

Arquitectura y Programación de Altas Prestaciones

Práctica 3: Algoritmo a Elegir



Especialidad Ingeniería de Computadores
Departamento de Arquitectura y Tecnología
de los Computadores
Maribel García Arenas
maribel@atc.ugr.es



Enunciado

- Desarrollar un programa paralelo con MPI que sea escalable partiendo de un algoritmo secuencial previamente contrastado con la profesora.
- Compilarlo y ejecutarlo en el Aula de prácticas y en atcgrid.ugr.es con una carga homogénea por proceso, comprobando la evolución del tiempo de ejecución.
- Representar gráficamente el tiempo de ejecución total y la ganancia en velocidad en función del número de procesos o nodos para tres cargas de trabajo diferentes.





Programación

- Modifica el algoritmo anterior incorporando alguna mejora o cambio y trata de explicar como afecta dicho cambio al desarrollo del algoritmo contestando a preguntas cómo:
 - ¿Qué cambios he llevado a cabo y por qué?
 - ¿Es mejor esta versión que la anterior?
 - ¿Cuánto tiempo de ejecución he mejorado y por qué?
- Incorpora mediciones de tiempos para cada una de las zonas del programa incluyendo como mínimo las zonas:
 - Creación y destrucción de procesos
 - Cálculo
 - Recepción de resultados





Programación II

- Recomendaciones de trabajo:
 - Establece y fija las cargas de trabajo (tamaño de la imagen, tamaño de los datos o carga computacional que va a llegar a cabo el algoritmo) con la que vas a trabajar
 - Mide el tiempo de ejecución para al menos tres configuraciones de carga diferente para cada uno de los conjuntos de datos utilizados, para ello varía el número de procesos enviados a cada nodo, ya sea en el aula o en atcgrid.
 - Calcula la ganancia en velocidad obtenida para cada una de las configuraciones para cada conjunto de datos
 - Representa una gráfica que ponga de manifiesto la ganancia obtenida por cada conjunto de trabajo. Ten en cuenta que se debe poder apreciar las diferentes ganancias para compararlas entre ellas.
 - Realiza las mediciones al menos tres veces cada una y utiliza siempre el menor tiempo para obtener el caso más favorable.



Recomendaciones

- Fuentes de inspiración:<http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf>
- Se recomienda analizar el problema mediante el algoritmo secuencial
- Variar la versión secuencial para aceptar parámetros en línea.
- Utilizar la definición de constante cuando sea adecuado, son más rápidas
- Realizar un análisis del problema intentando averiguar las tareas que pueden ser paralelizables, analizando las dependencias y estableciendo el tipo de paralelización más adecuada para una máquina de memoria distribuida como la que podéis montar en el laboratorio o atcgrid.
- Implementar una solución paralela inicial con MPI.
- Comprobar que los datos obtenidos son los válidos mediante la representación gráfica de los tiempos menores.
- La práctica se entrega:
 - Hoja de calculo donde se detalle al principio las diferentes cargas de trabajo a las que se va a someter el algoritmo. Por ejemplo si son imágenes, qué tamaño tienen las imágenes, si ejecuta algo con matrices, qué tamaño de matrices se va a tener en cuenta, etc.
 - Detalle de los experimentos realizados, dejando claro la carga de trabajo que realiza cada proceso en cada caso.
 - Tiempos de ejecución que no incluyan la Entrada/Salida
 - Gráficas comparativas de las ganancias obtenidas si las hubiera
 - Conclusiones obtenidas
 - Código secuencial
 - Código parametrizado paralelo



¿Fecha de Entrega?

