



UNIVERSIDAD DE ALMERÍA

Computación en procesadores gráficos

Programación con CUDA Ejercicios 4 - 7

José Antonio Martínez García
Francisco M. Vázquez López
Manuel Ujaldón Martínez
Ester Martín Garzón



Contenidos

- Arquitectura de las GPUs
- **Modelo de programación SIMT**
- **Claves computacionales para la programación de GPUs**
- **Programación con CUDA: Ejercicios 4 - 7**
- Supercomputación gráfica y arquitecturas emergentes



Programación con CUDA

- **Ejercicio 4:** `~ejercicios/reverseArray/singleblock`
 - `reverseArray_singleblock.cu`, `reverseArraysb_kernel.cu`
 - `nvcc reverseArray_singleblock.cu -o reverseArray`
 - ¿ Cuantos bloques se ejecutan ?
 - ¿ De qué tamaño es el bloque ?
 - ¿ Cuantos SMs trabajan ?
 - ¿ Qué hace cada thread ?
 - `d_out[255]=d_in[0]`
 - `d_out[254]=d_in[1]`
 - `d_out[253]=d_in[2]`
 - ¿ Se realiza un acceso coalescente en `d_in` y `d_out` ?
 - Para compute capability 1.0 y 1.1
 - Para compute capability 1.2 y 1.3



Programación con CUDA

- **Ejercicio 5:** ~ejercicios/reverseArray/multiblock
 - reverseArray_multiblock.cu, reverseArraymb_kernel.cu
 - `nvcc reverseArray_multiblock.cu -o reverseArray`
 - ¿ Tamaño de bloque ?
 - ¿ Número de bloques ?
 - ¿ Cuantos SMs trabajan ?
 - ¿ Qué valores toman inOffset y outOffset ?
 - ¿ Qué valores toman in y out ?
 - ¿ Coalescencia en d_in y d_out ?



Programación con CUDA

- **Ejercicio 6:** ~ejercicios/reverseArray/multiblockfast
 - reverseArray_multiblockfast.cu, reverseArraymbf_kernel.cu
 - `nvcc reverseArray_multiblockfast.cu -o reverseArray`
 - ¿ Qué se guarda en sdata ?
 - ¿ Se realiza la copia en d_out igual que en los ejercicios anteriores ?
 - `sdata[9]=d_in[0]`
 - `sdata[8]=d_in[1]`
 - `sdata[0]=d_in[9]`
 - `d_out[50]=sdata[0] // d_in[9]`
 - `d_out[51]=sdata[1] // d_in[8]`
 - `d_out[59]=sdata[9] // d_in[0]`



Programación con CUDA

■ Ejercicio 7: ~ejercicios/vecAdd

- `vecAdd.cu, vecAdd_kernel.cu` `nvcc vecAdd.cu -o vecAdd`
- `./vecAdd 100 256 1 //N = 100, Tamaño bloque = 256, verbose = 1`
- Flags entrada: N: Tamaño vector, MAX_BLOCKSIZE: Tamaño de bloque, verbose: Salida por pantalla
- El programa realiza la suma de dos vectores: $C = A + B$
- Medir tiempos para: Copia datos GPU, ejecución kernel, copia datos CPU.
¿ Se necesita algún punto de sincronización ?
 - N = 10000, 100000, 1e6
 - MAX_BLOCKSIZE = 16, 32, 64, 128, 256, 512
 - Un grupo con ***cudaEventCreate*** y otro con ***gettimeofday***
 - Iteraciones = 10000
- ¿ Porqué la duración del kernel no va en proporción al tamaño del vector ?
- Calcular la ocupación mediante Cuda Occupancy Calculator
- Calcular el ancho de banda para cada configuración: $(Br+Bw)/10^{e9}/time(secs)$



Programación con CUDA

■ Características 9500 GT/GTX 285

	9500GT	GTX 285
SMs	4	30
SPs/SM	8	8
Total Cores	32	240
Compute cap	1.1	1.3
Memoria	1 GB	2 GB
Peak BandWidth GBPS	25.6 GDDR3/16.0 GDDR2	159.0 GDDR3
Clock	1.38 Ghz	1.48 Ghz



Programación con CUDA

■ Ejercicio 7: Resultados para 9500 GT

sec	1e4	1e5	1e6
16	0,2477	1,9318	18,7047
32	0,1823	1,3419	12,8636
64	0,1781	1,3063	12,4601
128	0,1766	1,3008	12,4661
256	0,1721	1,3019	12,4591
512	0,1860	1,3163	12,5409



Programación con CUDA

■ Ejercicio 7: Resultados para GTX 285

sec	1e4	1e5	1e6
16	0,0808	0,4604	4,2757
32	0,0600	0,2497	2,1500
64	0,0506	0,1455	1,0851
128	0,0513	0,1327	0,9448
256	0,0510	0,1367	0,9524
512	0,0533	0,1428	0,9559



Programación con CUDA

■ Ejercicio 7: Cuda Occupancy Calculator

- `nvcc vecAdd.cu -o vecAdd -Xptxas=-v`
- 4 Registros + 44 bytes shared memory

	9500 GT	GTX 285
16	33 %	25 %
32	33 %	25 %
64	67 %	50 %
128	100 %	100 %
256	100 %	100 %
512	67 %	100 %



Programación con CUDA

■ Ejercicio 7: Bandwidth 9500 GT. Peak=16.0/25.6 GBPS

9500 GT	1e4	1e5	1e6
16	4,8446	6,2118	6,4155
32	6,5826	8,9425	9,3286
64	6,7378	9,1863	9,6307
128	6,7950	9,2251	9,6261
256	6,9727	9,2173	9,6315
512	6,4516	9,1165	9,5687



Programación con CUDA

■ Ejercicio 7: Bandwidth GTX 285. Peak=159.0 GBPS

GTX 285	1e4	1e5	1e6
16	14,8515	26,0643	28,0656
32	20,0000	48,0577	55,8140
64	23,7154	82,4742	110,5889
128	23,3918	90,4295	127,0110
256	23,5294	87,7835	125,9975
512	22,5141	84,0336	125,5361



Programación con CUDA

■ Ejercicio 7: Cuda Visual Profiler

Parámetro	9500 GT	GTX 285
gpu time	1248	98
cpu time	1266	118
occupancy	1	1
grid size X	3907	3907
branch	41	41
divergent branch	1	1
instructions	154	154
gld request	-	5
gst request	-	2



UNIVERSIDAD DE ALMERÍA