



PRIMER PARRCIAL

ESTUDIANTES

Javier Andrés Lasso Rojas – 2061149-3743

DOCENTE

John Alexander Sanabria Ordoñez

INFRA. PARALELAS Y DISTRIBUIDAS

UNIVERSIDAD DEL VALLE

FACULTAD DE INGENIERÍA

SANTIAGO DE CALI

2024

```
javierlasso@LAPTOP-3HKBS1ST:~/paralelas-distribuidas-1er-parcial$ lscpu
```

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:          48 bits physical, 48 bits virtual
Byte Order:             Little Endian
CPU(s):                 6
On-line CPU(s) list:    0-5
Vendor ID:              AuthenticAMD
Model name:             AMD Ryzen 5 4500U with Radeon Graphics
CPU family:             23
Model:                  96
Thread(s) per core:     1
Core(s) per socket:     6
Socket(s):              1
Stepping:               1
BogoMIPS:               4741.13
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr
                        _opt pdpe1gb rdtscp lm constant_tsc rep_good noopl tsc_reliable nonstop_tsc cpuid extd_apicid pni pclmulqdq ssse3 fma cx16 s
                        se4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm cmp_legacy svm cr8_legacy abm sse4a misalignsse 3dno
                        wprefetch osvw topoext perfctr_core ssbd ibrs ibpb stibp vmcall fsgsbase bmi1 avx2 smep bmi2 rdseed adx smap clflushopt cl
                        wb sha_ni xsaveopt xsavec xgetbv1 clzero xsaveerptr arat npt nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pause
                        filter pfthreshold v_vmsave_vmload umip rdpid

Virtualization features:
Virtualization:         AMD-V
Hypervisor vendor:      Microsoft
Virtualization type:    full
Caches (sum of all):
L1d:                    192 KiB (6 instances)
L1i:                    192 KiB (6 instances)
L2:                     3 MiB (6 instances)
L3:                     4 MiB (1 instance)
Vulnerabilities:
Gather data sampling:    Not affected
Itlb multihit:          Not affected
```

Las imágenes procesadas se encuentran en la carpeta del drive donde comparto todo el parcial.

Enlace al GitHub. <https://github.com/Jlroja/parcial1infraestructuras.git>

Tiempos de ejecución

Secuencial

10.788s

14.959s X

9.544s X

9.572s

12.572s

promedio 10.98s

numero de hilos = #nucleos (6)

9.791s X

8.565s

7.687s

7.630s X

8.920s

promedio 8.39 s

numero de hilos = #nucleos (12)

8.520s

9.208s

8.920s

9.503s X

8.396s X

promedio 8.83 s

Speedup

$$\text{Speedup} = T_{\text{secuencial}} / T_{\text{paralelo}}$$

Speedup con 6 hilos $10.98 \div 8.39 \approx 1.31$

Speedup con 12 hilos $10.98 \div 8.83 \approx 1.24$

Podemos observar que cuando ejecutamos el programa con 6 hilos se obtiene un tiempo mucho mas eficiente que en la versión secuencial entonces entiendo que la paralelización mejora de una muy buena manera el rendimiento. Y cuando lo ejecutamos utilizando el doble de hilos en este caso 12 se nota que el rendimiento mejora un poco más.

Al aumentar los hilos yo esperaba una gran disminución de tiempo por eso realice los cálculos varias veces, pero me doy cuenta de que no se puede notar una gran mejora en el rendimiento porque al momento de realizar estos cálculos estoy consumiendo más recursos al tener en uso diferentes aplicaciones y también que cuando creo mas hilos cada uno de estos consumiría recursos del sistema y se crearía una sobrecarga que no me ayudaría a tener un mejor rendimiento.

