



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3533 — Computación de alto rendimiento — 2/2024

Tarea 2

Viernes 15 de noviembre, 2024

Fecha de entrega: Lunes 25 de noviembre, hasta las 23:59

Composición: Parejas (2 personas)

Objetivos y entrega

Analizar y poder optimizar diferentes rasgos de un código, utilizando openMP en el lenguaje C.

La entrega consiste de un comprimido .zip que contenga los siguientes archivos:

- Un documento .pdf escrito **en latex** con la respuesta a las preguntas del enunciado.
- archivo .c que contenga la solución paralelizada diseñada.

La entrega debe realizarla solo un integrante mediante un buzón de canvas que se habilitará durante la semana y el comprimido debe llamarse `< nro_alumno1_nro_alumno2.zip >`. Es decir, si los números de alumno fueran 12345678 y 98765432. El archivo debe llamarse 12345678_98765432.zip.

Actividad y preguntas

Conteste cada pregunta de la manera más simple posible. Entregue suficiente información para cumplir con lo preguntado, pero no sea redundante. Responda de manera sucinta y precisa.

Para las siguientes preguntas, es necesario que entienda lo que se está realizando en el código entregado y realice los cambios necesarios para poder resolver el problema utilizando código paralelizado. La idea es **disminuir el tiempo de ejecución** de manera considerable. Como pista, puede que su solución paralelizada no dé el mismo resultado que el original si decide modificar el algoritmo a utilizar para resolver el problema, pero no se preocupen de eso.

El programa entregado consiste en una simulación de una red en la cual hay una fuente de calor en el centro. La simulación avanza, de manera que el calor se va transfiriendo por los demás lugares de la red a medida que el tiempo va pasando. Además, puede ser útil saber que el algoritmo utilizado (uno de los varios que existen) para resolver el problema es el método de Gauss-Seidel.

1. **(2 pts)** Explique qué cambios realizó en el código, cómo logró paralelizar la solución y toda suposición que realizó para lograrlo.
2. **(4 pts)** Pruebe diferente cantidad de threads para correr el código. Utilice una cantidad suficiente que le permita dibujar una curva en donde pueda verificar desde qué momento el speedup y, por lo tanto, la eficiencia disminuyen. Debe incluir una tabla y gráficos de tiempo de ejecución, speedup y eficiencia que evidencie tanto los tiempos y cantidad de threads, como la curva formada por estos datos.

Recuerde que todo lo que corra en el servidor debe hacerlo mediante SLURM. Quien sea sorprendido por alguno de los sistemas de monitoreo del servidor incumpliendo esto, será bloqueado y no volverá a tener acceso al servidor, quedando imposibilitado para poder seguir avanzando con la pregunta 2.