

Practica 2

1. Desarrolle un algoritmo paralelo que compute una suma de vectores: $A_i = B_i + C_i$. Para ello, considere lo siguiente:
 - Identifique las regiones de código que pueden ejecutarse en paralelo y las que no ¿Existen dependencias?
 - Analice cómo distribuir el trabajo entre los hilos. ¿El cómputo es regular? ¿La arquitectura es homogénea?Mida el tiempo de ejecución para diferentes valores de N y $T=\{2,4,8\}$. Analice el rendimiento.
- Cada hilo se va a encargar de realizar la suma de distintas porciones del vector, por lo tanto no existen dependencias entre los distintos hilos que se están ejecutando en paralelo.

Las regiones que no se pueden ejecutar en paralelo son:

 - Verificación de los argumentos
 - La inicialización de ciertas variables necesarias para la división del vector en varias porciones.
 - La inicialización de los vectores (en el caso del código en realidad si se podría hacer, cada hilo podría inicializar su porción de los vectores A y B en 1 pero creo que esa no es la idea del ejercicio)
 - La creación de los hilos y la espera a la terminación de los hilos hijos (join)
- La arquitectura es homogénea ya que se tratan de hilos trabajando sobre la misma memoria compartida, pero el cómputo no es regular particularmente en los casos que queda un resto en la división de N (tamaño vectores) / T (cantidad de hilos) , en este caso el hilo último hilo (T - 1) tendrá una porción del vector más grande y por lo tanto deberá realizar más sumas.

Comparación algoritmo secuencial con algoritmos paralelos de 2, 4 y 8 hilos

```

agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1 100
Suma de vectores de dimensión 100. Tiempo en segundos 0.000001
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 100 2
Suma de vectores de dimensión 100. Tiempo en segundos 0.000245
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 100 4
Suma de vectores de dimensión 100. Tiempo en segundos 0.000254
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 100 8
Suma de vectores de dimensión 100. Tiempo en segundos 0.000118
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1 1000
Suma de vectores de dimensión 1000. Tiempo en segundos 0.000006
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 1000 2
Suma de vectores de dimensión 1000. Tiempo en segundos 0.000240
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 1000 4
Suma de vectores de dimensión 1000. Tiempo en segundos 0.000117
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 1000 8
Suma de vectores de dimensión 1000. Tiempo en segundos 0.000131
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1 10000
Suma de vectores de dimensión 10000. Tiempo en segundos 0.000089
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 10000 2
Suma de vectores de dimensión 10000. Tiempo en segundos 0.000357
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 10000 4
Suma de vectores de dimensión 10000. Tiempo en segundos 0.000086
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 10000 8
Suma de vectores de dimensión 10000. Tiempo en segundos 0.000119
Suma de vectores resultado correcto

agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1 100000
Suma de vectores de dimensión 100000. Tiempo en segundos 0.000703
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 100000 2
Suma de vectores de dimensión 100000. Tiempo en segundos 0.000797
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 100000 4
Suma de vectores de dimensión 100000. Tiempo en segundos 0.001020
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 100000 8
Suma de vectores de dimensión 100000. Tiempo en segundos 0.001173
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1 1000000
Suma de vectores de dimensión 1000000. Tiempo en segundos 0.007382
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 1000000 2
Suma de vectores de dimensión 1000000. Tiempo en segundos 0.005285
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 1000000 4
Suma de vectores de dimensión 1000000. Tiempo en segundos 0.006114
Suma de vectores resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto1$ ./punto1paralelo 1000000 8
Suma de vectores de dimensión 1000000. Tiempo en segundos 0.006905
Suma de vectores resultado correcto

```

Speedup y Eficiencia

Tp → tiempo desde que empieza a ejecutarse el primer hilo hasta que termina el último
Ts → tiempo que tarda en ejecutarse la solución secuencial.

Ts/Tp → Si el resultado es > a 1 es mejor el paralelo

(A ver en teoría el speedup permite independizarnos del tamaño del problema pero no pienso calcularlo así que voy a usar los distintos tiempos de los diversos N)

La eficiencia permite saber qué tan cerca se está del speedup óptimo, que tan bien estamos aprovechando la arquitectura.

$$E = S/S_{\text{Optimo}}$$

El Speedup óptimo depende de la arquitectura, si es homogénea es la cantidad de procesadores. En mi caso estoy usando una máquina virtual con 2 procesadores, por lo tanto el S_{Optimo} sería 2.

El valor de la eficiencia es un número entre 0 y 1, si es más cercano a 1 indica que la arquitectura se está aprovechando, caso contrario es más cercano a 0.

$$N = 100$$

Tiempo en segundos secuencial 0.000001

Tiempo en segundos paralelo 2 0.000245

Tiempo en segundos paralelo 4 0.000254

Tiempo en segundos paralelo 8 0.000118

Comparación respecto a 2 hilos $\rightarrow 0.000001/0.000245 = 0.00408163265$

Eficiencia $\rightarrow 0.00408163265/2 = 0.002$

Comparación respecto a 4 hilos $\rightarrow 0.000001/0.000254 = 0.00393700787$

Eficiencia $\rightarrow 0.00393700787/2 = 0.002$

Comparación respecto a 8 hilos $\rightarrow 0.000001/0.000118 = 0.00847457627$

Eficiencia $\rightarrow 0.00847457627/2 = 0.004$

$$N = 1000$$

Tiempo en segundos 0.000006

Tiempo en segundos paralelo 2 0.000240

Tiempo en segundos paralelo 4 0.000117

Tiempo en segundos paralelo 8 0.000131

Comparación respecto a 2 hilos $\rightarrow 0.000006/0.000240 = 0.025$

Eficiencia $\rightarrow 0.025/2 = 0.0125$

Comparación respecto a 4 hilos $\rightarrow 0.000006/0.000117 = 0.05128205128$

Eficiencia $\rightarrow 0.05128205128/2 = 0.026$

Comparación respecto a 8 hilos $\rightarrow 0.000006/0.000131 = 0.04580152671$

Eficiencia $\rightarrow 0.04580152671/2 = 0.023$

$$N = 10000$$

Tiempo en segundos 0.000089

Tiempo en segundos paralelo 2 0.000357

Tiempo en segundos paralelo 4 0.000086

Tiempo en segundos paralelo 8 0.000119

Comparación respecto a 2 hilos $\rightarrow 0.000089/0.000357 = 0.24929971988$

Eficiencia $\rightarrow 0.24929971988/2 = 0.125$

Comparación respecto a 4 hilos $\rightarrow 0.000089/0.000086 = 1.03488372093$

Eficiencia $\rightarrow 1.03488372093/2 = 0.517$

Comparación respecto a 8 hilos $\rightarrow 0.000089/0.000119 = 0.04580152671$

Eficiencia $\rightarrow 0.04580152671/2 = 0.023$

N = 100000

Tiempo en segundos 0.000703

Tiempo en segundos paralelo 2 0.000797

Tiempo en segundos paralelo 4 0.001020

Tiempo en segundos paralelo 8 0.001173

Comparación respecto a 2 hilos $\rightarrow 0.000703/0.000797 = 0.88205771643$

Eficiencia $\rightarrow 0.88205771643/2 = 0.441$

Comparación respecto a 4 hilos $\rightarrow 0.000703/0.001020 = 0.68921568627$

Eficiencia $\rightarrow 0.68921568627/2 = 0.345$

Comparación respecto a 8 hilos $\rightarrow 0.000703/0.001173 = 0.59931798806$

Eficiencia $\rightarrow 0.59931798806/2 = 0.3$

N = 1000000

Tiempo en segundos 0.007382

Tiempo en segundos paralelo 2 0.005285

Tiempo en segundos paralelo 4 0.006114

Tiempo en segundos paralelo 8 0.006905

Comparación respecto a 2 hilos $\rightarrow 0.007382/0.005285 = 1.397$

Eficiencia $\rightarrow 1.397/2 = 0.6985$

Comparación respecto a 4 hilos $\rightarrow 0.007382/0.006114 = 1.207$

Eficiencia $\rightarrow 1.207/2 = 0.6035$

Comparación respecto a 8 hilos $\rightarrow 0.007382/0.006905 = 1.069$

Eficiencia $\rightarrow 1.069/2 = 0.5345$

Se puede ver como claramente a medida que aumenta N el algoritmo paralelo comienza a ser mejor que el secuencial y el más eficiente comienza a ser el que utiliza 2 hilos.

2. Desarrolle un algoritmo paralelo que compute la multiplicación de matrices cuadradas de NxN. Primero, considere a la versión optimizada del ejercicio 8 de la práctica anterior como algoritmo base. Luego, paralelice la versión que computa por bloques.

Mida el tiempo de ejecución para $N=\{512, 1024, 2048, 4096\}$ y $T=\{2,4,8\}$. Analice el rendimiento.

Comparación algoritmo secuencial con algoritmos paralelos de 2, 4 y 8 hilos

```
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2 512
Multiplicacion de matrices de 512x512. Tiempo en segundos 0.523177
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 512 2
Multiplicacion de matrices de 512x512. Tiempo en segundos 1.023079
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 512 4
Multiplicacion de matrices de 512x512. Tiempo en segundos 0.927933
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 512 8
Multiplicacion de matrices de 512x512. Tiempo en segundos 2.078978
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2 1024
Multiplicacion de matrices de 1024x1024. Tiempo en segundos 3.794026
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 1024 2
Multiplicacion de matrices de 1024x1024. Tiempo en segundos 7.037551
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 1024 4
Multiplicacion de matrices de 1024x1024. Tiempo en segundos 7.361160
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 1024 8
Multiplicacion de matrices de 1024x1024. Tiempo en segundos 14.474623
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2 2048
Multiplicacion de matrices de 2048x2048. Tiempo en segundos 28.035081
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 2048 2
Multiplicacion de matrices de 2048x2048. Tiempo en segundos 57.394064
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 2048 4
Multiplicacion de matrices de 2048x2048. Tiempo en segundos 57.936163
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 2048 8
Multiplicacion de matrices de 2048x2048. Tiempo en segundos 117.262283
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2 4096
Multiplicacion de matrices de 4096x4096. Tiempo en segundos 224.573556
Multiplicacion de matrices resultado correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto2$ ./punto2Paralelo 4096 4
Multiplicacion de matrices de 4096x4096. Tiempo en segundos 458.781515
Multiplicacion de matrices resultado correcto
```

Speedup y Eficiencia

$N = 512$

Tiempo en segundos 0.523177

Tiempo en segundos paralelo 2 1.023079

Tiempo en segundos paralelo 4 0.927933

Tiempo en segundos paralelo 8 2.078978

Comparación respecto a 2 hilos $\rightarrow 0.523177/1.023079 = 0.511$

Eficiencia $\rightarrow 0.511/4 = 0.127$

Comparación respecto a 4 hilos $\rightarrow 0.523177/0.927933 = 0.563$

Eficiencia $\rightarrow 0.563/4 = 0.140$

Comparación respecto a 8 hilos $\rightarrow 0.523177/2.078978 = 0.252$

Eficiencia $\rightarrow 0.252/4 = 0.063$

N = 1024

Tiempo en segundos 3.794026

Tiempo en segundos paralelo 2 7.037551

Tiempo en segundos paralelo 4 7.361160

Tiempo en segundos paralelo 8 14.474623

Comparación respecto a 2 hilos $\rightarrow 3.794026/7.037551 = 0.539$

Eficiencia $\rightarrow 0.539/4 = 0.134$

Comparación respecto a 4 hilos $\rightarrow 3.794026/7.361160 = 0.515$

Eficiencia $\rightarrow 0.515/4 = 0.128$

Comparación respecto a 8 hilos $\rightarrow 3.794026/14.474623 = 0.262$

Eficiencia $\rightarrow 0.262/4 = 0.065$

N = 2048

Tiempo en segundos 28.035081

Tiempo en segundos paralelo 2 57.394064

Tiempo en segundos paralelo 4 57.936163

Tiempo en segundos paralelo 8 117.262283

Comparación respecto a 2 hilos $\rightarrow 28.035081/57.394064 = 0.488$

Eficiencia $\rightarrow 0.488/4 = 0.122$

Comparación respecto a 4 hilos $\rightarrow 28.035081/57.936163 = 0.484$

Eficiencia $\rightarrow 0.488/4 = 0.122$

Comparación respecto a 8 hilos $\rightarrow 28.035081/117.262283 = 0.239$

Eficiencia $\rightarrow 0.239/4 = 0.059$

N = 4096

Tiempo en segundos 224.573556

Tiempo en segundos paralelo 4 458.781515

Comparación respecto a 4 hilos $\rightarrow 224.573556/458.781515 = 0.49$

Eficiencia $\rightarrow 0.49/4 = 0.123$

Claramente el algoritmo paralelo tarda más en todos los casos, probablemente sea porque el rendimiento de la implementación paralela se ve afectado por la cantidad de procesadores la cual es muy pequeña (4). A medida que hay más hilos el rendimiento cae y esto puede darse por varios factores:

1. **Limitaciones de hardware:** Si hay solo dos procesadores físicos y se está ejecutando un código paralelo con más de dos hilos, es probable que el sistema operativo tenga que hacer cambios de contexto más frecuentes, lo que podría llevar a un rendimiento inferior debido a la competencia por los recursos de la CPU.
 2. **Overhead de paralelización:** Existe un cierto costo asociado con la paralelización de un algoritmo. Esto incluye el tiempo necesario para dividir y distribuir la carga de trabajo entre los hilos, así como el tiempo requerido para combinar los resultados de nuevo en un resultado único al final de la ejecución.
 3. **Tamaño del problema:** Para problemas pequeños, el costo de la paralelización puede superar los beneficios. Es posible que para tamaños de problema pequeños, el tiempo requerido para dividir y distribuir la carga de trabajo sea significativo en comparación con el tiempo total de ejecución.
-
3. Desarrolle un algoritmo paralelo que cuente la cantidad de veces que un elemento X aparece dentro de un vector de N elementos enteros. Al finalizar, la cantidad de ocurrencias del elemento X debe quedar en una variable llamada ocurrencias. Para la sincronización emplee mutex-locks. Pruebe con diversos tamaños de N y $T=\{2,4,8\}$. Analice el rendimiento.

Comparación algoritmo secuencial con algoritmos paralelos de 2, 4 y 8 hilos

```

agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3 32768
Elemento 1 encontrado 32768 veces
Búsqueda de variable 1 en vector de dimensión 32768. Tiempo en segundos 0.000091
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 32768 2
Elemento 1 encontrado 32768 veces
Búsqueda de variable 1 en vector de dimensión 32768. Tiempo en segundos 0.000215
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 32768 4
Elemento 1 encontrado 32768 veces
Búsqueda de variable 1 en vector de dimensión 32768. Tiempo en segundos 0.000312
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 32768 8
Elemento 1 encontrado 32768 veces
Búsqueda de variable 1 en vector de dimensión 32768. Tiempo en segundos 0.000111
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3 65536
Elemento 1 encontrado 65536 veces
Búsqueda de variable 1 en vector de dimensión 65536. Tiempo en segundos 0.000141
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 65536 2
Elemento 1 encontrado 65536 veces
Búsqueda de variable 1 en vector de dimensión 65536. Tiempo en segundos 0.000407
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 65536 4
Elemento 1 encontrado 65536 veces
Búsqueda de variable 1 en vector de dimensión 65536. Tiempo en segundos 0.000295
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 65536 8
Elemento 1 encontrado 65536 veces
Búsqueda de variable 1 en vector de dimensión 65536. Tiempo en segundos 0.000163
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3 131072
Elemento 1 encontrado 131072 veces
Búsqueda de variable 1 en vector de dimensión 131072. Tiempo en segundos 0.000405
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 131072 2
Elemento 1 encontrado 131072 veces
Búsqueda de variable 1 en vector de dimensión 131072. Tiempo en segundos 0.000464
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 131072 4
Elemento 1 encontrado 131072 veces
Búsqueda de variable 1 en vector de dimensión 131072. Tiempo en segundos 0.000372
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 131072 8
Elemento 1 encontrado 131072 veces
Búsqueda de variable 1 en vector de dimensión 131072. Tiempo en segundos 0.000236
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3 262144
Elemento 1 encontrado 262144 veces
Búsqueda de variable 1 en vector de dimensión 262144. Tiempo en segundos 0.000749
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 262144 2
Elemento 1 encontrado 262144 veces
Búsqueda de variable 1 en vector de dimensión 262144. Tiempo en segundos 0.000689
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 262144 4
Elemento 1 encontrado 262144 veces
Búsqueda de variable 1 en vector de dimensión 262144. Tiempo en segundos 0.000434
Busqueda correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto3$ ./punto3Paralelo 262144 8
Elemento 1 encontrado 262144 veces
Búsqueda de variable 1 en vector de dimensión 262144. Tiempo en segundos 0.000306
Busqueda correcta

```

Speedup y Eficiencia

$N = 32768$

Tiempo en segundos 0.000091

Tiempo en segundos paralelo 2 0.000215

Tiempo en segundos paralelo 4 0.000312

Tiempo en segundos paralelo 8 0.000111

Comparación respecto a 2 hilos $\rightarrow 0.000091/0.000215 = 0.423$

Eficiencia $\rightarrow 0.423/4 = 0.105$

Comparación respecto a 4 hilos $\rightarrow 0.000091/0.000312 = 0.292$

Eficiencia $\rightarrow 0.292/4 = 0.073$

Comparación respecto a 8 hilos $\rightarrow 0.000091/0.000111 = 0.819$

Eficiencia $\rightarrow 0.819/4 = 0.204$

N = 65536

Tiempo en segundos 0.000141

Tiempo en segundos paralelo 2 0.000407

Tiempo en segundos paralelo 4 0.000295

Tiempo en segundos paralelo 8 0.000163

Comparación respecto a 2 hilos $\rightarrow 0.000141/0.000407 = 0.346$

Eficiencia $\rightarrow 0.346/4 = 0.086$

Comparación respecto a 4 hilos $\rightarrow 0.000141/0.000295 = 0.478$

Eficiencia $\rightarrow 0.478/4 = 0.119$

Comparación respecto a 8 hilos $\rightarrow 0.000141/0.000163 = 0.865$

Eficiencia $\rightarrow 0.865/4 = 0.216$

N = 131072

Tiempo en segundos 0.000405

Tiempo en segundos paralelo 2 0.000464

Tiempo en segundos paralelo 4 0.000372

Tiempo en segundos paralelo 8 0.000236

Comparación respecto a 2 hilos $\rightarrow 0.000405/0.000464 = 0.873$

Eficiencia $\rightarrow 0.873/4 = 0.218$

Comparación respecto a 4 hilos $\rightarrow 0.000405/0.000372 = 1.089$

Eficiencia $\rightarrow 1.089/4 = 0.272$

Comparación respecto a 8 hilos $\rightarrow 0.000405/0.000236 = 1.716$

Eficiencia $\rightarrow 1.716/4 = 0.429$

N = 262144

Tiempo en segundos 0.000749

Tiempo en segundos paralelo 2 0.000689

Tiempo en segundos paralelo 4 0.000434

Tiempo en segundos paralelo 8 0.000306

Comparación respecto a 2 hilos $\rightarrow 0.000749/0.000689 = 1.087$

Eficiencia $\rightarrow 1.087/4 = 0.271$

Comparación respecto a 4 hilos $\rightarrow 0.000749/0.000434 = 1.726$

Eficiencia $\rightarrow 1.726/4 = 0.431$

Comparación respecto a 8 hilos $\rightarrow 0.000749/0.000306 = 2.447$

Eficiencia $\rightarrow 2.447/4 = 0.611$

Se puede ver como claramente cuando aumenta N aumenta el rendimiento a medida que se tienen más hilos y a su vez aumenta el aprovechamiento de la arquitectura.

4. Desarrolle un algoritmo paralelo que calcule el valor promedio, mínimo y máximo de los números almacenados en un vector de tamaño N. Para la sincronización emplee semáforos. Pruebe con diversos tamaños de N y $T=\{2,4,8\}$. Analice el rendimiento.

Comparación algoritmo secuencial con algoritmos paralelos de 2, 4 y 8 hilos

```
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4 32768
Máximo: 32767
Mínimo: 0
Promedio: 16383
Calculo de máximo, mínimo y promedio en vector de dimensión 32768. Tiempo en segundos 0.000258
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 32768 2
Máximo: 32767
Mínimo: 0
Promedio: 16383
Calculo de máximo, mínimo y promedio en vector de dimensión 32768. Tiempo en segundos 0.000401
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 32768 4
Máximo: 32767
Mínimo: 0
Promedio: 16383
Calculo de máximo, mínimo y promedio en vector de dimensión 32768. Tiempo en segundos 0.000333
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 32768 8
Máximo: 32767
Mínimo: 0
Promedio: 16383
Calculo de máximo, mínimo y promedio en vector de dimensión 32768. Tiempo en segundos 0.000172
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4 65536
Máximo: 65535
Mínimo: 0
Promedio: 32767
Calculo de máximo, mínimo y promedio en vector de dimensión 65536. Tiempo en segundos 0.000536
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 65536 2
Máximo: 65535
Mínimo: 0
Promedio: 32767
Calculo de máximo, mínimo y promedio en vector de dimensión 65536. Tiempo en segundos 0.000641
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 65536 4
Máximo: 65535
Mínimo: 0
Promedio: 32767
Calculo de máximo, mínimo y promedio en vector de dimensión 65536. Tiempo en segundos 0.000435
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 65536 8
Máximo: 65535
Mínimo: 0
Promedio: 32767
Calculo de máximo, mínimo y promedio en vector de dimensión 65536. Tiempo en segundos 0.000283
Calculo correcto
```

```

Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4 16777216
Máximo: 16777215
Mínimo: 0
Promedio: 8388607
Calculo de máximo, mínimo y promedio en vector de dimensión 16777216. Tiempo en segundos 0.129269
Calculo correcto
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 16777216 2
Máximo: 16777215
Mínimo: 0
Promedio: -174
Calculo de máximo, mínimo y promedio en vector de dimensión 16777216. Tiempo en segundos 0.082694
Calculo erroneo
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 16777216 4
Máximo: 16777215
Mínimo: 0
Promedio: 345
Calculo de máximo, mínimo y promedio en vector de dimensión 16777216. Tiempo en segundos 0.065500
Calculo erroneo
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto4$ ./punto4Paralelo 16777216 8
Máximo: 16777215
Mínimo: 0
Promedio: 341
Calculo de máximo, mínimo y promedio en vector de dimensión 16777216. Tiempo en segundos 0.051242
Calculo erroneo

```

Speedup y Eficiencia

N = 32768

Tiempo en segundos 0.000258

Tiempo en segundos paralelo 2 0.000401

Tiempo en segundos paralelo 4 0.000333

Tiempo en segundos paralelo 8 0.000172

Comparación respecto a 2 hilos $\rightarrow 0.000258/0.000401 = 0.643$

Eficiencia $\rightarrow 0.643/4 = 0.160$

Comparación respecto a 4 hilos $\rightarrow 0.000258/0.000333 = 0.775$

Eficiencia $\rightarrow 0.775/4 = 0.193$

Comparación respecto a 8 hilos $\rightarrow 0.000258/0.000172 = 1.5$

Eficiencia $\rightarrow 1.5/4 = 0.375$

N= 65536

Tiempo en segundos 0.000536

Tiempo en segundos paralelo 2 0.000641

Tiempo en segundos paralelo 4 0.000435

Tiempo en segundos paralelo 8 0.000283

Comparación respecto a 2 hilos $\rightarrow 0.000536/0.000641 = 0.836$

Eficiencia $\rightarrow 0.836/4 = 0.209$

Comparación respecto a 4 hilos $\rightarrow 0.000536/0.000435 = 1.232$

Eficiencia $\rightarrow 1.232/4 = 0.308$

Comparación respecto a 8 hilos $\rightarrow 0.000536/0.000283 = 1.893$

Eficiencia $\rightarrow 1.893/4 = 0.473$

N = 16777216

Tiempo en segundos 0.129269

Tiempo en segundos paralelo 2 0.082694

Tiempo en segundos paralelo 4 0.065500

Tiempo en segundos paralelo 8 0.051242

Comparación respecto a 2 hilos $\rightarrow 0.129269/0.082694 = 1.563$

Eficiencia $\rightarrow 1.563/4 = 0.390$

Comparación respecto a 4 hilos $\rightarrow 0.129269/0.065500 = 1.974$

Eficiencia $\rightarrow 1.974/4 = 0.493$

Comparación respecto a 8 hilos $\rightarrow 0.129269/0.051242 = 2.523$

Eficiencia $\rightarrow 2.523/4 = 0.630$

Nuevamente cuando aumenta N aumenta el rendimiento a medida que se tienen más hilos y a su vez aumenta el aprovechamiento de la arquitectura.

5. Desarrolle un algoritmo paralelo que obtenga la intersección de dos conjuntos de números enteros. Pruebe con diversos tamaños de N y $T=\{2,4,8\}$ y analice el rendimiento. Tenga en cuenta la siguiente propiedad:

$$A \cap B = A \cap (B_0 \cup B_1 \cup B_2 \dots \cup B_n) = (A \cap B_0) \cup (A \cap B_1) \cup (A \cap B_2) \dots \cup (A \cap B_n)$$

Comparación algoritmo secuencial con algoritmos paralelos de 2, 4 y 8 hilos

```

agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersec 100
Calculo de intersección en vector de dimensión 100. Tiempo en segundos 0.000018
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 100 2
Calculo de intersección en vector de dimensión 100. Tiempo en segundos 0.000270
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 100 4
Calculo de intersección en vector de dimensión 100. Tiempo en segundos 0.000384
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 100 8
Calculo de intersección en vector de dimensión 100. Tiempo en segundos 0.000164
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersec 1000
Calculo de intersección en vector de dimensión 1000. Tiempo en segundos 0.001446
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 1000 2
Calculo de intersección en vector de dimensión 1000. Tiempo en segundos 0.001302
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 1000 4
Calculo de intersección en vector de dimensión 1000. Tiempo en segundos 0.000822
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 1000 8
Calculo de intersección en vector de dimensión 1000. Tiempo en segundos 0.000477
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersec 10000
Calculo de intersección en vector de dimensión 10000. Tiempo en segundos 0.134324
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 10000 2
Calculo de intersección en vector de dimensión 10000. Tiempo en segundos 0.104951
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 10000 4
Calculo de intersección en vector de dimensión 10000. Tiempo en segundos 0.061824
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 10000 8
Calculo de intersección en vector de dimensión 10000. Tiempo en segundos 0.044019
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersec 100000
Calculo de intersección en vector de dimensión 100000. Tiempo en segundos 12.041573
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 100000 2
Calculo de intersección en vector de dimensión 100000. Tiempo en segundos 8.925499
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 100000 4
Calculo de intersección en vector de dimensión 100000. Tiempo en segundos 5.124451
Intersección correcta
agusnfr@agusnfr-VirtualBox:~/Paralelos/Practica 2/Punto5$ ./vectorIntersecParalelo 100000 8
Calculo de intersección en vector de dimensión 100000. Tiempo en segundos 3.480211
Intersección correcta

```

Speedup y Eficiencia

N = 100

Tiempo en segundos 0.000018

Tiempo en segundos paralelo 2 0.000270

Tiempo en segundos paralelo 4 0.000384

Tiempo en segundos paralelo 8 0.000164

Comparación respecto a 2 hilos $\rightarrow 0.000018/0.000270 = 0.067$

Eficiencia $\rightarrow 0.067/4 = 0.016$

Comparación respecto a 4 hilos $\rightarrow 0.000018/0.000384 = 0.047$

Eficiencia $\rightarrow 0.047/4 = 0.011$

Comparación respecto a 8 hilos $\rightarrow 0.000018/0.000164 = 0.109$

Eficiencia $\rightarrow 0.109/4 = 0.027$

N = 1000

Tiempo en segundos 0.001446
Tiempo en segundos paralelo 2 0.001302
Tiempo en segundos paralelo 4 0.000822
Tiempo en segundos paralelo 8 0.000477

Comparación respecto a 2 hilos $\rightarrow 0.001446/0.001302 = 1.111$
Eficiencia $\rightarrow 1.111/4 = 0.277$
Comparación respecto a 4 hilos $\rightarrow 0.001446/0.000822 = 1.759$
Eficiencia $\rightarrow 1.759/4 = 0.439$
Comparación respecto a 8 hilos $\rightarrow 0.001446/0.000477 = 3.031$
Eficiencia $\rightarrow 3.031/4 = 0.757$

N = 10000

Tiempo en segundos 0.134324
Tiempo en segundos paralelo 2 0.104951
Tiempo en segundos paralelo 4 0.061824
Tiempo en segundos paralelo 8 0.044019

Comparación respecto a 2 hilos $\rightarrow 0.134324/0.104951 = 1.28$
Eficiencia $\rightarrow 1.28/4 = 0.32$
Comparación respecto a 4 hilos $\rightarrow 0.134324/0.061824 = 2.173$
Eficiencia $\rightarrow 2.173/4 = 0.543$
Comparación respecto a 8 hilos $\rightarrow 0.134324/0.044019 = 3.052$
Eficiencia $\rightarrow 3.052/4 = 0.763$

N = 100000

Tiempo en segundos 12.041573
Tiempo en segundos paralelo 2 8.925499
Tiempo en segundos paralelo 4 5.124451
Tiempo en segundos paralelo 8 3.480211

Comparación respecto a 2 hilos $\rightarrow 12.041573/8.925499 = 1.349$
Eficiencia $\rightarrow 1.349/4 = 0.337$
Comparación respecto a 4 hilos $\rightarrow 12.041573/5.124451 = 2.35$
Eficiencia $\rightarrow 2.35/4 = 0.587$
Comparación respecto a 8 hilos $\rightarrow 12.041573/3.480211 = 3.46$
Eficiencia $\rightarrow 3.46/4 = 0.865$

Podemos ver cómo a medida que aumenta N aumenta el rendimiento cuando se tienen más hilos de forma considerable, a sí mismo a medida que N tiende a infinito el tiempo de ejecución crece demasiado.