Outline

We implemented query 1 and query 3 with a hash table. We decided to modify the size of the hash table in order to see its impact of query execution time. Our basic hypothesis was that increasing the hash table size would reduce query execution time.

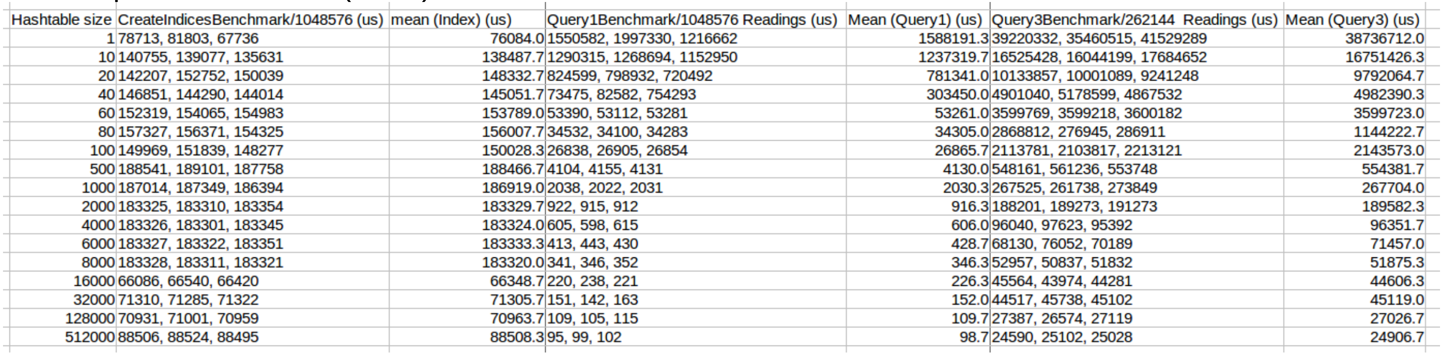
Experiment

- We ran the benchmark on a laptop with the following specs (found by running the benchmark): “(8 X 4000 MHz CPU s) CPU Caches: L1 Data 32K (x4) L1 Instruction 32K (x4) L2 Unified 256K (x4) L3 Unified 8192K (x1)”
- We varied the “`HASH_TABLE_SIZE`” variable in our code and recorded the execution time for creating the index, running Query1Benchmark/1048576, and running Query3Benchmark/262144.
- We carried out three repeats and then calculated the mean

Experiment Justifications:

- We chose to measure the performance on the query 1 and query 3 benchmarks that took the longest time in order to reduce percentage uncertainty in our readings.
- We didn’t measure the impact on query 2 as query 2 used a different implementation of a hash table.
- We took a mean average to reduce the impact of outliers and experimental error

Experiment Results (Table):

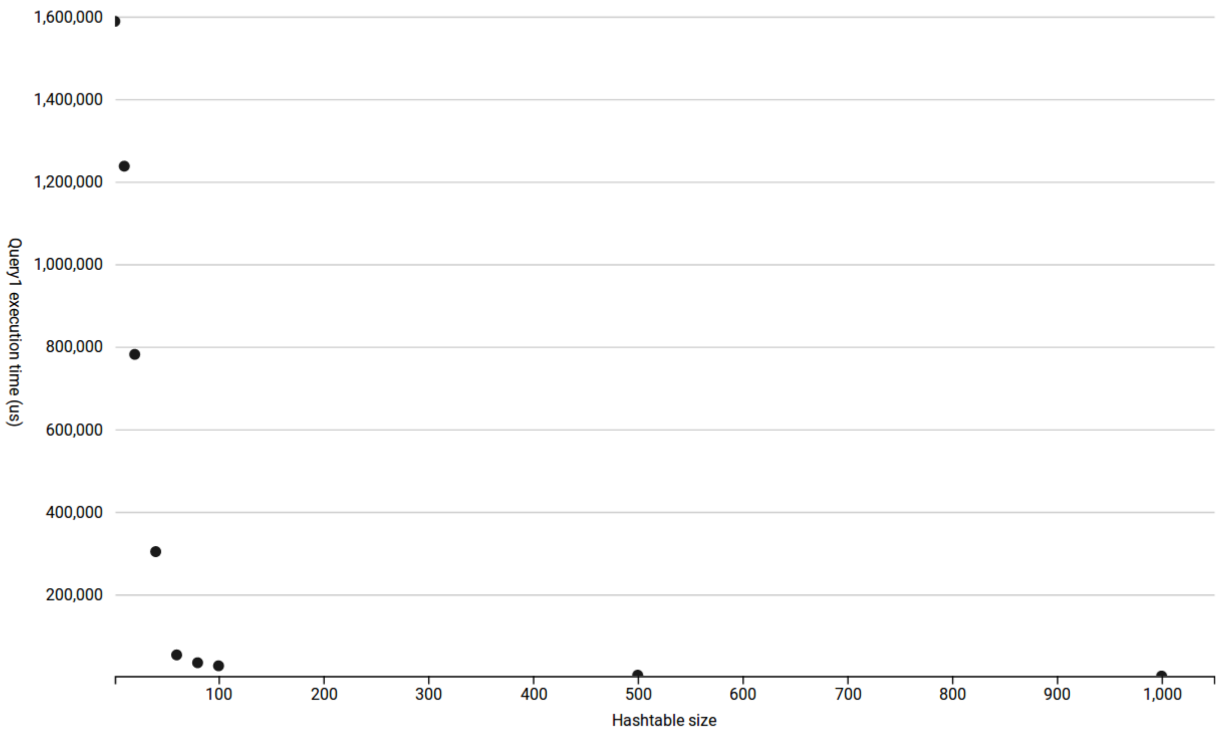


Hashtable size	CreateIndicesBenchmark/1048576 (us)	mean (Index) (us)	Query1Benchmark/1048576 Readings (us)	Mean (Query1) (us)	Query3Benchmark/262144 Readings (us)	Mean (Query3) (us)
1	78713, 81803, 67736	76084.0	1550582, 1997330, 1216662	1588191.3	39220332, 35460515, 41529289	38736712.0
10	140755, 139077, 135631	138487.7	1290315, 1268694, 1152950	1237319.7	16525428, 16044199, 17684652	16751426.3
20	142207, 152752, 150039	148332.7	824599, 798932, 720492	781341.0	10133857, 10001089, 9241248	9792064.7
40	146851, 144290, 144014	145051.7	73475, 82582, 754293	303450.0	4901040, 5178599, 4867532	4982390.3
60	152319, 154065, 154983	153789.0	53390, 53112, 53281	53261.0	3599769, 3599218, 3600182	3599723.0
80	157327, 156371, 154325	156007.7	34532, 34100, 34283	34305.0	2868812, 276945, 286911	1144222.7
100	149969, 151839, 148277	150028.3	26838, 26905, 26854	26865.7	2113781, 2103817, 2213121	2143573.0
500	188541, 189101, 187758	188466.7	4104, 4155, 4131	4130.0	548161, 561236, 553748	554381.7
1000	187014, 187349, 186394	186919.0	2038, 2022, 2031	2030.3	267525, 261738, 273849	267704.0
2000	183325, 183310, 183354	183329.7	922, 915, 912	916.3	188201, 189273, 191273	189582.3
4000	183326, 183301, 183345	183324.0	605, 598, 615	606.0	96040, 97623, 95392	96351.7
6000	183327, 183322, 183351	183333.3	413, 443, 430	428.7	68130, 76052, 70189	71457.0
8000	183328, 183311, 183321	183320.0	341, 346, 352	346.3	52957, 50837, 51832	51875.3
16000	66086, 66540, 66420	66348.7	220, 238, 221	226.3	45564, 43974, 44281	44606.3
32000	71310, 71285, 71322	71305.7	151, 142, 163	152.0	44517, 45738, 45102	45119.0
128000	70931, 71001, 70959	70963.7	109, 105, 115	109.7	27387, 26574, 27119	27026.7
512000	88506, 88524, 88495	88508.3	95, 99, 102	98.7	24590, 25102, 25028	24906.7

Experiment Results (Graphs)

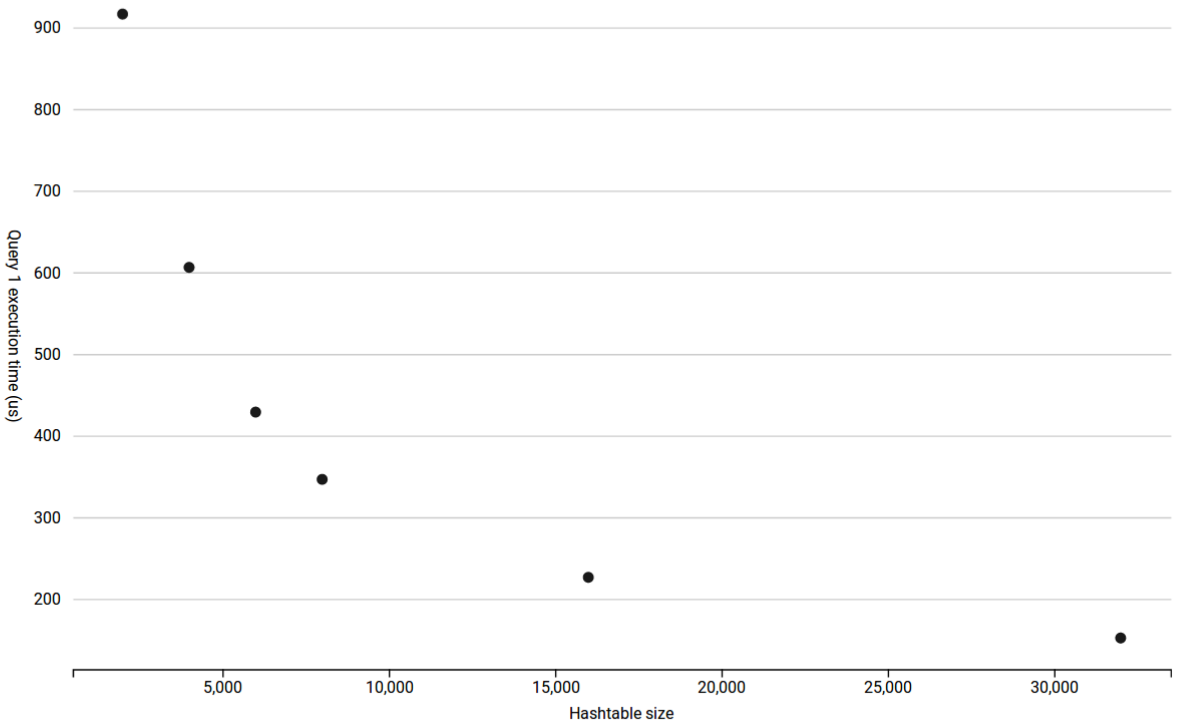
The impact of hashtable size on query 1's execution time

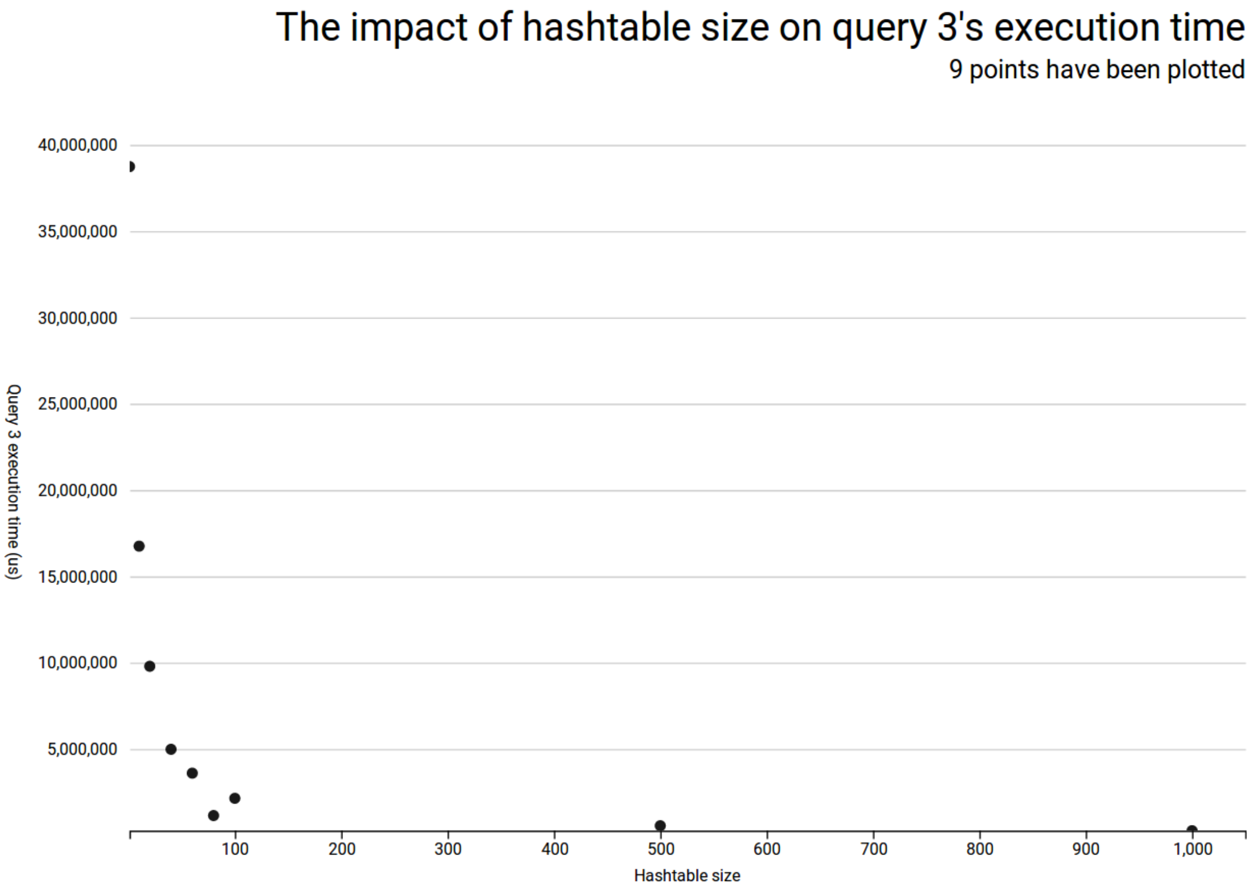
9 points have been plotted

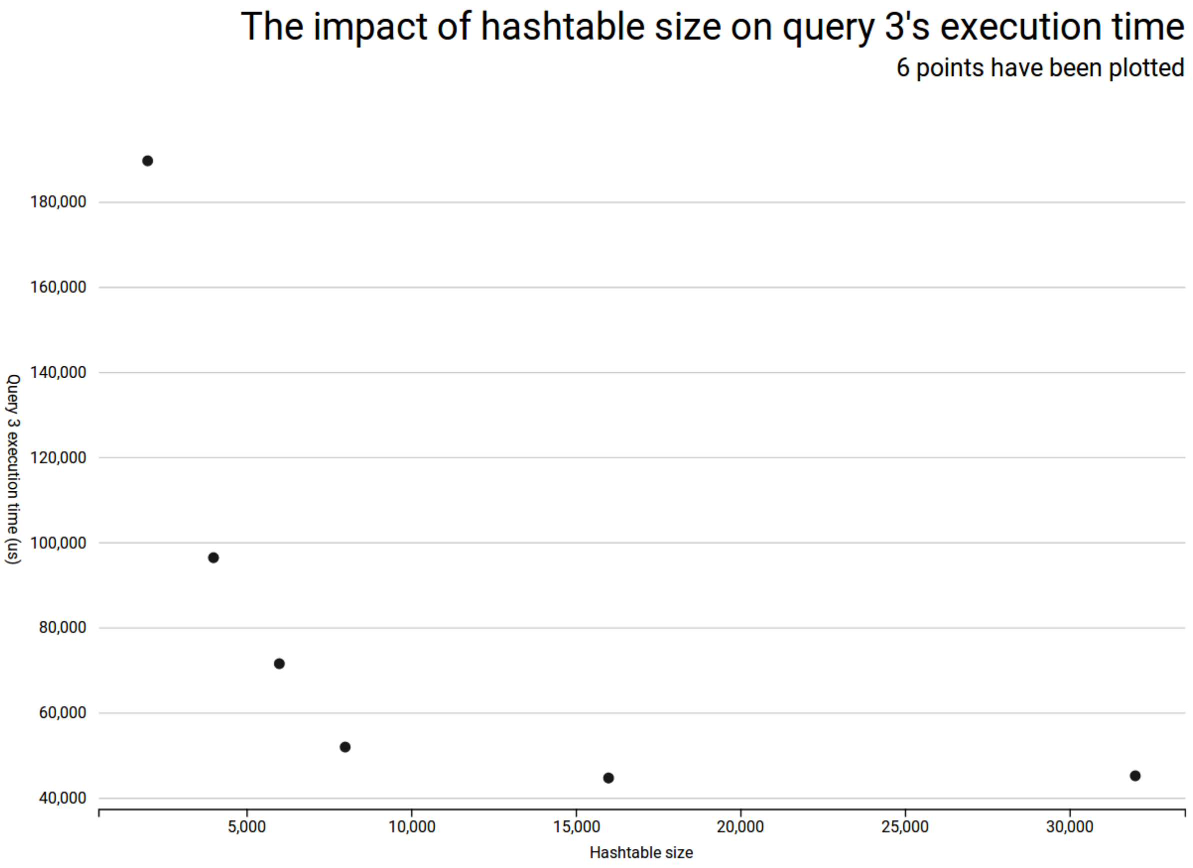


The impact of hashtable size on query 1's exeuction time

6 points have been plotted




The impact of hashtable size on query 3's execution time
9 points have been plotted
Query 3 execution time (us)
40,000,000
35,000,000
30,000,000
25,000,000
20,000,000
15,000,000
10,000,000
5,000,000
100
200
300
400
500
600
700
800
900
1,000
Hashtable size


The impact of hashtable size on query 3's execution time
6 points have been plotted
Query 3 execution time (us)
180,000
160,000
140,000
120,000
100,000
80,000
60,000
40,000
5,000
10,000
15,000
20,000
25,000
30,000
Hashtable size

Result findings:

- Our graphs for query 1 and query 3 both show an exponential decrease in query execution time as the hashtable size was increased.
- Increasing the bucket size above 128k had no drastic performance gains for our queries.
- Possible explaination of our findings: Increasing the hashtable size reduced the likelihood of hash collisions, and hence accessing items in the hashtable were more likely to be faster. Therefore, the overall query time was reduced.

- An interesting observation on index creation time was when we increased the hashtable size from 8000 to 16000. The index creation time went from roughly 180k to 66k.