

HW2 - Locked Probing Hash Table

MGP 2022 Spring
Due - 3.25 11:59 pm

[Homework Description]

- Submit a report via LearnUS
- Leave your source code in <home>/HW2

In this homework, your target is to improve the performance of a locked hash table. Unlike the lecture slides pages 83~87, you need to use an open addressing method.

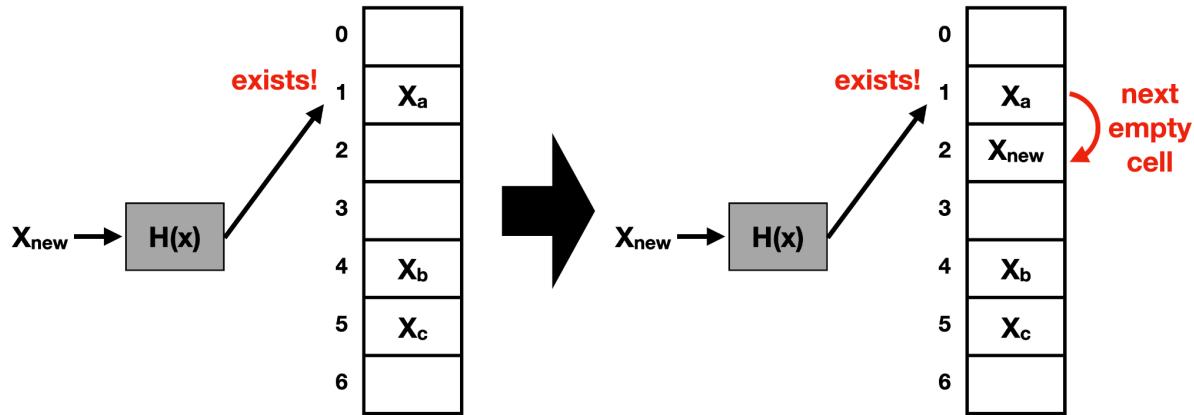


Figure 1. Linear probing.

The open addressing methods do not use chaining, i.e. there is only one element saved in a bucket. The main concept is that, if the hash collision occurs on the index $H(x)$, check whether the cell $H(x)+n$ is empty or not. In the case of linear probing, for example, n grows from 1 until it finds an empty cell. The item is then inserted into the index $H(x)+n$. Refer to Figure 1 for visual illustration.

You can use any open addressing method for this homework. Implement your own hash table in "better_locked_hash_table.h".

In the main function, we first initialize the hash table by inserting some elements in it. Then we measure the time to insert and read elements, i.e. we will operate an insert operation followed by 9 read operations.

In the given source code, there is a base version of a locked probing hash table.

```
// args:  
// ./HTtest HT_SIZE N_Init N_Test N_additional_read N_threads Use_custom  
$ make run  
.HTtest 10000000 4000000 4000000 9 16 0  
TABLE_SIZE 10000000 init: 4000000 new: 4000000 NT: 16 additional_reads: 9
```

```

use_custom: 0
baseline HT 0
start filling
init hash table took 1.65031 sec
start test
test 36000000 ops took 8.87275 sec
sanity check PASSED:
./HTtest 10000000 4000000 4000000 9 16 1
TABLE_SIZE 10000000 init: 4000000 new: 4000000 NT: 16 additional_reads: 9
use_custom: 1
better HT 1
...

```

You can assume we will check the performance for the default argument settings (and the provided pattern) only (i.e. 10M table size, 4M initial elements, 4M operations, 16 threads, use_custom:1).

(Below paragraph was modified at Mar. 20)

~~In this homework, there is no minimum performance bar. However, the amount of performance gain will be reflected in the score.~~

The performance bar is set to 0.6 sec. You will get a full point for the performance when your code achieves below 0.6 sec. In addition to that, if you reach somewhere between 0.4~0.2 sec, you will get additional 10% bonus points. Furthermore, if you succeed in reaching below 0.2 sec, you will take another 10% bonus points, i.e., 20% of bonus points. (Jinho worked on it for a couple hours, and his best was about **0.18 sec.**)

- TL; DR.:
 - [0.6 ,0.4) sec : full point
 - [0.4, 0.2) sec : full point + 10% bonus points
 - [0.2, 0.0) sec : full point + 20% bonus points

Just as HW1, we will use condor to measure the performance.

Use **make remote_base**, **make remote_better** commands, and refer to the HW1 description and rules for how to use it. For more information, check the "**readme.md**" file.

- You can use OpenMP if you want (I strongly think it will be useless)
- You can use SSE/AVX (I think it will be useless)
- You can change the hashtable structure (hash function, etc) as long as it is a kind of probing.
- You CANNOT change the main.cc or bucket.h
 - If you think you need to, contact us first
 - If you want to make a version that changes main.cc or bucket.h for experiments and include that only in the report, that is fine. Just don't do it in your main dir (i.e., <home>/HW2)

[Report]

Write a short 'scientific report' including the following contents. **The strict limit is 3 pages.** You should upload your report to **LearnUS** in time.

- PDF ONLY (penalties will be given to doc, hwp, etc formats)
- Self introduction (half-page)
 - Required: Photo, affiliation, Name, ID, e-mail address
 - Optional: Whatever you want to put for your introduction
- Main report
 - No fixed format, but please refer to existing scientific papers (e.g.,
<https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final197.pdf>) for examples of format and contents.
 - Include at least one proper references
 - We recommend using Latex. It is definitely prettier than MSword or hwp.
 - If 3 pages are too short, consider using double-column format
 - https://ieeecs-media.computer.org/assets/zip/Trans_final_submission.zip
 - https://ieeecs-media.computer.org/assets/zip/ieetran-final_sub.zip

We do grade the reports, and its quality matters.

- Basically, a report should tell the reader what problem has been solved, how you solved it, and what the experimental results are. Within its content, it is expected to have some take-away lessons from your experiments. We highly recommend including results from various experiment settings (different designs you tested, table size, number of threads, workload mix, etc), in forms of graphs or figures.
 - BTW, if your report is formatted as [1. What problem has been solved], [2. How I solved it], [3. what the experimental results are], etc... I promise that you will get a very low score :(
- You are required to include at least one reference in a proper format. Please read the [References] section at the end of this doc for more information.

[Note]

You will encounter the error below at the very beginning.

```
./HTtest 10000000 4000000 4000000 9 16 1
TABLE_SIZE 10000000 init: 4000000 new: 4000000 NT: 16 additional_reads: 9 use_custom:
```

```
1
better HT 1
start filling
HTtest: main.cc:44: void init_worker(hash_table*, int, int, int): Assertion `res == true' failed.
HTtest: main.cc:44: void init_worker(hash_table*, int, int, int): Assertion `res == true' failed.
HTtest: main.cc:44: void init_worker(hash_table*, int, int, int): Assertion `res == true' failed.
HTtest: main.cc:44: void init_worker(hash_table*, int, int, int): Assertion `res == true' failed.
.... // So many scary errors....
```

No worries.

It is just because you have not implemented the better_locked_probing_hash_table yet. The error will automatically disappear when the function is implemented correctly.

[References]

Just listing a few references that you've really "referenced" during your lab is NOT the correct way. Let's take an example.

=====BAD example=====

1. Body

I made a multi-threaded problem, which runs hashtables.

It uses cuckoo hashing, and I made it with fine-grained locks.

it works well!

2. References

[1] MGP 2022 Fall Lecture Notes, Jinho Lee

[2] Contemporary Logic Design, 2nd edition, Randy Katz

[3] Somebody else's lab report,
<https://www.coursehero.com/file/74974314/CECS-326-Lab-1-Report-1pdf/>

=====BAD example end=====

If this is your report, you're going to get at most 0 points for using references.

First of all, please DO NOT CITE my lecture notes.

You're supposed to cite documents that talked about something for the first time.

and my lecture notes are 95% from the textbook or someone else's papers.

Secondly, ALL your references should appear within the body text.

One of the core purposes of the references is to show something you've said in the report/paper is not your own thought but is already said by someone else.

It is not meant to be a meaningless list of stuff you've read.

If you cannot find a place that a certain reference should go into, you should probably remove that reference.

Lastly, When you cite a book, you should tell which page, or at least which chapter. A book is usually composed of a few hundred pages, and it's almost impossible to find the content if you don't tell the reader.

so, a better example would be as below

=====better example=====

1. Body

I made a multi-threaded problem, which runs hashtables.

It uses cuckoo hashing [1], and I did my thinking according to inductive reasoning similar to that of Newton's [2].

it works well!

2. References

[1] Pagh, Rasmus; Rodler, Flemming Friche (2001). "Cuckoo Hashing". Algorithms – ESA 2001. Lecture Notes in Computer Science. Vol. 2161.

[2] Isaac Newton: "In [experimental] philosophy particular propositions are inferred from the phenomena and afterwards rendered general by induction": "Principia", Book 3, General Scholium, at p.392 in Volume 2 of Andrew Motte's English translation published 1729.

=====better example end=====

for further details, you can have a look at other resources such as
<https://www.acm.org/publications/authors/reference-formatting>