PRÁCTICA 1. MPI

1. INTRODUCCIÓN

En este documento se describe la práctica 1 de las asignaturas Arquitectura de Altas Prestaciones del Grado de Ciencia e Ingeniería de Datos y Computación de Altas Prestaciones del Grado de Ingeniería de Computadores.

Esta práctica es la primera de las dos prácticas obligatorias de la asignatura.

2. DESCRIPCIÓN GENERAL

Esta práctica tiene como objetivo poner en práctica los conocimientos adquiridos sobre paralelización en memoria compartida y en memoria distribuida en CPU, durante el transcurso de la asignatura.

Para ello será necesario desarrollar un programa que analice una gran cantidad de datos en el mínimo tiempo posible. Concretamente, los alumnos deberán agrupar los datos utilizando el algoritmo k-medias para generar grupos, y calcular las siguientes estadísticas para cada una de las columnas de los datos: media, mínimo, máximo y varianza.

Los alumnos deberán seleccionar y justificar las estrategias de paralelización, distinguiendo las secciones de grano fino y grano grueso, y combinando paralelización en memoria distribuida mediante OpenMPI con memoria compartida mediante OpenMP.

3. ALGORITMO K-MEDIAS

Se trata de un algoritmo de agrupación de datos en un número predefinido de grupos tal que cada dato pertenece al grupo con el valor medio más cercano (salvo que se realice el apartado optativo N, ese número de grupos coincidirá con el número de nodos de MPI). Se trata de un algoritmo muy utilizado en minería de datos, y su descripción puede encontrarse fácilmente online.

FORMATO DE LOS DATOS DE ENTRADA

El programa deberá ser capaz de aceptar cualquier volumen de datos que cumplan con el siguiente formato binario:

UINT32 – Number of rows - 04 bytes

UINT32 – Number of columns - 04 bytes

foreach row:

  foreach column:

    REAL32 -Value - 04 bytes

  end

end

ESTRUCTURAS DE DATOS DEL ALGORITMO

Es un algoritmo iterativo que trabaja con un conjunto de centroides (valores medios de los grupos), que va desplazando, y un conjunto de grupos de puntos que intercambian puntos entre iteraciones. Puede inicializarse con un valor inicial para cualquiera de los dos conjuntos.

En este caso, se utilizará una distribución inicial del conjunto de datos en grupos (equitativa y basada en el índice de los datos), y se procederá a calcular los centroides

ACTUALIZACION DE LOS CENTROIDES

Cada vez que el conjunto de datos de un grupo cambie es necesario actualizar su centroide (un punto formado por el valor medio en cada columna de los datos que forman el grupo).

Todos los nodos deberán comunicarse las actualizaciones de los centroides para el siguiente paso.

ACTUALIZACION DE LOS GRUPOS

Cada vez que los centroides se actualicen, se deberá calcular para cada punto la distancia al centroide, y asignar el punto al grupo del que tenga mínima distancia (recuerda que, si la distancia es mínima, la distancia al cuadrado también lo es, y viceversa). Si cada nodo está manejando uno de los grupos, deberá comunicarse con otros nodos para enviarles los puntos que dejen de pertenecerle.

CRITERIO DE FINALIZACION

En función del conjunto de datos, y de la inicialización, el algoritmo puede estancarse en un ciclo iterativo infinito, sin que dejen de moverse pequeñas cantidades de datos entre los grupos, y desplazamientos poco significativos de los centroides.

El algoritmo debe incluir los siguientes criterios de finalización:

- Que el número total de puntos desplazados (que cambian de grupo) sea menor al 5% (O el total de desplazados respecto al total de datos, o que todos los nodos cumplan que sus desplazados sean menores que el 5% de sus datos).

- Que el total de iteraciones sea inferior a 2000

EJEMPLO

Para un conjunto de 1000 filas y tres columnas (x,y, z) y 4 grupos gestionados por 4 nodos:

Se dividen los datos entre los nodos (el 0 los datos 0 a 249, el 1 los datos del 250 al 499…).

Cada nodo calcula su centroide (xCentroide = suma de las x / número de datos, igual para yCentroide…)

Los nodos se comunican los centroides, de forma que todos conocen todos los centroides.

Cada nodo calcula, para cada punto, su distancia a todos los centroides. Si el punto es más cercano a otro centroide, deberá enviarse al nodo correspondiente. Después de mover todos los puntos se actualizarán los centroides, y se continuará el bucle.

ANALISIS BASICO DE DATOS

Además de aplicar el algoritmo k-medias, el programa deberá calcular las estadísticas previamente mencionadas. Para el ejemplo anterior, habría que calcular la "x" mínima, máxima, media, y la varianza en "x", y lo mismo para "y" y para "z".

5. OBJETIVO DE LA PRÁCTICA

Tal y como se ha ido comentando en apartados anteriores, lo que se pide es el desarrollo de una aplicación con una algoritmia simple en la que se enfatiza el rendimiento, y se pondrán en práctica todos los conocimientos adquiridos en las áreas de paralelización en CPU.

El alumno deberá justificar sus decisiones de diseño, bien basándose en la teoría vista en clase, o en pruebas experimentales. Esto incluye las estructuras de datos utilizadas, el reparto de carga en paralelización, y el uso en cada caso de OpenMP u OpenMPI.

6. CRITERIOS DE EVALUACIÓN

Aparte de los requisitos descritos, se valorará la claridad, limpieza, organización y planteamiento de la memoria: se tendrá en cuenta la calidad de esta (ver apartado 9), ya que la memoria representa el resultado del trabajo realizado. Como consecuencia, esta práctica podrá ser considerada suspensa si la memoria no alcanza un nivel mínimo de calidad. La puntuación máxima es de 10 puntos, pero los apartados de evaluación suman un total de 11. Esto significa que la nota máxima puede alcanzarse por varias vías. Cualquier puntuación por encima de 10 se truncará a un 10.

Leer los datos en binario, y desarrollar el algoritmo en serie (sin usar OpenMPI ni OpenMP), presentando los resultados en pantalla. Para presentar la distribución en grupos, se calculará la distancia cuadrática media de los puntos a su centroide (un valor por centroide). (3 puntos)

Paralelizar mediante OpenMP las secciones de código apropiadas. Se penalizarán las condiciones de carrera, y los bloqueos innecesarios de hilos (2 puntos).

Paralelizar mediante OpenMPI las secciones de código apropiadas. Se penalizará el exceso de mensajes, y los bloqueos innecesarios de nodos. (2 puntos)

Estudiar en un entorno de memoria distribuida distintas configuraciones y repartos entre nodos e hilos. (1 punto)

Claridad, limpieza y organización de la memoria. Se penalizará la aparición de figuras sin referenciar en el texto, o de resultados numéricos sin explicar. (2 puntos)

Se podrá obtener hasta un punto extra en función de la velocidad final de la aplicación (medida por el profesor en un entorno y con una configuración que se comunicará previamente a los alumnos). La aplicación más lenta recibirá 0 puntos en este apartado, y la más rápida un punto, asignando valores intermedios proporcionales al tiempo conseguido al resto de prácticas. (1 punto)

7. METODOLOGÍA

La práctica se realiza individualmente y se entrega a través del campus virtual antes de la fecha límite fijada para ello. Al igual que cualquier otro trabajo, se aplicará, llegado el caso, la normativa anticopia de la universidad.

8. ENTREGABLES

Se deben entregar, por cada grupo, dos archivos comprimidos (zip o 7z):

- Memoria de la práctica (*pdf)*, debe seguir las recomendaciones descritas en el apartado 9.
- Archivo comprimido que incluya el código desarrollado, y un archivo README indicando los pasos necesarios para compilar y ejecutar la práctica en una consola Linux .
- En caso de utilizar una IA para la generación total o parcial del código de la práctica, se deberá indicar en la memoria y adjuntar así mismo un documento con la conversación/conversaciones con la IA que han dado como producto el código entregado.

9. RECOMENDACIONES PARA LA REDACCIÓN DE MEMORIAS DE PRÁCTICAS

Se recomienda seguir las recomendaciones de escritura de documentos técnicos en general y de memorias de trabajos prácticos en particular. Principalmente:

- Las memorias se entregan en formato *pdf*.
- El documento debe incluir:
    - Portada: con información sobre curso, grado, asignatura, autores y título.
    - Índice. Opcionalmente, índice de tablas, de figuras y de expresiones matemáticas.
    - Introducción: con una descripción breve de la finalidad del documento, contexto y contenido de la memoria.
    - Conclusiones: consideraciones sobre lo que ha supuesto la realización del trabajo.
    - Bibliografía: listado de referencias empleadas en el documento (usar referencias cruzadas para referirse a ellas en el documento).
- Usar un lenguaje basado en formas no personales.
- Incluir siempre títulos en las figuras, tablas y expresiones matemáticas (si se diera el caso), de forma que se pueda hacer referencia a ellas en cualquier parte del texto.
- Incluir paginación en el documento.
- Numerar los apartados, capítulos o secciones para facilitar la referencia a ellos si fuera necesario.
- Si se incluye código en la memoria, no incluirlo como imagen sino como texto aplicando un formato distinto.
- Revisar a conciencia la ortografía y gramática, incluyendo los signos de puntuación.
- Justificar el texto.
- Si el documento incluye gráficas, estas deben incluir títulos y unidades de cada eje y una leyenda con la descripción de cada serie de datos. También título y opcionalmente un subtítulo.
- Se deben minimizar los espacios en blanco entre figuras o tablas, el texto debe rellenarlos y hacer referencia a las figuras o tablas cuando corresponda.