COMP9315 Assignment 2 Report

Effects of Sampling on the Accuracy of Aggregate Queries

Mathew Chong (z3330186/mcho421)
Tian Lan (z3466842/tlan184)

Hypothesis

We have a couple of hypothesis regarding the performance and accuracy of using sampling for aggregate queries. These are:

- The number of page reads accounts for the majority of the time spent within the query whereas the cost of processing a tuple within a page is minimal, i.e. Page-based sampling will result in significantly better performance than tuple-based sampling.
- For fields with values that are independent and identically distributed, the accuracy will be acceptable if the number of tuples returned from sampling is large.

Experiment Design

Sampling

We tested sampling rates from 10% to 100% in 10% increments. This was performed for all queries and for both tuple-based and page-based sampling. The tests were performed by a script that looped through sampling types, then sampling rates, then each database, and then each query, while restarting the server after each sampling configuration change.

The page in page-based sampling or the tuple in tuple-based sampling was chosen to ensure an even spread throughout the table. For example, if a sampling rate of 50% was used, we select the 1st, 3rd, 5th, etc. tuples/pages from the table.

Test data

We used gendata.c to generate the test data for our databases. For each database, we generate 10000 tuples in R and for each tuple in R it generates X tuples in S linked by a foreign key where X is a uniformly random integer in the range 0..5, which on average would result in 25000 tuples in S. Each S.a is a normally distributed random number between 1 and 100000, which would result in an average of 50000.

We created 10 databases of 10000 tuples each which we averaged our results over. We modified gendata.c to use different seeds to achieve this. Below are some of the properties of the databases generated:

Database	**#Pages in R**	**#Pages in S**	**#Tuples in R**	**#Tuples in S**
0	112	293	10000	24982
1	113	288	10000	24688
2	112	290	10000	24873
3	112	292	10000	25017

4	113	290	10000	24874
5	112	294	10000	25014
6	113	296	10000	25340
7	112	291	10000	24828
8	112	288	10000	24604
9	113	292	10000	24935
Average	112.4	291.4	10000	24915.5

Test queries

We tested 2 queries:

1. SELECT avg(S.a), count(*) FROM S;
2. SELECT count(*) from R join S on R.x = S.c;

The first query is to determine the accuracy of the avg() aggregate which would be determined by the number of tuples returned at the sampling rate. For the second query, we are interested in estimating the true number of tuples resulting from a join. We would multiply the result of the second query by $1/(\text{sampling\_rate}/100)^2$.

Test machine

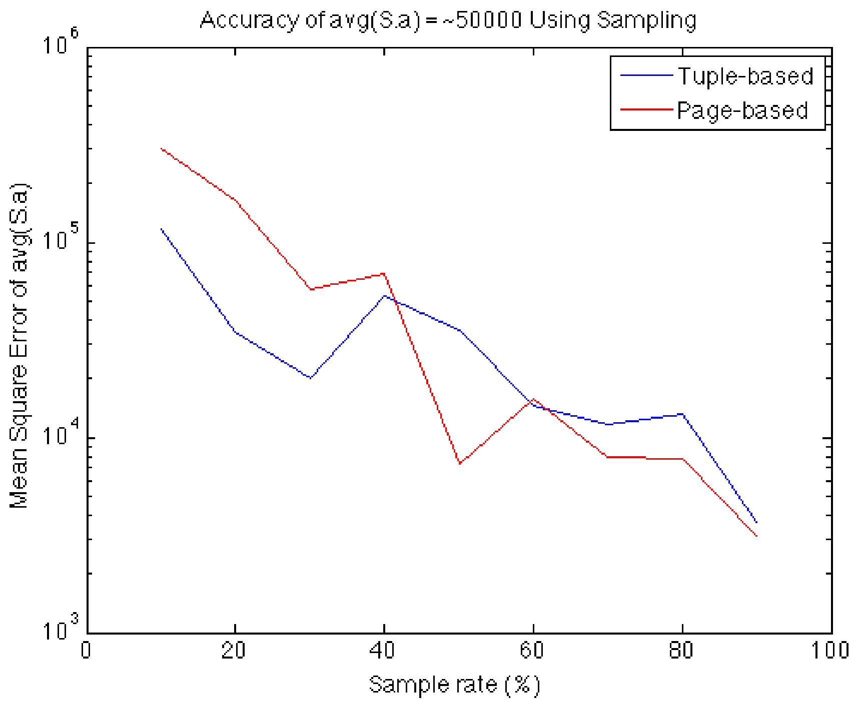
The experiment was run on a Macbook Pro (Early 2011) running OS X 10.9.2, with a 5400-rpm Serial ATA hard drive and 2.3GHz dual-core Intel Core i5 processor with 3MB shared L3 cache. We were running a modified PostgreSQL 9.3.4 with sampling settings for the experiment.

Results

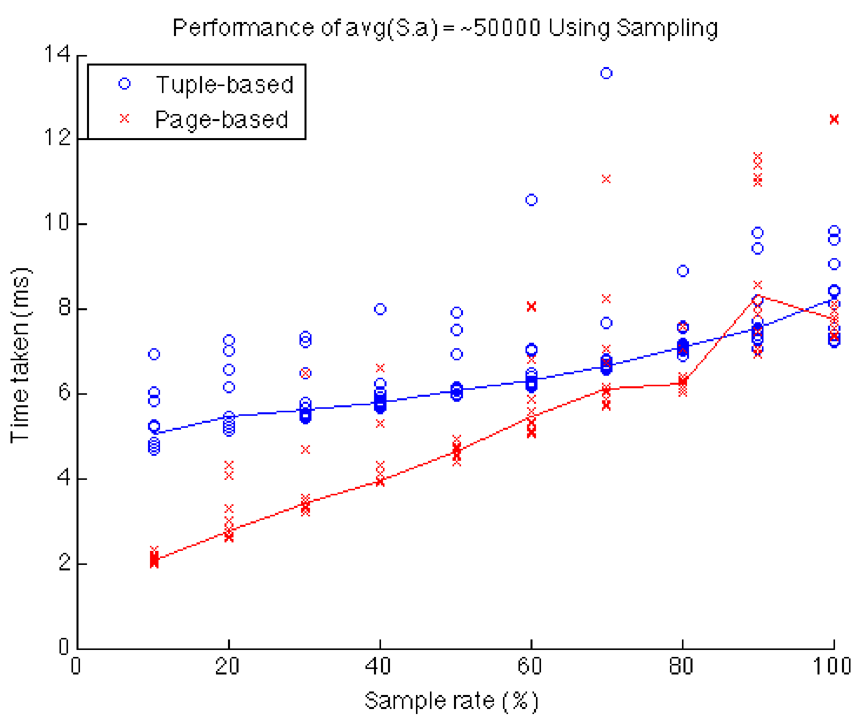
Results for single table aggregate - avg()

The following are the results of the query: SELECT avg(S.a), count(*) FROM S;
The table and graphs below show the average results of the 10 databases generated.

Type	**Sample Rate (%)**	**Median Time (ms)**	**Avg. #Page Requests**	**Avg. Tuple Count**	**Avg. avg(S.a)**
tuple-based	10	5.038	291.4	2492	50004.39435
	20	5.454	291.4	4983.4	50108.60808
	30	5.6125	291.4	7475.2	50017.72546
	40	5.8035	291.4	9966.6	50033.92617
	50	6.083	291.4	12457.9	49893.81605
	60	6.3415	291.4	14949.7	50059.46351
	70	6.6605	291.4	17441.2	49982.98812
	80	7.1225	291.4	19932.9	50034.40627
	90	7.536	291.4	22424.4	49992.18881
	100	8.2635	291.4	24915.5	49997.76642
page-based	10	2.0715	29.6	2529.7	49953.81403
	20	2.771	58.7	5006.2	49942.22959
	30	3.4	87.8	7496.8	50037.14208
	40	3.9625	117	9991.9	49938.35483
	50	4.6335	145.8	12493.3	50028.13119
	60	5.4525	175.2	14975	49991.53518
	70	6.102	204.4	17490.9	49999.37213
	80	6.236	233.5	19960.7	49977.05196
	90	8.3215	262.6	22458	50017.30879
	100	7.746	291.4	24915.5	49997.76642



Above: Shows the Mean Square Error of the avg() reported by the 10 databases on the first query. The expected result for avg() is roughly 50000.

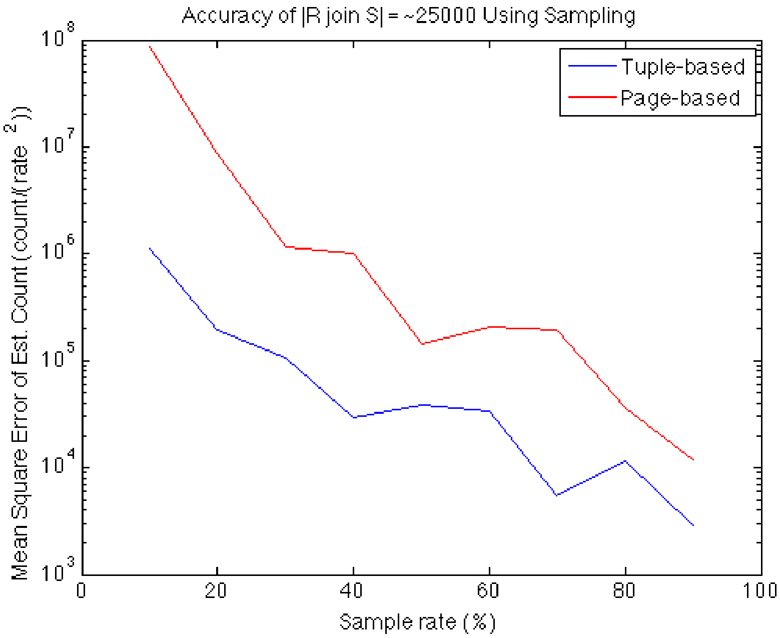


Above: Show the time taken for the 10 databases on the first query. The dots show the individual times for each database while the line connects the median times.

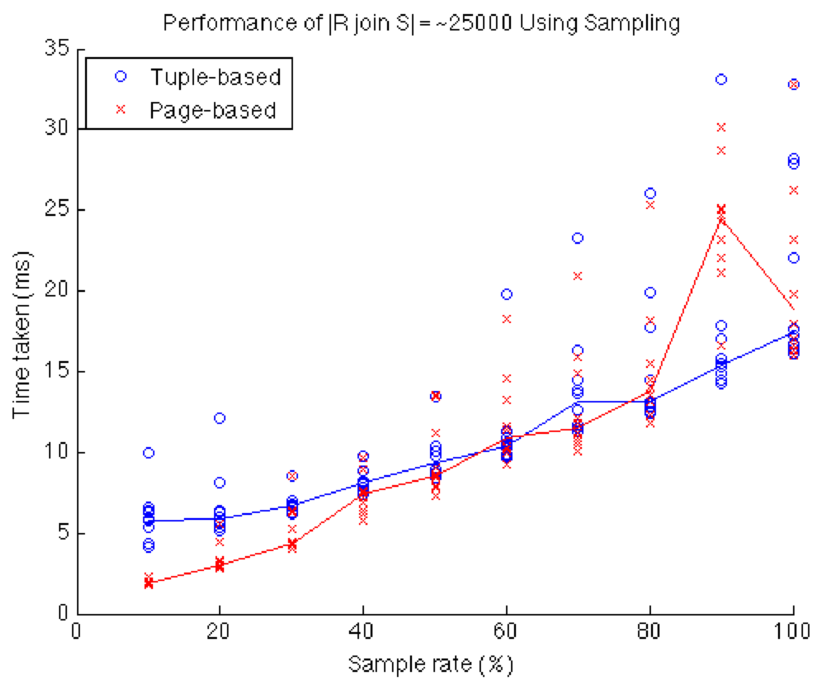
Results for count aggregate on R join S

The following are the results of the query: SELECT count(*) from R join S on R.x = S.c;
The table and graphs below show the average results of the 10 databases generated.

Type	Sample Rate (%)	Median Time (ms)	Avg. #Page Requests S	Avg. #Page Requests R	Avg. count(*)	Est. Count
tuple-based	10	5.8055	112.4	291.4	250.8	25080
	20	5.856	112.4	291.4	989.7	24742.5
	30	6.744	112.4	291.4	2233.2	24813.33333
	40	8.096	112.4	291.4	3974.9	24843.125
	50	9.3525	112.4	291.4	6225.4	24901.6
	60	10.405	112.4	291.4	8978.8	24941.11111
	70	13.161	112.4	291.4	12206.6	24911.42857
	80	13.123	112.4	291.4	15973.8	24959.0625
	90	15.425	112.4	291.4	20198.3	24936.17284
	100	17.452	112.4	291.4	24915.5	24915.5
page-based	10	1.88	12	29.6	316.4	31640
	20	3.018	23	58.7	1092.4	27310
	30	4.396	34	87.8	2210.4	24560
	40	7.3905	45.4	117	4122.2	25763.75
	50	8.575	56.4	145.8	6273.4	25093.6
	60	10.9055	68	175.2	9097	25269.44444
	70	11.531	79.4	204.4	12394.8	25295.5102
	80	13.7795	90.4	233.5	16007	25010.9375
	90	24.4985	101.4	262.6	20234.2	24980.49383
	100	18.8775	112.4	291.4	24915.5	24915.5



Above: Shows the Mean Square Error of the estimated total count of the join reported by the 10 databases on the second query. The expected result for avg() is roughly 25000.



Above: Show the time taken for the 10 databases on the second query. The dots show the individual times for each database while the line connects the median times.

Discussion

Discussion for single table aggregate - avg()

This query took about 8 ms for an aggregation of ~25000 tuples at 100% sampling rate. Both the tuple-based and page-based sampling returned roughly the same number of tuples for the query.

From the graph it appears that the time taken for the tuple-based and page-based queries decreased linearly for decreasing sampling rates. This should be expected because reading and processing the tuples/pages should each take linear time.

Although not as much as the page-based sampling (0.06ms/percent) the tuple-based sampling (0.03ms/percent) does noticeably reduce the run time of the query. The page-based sampling would be faster due to less page reads but the tuple-based sampling would still have less tuples to aggregate compared to 100% sampling resulting in some performance gains. We also note that tuple-based sampling is not as slow as we expect, so we suspect that the operating system buffer cache might be caching pages of the databases.

In terms of accuracy, the number of errors appears to increase logarithmically with decreasing sampling rate. This makes sense as there would be more variance with less tuples.

Although there is a large amount of variance in the Mean Squared Error, the error for tuple-based and page-based appears roughly the same. This is because the distribution is independent so it doesn’t matter how we select the tuples.

Based on the results of sampling on ~25000 tuples, the aggregation error is not too bad. At 10% the Root Mean Squared Error is around 500 which is only 1% of 50000, the true average value. This is probably because 10% of 25000 tuples is 2500, still a statistically significant sample size.

Discussion for count aggregate on R join S

This query took about 18 ms for a count aggregate on a join between a 10000 tuple table and a ~25000 tuple table.

The graph for this query also shows that the time taken for tuple-based and page-based queries decreased linearly for decreasing sampling rates.

Compared with the first query, the gradient (time/percent) for this query was 3-4x higher for both the tuple-based (0.13ms/percent) and page-based (0.18ms/percent) sampling. This might be because of the additional 1000 tuples to consider and the query plan having 2 sequential scan nodes.

The accuracy of the tuple-based sampling was consistently better than the page-based sampling. This is probably due to the way we calculate the estimated count, which is count/(sampling rate^2). Since the page-based sampling chooses pages in proportion to the sampling rate, this might be a too coarse number to base the join count estimation on. The page-based count was consistently overestimated, with a Root Mean Squared Error of 10000 (40% of the expected 25000) at 10% sampling rate compared to 1000 (4%) in tuple-based sampling.

Conclusion & Recommendations

Based on our implementation and testing, it appears that page-based sampling does outperform tuple-based sampling in terms of performance, but not as much as we anticipated. In terms accuracy, aggregation on 25000 tuples at a 10% sampling rate, the accuracy was acceptable for both forms of sampling, but the accuracy for the join count estimation was poor for the page-based sampling.

It would be recommended to do further tests to see if the operating system buffer cache is having a significant effect on the performance of the sampling and possibly using larger datasets.