

24 febrero 2015

Práctica 1

Utilizando la herramienta Parallel Amplifier identifica las regiones que hacen uso intensivo del CPU.

1. ¿Cuáles son las líneas de código que demandan mayor tiempo del CPU?

Se puede apreciar que el 99.1% del tiempo fue invertido en el código que conforma los threads, dentro el ciclo for, con la llamada a la función blackandwhite como la más tardada dentro del ciclo.

View Source				
Function Call Sites and Loops	Total Time %	Total Time	Self Time	Hot Loops
Total	100.0%	1.0997s	0s	
RtlInitializeExceptionChain	99.1%	1.0897s	0s	
threadProcessBMP	99.1%	1.0897s	0s	
[loop at main.cpp:51 in threadProcessBMP]	99.1%	1.0897s	0s	
[loop at main.cpp:52 in threadProcessBMP]	99.1%	1.0897s	0.9946s	
blackandwhite	8.6%	0.0951s	0.0951s	
pre_c_init	0.9%	0.0100s	0s	
RtlInitializeExceptionChain	0.9%	0.0100s	0s	
_tmainCRTStartup	0.9%	0.0100s	0s	

2. Basándonos en la información que obtenemos en la pregunta anterior, ¿cuál es la mejor forma de paralelizar?

Asignando tareas iguales y estáticas, con el menor tiempo de inicialización posible.

3. Basándonos en la ley de Amdahl y el reporte que nos muestra Parallel Amplifier, ¿Cuál es la aceleración esperada a lograr con 2 y 4 procesadores?

Se espera casi (99.1%) duplicar y cuatruplicar los tiempos de ejecución.