

Rubrica de evaluación de laboratorios.

Programación Paralela 2021-I - Prof. Roger Nieto.

La evaluación de los laboratorios en el curso se hará de la siguiente manera:

- 10 % Diseño: Debe incluir, por lo menos,
 - Explicación detallada del código del algoritmo o aplicación que se va a paralelizar. Realizar un esquema en pseudo-código con las principales partes o tareas metodológicas del algoritmo o código elegido, use un flowchart para la representación diagramática del algoritmo. Enunciar las entradas y salidas del mismo, así como el tipo de datos y dimensión máxima que se usarán para las entradas. Enuncie por qué definió ese límite en el tamaño de datos, la idea es que llegue al worst-case (límite) en algún elemento de la arquitectura del hardware usado.
 - Análisis de complejidad computacional del algoritmo o código seleccionado.
 - Análisis de tiempo del algoritmo serial con las herramientas time, wtime y Gprof. Use gráficas para evaluar estos tiempos en función del tamaño de los datos de entrada. Recuerda reportar la media aritmética, junto con su desviación estándar, para mínimo 5 ejecuciones. Cuando tome este tiempo, intente no ejecutar otras aplicaciones demandantes en segundo plano en su equipo.
 - Propuesta de paralelización (que partes metodológicas del algoritmo se van a paralelizar y por qué). Explicar detalladamente.
 - Análisis de independencia de datos, y como piensa realizar la partición (por datos, por tareas, por hilos, etc.).
 - Características del hardware donde se ejecutaron las pruebas (RAM (size- speed), CPU (caches size, number of cores, etc), SSD-HDD disks (speed)).
- 30 % Funcionamiento.
 - El funcionamiento será evaluado en función de la mejora en tiempo de ejecución que se obtenga respecto a la versión secuencial del algoritmo y el número de cores/threads utilizados en la ejecución. Se tendrá en cuenta también la eficiencia y potencial escalabilidad de su propuesta de paralelización.
- 20% Informe: debe estar escrito en inglés y contener, además todo lo que contiene el diseño realizado, junto con potenciales actualizaciones y/o cambios:
 - Verificar si hay incidencia del contenido de los datos en el rendimiento del algoritmo.
 - Verificar que no existan data-races usando la herramienta Coderrect (e.g. using OpenMP).
 - Realizar análisis de escalamiento fuerte y débil, junto con la gráfica de Isoefficiency.

- Las gráficas y tablas de speedup vs threads/cores, tiempo vs threads/cores (el número máximo de cores/threads que permita su laptop), time vs data size (para varios cores/threads) y uso de RAM vs data size. Las tablas deben contener la desviación estándar de las 5 medidas realizadas para calcular la media que reporta.
- Análisis del tamaño de la pila (stack) per thread.
- La explicación detallada del código realizado.
- Problemas que se encontraron en la paralelización y como lo resolvieron.
- **Enlaces y/o material bibliográfico que haya usado para su solución. Si se encuentra que el código usado se copia de otra fuente y no se cita, se aplicará la penalización que corresponda según el reglamento de la Universidad.**
- Importancia o aplicabilidad real del algoritmo elegido.
- Sustentación: 40%
 - La sustentación será individual. El profesor podrá realizar preguntas acerca del laboratorio y su implementación, la temática necesaria para su resolución vista en clase, además de las lecturas sugeridas.

El laboratorio debe ser enviado como máximo el día antes de la sustentación. A partir de esa fecha, se tendrá una penalización de una unidad (1.0) en la nota final, por cada día de retraso.