COMPARACIONES EN MPI

Linked List

La siguiente tabla muestra las comparaciones entre la implementación de listas enlazadas con diferente funcionalidad: Mutex, Locks y un solo Mutex por lista.

Insert = 1000 Delete = 100 Search = 100

N.º THREAD	MUTEX NODO	MUTEX LISTA	READ-WRITE
1	0.82	0.588	1.075
2	0.628	1.055	1.044
4	0.518	1.029	0.871
8	0.508	1.016	1.069

Insert = 20200 Delete = 100 Search = 100

N.º THREAD	MUTEX NODO	MUTEX LISTA	READ-WRITE
1	1.901	3.935	4.584
2	1.849	4.33	4.561
4	1.891	4.547	4.542
8	2.008	4.428	4.444

Multiplicación Matrix Vector

Threads	Matrix Dimension					
	$8,000,000 \times 8$		8000×8000		$8 \times 8,000,000$	
	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

Implementación	8000000x8	2	8000x8000	8	8x8000000	
threads	tiempo	efi	tiempo	efi	tiempo	efi
1	0.18	1.00	0.21	1.00	0.20	1.00
2	0.11	0.85	0.12	0.48	0.12	0.23
4	0.09	0.88	0.08	0.61	0.09	0.28

Thread Safety

```
Pease porridge hot.
Thread 0 > my line = Pease porridge hot.

        Thread 0 > string 1 = Pease
        Thread 0 > string 2 = porridge
        Thread 0 > string 3 = hot.
Pease porridge cold.
Thread 1 > my line = Pease porridge cold.

        Thread 1 > string 1 = Pease
        Thread 1 > string 2 = porridge
        Thread 1 > string 3 = cold.
Pease porridge in the pot
Thread 0 > my line = Pease porridge in the pot

        Thread 0 > string 1 = Pease
        Thread 0 > string 2 = porridge
        Thread 0 > string 3 = in
        Thread 0 > string 4 = the
        Thread 0 > string 5 = pot
Nine days old.
Thread 1 > my line = Nine days old.

        Thread 1 > string 1 = Nine
        Thread 1 > string 2 = days
        Thread 1 > string 3 = old.
```

```
Pease porridge hot.
Thread 0 > my line = Pease porridge hot.

        Thread 0 > string 1 = Pease
        Thread 0 > string 2 = porridge
        Thread 0 > string 3 = hot.
Pease porridge cold.
Thread 1 > my line = Pease porridge cold.

        Thread 1 > string 1 = Pease
        Thread 1 > string 2 = porridge
        Thread 1 > string 3 = cold.
Pease porridge in the pot
Thread 0 > my line = Pease porridge in the pot

        Thread 0 > string 1 = Pease
        Thread 0 > string 2 = porridge
        Thread 0 > string 3 = in
        Thread 0 > string 4 = the
        Thread 0 > string 5 = pot
Nine days old.
Thread 1 > my line = Nine days old.

        Thread 1 > string 1 = Nine
        Thread 1 > string 2 = days
        Thread 1 > string 3 = old.
```

El método strok suele tener fallos debido a que su implementación no está pensada para ser utilizada con muchos procesadores, esto trae algunos problemas con la sincronización de los threads y fallos en los resultados, para evitar tener problemas la librería nos plantea un método llamado strtok_r que está pensado para una ejecución en paralelo

Codigo: https://github.com/EdherCerdan/Paralelos2019