Universidad Técnica Federico Santa María
Campus Santiago San Joaquín
Departamento de Informática
2° semestre 2020

Taller 1: Promediando imágenes

Taller de Modelos y Métodos Cuantitativos
Profesor Álvaro Salinas
Estudiantes Ethiel Carmona 201773533-3,
Joaquín Castillo 201773520-1,
María Paz Morales 201773505-8,
Axel Reyes 201773502-3

29 de octubre de 2020

1. Especificaciones

   En el presente taller se utilizó la siguiente tarjeta gráfica:

   - **Nombre:** NVIDIA GeForce
   - **Modelo:** GTX 1050 TI
   - **Compute Capability:** Versión 6.1
   - **Arquitectura:** Pascal
   - **Cuda Cores:** 768
   - **Frame Buffer:** 4 GB GDDR5
   - **Velocidad de memoria:** 7 Gbps

2. Desarrollo

   [**Pregunta**] Antes de ejecutar su código, ¿cuál de las dos implementaciones cree usted que presentará un mejor desempeño? ¿Depende de algo? Argumente su respuesta.

   Creemos que la implementación con GPU tendrá un mejor desempeño en los archivos con imágenes grandes (más pixeles). Esto porque se aprovecha de mejor manera a las hebras. Por lo mismo, se espera que la CPU tenga un mejor desempeño con gran cantidad de imágenes de menor tamaño.

   [**Informe**] A continuación se muestran las 6 imágenes resultantes de cada conjunto de archivos de entrada:

Figura 1: Imagen resultante del primer conjunto de entradas.

Figura 2: Imagen resultante del segundo conjunto de entradas.

Figura 3: Imagen resultante del tercer conjunto de entradas.

Figura 4: Imagen resultante del cuarto conjunto de entradas.

Figura 5: Imagen resultante del quinto conjunto de entradas.

Figura 6: Imagen resultante del sexto conjunto de entradas.

[**Informe**] En la Tabla 1 se muestran los tiempos de ejecución de los 6 archivos de entrada tanto con CPU como GPU.

Entrada	Dimensión [px]	Tiempo CPU [ms]	Tiempo GPU [ms]
1	2x2	18	67,051521
2	4x4	47	13,626368
3	16x16	126	0,957888
4	72x128	62	0,335872
5	144x256	32	0,340992
6	720x1280	23	0,452608

Tabla 1: Tabla de tiempos de ejecución de las 6 entradas con imágenes tanto en CPU como GPU.

[**Pregunta**] Analice los tiempos obtenidos. ¿Qué puede concluir? Comente sobre el comportamiento observado y relaciónelo al enfoque de paralelismo (trabajo de cada hebra) que utilizó.

A excepción del conjunto de imágenes de 2x2, la GPU resultó ser más eficiente que la CPU, superando los tiempos hasta por 4 órdenes de magnitud en algunos casos.

En los dos primeros conjunto de imágenes, que son los de dimensiones más pequeñas, la GPU tomó más tiempo de lo habitual en ejecutar el programa, incluso más que la CPU para el primer conjunto. Esto se puede explicar debido a que, al ser pocos píxeles por imágenes, se tienen muchas menos hebras y se aprovecha muy poco el paralelismo de estas. Finalmente, para el último conjunto de imágenes (720x1280) se esperaba que fuese la mejor en cuanto al rendimiento de la GPU, ya que al ser pocas imágenes de mayor dimensión se aprovecha de mejor manera el paralelismo de las hebras. Sin embargo, la que obtuvo mejores resultados fue la del conjunto 4. Esto se puede deber probablemente a que al crear tantas hebras para las imágenes del conjunto 6, la SM se llenó y provocó que bloques de hebras tuviesen que esperar a que se desocupara.

3. Conclusiones

En este taller se pudo apreciar que no siempre la CPU tendrá peores tiempos de ejecución. Hay casos en los que funcionó mejor porque el paralelismo de la GPU no se aprovechó bien. En este sentido, cuando se enfrente un problema, conviene analizarlo bien para entender cómo sacar provecho del paralelismo, ya que para distintos casos de un mismo problema se podrían utilizar diferentes acercamientos. Por ejemplo, tal vez en este taller hubiese sido mejor tratar los primeros archivos con hebras donde cada una analizara una imagen pequeñita por sí sola, realizando operaciones atómicas. Así, se tomarían las múltiples imágenes en paralelo y se aprovecharía el potencial de las hebras.