

hernandez_pedro_project5_part2_report

Name: Pedro Hernandez Carranza

Date: 12-04-2025

Figure 1 : Elapsed Time



Figure 2 : Throughput

Hash Tables — LRU Profile

CSV file lru_profile.csv

Load example Metric Throughput (ops/ms) ▾

N log N baseline (for elapsed time) Overall throughput: operations completed per millisecond (higher is better).

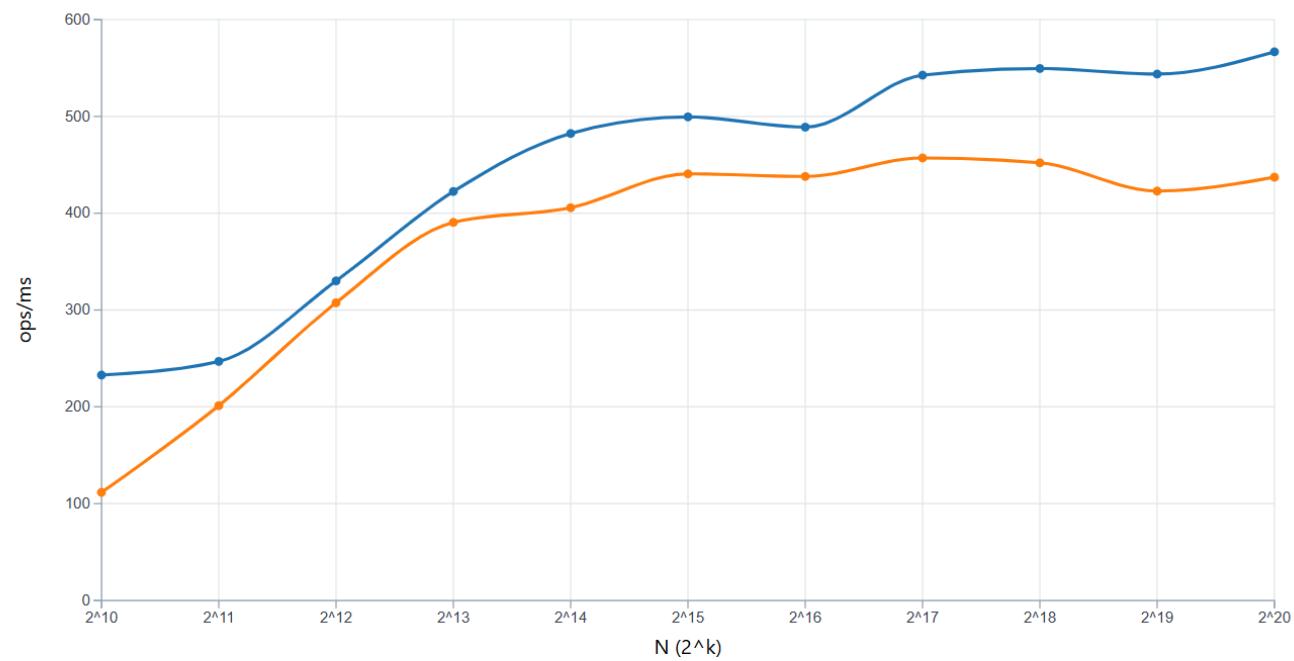


Figure 3 : Latency

Hash Tables — LRU Profile

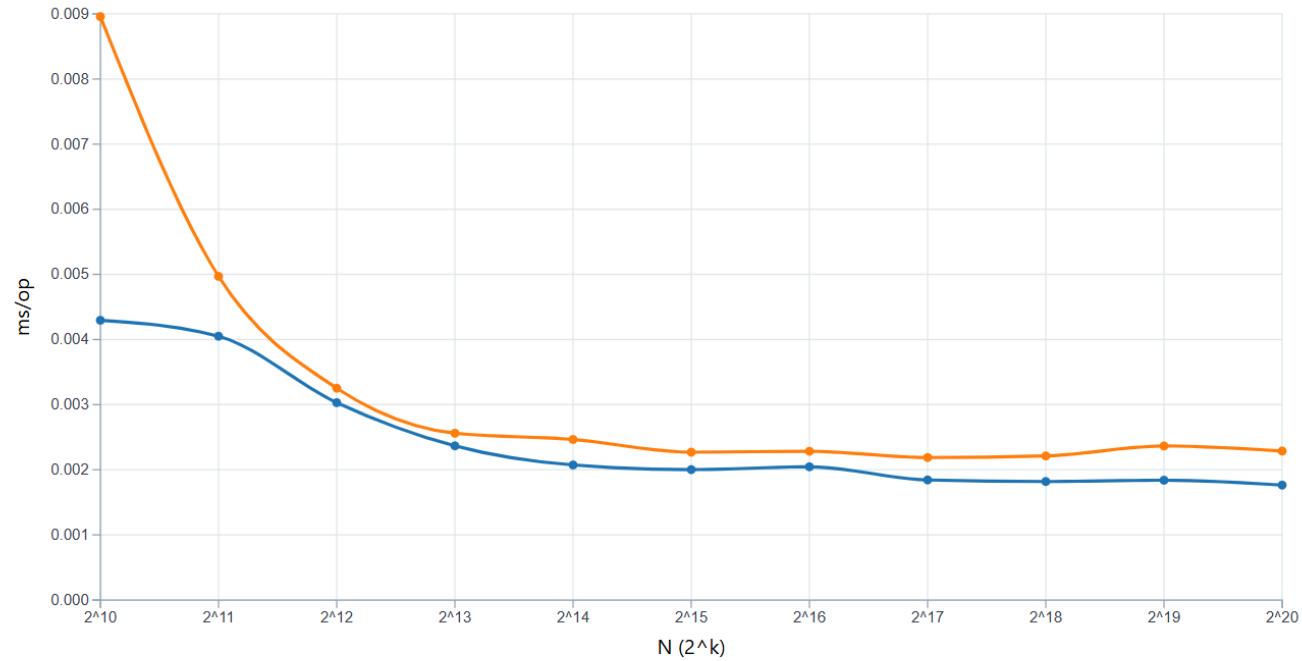
CSV file lru_profile.csv

Metric

Latency (ms/op)



N log N baseline (for elapsed time) Overall latency: milliseconds per operation (lower is better).



hash_map_double — compaction_on

hash_map_single — compaction_on

Figure 4 : Average Probes

Hash Tables — LRU Profile

CSV file lru_profile.csv

Metric

Average probes

N log N baseline (for elapsed time) Average probes per op (lower is better).

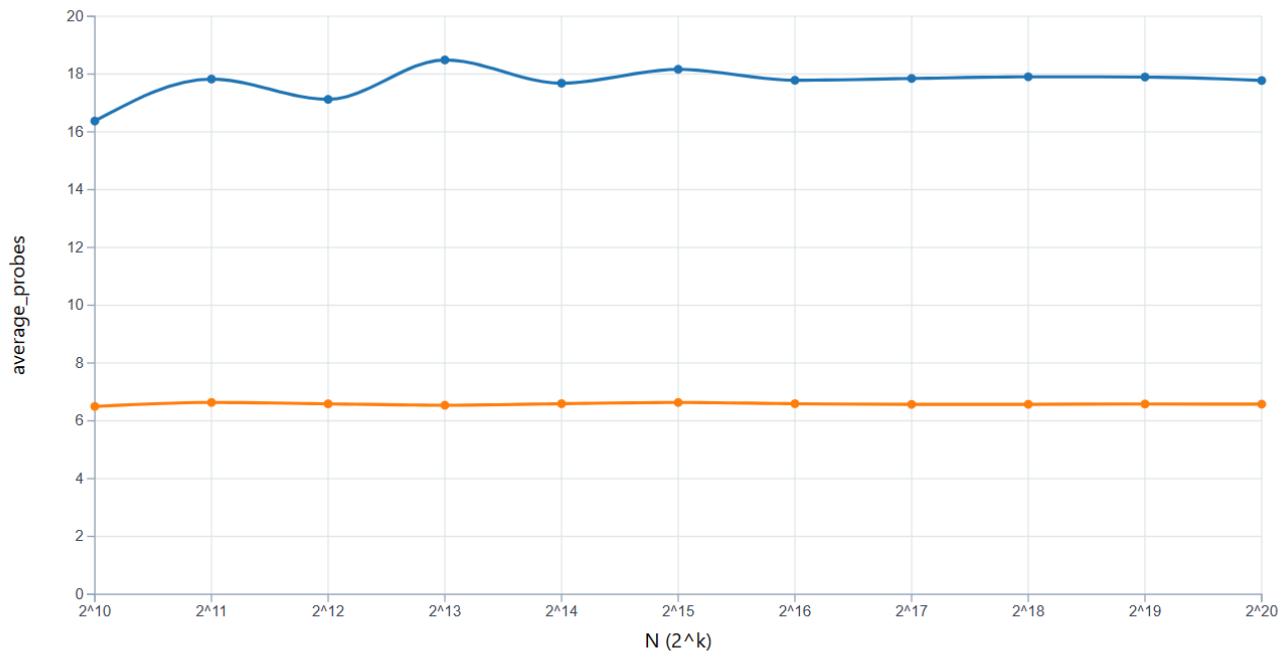


Figure 5 : Load Factor

Hash Tables — LRU Profile

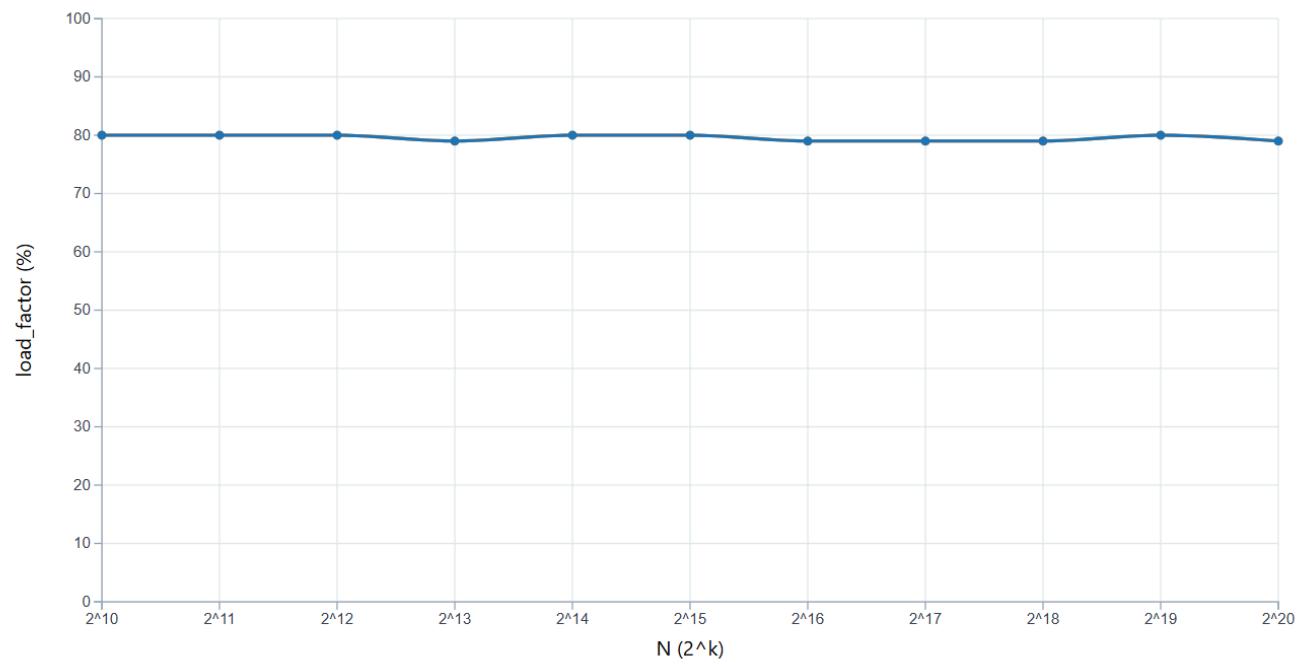
CSV file lru_profile.csv

Load example

Metric

Load factor (%)

N log N baseline (for elapsed time) Single occupancy metric (%).



hash_map_double — compaction_on hash_map_single — compaction_on

Figure 6 : Effective Load

Hash Tables — LRU Profile

CSV file

Choose File

Iru_profile.csv

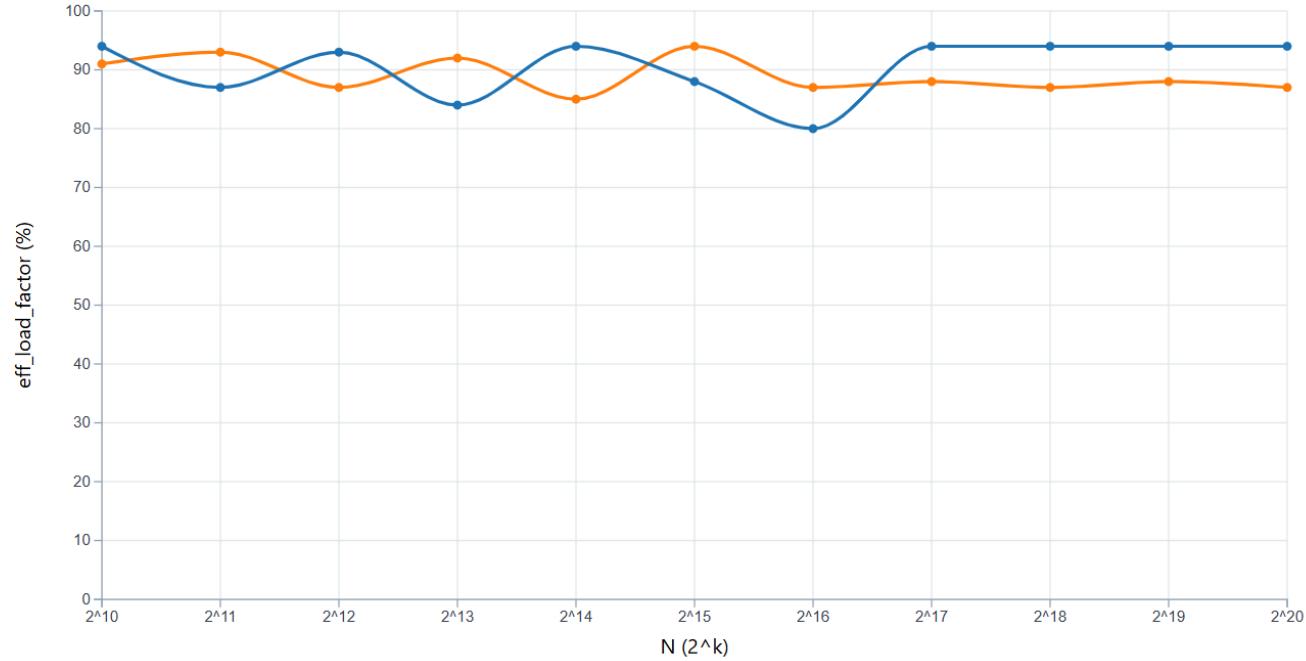
Load example

Metric

Effective load factor (%)

▼

N log N baseline (for elapsed time) Single occupancy metric (%).



hash_map_double — compaction_on

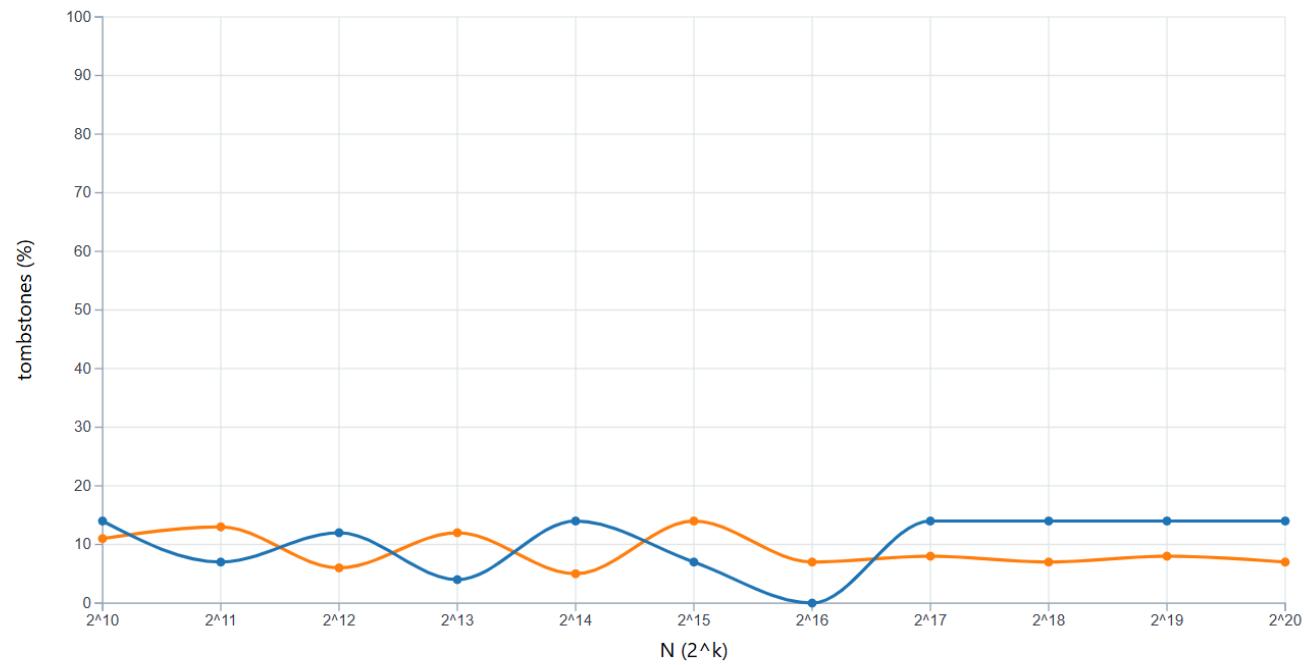
hash_map_single — compaction_on

Figure 7 : Tombstone(%)

Hash Tables — LRU Profile

CSV file lru_profile.csv Load example Metric

N log N baseline (for elapsed time) Single occupancy metric (%).



hash_map_double — compaction_on hash_map_single — compaction_on

Figure 8 : Occupancy

Hash Tables — LRU Profile

CSV file lru_profile.csv

Load example

Metric

Occupancy trio (eff/load/tomb, %) ▾

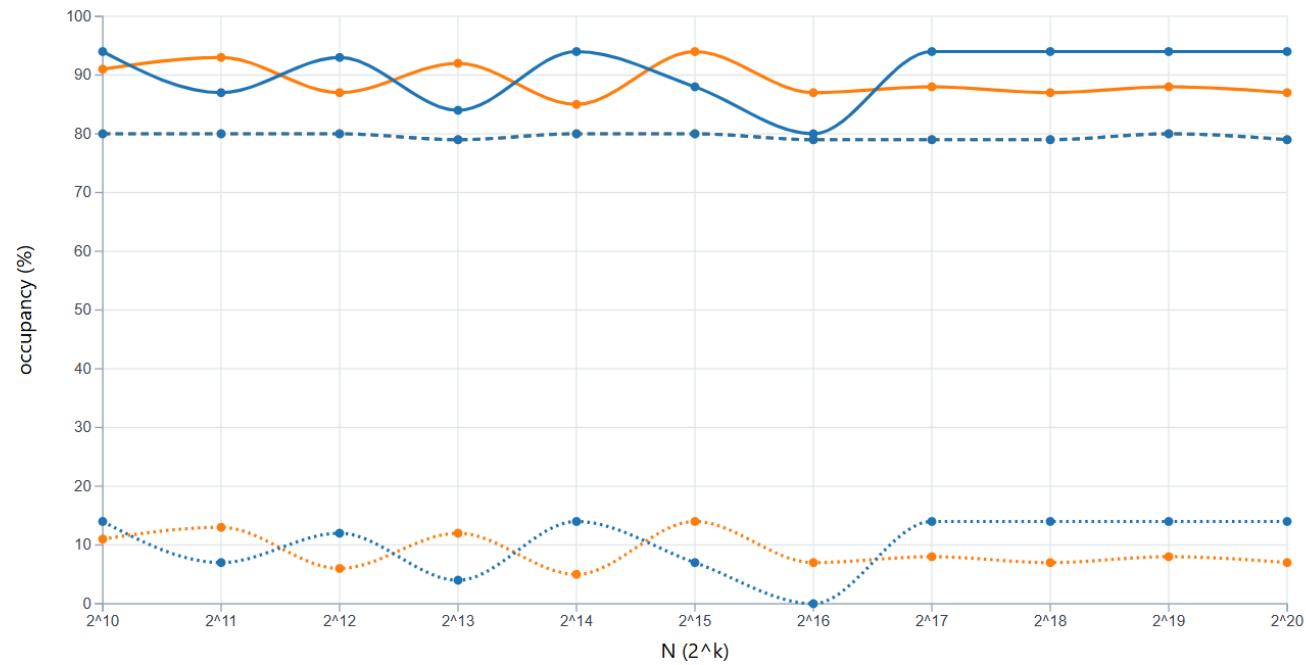
N log N baseline (for elapsed time)

— effective

- - - load

— tombstones

Three occupancy metrics together: effective (solid), load (dashed), tombstones (dotted).



hash_map_double — compaction_on

hash_map_single — compaction_on

Figure 9 : Compaction Counts

Hash Tables — LRU Profile

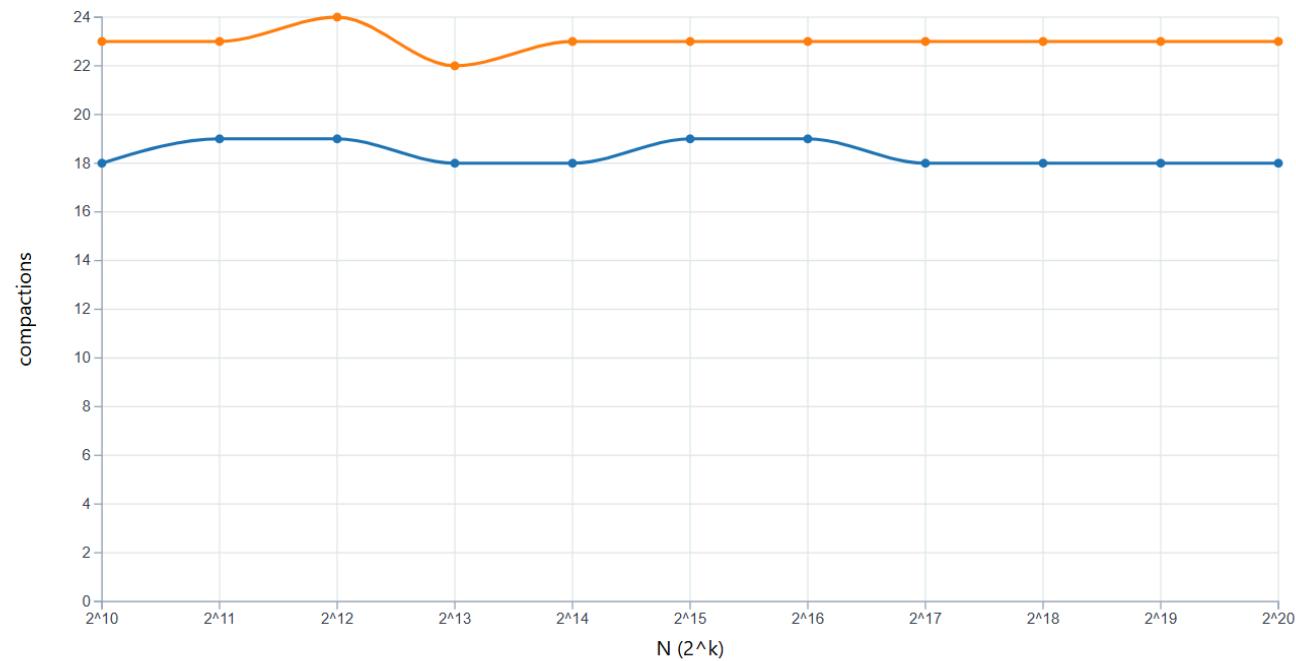
CSV file Iru_profile.csv

Metric

Compactions (count)



N log N baseline (for elapsed time) Number of table compactions triggered for this run.

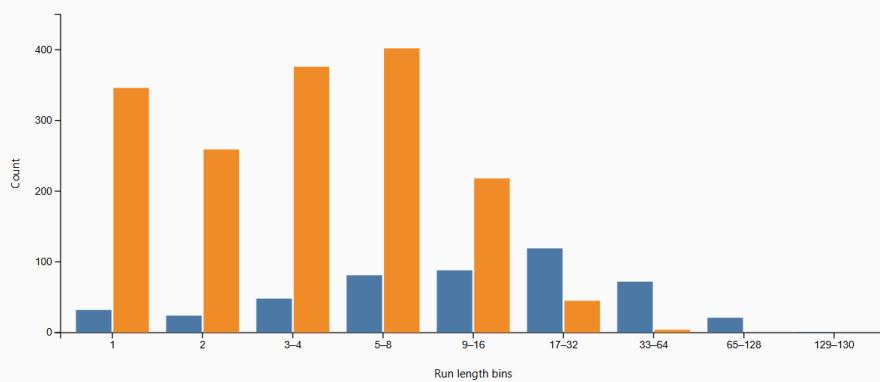


● hash_map_double — compaction_on

● hash_map_single — compaction_on

Figure 10 : Histogram of double probe (N=10273)

■ Before compaction ■ After compaction

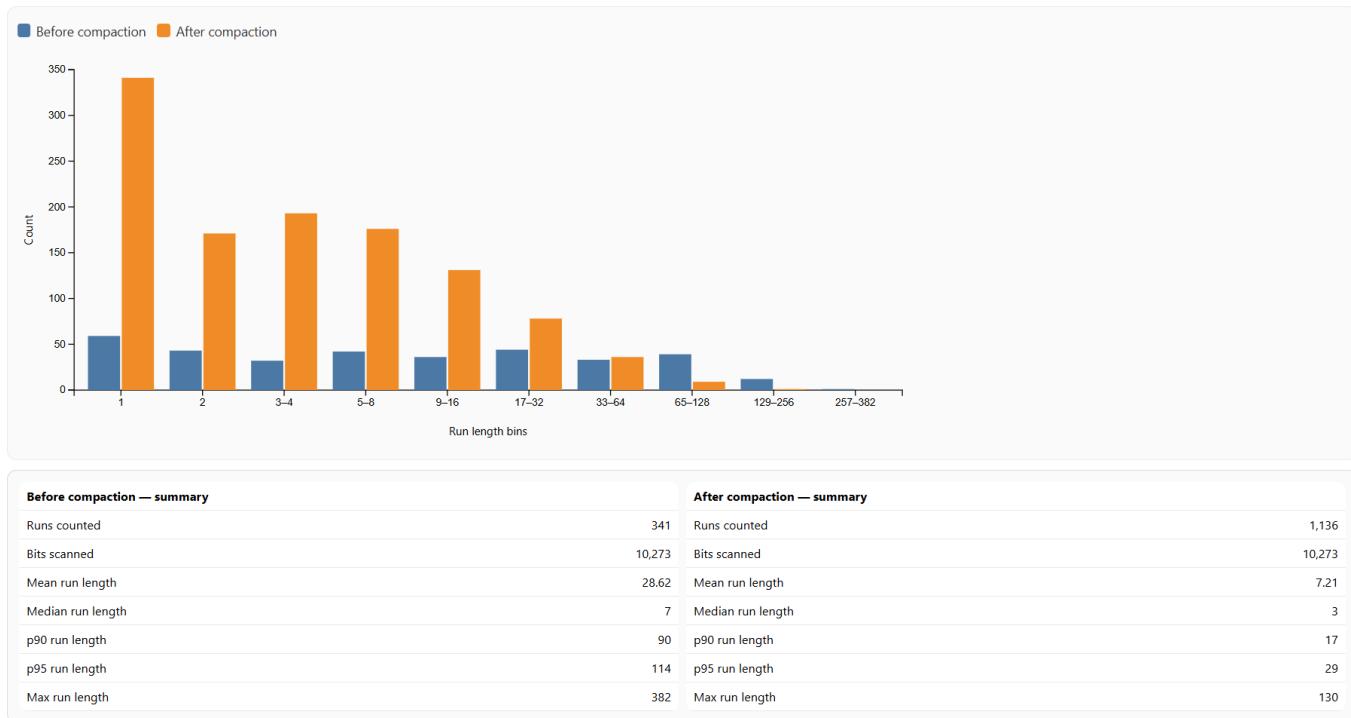


Before compaction — summary

Runs counted	486	Runs counted	1,650
Bits scanned	10,273	Bits scanned	10,273
Mean run length	20.08	Mean run length	4.96
Median run length	13	Median run length	4
p90 run length	47	p90 run length	11
p95 run length	58	p95 run length	14
Max run length	130	Max run length	40

After compaction — summary

Figure 11 : Histogram of single probe (N=10273)



1. Time & rate: elapsed time, throughput (ops/ms), and latency (ms/op).
 1. We expect the double probe to take more time per probe due to the look up having two steps instead of just one, so we expect the double probe to have a higher elapsed time and throughput. That being said, we expect double probes to take less time relative to single probes as the table becomes more full due to single probes having to probe more locations compared to double probes.
 2. Looking at Figure 1 we can see that double probes do in fact take longer especially for N values above 2^{17} . Looking at Figure 2 we can also see that a single probe throughout all N values has a higher ops/ms. Similarly we can see in Figure 3 that double probing fairly constantly has a higher latency again due to the extra step in look up. Ignoring the large discrepancy at N of 2^{10} in Figure 3, there is about a difference of 0.0001ms/op.
2. Work per op & occupancy: average probes, load factor vs. effective load, and tombstone %.
 1. We expect the load factor to be the same for both implementations as they are being run with the same trace files. For average probes

we expect a single probe to have a higher average due to it having to probe linearly. This should be the case especially for sections of the table with large continuous runs. For effective load we expect the double probe to be lower as its double probe approach would result in runs being shorter with more gaps in between. For tombstones we expect for a single probe to have more of them due to requiring them to move linearly as it searches for an available spot.

2. Looking at Figure 5 we can see that the load factors of single and double probes are the same regardless of the N being tested. This is expected as they are running the work load profile (trace file). Despite that the average probe count of single probe was on average 10 probes larger than that of double probe as shown in Figure 4. This is due to single probes' linear probing nature while double probes can move around more with its second hash option. The effective load is an interesting case as the hypothesis only partially holds. Looking at Figure 6 we can see that from N_s 2^{10} to 2^{16} single and double alternate on which has a higher effective load. Past N 2^{16} the loads appear to remain constant and as predicted with double probing having a smaller effective load of about 87% compared to single probes 94%.

3. Structure on the table: visual “maps” and run-length histograms that reveal clustering and tombstone buildup.

1. Taking a look at Figures 10 and 11 we can compare post compaction results for run data. We can see that for double prob the mean run length is 4.96 with a max run length of 40. Now looking at single probes we can see the mean run length is 7.21, roughly 68% larger than double probe's, and has a max run of 130, that's 325% higher than double probing. The large increase of runs in a single probe is due to it linearly probing after the first location is checked. This includes checking tombstones. The linear style of probing results in much of the data being

stuck together.

5.1 Expectations vs. observations

4. What did you expect the ordering to be (single vs. double) across small N and large N? Give a sentence of reasoning for each region.
 1. I expect for single and double to be very close at small N values. Single may even be faster for small N values since double prob uses more time in using its second hash function. I expect that as N gets larger, a single probe will begin to use more time and it would need to linearly probe through increasingly long runs or data, cache efficiently but time none the less.
 5. After plotting elapsed time vs. N, where does the data agree or disagree with your expectation? Identify any N-range where the gap widens or narrows notably.
 1. Looking Figure 1 we can see that the graphed lines are overlapped up to $N = 2^{15}$. Beyond this point double probing starts to take more time. Both follow the $N \log N$ shape however double probes increase more closely aligned with the increase. I think this increase in time compared to single probing may be due to more calls to the second hash function. The reason for single being faster than double may also be due to how cache friendly single probing is. Double probing jumps around memory which is not cache friendly. Interesting for the same points that double probe deviates from single, double probe also has less tombstones and lower effective load.
 6. Pick one N where the separation is largest. Roughly how big is the difference (percentage or ratio)?
 1. Looking at Figure 1, the largest difference between double and single probe takes place at $N=2^{20}$. With double elapsing 48946.100ms and single probe elapsing 37759.100ms, their difference being 11187ms. That's 11 seconds, a noticeable amount of time.
- #### 5.1.2 “Work” per operation vs. wall-clock time

7. Compare average_probes vs. elapsed time across N for both methods.
Do fewer probes consistently correlate with lower time in your runs?
 1. No, looking at Figure 4 the average number of probes stayed fairly consistent despite the N value. Single probe averaged roughly 17.8 probes for all Ns while double probe averaged 6.6 probes
8. Find any N where the ordering by average_probes and by time disagree.
Which other recorded metrics (e.g., hashing overhead, occupancy, compactions) could explain the mismatch?
 1. Looking at Figure 1 we can see that the two implementations start to differ at $N=2^{17}$. At and after this N value effective load of single and double probe remains 94% and about 87.5% respectively (Figure 6). The tombstone percentage also stabilizes with single probe having 14% and double probe having roughly 7.5% (Figure 7). The other metrics don't correlate with the disagreement. The values that do correlate suggest the opposite of the elapsed time graph so the difference must be in the cache friendliness of single probe compared to double probe.
- 5.1.3 Hashing cost and memory locality
9. Based on your implementation, how do the per-operation hashing costs differ? For which N-range would you expect constant factors to matter most, and why?
 1. For a single probe the hash function is called once and then steps forward linearly till an available spot is found. The cost of this pre operation is a constant factor of 1. Double probe has the same pre operation except that instead of linearly stepping forward it calls a second hash function so it has twice the constant factor of single probe. These factors play a small role but become increasingly important as the load of the table increases and more collisions occur. This would be the case for large N values in Figure 1 this is a contributing factor as to why double probing starts to take more time from $N=2^{17}$ and onward.
10. Look for signs of locality effects: at which N does the more contiguous method track more closely with time despite similar probe counts?

Where does the more “de-clustered” method start to pull ahead?

1. A single probe starts to pull ahead starting at $N=2^{17}$ as shown in Figure 1. Double prob in my test never overtakes single probe even after redoing the trials, but theoretically its shorter runs and "de-clustered" nature should allow it at greater Ns to push ahead of single probe.

5.1.4 Compaction effects (indirect inference)

11. Across N, which method triggers more compactions? Does that align with higher tombstones_pct or higher eff_load_factor_pct before compaction?

1. Looking at Figure 9 we can see that the double probing consistently did more compactions. While this should theoretically lead to double probing having consistently lower tombstone and effective load percentages in my testing this was not the case. The tombstone % in Figure 7 create a helix like structure and double probing only consistently had a lower percentage past $N=2^{17}$. The same can be seen in Figure 6 for effective load.

12. At the same N, do runs with more compactions also show higher elapsed time? Point to one example where this correlation is strongest (or weakest).

1. Runs with more compactions should have shorter elapsed times as they would have shorter continuous runs leading to less colossians, but the above testing illustrates that compaction was not the cause of double probing taking more time. Figure 4 shows that for all Ns double probe had more competitions yet looking at Figure 1 for $N \geq 2^{17}$ the elapsed time was higher than single probing.

13. Using the before/after maps for the final compaction (see the section related to histograms), do you see a structural change that would plausibly explain changes in average_probes?

1. Looking at the histograms, Figures 10-11, after the final compaction runs were greatly shortened for both single and double probes. This should have lowered the average probe but I was unable to capture that

in my testing.

5.1.5 Throughput and latency cross-check

14. Do throughput and latency plots tell the same story as elapsed time? If not, where do they diverge, and what's your explanation?
 1. Looking at Figure 2 we see that single probing has higher throughput at all N values with the largest difference being at $N=2^{20}$ with single being 566.710 ops/ms and double being 437.184 ops/ms. That's a large difference of 129.526 ops/ms. This large difference in ops/ms may be caused by the extra time spent using the second hash function. The Latency metric also supports this idea. Looking at Figure 3 we can see that for all Ns double probing had a higher latency than single probe. On average, excluding outliers at $N=2^{10}$ and $N=2^{11}$, is 0.000392ms/op. This is a small value but considering how large N gets difference leaves a noticeable impact on the performance of double probing.
15. Which metric makes the separation between single and double clearer at larger N, and why might that be?
 1. In my opinion the metric with the largest separation for large Ns is effective load percentage (Figure 6). Single and double probes remain 94% and about 87.5% respectively past $N \geq 2^{17}$. This metric illustrates the clustering of single probing and the effects of compacting.
16. For each N, note load_factor_pct, eff_load_factor_pct, and tombstones_pct. Do you see much change in these three measures?
 1. Looking at Figure 8 we can see that tombstones and effective load are directly correlated while the load factor stays constant for both single and double probing.
17. Choose one N where the time gap is largest. How do the occupancy percentages differ between methods there? Can you explain the gap?
 1. Looking at Figure 1 the largest time difference occurs at $N=2^{20}$. At this value the load factor is still the same. The single probe's effective load is 94% and its tombstone was 14%. The double probe's effective

load is 87% and its tombstone was 7%. These metrics suggest that a single probe would take more time but that was not the case for my trials. Most likely double probes increase latency and single probes cache friendliness play a large role resulting in double probes taking more time.

5.1.7 Before/after compaction structure (one snapshot)

18. How did the distributions shift across compaction? Relate the shift to any change in average_probes you observe for that run.

1. See Figures 10 and 11 for stats on the last compaction. Looking at Figure 4 we see that this particular compression did not play a significant role, but we can see that in Figures 10 and 11 that for both single and double the runs shifted left ward (got smaller).
- Where are the longest 1-runs in ACTIVE+DELETED—few big blocks or many medium ones?
 - Looking at Figure 10-11 roughly 30% of the runs are now in the 1-2 row bins, most of the others appear to be medium especially for the single probe histogram.
 - At the same N, do single and double probing show visibly different clustering?
 - The clustering is still more apparent in the single probe but to a far less extent as shown by Figure 11.
Histograms (overlay: before vs after)
 - Single vs double at the same N: which has more long runs in ACTIVE+DELETED?
 - Single has longer runs on average with a mean run of 7.21 which can be found at the bottom of Figure 11.
 - When you see longer runs in ACTIVE+DELETED, do you also see higher average_probes and higher elapsed_ms?

- The average probes stay the same, but the elapsed time continues to increase as the effective load stays fairly high especially for large N values.