

C1 - ParComp - 2021-I

{Prof: jgonzalez} @utec.edu.pe

May 7, 2021

1 Indicaciones

1. Escribir un informe explicando de forma breve y directa el diseño de cada ejercicio.
2. Separar en carpetas para cada ejercicio **con los nombres:** Ejercicio 1, Ejercicio 2, Ejercicio 3, Ejercicio 4. Subir en un archivo comprimido (.zip) en Gradescope **solamente:** 1) source code, 2) informe en .pdf (fuera de las carpetas).
3. Fecha de **entrega:** Verificar Sección de Tareas C01 de Canvas.
4. Las entregas son realizadas **solamente por Gradescope.**
5. Cualquier intento de plagio parcial o total es sancionado por UTEC en todas sus atribuciones.

2 Ejercicios

Ejecutar las implementaciones en el cluster Khipu. Medir los tiempos de las regiones de interés (ROI) de cada programa.

1. (1 point) Implementar una multiplicación MatrixVector serial con N_{rows} : 2048 y 4096.
 - Implementación serial naive.
 - Implementación serial con vector transpuesto.
 - Implementación serial con vector transpuesto para block tilling en cache.
2. (2 points) Implementar una multiplicación MatrixVector con $num_threads$ con N_{rows} y $N_{columns}$: 2048 y 4096, 8192.
 - Implementación con vector transpuesto. Determinar el $num_threads$ para obtener el menor tiempo de ejecución.
 - Implementación con vector transpuesto para block tilling. Determinar el $num_threads$ para obtener el menor tiempo de ejecución.
 - Extender su implementación para una multiplicación de matrices. Determinar el $num_threads$ para obtener el menor tiempo de ejecución.
3. (2 points) Implementar una multiplicación MatrixVector en vectorial SIMD con N_{rows} : 2048 y 4096, 8192.
 - Usar instrucciones AVX o SSE del procesador. <https://software.intel.com/sites/landingpage/IntrinsicsGuide/>
 - Comparar con una implementación serial de la primera pregunta item a) usando el flag `-fopt-info-vec-all`.