión utilizando entre 2 y 32 procesadores.
- P7. ¿A que se debe la poca escalabilidad obtenida al aumentar el número de procesadores?

Otras posibles estrategias de paralelización

En esta última opción, se os pide que penséis en alguna transformación simple en el código que os permita mejorar la eficiencia de las paralelizaciones anteriores (se pueden buscar otros esquemas de paralelización, variar tamaño de bloque, ...).

PREGUNTAS:

- P8. Modificar las versiones paralelas adecuadamente para reflejar las transformaciones que hayáis pensado.
- P9. Completar la gráfica de escalabilidad para la ejecución paralela de esta versión utilizando entre 2 y 32 procesadores.

