



CIENCIA DE LA COMPUTACIÓN

“Pthreads”

Algoritmos Paralelos CCOMP 7-1

KARELIA ALEXANDRA VILCA
SALINAS

SEMESTRE VII

2016

“La alumna declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo”

FIRMA

Tiempos de Lista Simple Enlazada Paralela

Especificaciones

100,000 ops/thread

99.9% Member

0.05% Insert

0.05% Delete

Tiempo en segundos

Implementation	Number of Threads			
	1	2	4	8
Read-Write Locks	0.213	0.123	0.098	0.115
One Mutex for Entire List	0.211	0.450	0.385	0.457
One Mutex per Node	1.680	5.700	3.450	2.700

Implementación	Numero de hilos			
	1	2	4	8
Read-Write Locks	0.682828903198 2422	0.344566106796 2646	0.215972185134 8877	0.243134021759 0332
One Mutex for Entire List	0.674429178237 915	0.807137966156 0059	1.351937055587 769	1.389462947845 459
One Mutex per Node	2.826874971389 771	1.720713853836 06	1.217756032943 726	1.581643819808 96

Especificaciones

100,000 ops/thread

80% Member

10% Insert

10% Delete

Tiempo en segundos

Implementation	Number of Threads			
	1	2	4	8
Read-Write Locks	2.48	4.97	4.69	4.71
One Mutex for Entire List	2.50	5.13	5.04	5.11
One Mutex per Node	12.00	29.60	17.00	12.00

Implementación	Numero de hilos			
	1	2	4	8
Read-Write Locks	1.058800220489502	0.9292051792144775	0.9012210369110107	1.267345905303955
One Mutex for Entire List	1.058229923248291	1.162974119186401	1.278278112411499	1.292840957641602
One Mutex per Node	2.802820205688477	1.616665840148926	1.085551977157593	2.58164381980896

Tiempo de ejecución y eficiencia de la multiplicación matrix-vector

Threads	Matrix Dimension					
	8,000,000 × 8		8000 × 8000		8 × 8,000,000	
	Time	Eff.	Time	Eff.	Time	Eff.
1	0.393	1.000	0.345	1.000	0.441	1.000
2	0.217	0.906	0.188	0.918	0.300	0.735
4	0.139	0.707	0.115	0.750	0.388	0.290

En segundos

Hilos	Dimenciones de la Matriz					
	8 000 000 * 8		8 000 * 8 000		8 * 8 000 000	
	Time	Eff.	Time	Eff.	Time	Eff.
1	0.425	1.118	0.337	1.159	<u>0.425</u>	1.000
2	0.217	1.001	0.231	0.866	0.417	0.901
4	0.195	0.487	0.172	0.473	0.285	0.287

Token

Programa con uso de strtok

- **Archivo:** strmal.c
- **Propósito:** Tratar de usar threads para tokenizar el texto entrante. Ilustrar los problemas con la función que no es segura para threads.
- **Advertencia:** Este problema definitivamente tiene problemas.
- **Input:** Lineas de texto
- **Output:** Para cada linea del input:
 - La linea es leida por el programa, y los tokens son identificados por strtok.
- **Compilación:** gcc -g -Wall -o pthread_tokenize strmal.c -lpthread
- **Uso:** pthread_tokenize <thread_count> <> <input>
- **Algoritmo:** Para cada linea del input, el siguiente hilo lee la linea y la "tokeniza".

Programa con uso de strtok_r

- **Archivo:** str.c
- **Propósito:** Tratar de usar threads para tokenizar el texto entrante. Esta función usa la función de seguridad de hilos, strtok_r.
- **Input:** Lineas de texto
- **Output:** Para cada linea del input:
 - La linea es leida por el programa, y los tokens son identificados por strtok_r.
- **Compilación:** gcc -g -Wall -o pthread_tokenize_safe1 str.c -lpthread
- **Uso:** pthread_tokenize_safe1 <thread_count> <> <input>
- **Algoritmo:** Para cada linea del input, el siguiente hilo lee la linea y la "tokeniza".

Conclusiones

No se recomienda el uso de strtok para multiples procesos ya que no es segura con hilos. Su correcto funcionamiento depende de que sea llamado en el ámbito de una función y no con variables globales, dado que la variable de strtok puede ser reemplazada en otro proceso o hilo.

La solución seria usar strtok_r que hace uso de una variable que almacena varios punteros para evitar la sobre escritura de variables en desuso. También se podría usar un mutex general al momento de tokenizar, pero el proceso seria mas lento , por ser serial.

Problemas:**4.4**

El tiempo de creación de hilos se ve afectado por la cantidad de estos si han sido liberados y ejecutan alguna tarea, pero la creación en si no debería demandar un tiempo mayor, aunque depende de la estructura en la que se guarda la información de los hilos, como son agregados, etc. Todo esto lo regula el sistema operativo.