Lock-Free SkipList-based concurrent Priority Queue

Harold Alejandro Villanueva Borda

Lock-Free SkipList

- Cola de prioridad concurrente.

- "A Pragmatic Implementation of Non-Blocking Linked-Lists" de Maged M. Michael

- Fast and Lock-Free Concurrent Priority Queues for Multi-Thread Systems de Håkan Sundell y Philippas Tsigas

Lock-Free SkipList

- `struct Markable Reference`
- `struct AtomicMarkableReference`
- `class SkipList`
- `#define SKIPLIST_TEMPLATE_ARGS template <typename KeyType, typename ValueType>// template arguments for the skip list class`
- `#define SKIPLIST_TYPE SkipList<KeyType, ValueType> // type of the skip list class`

`class SkipList`

- `struct SkipNode`
  - `void intialize_forward(const int forward_size, SkipNode *forward_target)`

Métodos principales de la SkipList:

- `bool SKIPLIST_TYPE::find_with_gc(const KeyType search_key, SkipNode **preds, SkipNode **succs)`
- `void SKIPLIST_TYPE::insert(const KeyType key, const ValueType &val)`
- `bool SKIPLIST_TYPE::remove(const KeyType key)`

Concurrent Priority Queue

- Evita la exclusión mutua
- La implementación de basa en la estructura skiplist
- `template <typename K, typename T, class Container = SkipList<K,T>>`
- `void push(K&& key, T&& value)`
- `std::shared_ptr<T> try_pop()`

Tabla secuencial y tabla 4 threads

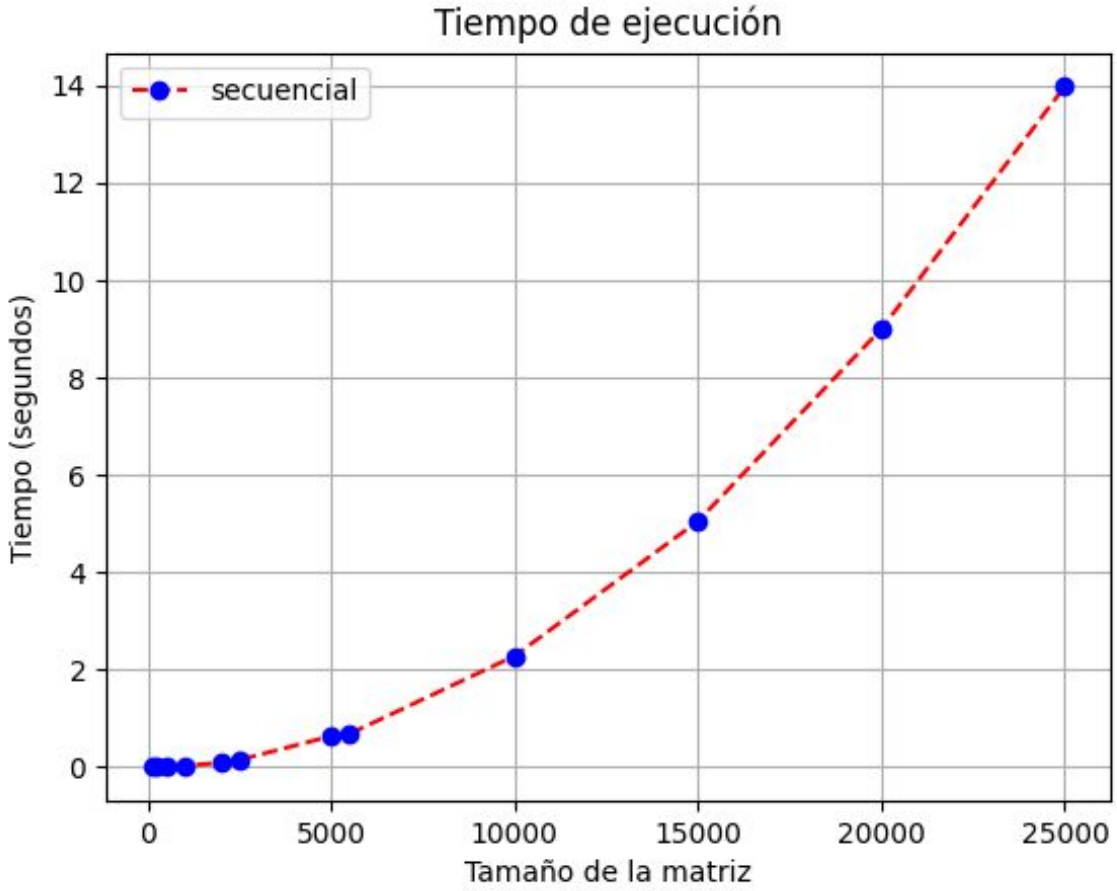
N	Tiempo (s)
100.0	0.00102
250.0	0.002
500.0	0.01197
1000.0	0.02707
2000.0	0.09082
2500.0	0.1478
5000.0	0.63797
5500.0	0.68371
10000.0	2.28364
15000.0	5.05393
20000.0	9.00105
25000.0	13.9796

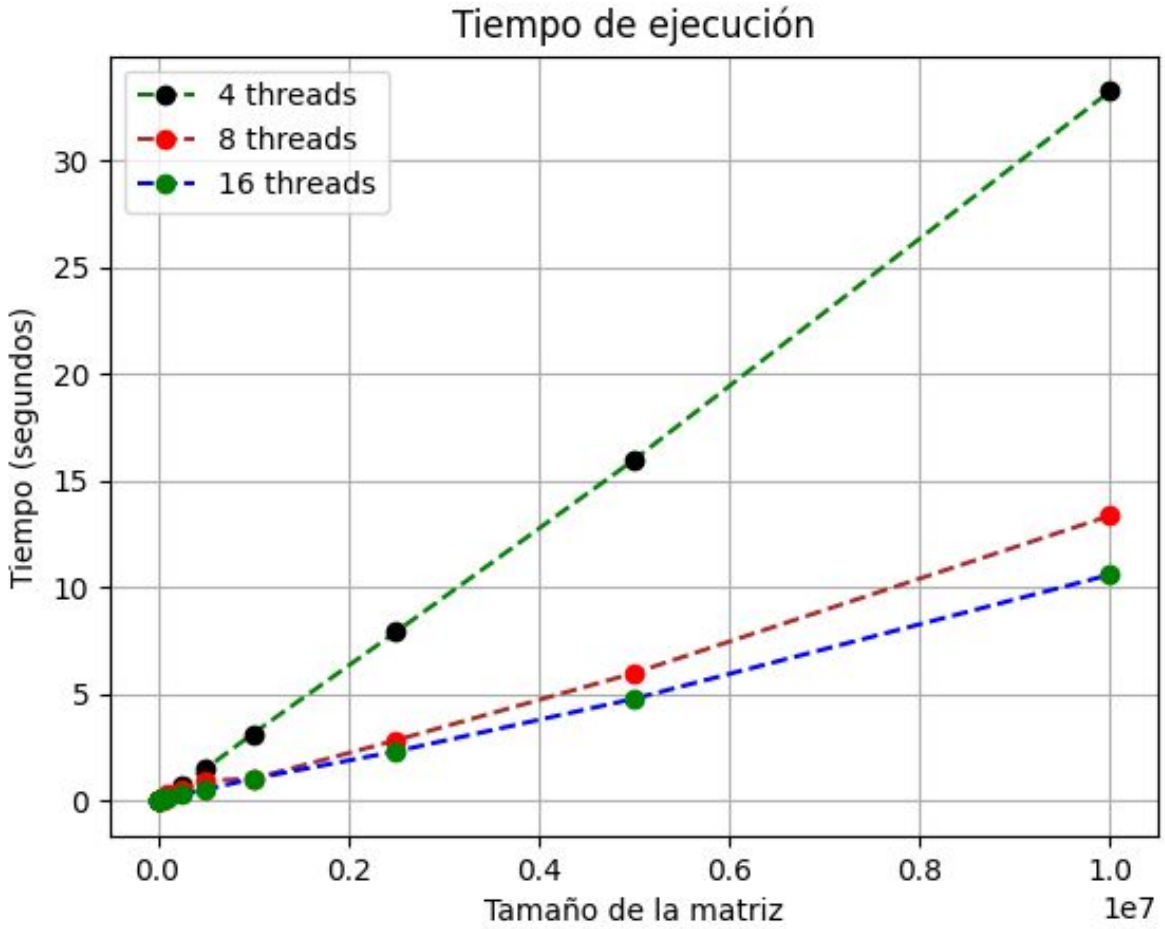
N	Tiempo (s)
100.0	0.003
500.0	0.00299
1000.0	0.00299
5000.0	0.00997
10000.0	0.02094
50000.0	0.1616
100000.0	0.20745
250000.0	0.70085
500000.0	1.49331
1000000.0	3.1568
2500000.0	7.91236
5000000.0	15.9778
10000000.0	33.2494

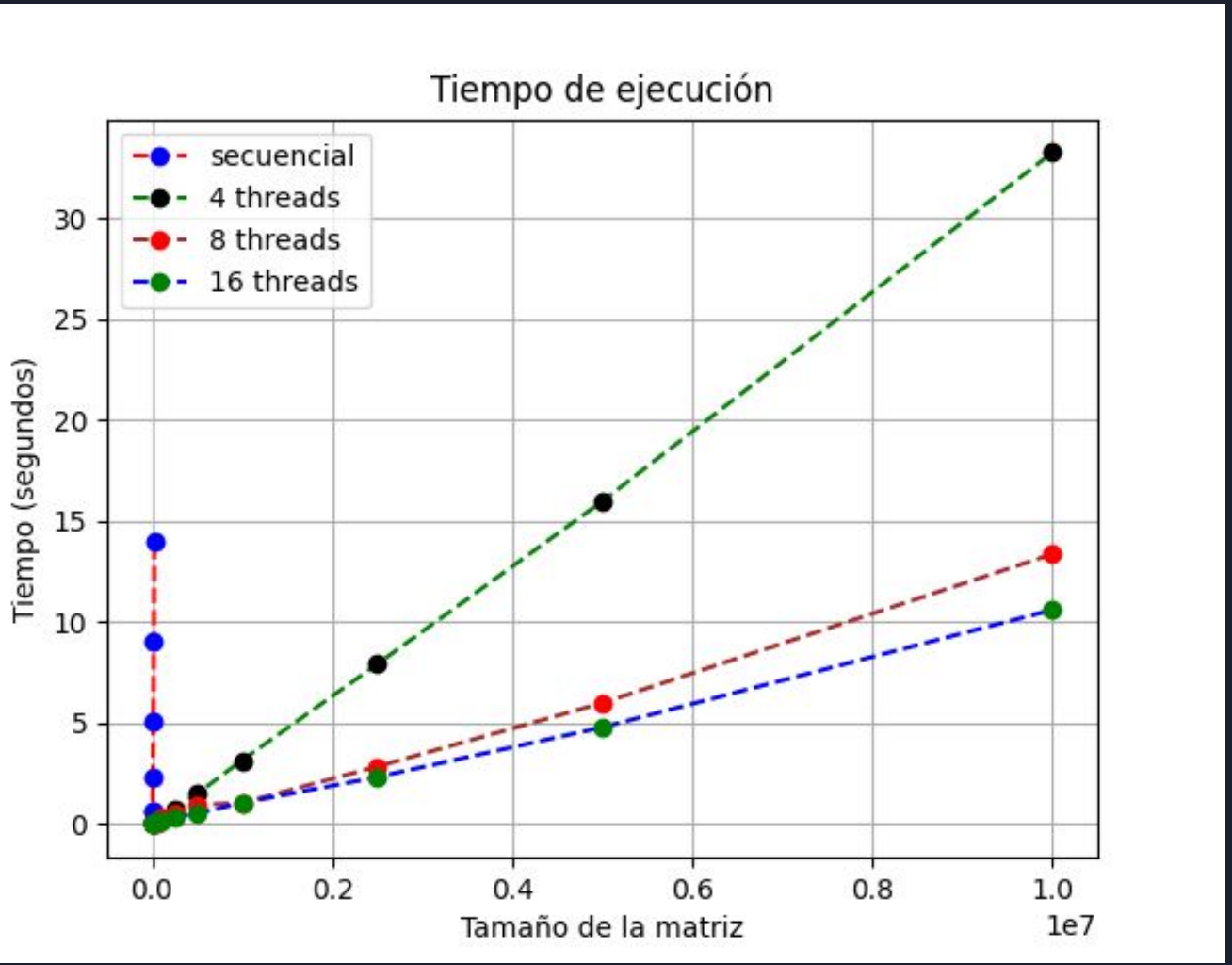
Tabla 8 y 16 threads

N	Tiempo (s)
100.0	0.00199
500.0	0.00299
1000.0	0.00399
5000.0	0.01496
10000.0	0.01695
50000.0	0.08129
100000.0	0.33091
250000.0	0.51721
500000.0	0.95013
1000000.0	1.04122
2500000.0	2.84187
5000000.0	5.99157
10000000.0	13.3678

N	Tiempo (s)
100.0	0.00399
500.0	0.00499
1000.0	0.00499
5000.0	0.02294
10000.0	0.02593
50000.0	0.12467
100000.0	0.11768
250000.0	0.29422
500000.0	0.53582
1000000.0	1.04155
2500000.0	2.32286
5000000.0	4.78649
10000000.0	10.604


Tiempo de ejecución
secuencial
Tiempo (segundos)
Tamaño de la matriz
0
2
4
6
8
10
12
14
0
5000
10000
15000
20000
25000


Tiempo de ejecución
4 threads
8 threads
16 threads
Tiempo (segundos)
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0
1e7
Tamaño de la matriz


Tiempo de ejecución
secuencial
4 threads
8 threads
16 threads
Tiempo (segundos)
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0
1e7
Tamaño de la matriz