INF11011 - Advanced Databases (Spring 2019)
Programming Assignment 2
S1884908

EXPERIMENT 1 - Comparative performance of the join algorithms

SETUP

For studying the relative performance of the two join algorithms, each of these (algorithms) were executed for a total of five iterations. ***This was done by keeping the number of buffer pages (NUM_OF_BUF_PAGES) fixed at 50, the block size in block join fixed at 50 and the number of tuples in the inner relation fixed at 2500 and the outer relation fixed at 10000***.

OBSERVATIONS

Iteration #	Execution Time [seconds] (Tuple Join)	Page Misses (Tuple Join)	Execution Time [seconds] (Block Join)	Page Misses (Block Join)
1	2.05	485096	0.28	9763
2	2	485096	0.28	9763
3	2.13	485096	0.27	9763
4	2.09	485096	0.34	9763
5	1.99	485096	0.27	9763

Table 1

Average Execution Time [seconds] (Tuple Join): 2.052
Average Page Misses (Tuple Join): 485096
Average Execution Time [seconds] (Block Join): 0.288
Average Page Misses (Block Join): 9763

CONCLUSION

1. From the empirical results, it can be concluded that block join is a much faster algorithm than tuple join. In this experiment, on an average block join outperformed the tuple join by a factor of 7 times (2.052/0.288). Another observation was that, block join seemed to hover around 0.28-0.27 in it's execution time for all of the five iterations, with an unusual spike in the 4th Iteration. The execution time of the tuple join seemed to reduce relatively from the third iteration onwards.
2. The number of page misses by the tuple join far exceeded the number of page misses by the block join. The number of page misses in the tuple join algorithm were greater than the block join by a factor of 50 times (485096/9763).
3. Hence it can be concluded, that block join is much faster than tuple join as it has a much lesser execution time and also produces a much lesser number of page misses when compared to tuple join.

EXPERIMENT 2 - Effect of varying buffer pool size

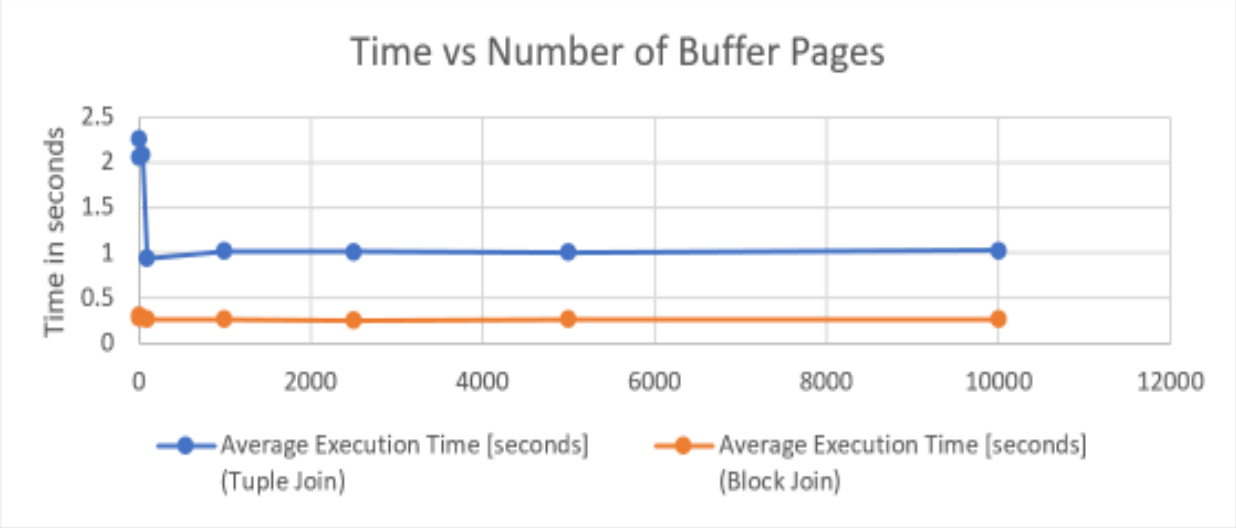
SETUP

For studying the effect of varying the buffer pool size on the performance of the two join algorithms, each of these (algorithms) were executed for a total of five iterations. The five iterations were repeated for each of the different number of buffer pages (*NUM_OF_BUF_PAGES*). ***This was done by keeping the number of tuples in the outer relation fixed at 10000, the inner relation fixed at 2500 and the block size in block join fixed at 50.*** The average execution time and average number of page misses for the repeated iterations were computed and tabulated in Table 2. ***Note:*** For brevity, the average time and and average number of page misses have been reported.

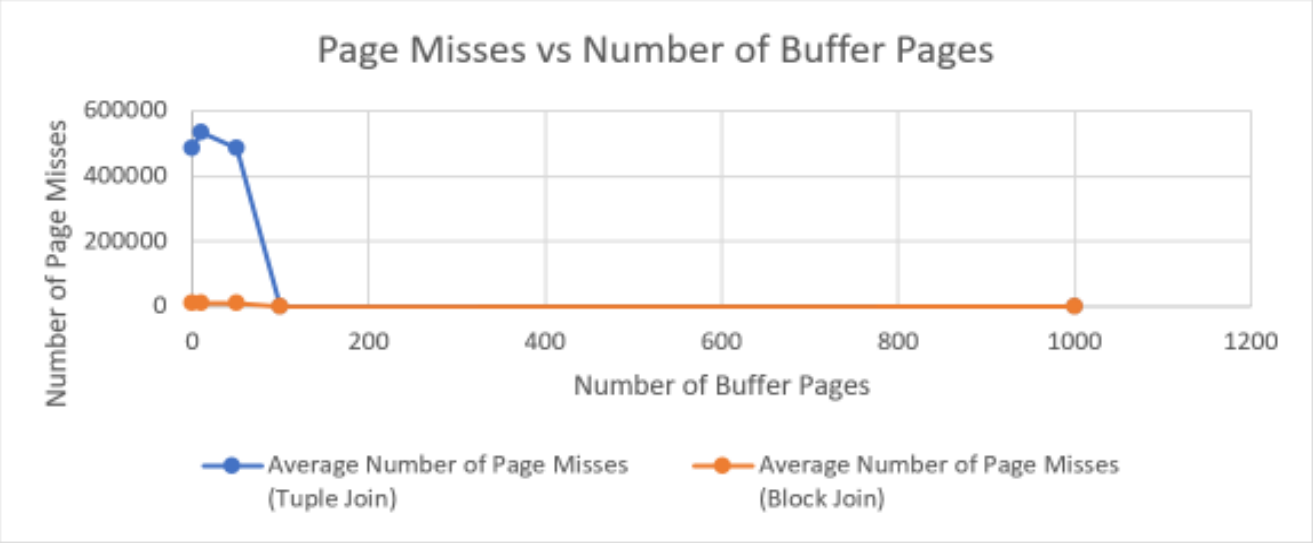
OBSERVATIONS

Number of Buffer Pages	Average Execution Time [seconds] (Tuple Join)	Average Number of Page Misses (Tuple Join)	Average Execution Time [seconds] (Block Join)	Average Number of Page Misses (Block Join)
0	2.058	485096	0.282	9763
10	2.258	534049	0.3	12368
50	2.08	485096	0.286	9763
100	0.944	747	0.266	748
1000	1.022	459	0.262	459
2500	1.012	459	0.252	459
5000	1.006	459	0.262	459
10000	1.026	459	0.266	459

Table 2



Graph 1



Graph 2

CONCLUSION

1. On increasing the number of buffer pages the execution time does not keep decreasing. From Graph 1 it can be seen that in the case of tuple join there is a sharp decrease in the execution time when the number of buffer pages are increased from 50 to 100. In the case of block join however, the decrease in execution time is not so drastic. When the number of buffer pages are increased to beyond 100 (1000 and above) the execution time increases in the case of tuple join. In the case of block join the execution time decreases and is lowest when the number of buffer pages are increased to 2500 (which is also the number of tuples in the inner relation) and the execution time increases thereafter. These observations suggest that for a fixed number of tuples in the inner and outer relation, the optimum number of buffer pages (based on time) vary based on the algorithm used for the join operation.

2. In the case of tuple join there is again a sharp decrease in the number of page misses when the number of buffer pages are increased from 50 to 100. Further reduction in the number of page misses occur when increasing the number of buffer pages to 1000 and thereafter there is no further improvement beyond 459 page misses (even on increasing the number of buffer pages). Similarly even for the case of block join when the number of buffer pages are set to 100 from 50, there is a drastic reduction in the number of page misses. Just like tuple join, block join also shows no further improvement in the number of page misses (459) when increasing the number of buffer pages to beyond 1000.

EXPERIMENT 3 - Effect of varying relation size

SETUP

For studying the effect of relation size, on the performance of the two join algorithms, each of the algorithms were executed for a total of five iterations, given a particular number of tuples in the outer and inner relation. ***This was done by keeping the number of buffer pages (NUM_OF_BUF_PAGES) fixed at 50 and the block size in block join fixed at 50***. Then given a particular number of records in the inner and outer relation, the average execution time and average number of page misses for the 5 iterations were recorded. These observation are tabulated in Table 3.

Note: For brevity, the average time and and average number of page misses have been reported.

OBSERVATIONS

Number of Tuples in the Outer Relation (R)	Number of Tuples in the Inner Relation (S)	Average Execution Time [seconds] (Tuple Join)	Average Number of Page Misses (Tuple Join)	Average Execution Time [seconds] (Block Join)	Average Number of Page Misses (Block Join)
10000	2	0.064	747	0.052	748
10000	50	0.084	747	0.058	748
10000	2500	2.052	485096	0.288	9763
10000	7000	6.192	1441288	0.792	29538
10000	10000	8.124	2052709	1.016	41763
10000	20000	16.462	4073901	1.878	82433
20000	20000	32.838	8183536	3.864	164563

TABLE 3

CONCLUSION

The execution time increases with the increase in number of tuples (increasing both the outer and inner relation). The increase in execution time for tuple join, however is very drastic as compared to the block join, as is evident from the last observation in Table 3. Although initially block join had one more page miss (748) when compared to the tuple join (747), subsequent iterations have much lesser page miss in block join as compared to tuple join.

EXPERIMENT 4 - Studying the effect of swapping the inner and outer relations

SETUP

For studying the effect of swapping the inner and outer relation, the experiment was divided into two parts.

Experiment 4.a. The first part studies the effect of swapping the inner and outer relation by keeping the number of ***buffer pages (NUM_OF_BUF_PAGES) fixed at 50*** and varying the number of tuples in the inner and outer relation. These results are tabulated in Table 4.

Experiment 4.b. The second part studies the effect of swapping the inner and outer relation by keeping the ***outer and inner relation size constant, 10000 and 2500 tuples respectively***, and varying the number of buffer pages (NUM_OF_BUF_PAGES). Results are tabulated in Table 5.

Note: The block size in block join is fixed at 50 for both the experiments 4.a. and 4.b.

For brevity, the average time and and average number of page misses have been reported and these averages have been collected by 5 repeated iterations of each sample point.

OBSERVATIONS

Number of Tuples in the Outer Relation (R)	Number of Tuples in the Inner Relation (S)	Average Execution Time [seconds] (Tuple Join) [R is Outer], [S is Outer]	Average Number of Page Misses (Tuple Join) [R is Outer], [S is Outer]	Average Execution Time [seconds] (Block Join) [R is Outer], [S is Outer]	Average Number of Page Misses (Block Join) [R is Outer], [S is Outer]
10000	2500	[2.034],[2.628]	[485096],[738279]	[0.296],[0.282]	[9763],[15260]
10000	5000	[4.066],[5.07]	[1028516], [1468250]	[0.504],[0.512]	[21099],[30193]
10000	10000	[7.988],[9.956]	[2052709], [2918213]	[0.96],[0.95]	[41763],[59209]
10000	15000	[11.626], [14.404]	[3063605], [4365251]	[1.416],[1.376]	[62114],[88282]
15000	15000	[17.678], [22.28]	[4608722], [6552587]	[2.264],[2.086]	[93021], [132116]

Table 4

Number of Buffer Pages	Average Execution Time [seconds] (Tuple Join) [R is Outer], [S is Outer]	Average Number of Page Misses (Tuple Join) [R is Outer], [S is Outer]	Average Execution Time [seconds] (Block Join) [R is Outer], [S is Outer]	Average Number of Page Misses (Block Join) [R is Outer], [S is Outer]
0	[2.028],[2.656]	[485096],[738279]	[0.276],[0.276]	[9763],[15260]
10	[2.208],[2.808]	[534049],[754412]	[0.288],[0.3]	[12368],[18031]
50	[2.064],[2.704]	[485096],[738279]	[0.296],[0.29]	[9763],[15260]
100	[0.972],[2.66]	[747],[729084]	[0.28],[0.298]	[748],[15254]
2500	[1.018],[1.218]	[459],[459]	[0.27],[0.252]	[459],[459]
10000	[1.018],[1.22]	[459],[459]	[0.262],[0.246]	[459],[459]

Table 5

[R is Outer: R is the outer relation]
[S is Outer: S is the outer relation]

CONCLUSION

1. From Table 4, in tuple join, swapping the outer and inner relation always led to an increase in execution time and page misses. For block join the execution times were comparable although the number of page misses increased on swapping the outer and inner relation.
2. From Table 5, in tuple join, execution time and number of page miss increased on swapping the inner and outer relation. In block join the execution times remained comparable, but the number of page miss increased. The page misses remained constant for [2500,5000] buf pages.