

2023, INFO188 Tarea 2: Matriz Dispersa x Vector

Profesor: Cristóbal A. Navarro, **Ayudante:** Alejandro Villagrán

En esta tarea, usted implementará el producto de una matriz dispersa de $n \times n$, por un vector de $n \times 1$. Una matriz dispersa (*sparse matrix*), aquí llamada **Md**, es una matriz que tiene pocos valores distintos de cero. Cuando se trabaja con matrices dispersas, es posible ahorrar memoria ya que no es necesario tener los ceros explícitamente almacenados en RAM, solo es necesario tener un correcto registro de los valores distintos a cero. En el caso del producto **Md x v**, el esquema es:

$$\begin{bmatrix} & 6 & 1 & \\ 2 & & 8 & 3 \\ & & 4 & \\ & 7 & 5 & \end{bmatrix} \times \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix} = \begin{bmatrix} 30 \\ 76 \\ 24 \\ 58 \end{bmatrix}$$

En esta tarea, se pide programar una solución de Matriz dispersa por vector que sea paralela, y eficiente en el uso de memoria comparado a almacenar la matriz completa como normalmente se suele hacer. En particular se pide lo siguiente:

1. Proponer un diseño de estructura de matriz dispersa que sea paralelizable en CPU y GPU.
2. Implementar una solución en CPU usando OpenMP y otra en GPU usando CUDA.
3. El programa debe poder ejecutarse de la siguiente forma:
 - o `./prog <n> <d> <m> <s> <nt>`
 - **<n>**: lado de la matriz de $n \times n$, y a la vez largo del vector.
 - **<d>**: densidad de valores distintos de cero, valor entre 0 y 1.
 - **<s>**: semilla para valores aleatorios
 - **<m>**: modo CPU (0) o GPU (1)
 - **<nt>**: numero de threads OpenMP (para el modo CPU)
 - **Ejemplo CPU con 8 threads:** `./prog $((2**13)) 0.10 0 8`
 - **Ejemplo GPU:** `./prog $((2**13)) 0.07 1 8`
4. Los valores de **Md** y **v** deben ser **floats** aleatorios en base a la semilla **<s>** (para otorgar reproducibilidad)
5. El programa debe entregar en el terminal el tiempo que tarda el cálculo de **Md x v**
6. Generar gráficos (deben ir en el informe):
 - o **CPU:** Speedup con respecto a CPU 1 thread (Y) vs numero de threads (X)
 - o **CPU y GPU:** Speedup con respecto a CPU 1 thread (Y) vs **n** (X)
 - o **Memoria usada (Y) vs n (X):** 2 Curvas → la del método de uds, y la de uno tradicional que almacena todo.
 - o ****BONUS**** → investigar a partir de qué valor de **<d>** la técnica propuesta deja de ser conveniente comparado a un cálculo convencional, en términos de velocidad y memoria.

Grupos, entregables y fecha de entrega

- **[GRUPOS]** De 3 o 4 personas.
- **[ENTREGABLES]** Archivo *.zip con código fuente + informe PDF de máximo 4 páginas.
- **[PLAZO]** 19 de Diciembre 2023, todo el día.