

Reporte PDF

William Aguilar Calvo

1 ¿Cuál fue su estrategia de paralelización?

Se utilizo los bucles paralelos con ayuda de libreria CILK de C y C++, con la palabra reservada de *cilk_for*, especificamente se penso en ponerlo en el segundo nivel de los ciclos anidados, ya que si se ponía en el primero no iba a ser posible ya que genera una dependencia de datos, y si se ponía en el ultimo se abrirían demasiados hilos y el costo computacional seria mayor ademas del tiempo de apertura y conclusion de los hilos

2 ¿Qué patrones de programación paralela usó en la solución?

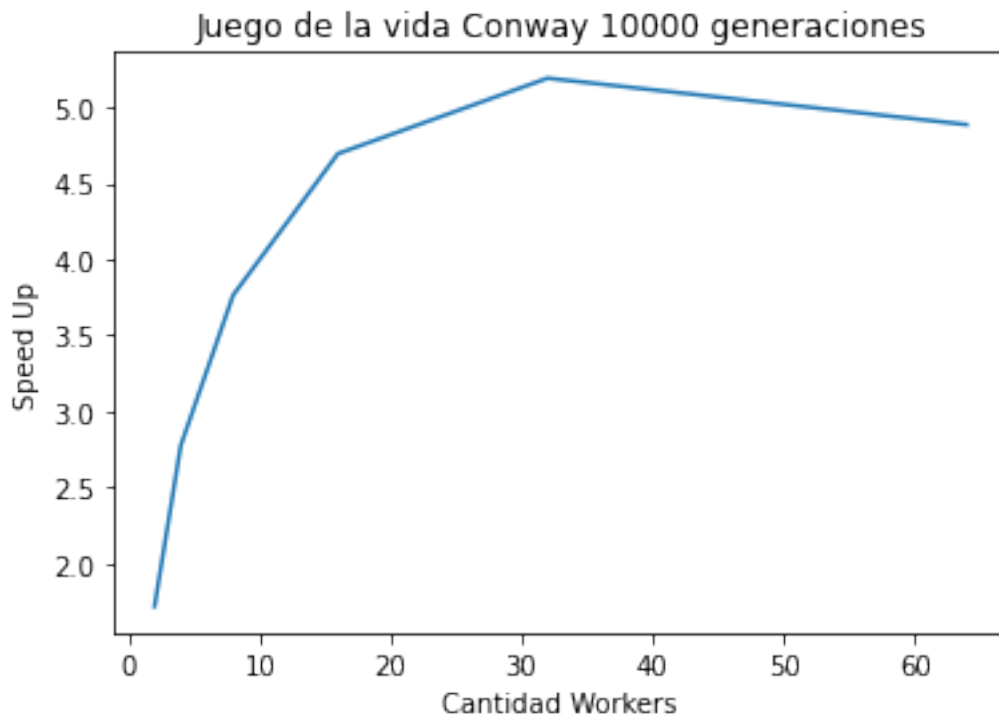
Se utilizo el patron Stencil y Recurrence para contar los vecinos y la asignacion de vida o muertes para cada elemento de la matriz

3 ¿Que operaciones parecieron las más apropiadas para el programa?

Aplicacion de bucles de paralelos y verificacion de espacios memoria, validacion de dumpsets de C++ y recorridos de las matrices para la verificacion y asignacion de vida o muerte

4 Análisis de Speed Up

Relacionado con el Speed Up, basicamente el ejemplo que se utilizo fue usar una matriz de 100 elementos, con 10000 iteraciones y este es el grafico de Speed Up y podemos ver que a mayor cantidad de workers, el speed up va a ser mayor, lo que significa que el tiempo es mucho menor mientras mas workers esto hasta masomenos los 32 workers, cuando se superan los 32 se puede notar que el Speed Up baja



5 Análisis de Eficiencia

Relacionado con la eficiencia, básicamente el ejemplo que se utilizó fue usar una matriz de 100 elementos, con 10000 iteraciones con 64 workers, básicamente podemos ver que la eficiencia sube hasta que se tienen 32 workers, luego va bajando

