Comparación tiempos de ejecución entre C y CUDA

William Enrique Choque Valderrama.

Cristian camilo Salazar Salazar.

Ingeniería de Sistemas

Universidad Tecnológica de Pereira

28 septiembre de 2016.

**Resumen**

En este informe analiza el desempeño que se obtiene al multiplicar dos matrices NxM \* MxL al ejecutar un algoritmo secuencial y otro paralelo con apoyo de la tecnología CUDA, se compararán los tiempos de cada algoritmo.

**Contenido**

Las siguientes tablas muestran los tiempos de 5 ejecuciones de cada algoritmo para cada multiplicación con su respectivo promedio y la aceleración obtenida por el algoritmo paralelo en CUDA

Matrices 5x7 \*7x3 tiled Width = 1

|  |  |  |
| --- | --- | --- |
| # Ejecución | Tiempo algoritmo secuencial (Segundos) | Tiempo algoritmo paralelo (Segundos) |
| 1 | 0.000006s | 0.000059s |
| 2 | 0.000006s | 0.000059s |
| 3 | 0.000006s | 0.00006s |
| 4 | 0.000006s | 0.000058s |
| 5 | 0.000006s | 0.000059s |
| Promedio | 0.000006s | 0.000059s |

Matrices 50x60 \*60x65 tiled Width 32

|  |  |  |
| --- | --- | --- |
| # Ejecución | Tiempo algoritmo secuencial (Segundos) | Tiempo algoritmo paralelo (Segundos) |
| 1 | 0.000809s | 0.000755s |
| 2 | 0.00081s | 0.000757s |
| 3 | 0.00081s | 0.000755s |
| 4 | 0.00081s | 0.000758s |
| 5 | 0.000812s | 0.000756s |
| Promedio | 0.0008102 | 0.0007565 |

Matrices 200x210 \*210x200 tiled Width 32

|  |  |  |
| --- | --- | --- |
| # Ejecución | Tiempo algoritmo secuencial (Segundos) | Tiempo algoritmo paralelo (Segundos) |
| 1 | 0.034532s | 0.000161s |
| 2 | 0.034621s | 0.000166s |
| 3 | 0.034529s | 0.000755s |
| 4 | 0.034529s | 0.00016s |
| 5 | 0.034518s | 0.00016s |
| Promedio | 0.0345458 | 0.0002804 |

Matrices 1000x1001 \*1001x1010 tiled Width 32

|  |  |  |
| --- | --- | --- |
| # Ejecución | Tiempo algoritmo secuencial (Segundos) | Tiempo algoritmo paralelo (Segundos) |
| 1 | 5.83502s | 0.002342s |
| 2 | 4.3057s | 0.002354s |
| 3 | 4.28207s | 0.002327s |
| 4 | 4.29267s | 0.002669s |
| 5 | 4.3348s | 0.002372s |
| Promedio | 4.610052 | 0.0019442 |

La aceleración obtenida es de aproximadamente 1827.49x

Matrices 1500x1550 \*1550x1506 tiled Width 32

|  |  |  |
| --- | --- | --- |
| # Ejecución | Tiempo algoritmo secuencial (Segundos) | Tiempo algoritmo paralelo (Segundos) |
| 1 | 18.5835s | 0.006309s |
| 2 | 18.6572s | 0.006117s |
| 3 | 19.5045s | 0.006464s |
| 4 | 18.5485 | 0.006215s |
| 5 | 18.741s | 0.006437s |
| Promedio | 18.75294 | 0.0063084 |

|La aceleración obtenida es de aproximadamente 2945.55x

El siguiente grafico muestra el aumentó el tiempo de ejecución en segundos del algoritmo secuencial hasta los 18.7 segundos, mientras el algoritmo paralelo se mantiene por debajo de 1 segundo

Azul : Algoritmo secuencia Rojo : Algoritmo paralelo

El aumento de tiempos apreciados en el algoritmo secuencial es de orden exponencial y el paralelo logra aumentar su desempeño logrando un orden lineal

**Conclusiones**

En esta práctica se pude observar que al multiplicar matrices pequeñas ambos algoritmos tienen un desempeño en tiempo muy similar, se nota que es un poco más rápido el algoritmo secuencial por el “cuello de botella” que es generado al enviar los datos hacia el device y esperar su resultado.

A medida que se aumentaba el tamaño de las matrices a multiplicar, se puede apreciar el desempeño del proceso en paralelo. Con las matrices de 50x60 \*60x65 a pesar de ser de pequeñas dimensiones se nota un aumento de desempeño en el paralelo, en cambio con matrices de 1000x1001 \*1001x1010, donde se aprecia un aumento de tamaño considerable, el paralelo aumentó el desempeño al secuencial con 1827.49X .

Al probar con diferentes tamaños de tiles se aumenta el paralelismo y por ende se vio un incremento en el desempeño del algoritmo.