

Acumulación de submatrices al cuadrado

Definición del problema

Dada una matriz A de $N \times N$ números reales y un conjunto de ternas del tipo (x, y, t) que definen submatrices dentro de A en la posición inicial (x, y) con tamaño t , se deberán obtener todas las submatrices, calcular su cuadrado y sumar el resultado a A. La acumulación de una submatriz en A no se debe tener en cuenta para la siguiente submatriz. Por ejemplo:

Matriz A:

-9,24	-7,69	-2,98	2,61	4,98	-9,98
-4,00	1,33	2,65	2,15	8,70	9,63
7,99	-5,10	2,12	6,86	9,64	-1,78
3,31	9,19	6,56	9,68	-7,36	5,82
-2,71	6,49	6,42	7,53	7,23	8,43
5,13	7,89	-9,25	-7,85	0,60	5,72

Lista de ternas:

(0, 2, 3), (2, 1, 3)

Sintaxis:

$A(x,y,t) \rightarrow$ Submatriz con origen en (x,y) y tamaño t

$$A'=A$$

Obtenemos la submatriz $S_1=A(0,2,3)$:

7,99	-5,10	2,12
3,31	9,19	6,56
-2,71	6,49	6,42

Calculamos S_1^2 :

41,21	-73,86	-2,91
39,09	110,15	109,42
-17,57	115,13	78,05

Acumulamos en la matriz $A'(0,2,3)=A(0,2,3)+S_1^2$

-9,24	-7,69	-2,98	2,61	4,98	-9,98
-4,00	1,33	2,65	2,15	8,70	9,63
49,20	-78,96	-0,79	6,86	9,64	-1,78
42,40	119,34	115,98	9,68	-7,36	5,82
-20,28	121,62	84,47	7,53	7,23	8,43
5,13	7,89	-9,25	-7,85	0,60	5,72

Obtenemos la submatriz $S_2=A(2,1,3)$

2,65	2,15	8,70
2,12	6,86	9,64
6,56	9,68	-7,36

Calculamos S_2^2 :

68,65	104,66	-20,25
83,40	144,93	13,62
-10,38	9,26	204,56

Acumulamos en la matriz $A'(2,1,3)=A'(2,1,3)+S_2^2$

-9,24	-7,69	-2,98	2,61	4,98	-9,98
-4,00	1,33	71,30	106,81	-11,55	9,63
49,20	-78,96	82,61	151,79	23,26	-1,78
42,40	119,34	105,60	18,94	197,20	5,82
-20,28	121,62	84,47	7,53	7,23	8,43
5,13	7,89	-9,25	-7,85	0,60	5,72

Para obtener una submatriz, se considerará la matriz A de forma cíclica, es decir, que si A es de 6×6 y tenemos la terna $(3,0,5)$, las tres primeras columnas de la submatriz serán

las columnas 3, 4 y 5 de A, mientras que las otras dos columnas de la submatriz serán las columnas 0 y 1 de A. Lo mismo ocurriría con las filas.

Ficheros suministrados

Se suministran los siguientes ficheros:

- Submatrices_sec.cpp → Esqueleto para la implementación de la versión secuencial.
- Submatrices_ocl.cpp → Esqueleto para la implementación de la versión OpenCL.
- Submatrices_mpi_ocl.cpp → Esqueleto para la implementación de la versión híbrida.

Los tres ficheros Submatrices_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.
- Submatrices → 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++ Submatrices_sec.cpp [OtroFich.cpp] -o ejecutable [-lOtraLib]
- OpenCL: g++ Submatrices_ocl.cpp [OtroFich.cpp] -o ejecutable -lOpenCL [-lOtraLib]
- Híbrida: mpic++ Submatrices_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

Número de submatrices 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).

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

5 102 5

600 3427 12

El fichero contiene dos experimentos:

1. Se generan una matriz de 5×5 y 5 ternas (submatrices) de forma aleatoria usando la semilla 102.
2. Se generan una matriz de 600×600 y 12 ternas (submatrices) de forma aleatoria usando la semilla 102.

A la hora de realizar experimentos, el valor que más influirá en el tiempo de ejecución es el número de submatrices (a más submatrices, más operaciones). También influirá el tamaño de la matriz A, ya que, a mayor tamaño, las submatrices podrán ser más grandes.