Taller de Sistemas Operativos, 2020

Taller 02

Escuela de Ingeniería Informática

Rodrigo Montenegro Farias

rodrigo.montenegro@alumnos.uv.cl

Resumen.

El objetivo principal de este taller es solucionar la problemática, que consiste en la creación de un programa en "c++", el cual, permite implementar dos módulos, ambas tareas se realizan de forma paralela e independiente mediante la implementación de threads POSIX, para realizar pruebas de desempeños que generen datos del comportamiento del tiempo de ejecución de ambos módulos y sus procesadores.

1. Introducción

El estudio generado por el taller 2, hace énfasis a un programa hecho en lenguaje c++[3]. Básicamente el código a implementar tiene como función desarrollar una forma concurrente de hilos en base a dos módulos para descomponer el trabajo de una forma simple y eficiente, el primero tiene como función llenar un arreglo de números enteros, son de de tipo uint32_t, el segundo módulo corresponde a la función sumar el contenido, ambos ocurren en forma paralela, se realiza para pruebas de desempeño. Los números implementados son aleatorios generados por la función thread safe[], este programa se compila con makefile y su forma de uso para ejecutar es como se muestra en la siguiente tabla 1.

./sumArray -N <nro> -t <nro> -l <nro> -L <nro> [-h]

tabla 1 forma de uso.

Para ejecutar el archivo en el bash[1], se utilizan estos parámetros como se representa en la tabla 2.

-N : tamaño del arreglo. -t : número de threads. -l : límite inferior rango aleatorio. -L : límite superior rango aleatorio. [-h] : muestra la ayuda de uso y termina.

tabla 2 parámetros para su funcionamiento.

Cada parámetro será utilizado para perfecto funcionamiento del código, es decir especificando la cantidad de hilo, tamaño de arreglo, y límites de números enteros randómicos.

1.1 conceptos previos

Lenguaje c++ [3], este lenguaje de programación corresponde a la extensión del lenguaje c.

Comandos Bash [1], son un conjunto de parámetros utilizados para la administración y configuración del sistema.

Pthreads [2], es un modelo de ejecución de forma concurrente junto a otros hijos.

thread safe[5], puede invocar múltiples hilos en ejecución sin preocuparnos de los datos que acceda.

taskset[4], permite establecer mediante comando en que core se ejecutará el proceso.

2. Descripción del problema

2.1 contexto del programa

El contexto del problema a enfrentar es en base a realizar un código, la funcionalidad del programa debe estar compuesto de dos módulos, uno que llene un arreglo de números enteros aleatorios del tipo “uint32_t”(enteros sin signo de 32 bits) y otro que sume el contenido del arreglo. Se hacen pruebas de desempeño que generan datos que permitan visualizar el comportamiento del tiempo de ejecución de ambos módulos dependiendo del tamaño del problema y de la cantidad de hilos utilizados, en la siguiente figura 1, se podrá visualizar el diseño en general.

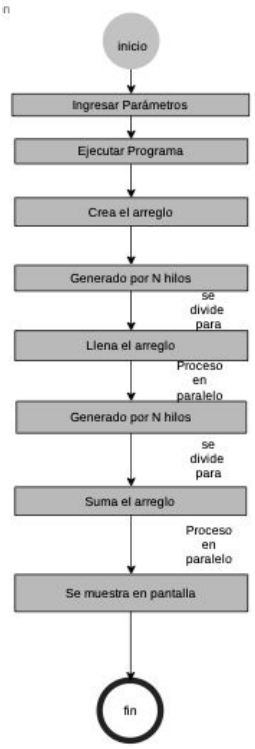


Figura 1, Visualización del diagrama general.

3. Descripción de solución

Mediante el anterior diagrama general tenemos una nocion de como es nuestro programa y cómo se comporta en base a su resultado final, nuestro programa da inicio a calcular el primer módulo, el cual llenar un arreglo de números enteros random y de forma concurrente el segundo módulo calcula la suma del arreglo generando la suma total, luego calcula el desempeño y resultado final en función del tiempo para finalizar mostrándolo por pantalla. En la siguiente figura se podrá visualizar el diagrama de alto nivel que aborda el problema.

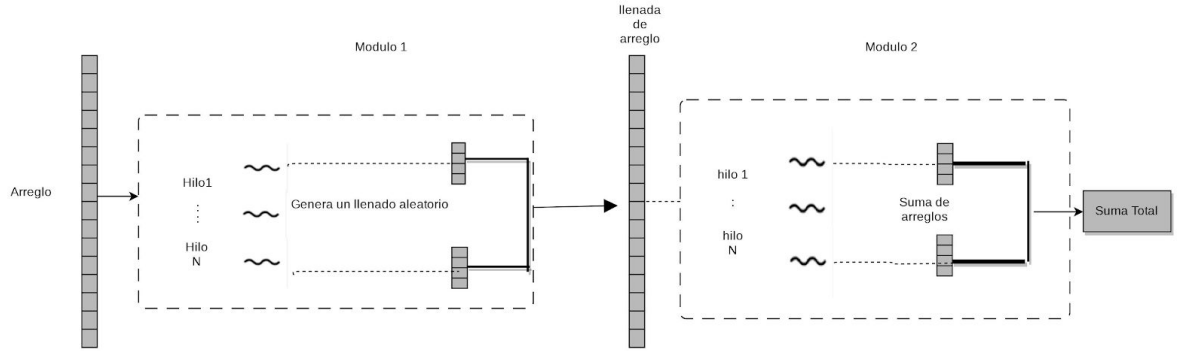


Figura 2, Visualización del diagrama de alto nivel.

Mediante el diagrama es posible ver el arreglo y sus funciones al pasar por los dos módulos. Su primera función será llenar cada arreglo vacío que poseerán los N hilos y serán sumados en el módulo siguiente, para mostrar por pantalla, la suma total generada.

La cantidad de hilos creados empiezan cuando el programa es ejecutado con sus parámetros, donde ambos módulos deben tener al menos 1 hilo cada uno que los ejecute. Mientras más hilos creados haya, en cada tarea de los módulos debe dividirse en pequeñas partes para cada hilo, para generar un programa más eficiente en los resultados finales.

El módulo 1, tiene como función llenar un arreglo de números random , el cual divide o troza la tarea en n números de hilos, los cuales son aleatorios y establecidos por parámetros de rangos.

El módulo 2. tiene como función sumar el arreglo de forma paralela y medite n hilos, luego se mostrará en pantalla la suma general.

3.1 Pruebas del programa

Estas pruebas nos permiten comprobar que efectivamente funciona la solución propuesta de hilos para nuestro problema, es posible visualizar con cuantos core los queremos hacer funcionar, mediante una herramienta nombrada taskset[4], la cual permite establecer en que core se ejecutará nuestro programa.

Es de esperar en la figura 3, la cual en su tiempo total nos muestra 5 este no contiene una mejora

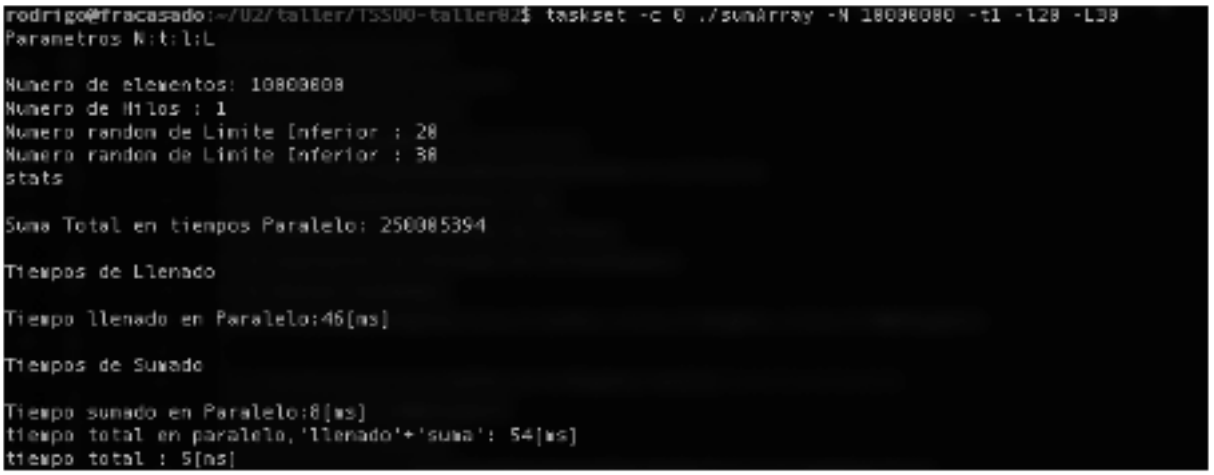
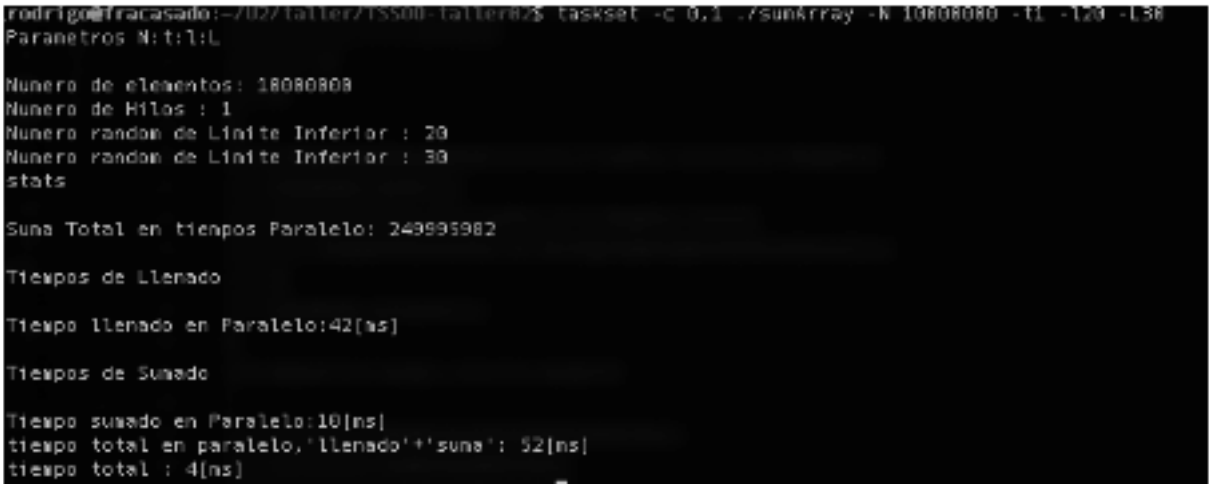


Figura 3 nos muestra la ejecución con 1 core,

En la figura 4, esta contiene 2 core, se puede ver en el tiempo total es igual a 4, que este disminuye a comparación del anterior.



4. Referencias

[1] “Ayuda Linux”, Listado de comandos importantes para Linux, Unix. Retrieved from

https://www.por-correo.com/index.php/articulos-de-interes/51-ayuda-linux-listado-de-comandos-importantes-para-linux-unix.html.

[2] Poxis “Pthreads” ,(2003). Retrieved from https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html.

[3] Programación “Que es c++” (2015). Retrieved from https://openwebinars.net/blog/que-es-cpp/ .

[4] taskset “taskset” (2009). Retrieved from https://www.oxfordscholarship.com/view/10.1093/acprof:osobl/9780199921959.001.0001/acprof-9780199921959-chapter-2

[5] Thread safe “Que es thread safe” (2000). Retrieved from https://esacademic.com/dic.nsf/eswiki/1149762

5. Conclusión

Como conclusión la programación multihilo, sin duda tiene mucho campo de aplicación, desde los sistemas operativos hasta en las tecnologías actuales que ocupamos cotidianamente, como por ejemplo; celulares y cajeros automáticos.
En la elaboración de este trabajo obtuvimos conceptos que nos darán la fluidez para desarrollar aplicaciones más contundente en c++ u otro lenguaje similar.
Siempre que utilizamos hilos es con la finalidad de poder realizar más de una tarea a la vez, administrar los recursos de la mejor forma posible y generar un programa mucho más rápido de lo habitual, aprendí a desarrollar un programa, el cual, está compuesto por hilos, acorde a la dificultad, este código era la necesidad de compartir su trabajo de lo contrario tardaría mucho, así que este método funciona cuando tenemos muchos procedimiento y podemos realizarlos de forma concurrente.