Emanuel Esquivel Lopez 2016133597
Taller 2
Arquitectura de computadores II

Investigación:

1. **En qué consiste OpenMP**

   Es un API especializada para el trabajo con multiproceso de memoria compartida a la hora de programar como principal ventaja es la realización de paralelismo de alto nivel.

2. **¿Como se define la cantidad de hilos en OpenMP**

   `omp_set_num_threads(int num_hreads)`

   Con esta función podemos definir el número total de hilos a usar.

   `omp_get_num_threads()`

   Con esta función podemos ver cantidad de hilos utilizados.

3. **¿Cómo se crea una región paralela en OpenMP?**

   Es delimitada con #pragma omp papallel

4. **¿Cómo se compila un código fuente c para utilizar OpenMP y qué encabezado debe incluirse?**

   Mediante el comando

   `gcc -o ejecFile file.c -fopenmp`

5. **¿Cuál función me permite conocer el número de procesadores disponibles para el programa?**

   Se hace mediante la función

   `omp_get_num_procs()`

   Realice un print con la función con los procesadores de su computadora.

   ```
   ema@ubuntu:~/Documents/Arqui2/Taller2/codigos$ gcc -o proc proc.c -fopenmp
   ema@ubuntu:~/Documents/Arqui2/Taller2/codigos$ ./proc
   Numero de procesadores disponibles: 4
   ema@ubuntu:~/Documents/Arqui2/Taller2/codigos$
   ```

6. **¿Cómo se definen las variables privadas en OpenMP?¿Por qué son importantes?**

   Se puede definir con la función `private(valor-i, …, valor-i_N)`

   Esto radica en cada hilo que hace uso de estas variables temporales, si una variable no es inicializada tampoco se mantiene fuera de la región paralela.

7. **¿Cómo se definen las variables compartidas en OpenMP?¿Cómo se deben actualizar valores de variables compartidas?**

   Mediante la función `shared(valor-i, …, valor-i_N)`

   Esto genera que todos los datos sean visibles para todos los hilos y accesibles por estos, todas por norma son compartidas en la región paralela excepto contadores.

8. **¿Para qué sirve flush en OpenMP?**

   Se emplea para verificar consistencia, esta permite exportar los hilos con valor modificado.

9. **¿Cuál es el propósito del pragma omp single? ¿En cuáles casos debe usarse?**

   Permite la ejecución de parte del código sea ejecutada en un solo hilo.

10. **¿Cuáles son tres instrucciones para la sincronización? Realice una comparación entre ellas donde incluya en cuáles casos se utiliza cada una.**

    a. static: se reparten todas las hiteraciones entre los hilos antes que se ejecuten en el bucle.
    b. dynamic: se reparte igual entre hilos.
    c. guided: múltiples iteraciones son asignadas a un mismo hilo.

11. **¿Cuál es el propósito de réduction y cómo se define?**

    Como objetivo tiene la de privatizar las variables listadas por cada hilo, al final la sección los hilos actualiza la variable global

Análisis

`pi`

1. **Identifique cuáles secciones se pueden paralelizar, ası´ como cuáles variables pueden ser privadas o compartidas. Justifique.**

   Se puede realizar en la sección del for, las variables compartida son pi, x, sum, la variable privada es la iteración i

2. **¿Qué realiza la función omp_get_wtime()?**

   Permite observar el tiempo del cronómetro, la suma de tiempos actividad de los hilos.

3. **Compile haciendo uso de OpenMP y ejecute el código modificando el parámetro de número de steps, realice una gráfica comparando.**

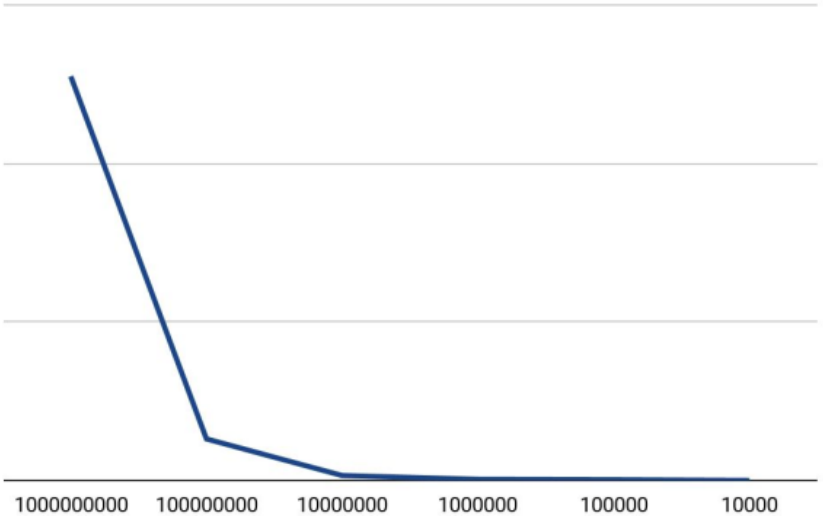


Figura 1: Steps vs time pi.

Como se ve al disminuir el número de steps disminuye el tiempo, lo cual era de esperarse ya que se ve que son directamente proporcionales, de manera casi exponencial.

`pi_loop`

1. **Explique cuál es el fin de los diferentes programas que se encuentran?**
    a. Primerao emplea openmp para crear los hilos solicitados
    b. Segundo pragma imprime la cantidad de hilos utilizados.
    c. Tercero se emplea para dividir iteraciones entre los hilos en proceso.
2. **¿Qué realiza la función omp get_num_threads()?**
   Permite obtener el número de hilos con lo que cuenta al realizar el proceso.
3. **En la línea 41, el ciclo se realiza 4 veces. Realice un cambio en el código fuente para que el ciclo se repita el doble de la cantidad de procesadores disponibles para el programa. Incluya un screenshot con el cambio.**

```
41          step = 1.0/(double) num_steps;
42          int proc = omp_get_num_procs();
43          for (i=1;i<=2*proc;i++){
44                sum = 0.0;
45                omp_set_num_threads(i);
46          start_time = omp_get_wtime();
47          }
```

Figura 2: Cambio en el codigo.

4. **Compile haciendo uso de OpenMP y ejecute el código modificando el parámetro de número de steps.**

5. **Mediante un gráfico de tiempo vs steps, pruebe 6 diferentes valores de steps. Explique brevemente el comportamiento ocurrido.**

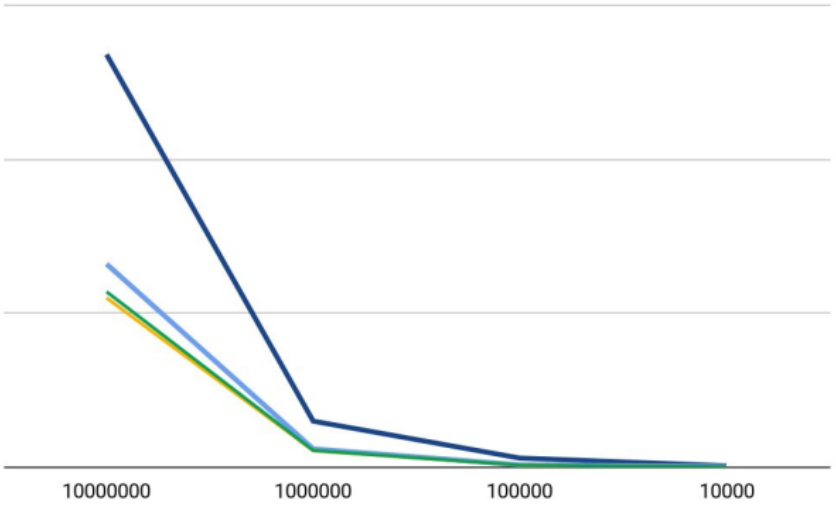


Figura 3: Steps vs time / diferente cantidad de hilos 1.2.3.4

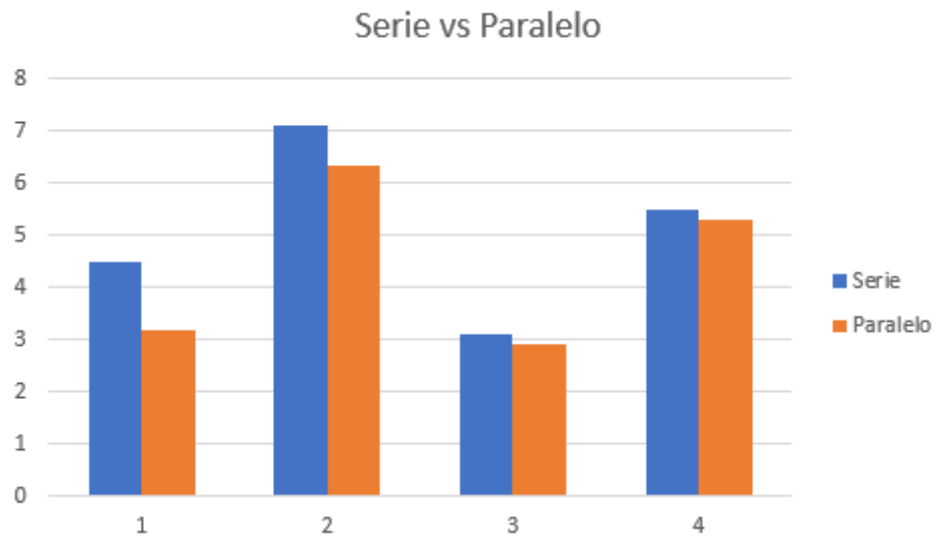
Nota: Azul:1, Celeste:2, Verde: 3, Amarillo: 4 (Hilos)

6. **Compare los resultados con el ejercicio anterior**

   Se aprecia igualmente, al disminuir la cantidad de steps disminuye el tiempo, y también se logra ver que si se aumenta el número de hilos el tiempo disminuye el tiempo de manera significativa.

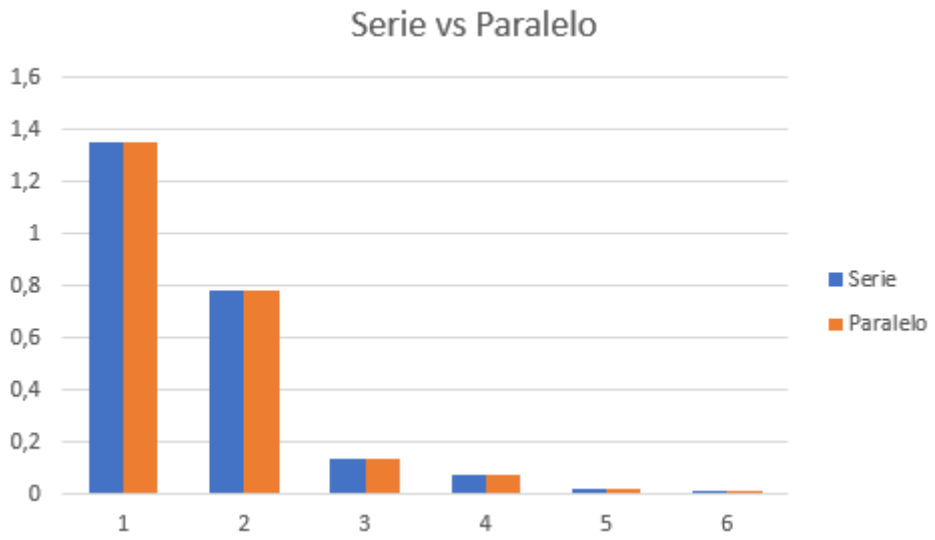
Ejercicios

1. **1. Realice un programa en C que aplique la operación SAXPY de manera serial y paralela (OpenMP). Compare el tiempo de ejecución de ambos programas para al menos tres tamaños diferentes de vectores.**



Como se ve en la gráfica, se compara los tiempos entre paralelo y serie, como se ve en todos los casos el paralelo cumple de mejor manera que el serie, los valores de vectores utilizados son 1M, 500k, 250k 150k respectivamente, este resultado es debido a la cantidad de hilos de ejecución utilizados en paralelo, los cuales son 4.

2. **Realice un programa en C utilizando OpenMP para calcular el valor de la constante e. Compare los tiempos y qué tan aproximado al valor real para 6 valores distintos de n**



Para este ejemplo se tomaron valores que van desde 100000000 hasta 500000, como se ve en la imagen los valores de este proceso en serie y en paralelo son prácticamente iguales por lo que en sí se puede ver que no hay mucha afectación, tambien por que el uso de hilos de limitar, pero en general tenemos un valor esperado.