Alumno: Christian Flores Meléndez

Informe: Algoritmos Paralelos

Multiplicación Matriz-Vector:

(Segundos)

Threads	Matrix Dimension					
	8000000x8		8000x8000		8x8000000	
	Time	Eff.	Time	Eff.	Time	Eff.
1	0.372	1.000	0.037	1.000	0.368	1.000
2	0.231	0.805	0.021	0.866	0.221	0.833
4	0.150	0.620	0.015	0.632	0.267	0.345

Lista enlazada Multi-threading:

100000 operaciones de las cuales el 99% son Member() , 0.5% Insert() y 0.5% Delete().

(Segundos)

Implementation	Number of threads			
	1	2	4	8
R-WLOCKS	1.752	1.212	1.02	1.506
OneMutexForEntireList	1.761	1.347	1.193	1.627
OneMutexPerNode	2.265	1.927	1.799	4.775

100000 operaciones de las cuales el 80% son Member(), 10% Insert() y 10% Delete().

(Segundos)

Implementation	Number of threads			
	1	2	4	8
R-WLOCKS	2.465	2.711	2.273	2.824
OneMutexForEntireList	2.812	2.623	2.494	5.167
OneMutexPerNode	5.602	5.527	5.279	9.082

Función strtok:

```
char *strtok(char *str, const char *delim);
```

No se debe usar strtok en un ámbito multithreading, debido a que el parámetro “str” es una variable en caché compartida entre todos los threads. Por lo cual se puede dar el caso que un thread sobrescriba los datos que eran usados por otro thread.
Una solución para este problema es el uso de strtok_r el cual dispone de un puntero para no perder de vista que la función se encuentra en la cadena de entrada.

4.4

El número de threads si afecta el tiempo de ejecución, esto se debe a que la creación de un thread tiene un costo ya que implica la asignación de un contador de programa, una pila de ejecución, separar espacio de memoria para el uso del thread, etc. De igual manera el destruir un thread conlleva un costo. Es por eso mientras más es el número de threads el tiempo de ejecución promedio se ve más afectado.

Alumno: Christian Flores Meléndez

4.6 Después de haber hecho las pruebas, concluyo que el rendimiento es similar si usamos un read write lock con preferecia al reader o con preferencia al writer. Pero también noté que dar preferencia al writer es más equitativo que dar preferencia al reader, pues al dar preferencia al reader aplazarán a los writers hasta que terminen todos los readers.