

Múltiplos de las medianas de una matriz

Definición del problema

Dada una matriz A de $(N+1) \times N$ números enteros, se deberá calcular la mediana de cada una de las filas de la matriz. Para obtener la mediana de un vector (fila de la matriz), se debe ordenar el vector y seleccionar el elemento que esté en el punto central. Una vez obtenidas las medianas, se debe calcular cuántos múltiplos de cada una de ellas hay en la matriz. En caso de que la mediana sea 0, el resultado será 0. Por ejemplo:

Matriz A

6	-2	5	5	8	-6	-3
5	-4	1	0	2	-2	0
9	8	2	4	-7	2	-9
-5	-7	6	-6	-1	-7	-7
9	-5	0	6	2	-4	2
0	0	9	-4	5	0	5
-1	7	-4	-2	6	8	-8
8	0	-8	-6	3	7	-3

Matriz A con las filas ordenadas

6	-2	5	5	8	-6	-3
5	-4	1	0	2	-2	0
9	8	2	4	-7	2	-9
-5	-7	6	-6	-1	-7	-7
9	-5	0	6	2	-4	2
0	0	9	-4	5	0	5
-1	7	-4	-2	6	8	-8
8	0	-8	-6	3	7	-3

Medianas y múltiplos

5	0	2	-6	2	0	-1	0
14	0	33	14	33	0	56	0

Para realizar la ordenación de las filas de la matriz deberá implementarse un método propio (no se pueden usar las funciones de ordenación disponibles en c/c++).

Ficheros suministrados

Se suministran los siguientes ficheros:

- MedianasMultiplos_sec.cpp → Esqueleto para la implementación de la versión secuencial.
- MedianasMultiplos_ocl.cpp → Esqueleto para la implementación de la versión OpenCL.
- MedianasMultiplos_mpi_ocl.cpp → Esqueleto para la implementación de la versión híbrida.

Los tres ficheros `MedianasMultiplos_xxx.cpp` se encargan de comprobar que los parámetros de entrada son correctos, obtener los parámetros de ejecución a partir del fichero de entrada, inicialización de estructuras y medición de tiempos de ejecución. Se debe añadir el código necesario para realizar las tareas concretas del problema entre “**** IMPLEMENTACIÓN ****” y “**** FIN IMPLEMENTACIÓN ****” (esto es una sugerencia, si se creyera necesario añadir cosas fuera de estas zonas, puede hacerse).

- `fichEntrada` → Fichero de entrada de ejemplo.
- `MedianasMultiplos` → Ejecutable ya compilado con una versión serie del problema para poder probar que la implementación realizada funciona correctamente.

Compilación

A continuación se muestran los comandos básicos de compilación. En caso de que se utilice alguna librería y/o archivo de código extra, deberán incluirse en la compilación:

- Secuencial: `g++ MedianasMultiplos_sec.cpp [OtroFich.cpp] -o ejecutable [-lOtraLib]`
- OpenCL: `g++ MedianasMultiplos_ocl.cpp [OtroFich.cpp] -o ejecutable -lOpenCL [-lOtraLib]`
- Híbrida: `mpic++ MedianasMultiplos_ocl.cpp [OtroFich.cpp] -o ejecutable -lOpenCL [-lOtraLib]`

Donde *ejecutable* es el nombre del ejecutable que se generará, *OtroFich.cpp* sería otro fichero necesario para la compilación (se añadirían todos los necesarios) y *-lOtraLib* sería otra librería necesaria para la compilación (se añadirían todas las necesarias). Tanto *OtroFich.cpp* como *-lOtraLib* son optativos y dependerán únicamente de las implementaciones concretas de cada trabajo, por lo que habrá trabajos que requieran de uno o más de estos valores y otros que no requerirán ninguno.

Ejecución

La ejecución del programa será de la siguiente forma:

```
[mpiexec [-n num_proc] [-f lista_hosts]] ./ejecutable fichEntrada [-d] [-h num_hilos]
```

donde:

- `mpiexec [-n num_proc] [-f lista_hosts]` → Para ejecuciones MPI, donde *-n num_proc* (opcional) indica el número de procesos a lanzar y *-f lista_hosts* (opcional) indica el fichero que contiene las direcciones de los nodos que ejecutarán los procesos MPI.
- *fichEntrada* → Fichero con los parámetros de ejecución del programa.

- *-d* → (Opcional) bandera que hará que el programa muestre los valores de los datos iniciales y finales de cada experimento así como su tiempo de ejecución.
- *-wi work_items* → Opcional. Si se indica, se lanzarán tantos work items como se indique en *work_items* (para ejecuciones OpenCL e híbrida).
- *-wi_wg workitems_por_workgroup* → Opcional. Si se indica, se lanzarán tantos work items en cada work group como se indique en *WorkItems_por_WorkGroup* (para ejecuciones OpenCL e híbrida).

Al finalizar, el programa mostrará por pantalla el tiempo de ejecución acumulado de todos los experimentos realizados.

Formato del fichero de entrada

Los valores de los diferentes elementos que componen el ejercicio deberán ser indicados mediante un fichero de entrada. El formato de dicho fichero será el siguiente:

Número de experimentos → Entero

Tamaño de la matriz del experimento 1 → Entero

Semilla del experimento 1 → Entero

Límite inferior de los valores de la matriz del experimento 1 → Entero

Límite superior de los valores de la matriz del experimento 1 → Entero

Los valores a partir de la línea 2 se repetirán tantas veces como experimentos se deseen realizar. Los valores pueden estar separados por saltos de línea o por espacios (se recomienda, para que sea más fácil de entender, usar una línea por experimento). Por otra parte, para poder calcular correctamente el valor mediano, el número de columnas de la matriz debe ser impar (en caso de ser un valor par, no habría un dato en el centro).

IMPORTANTE: No se deberán usar letras ni líneas en blanco ya que el programa principal no hace ningún control sobre el formato del fichero, simplemente lee números de forma consecutiva. Además, se leerán tantos grupos de parámetros como se indique en "Número de experimentos", por lo que si se añaden grupos adicionales, no se tendrán en cuenta, y si faltan grupos, el comportamiento será impredecible.

Ejemplo de fichero de entrada:

```
2
11 1 -1000 1000
1021 4 -994 994
```

El fichero contiene dos experimentos:

1. Se genera una matriz de 12x11 de forma aleatoria usando la semilla 1 con valores comprendidos en el rango (-1000, 1000).
2. Se genera una matriz de 1022x1021 de forma aleatoria usando la semilla 4 con valores comprendidos en el rango (-994, 994).

A la hora de realizar experimentos, el valor que más influirá en el tiempo de ejecución es el tamaño de la matriz, ya que influye tanto en la ordenación de cada fila (a más elementos a ordenar, más tiempo), como en el cálculo de los múltiplos (a más números, más comprobaciones).