

Examining ASE's New Sampling Feature for Update Statistics

By Eric Miner

ASE shortens the maintenance window for DBAs



Eric Miner has been with Sybase since 1992, working with the optimizer team in engineering as well as focusing on optimizer issues for technical and product support engineering. He can be reached at eric.miner@sybase.com.

Since time immemorial, or at least since the beginnings of ASE, the time update statistics takes to run has been a problem for many users. This has been particularly true as dataset sizes have grown over time. In fact, there are some that have grown so large that running update statistics simply doesn't fit into the available maintenance window. Decreasing the time that it takes to run update statistics has been a long-standing feature request for ASE.

In response, the new ASE 12.5.03 feature *Sampling for Update Statistics* is an optional method of gathering column-level statistics. Designed to shorten the time it takes to run update statistics, this new feature should decrease the size of your maintenance window and lower the cost of owning and using ASE.

What is Sampling for Update Statistics?

The dictionary says:

sample - noun: 1a. A portion, piece, or segment that is representative of a whole. b. An entity that is representative of a class; a specimen. 2. Statistics A set of elements drawn from and analyzed to estimate the characteristics of a population. Also called sampling.

The use of various sampling techniques to gather statistics for a large number or

purposes is standard practice everywhere statistics are used.

In a data server such as ASE, there are two approaches that can be taken to sampling values in a column—random row and random page, a.k.a. “random block” techniques. Sampling random rows is the most random form of sampling values. However, the I/O costs would be almost as high, or higher, than update statistics has been. Sampling random pages or “blocks” has been found to be more efficient because of the I/O savings. This I/O savings translates into time savings.

Keep in mind that the very nature of sampling means that not all rows will be read when statistics are gathered. We'll go into this in detail soon.

Some Terms and Definitions

Sampling rate: The percentage of pages to sample as specified with new update statistics syntax

'Full scan' Update Statistics: When update statistics reads all rows of the column. This is how update statistics has always functioned.

Histogram: A set of values that describe the distribution of data in a column.

Density (Range cell and Total): Statistical values used to describe the average number of duplicates in the column. The optimizer uses both differently.

Step: The row from which update statistics reads a value to create a boundary value in the histogram.

Boundary Value: A value read from the column used to establish the upper boundary for a histogram cell. The upper boundary of a range of values represented by a cell

Cell: Represents a range of values in the column that fall between its boundaries, or represents one highly duplicated value.

Cell weight: The percentage of the rows in the column that are represented by the cell.

Range cell: A histogram cell that represents more than one value in the column

Frequency count cell: A histogram cell that represents only one highly duplicate value in the column

SARG: Search argument, i.e., where column = value

How to Run Update Statistics with Sampling

Sampling is an option of update statistics. You can specify any percentage between 100% and 1%:

```
update statistics table_name (column_name) with sampling = X percent
```

This syntax will update or create statistics on the specified column using the specified sampling rate.

```
update index statistics table_name [index_name] with sampling
= X percent
```

This syntax will update the statistics of the leading column of the indexes or the specified index using a full scan update statistics, and it will update/build statistics on the inner columns using the specified sampling rate.

```
update all statistics table_name with sampling = X percent
```

This syntax will update or create statistics on all columns of the table. It will use full scans for leading columns of indexes, and will use the specified sampling rate for all other columns.

Some Unofficial Time Tests

The simple test below is by no means scientific, but it does give a good idea of the time savings that sampling can provide. Keep in mind that your mileage may vary. The table used had one million rows and four columns. The columns were either numeric (column "id") or int (columns "a," "b," and "c") datatypes. In-house tests show time savings in the 50-90%, range depending on the sampling rate.

```
update statistics test1(id)
```

cpu time: 16500 ms. elapsed time: 541080 ms. = 9.01 minutes.

```
update statistics test1(id) with sampling = 10 percent
```

cpu time: 1100 ms. elapsed time: 33063 ms. = 33 seconds.

What are the Trade-Offs of Sampling?

As with most things in life and performance and tuning, sampling has its trade-offs. Since any form of sampling will only read a subset of the values in a column, the resulting statistics will vary from those that result from reading all rows in the column. The lower the percentage of sampling requested, the higher these variations are likely to be. Such variations in the statistics may have an affect on the optimizer.

However, in most cases the effects, if any, should be minimal. It will take some trial-and-error testing to determine where the balance lies between the time it takes to run update statistics and the accuracy of the statistics for a given dataset.

Effects of Sampling on the Statistics

Sampling can have distinct effects on the resulting column-level statistics. In this section I'll discuss these and give some examples.

The effect of sampling you are most likely to see are values that are in a full-scan histogram falling out of the boundaries of a sampled histogram. This will affect the optimizer if such values are used as search values. As the percentage of sampling drops, the greater chance that more values will fall outside of either or both ends of the histogram. Below is an example out-of-bounds values:

Step	Weight		Value
1	0.00000000	<=	462
2	0.05263108	<=	55965
Some cells edited			
19	0.05263108	<=	1000132
20	0.05264055	<=	1055640

In this histogram segment, the minimum boundary value is 463 (remember that the boundary value of the first cell is one significant bit less than the minimum value in the column). The maximum boundary value is 1055560.

If you were to use a value less than 463 or greater than 1055640 in a search argument, the optimizer would have no

statistics to use in estimating the cost of a query plan. It would have to use “out of bounds costing,” which may or may not result in an efficient plan. If your search argument values stay within the histogram, the statistics can be used by the optimizer.

As you use various sampling rates, check to see if commonly used SARG values are falling out of bounds. This is particularly important in very dynamic columns that are often queried during the day. How many values fall out of bounds is an important consideration when determining the best sampling rate for the given column.

Below is an example of the effects of sampling on out-of-bounds costing by the optimizer. A full scan of column “id”:

Step	Weight	Value
1	0.00000000	<= 1000000
2	0.05263100	<= 1052631
3	0.05263100	<= 1105262

The traceon 302 output:

```
Selecting best index for the SEARCH CLAUSE:
tw1.id = 1000010

Estimated selectivity for id,
selectivity = 0.000001, upper limit = 0.052631.
```

The search value 1000010 falls into the full scan histogram and the statistics can be used to estimate the selectivity of the column. A 10% sample of column “id”:

Step	Weight	Value
1	0.00000000	<= 1000682
2	0.05262368	<= 1052371
3	0.05262368	<= 1111067

```
Selecting best index for the SEARCH CLAUSE:
tw1.id = 1000010
```

```
Estimated selectivity for id,
selectivity = 1.000000, upper limit = 1.000000.
```

```
Equi-SARG search value '1000010' is less than the smallest value in
sysstatistics for this column.
```

At a sampling rate of 10%, the search value falls outside the histogram—in this case, the lower end of the histogram. When this occurs, the optimizer can use only one of two values for the selectivity: 1 or 0. Which it uses depends on which end of the histogram the search value falls outside of and what the relational operator is.

Variations of Values in the Histogram

Another normal effect of sampling on the column’s statistics is variations in the histogram values, both the boundary values and the cell weights. Again, the smaller the sampling rate, the greater the potential for variations. In all the following examples, the columns contain 1 million rows.

Sampling and Range Cells – Example of Sampling a Unique Column

In this first example, column “id” is a completely unique numeric datatype column with 20 (default number) cells/steps. The minimum value in the column is 1 and the maximum value is 1055640.

In the full-scan histogram below, we see that the maximum boundary value is the max value for the column. The minimum boundary value is one bit less than the minimum; in this case, it is 0. This is all completely normal.

In the case of a unique column, or a column with a very even distribution of value, the weights of each cell are all the same, because the distribution of values is perfectly uniform. It is possible for the weight of the last cell to vary. This is due to values being “left over” after all previous cells are created.

Histogram for column:		"id"
Step	Weight	Value
1	0.00000000	<= 0
2	0.05263158	<= 55560
3	0.05263158	<= 111120
4	0.05263158	<= 166680
5	0.05263158	<= 222240
(Some cells edited)		
14	0.05263158	<= 722280
15	0.05263158	<= 777840
16	0.05263158	<= 833400
17	0.05263158	<= 888960
18	0.05263158	<= 944520
19	0.05263158	<= 1000080
20	0.05263158	<= 1055640

Two histograms from 10% samplings of column “id”:

The first thing we see in the histograms below is that the boundary values vary from those of the full scan histogram. This is normal and expected, especially at a sampling rate of 10%. As the sampling percentage decreases, the deviations from the full scan histogram do as well. At this sampling rate, the weights of the cells are still generally uniform, but do vary from the full scan. In the second histogram, the weight of the last cell is slightly different. This is not uncommon, but it

does differ from the full scan.

Since the boundary values differ it is likely that the upper and lower boundaries of the histogram may not correspond to the actual minimum (minus one bit) and maximum values in the column. In these examples, the minimum boundary value of the first histogram does indeed match that of the full scan. Random sampling read a page containing the value 1.

But, in the second histogram, the value is 616, meaning that the lowest value read during sampling was 617. In both cases, the upper boundary values differ from the full scan and from each other. Again, this is a normal result of sampling. See the section “Out of Bounds Values” for how this can affect the optimizer.

The first 10% sample of column “id”:

Step	Weight	Value
1	0.00000000	<= 0
2	0.05262959	<= 53008
3	0.05262959	<= 107248
4	0.05262959	<= 162720
5	0.05262959	<= 219424
(Some cells edited)		
14	0.05262959	<= 718672
15	0.05262959	<= 772912
16	0.05262959	<= 828384
17	0.05262959	<= 883856
18	0.05262959	<= 940560
19	0.05262959	<= 997264
20	0.05266745	<= 1052744

The second 10% sample of column “id”:

Step	Weight	Value
1	0.00000000	<= 616
2	0.05261801	<= 56131
3	0.05261801	<= 110855
4	0.05261801	<= 165579
5	0.05261801	<= 223318
(Some cells edited)		
14	0.05261801	<= 721864
15	0.05261801	<= 776588
16	0.05261801	<= 831312
17	0.05261801	<= 889051
18	0.05261801	<= 943775
19	0.05261801	<= 998499
20	0.05287587	<= 1053240

Example of Sampling a Column with a Less Even Distribution

Let’s take a look at a column whose values are less evenly distributed. Here the effects of sampling can be more pronounced. In the real world, it’s far more common for a column to be non-unique with an uneven distribution of values. The example below is primarily made up of Range cells. One cell that is a Frequency count cell (highlighted). Two values occupy nearly half of the rows represented by the cell. You can see these by the higher weights of their cells. The minimum value for the columns is 154143024, the maximum value is 960051513.

Here we see more obvious changes to the cell weights and boundary values when sampling is used than we did in the column with a more even distribution. The boundary values of the Range cells do vary in the 10% sampled histogram. Those values with a high number of duplicates appear in the boundary values of both the full scan and 10% sampled histograms, but their weights vary as expected.

Full scan of column “c”:

Step	Weight	Value
1	0.00000000	<= 154143023
2	0.05297200	<= 154143025
3	0.05810500	<= 154143289
4	0.05266400	<= 154678324
5	0.05266200	<= 171390774
6	0.08452200	<= 805908490
7	0.05315900	<= 805975865
8	0.06440500	<= 810000000
9	0.05263300	<= 825701426
10	0.05274600	<= 842347017
11	0.05264700	<= 858992694
12	0.05268800	<= 875628851
13	0.06491800	<= 880000000
14	0.05263400	<= 892811577
(Some cells edited)		
19	0.05256000	< 960051513
20	0.00002000	= 960051513

A 10% sampling of column “c”:

Step	Weight	Value
1	0.00000000	<= 154143023
2	0.05915578	<= 154143281
3	0.05262737	<= 154208823
4	0.05269137	<= 154679090
5	0.05265937	<= 171454775

```

6 0.08160458 <= 805908490
7 0.05323541 <= 805975864
8 0.06811636 <= 810000000
9 0.05267537 <= 825700914
10 0.05265937 <= 842347572
11 0.05264337 <= 858992649
12 0.05262737 <= 875639096
13 0.06274001 <= 880000000
14 0.05262737 <= 892811059
    =(some cells edited)=
19 0.05265937 <= 960051248
20 0.00068804 <= 960051510

```

Effects of Sampling on the Density Values

The two density values, *Range cell density* and *Total density*, represent the average percentage of duplicates in the column. Total density is a measure of the duplicates in the entire column. Range cell density is a measure of duplicates of all values that are in Range cells and are not represented by Frequency count cells; in other words, values that are not highly duplicated. Both are used in different ways by the optimizer. The Range cell density is used by the optimizer in its costing of search arguments. The Total density is used to cost a join when there are not SARGs on the column or on the column to which it is joining.

The density values are gathered when a full-scan update statistics is run. However, they are not currently gathered when sampling is used. If density values from a previous run of update statistics exist, sampling will not overwrite them. If there are no existing density values, a default value of 0.10 will be used. This value may not be completely accurate for the given column.

There are a few ways to assure the optimizer is not using the default value while it is costing the query. First, periodically run a full-scan update statistics without using sampling. Recreating and index will also do this for the leading column. Second, use `sp_modifystats` to insert a more accurate value or to factor the default value by adding zeros to the right of the decimal point. Reading in an edited `optdiag` file can also change the value.

```

Range cell density: 0.0009161693724066
Total density:      0.0070596353081496

```

The density values above are the “real” values that were obtained via a full scan of the column. The two values in this

example vary because there are some highly duplicated values in the column, but not too many. If sampling were to be used to update the statistics of this column, these values would not change.

```

Range cell density: 0.1000000000000000
Total density:      0.1000000000000000

```

The density values above were written when sampling was used to gather statistics on the column without previously existing statistics. These default values are not likely to be very accurate. When the default density values are in place, they may adversely affect the optimizer’s costing of SARGs and joins. It is highly recommended that you run a full-scan update statistics on your column from time to time to ensure reasonable density values. Sybase plans to obtain density values from sampling in the future.

Sampling and Indexes

Sampling cannot be used with `create index` because all pages have to be read in order to build the index. Also, the major attributes (leading columns) of index are not sampled unless you specifically update statistics on the individual column:

```
update statistics table_name (column_name)
```

If you run `update statistics table_name [index_name]`, the statistics for all the columns of the index(es) will be updated using a full scan of each column. If you add the sampling syntax, the leading column of the index will still be updated via a full scan. But, the inner columns will be updated using the sampling rate you specified.

```

run update statistics table_name [index_name] with sampling
= X percent

```

This will save a great deal of time. The more fields contained in an index, the more time is saved. Also, the amount of tempdb space need for the worktables to sort the values of the inner columns will be much smaller as the sampling rate decreases.

If you run `update all statistics table_name`, all columns of the table will be updated using a full scan. If you add sampling syntax, all columns of the table will be updated using the sampling rate you specify—except for the leading columns of indexes, which will be updated via a full scan.

Effects of Sampling on Query Plans

Of course, the most important factor in determining whether sampling will work well for you and the sampling rate to use is the effect, if any, on the query plans chosen by the optimizer. As we've seen in the earlier examples, sampling will create column-level statistics that vary from those of a full scan. But, will these variations have an adverse effect, or no effect, on the query plans the optimizer creates?

The cell weights and the Range cell density are both used to estimate the selectivity of a column that is referenced by a SARG. The column selectivity has a direct effect on estimating the cost of using an index. Thus, changes to the weights caused by variations will have an effect on the estimated costs.

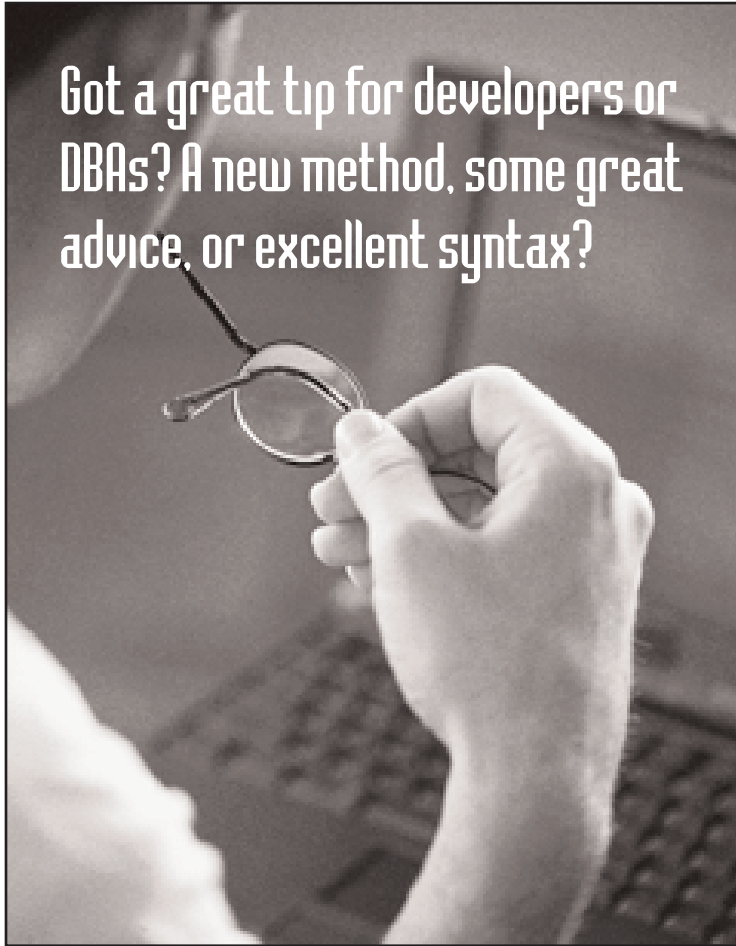
There is no way to tell by simply looking at the statistics if a variation will result in an inefficient plan. Instead, you'll need to run tests of common queries. I suggest running the queries against full scan statistics, and gathering traceon 302 and 310 and showplan outputs. Then, update the statistics using sampling and gather the outputs again. You can do

this with or without set noexec on (if it's a procedure use set fmtonly on).

If the outputs are the same, sampling has had no affect and you can safely use the sampling rate you used in the tests. If there is a change in the query plans, go back and try a different sampling rate; in general, try increasing it. There is a chance that sampling may not work for you because the changes in the statistics don't result in efficient query plans. But I suspect this will be a very rare situation.

Conclusion

Sampling for Update Statistics has been designed to help users maintain ASE databases in a much shorter time, in a smaller maintenance window. Faster maintenance will help lower the cost of using ASE. By its nature, sampling will gather statistics that vary from those obtained by reading all pages of the table. These variations can have an effect on the optimizer. While the effects are likely to be minimal, users should run tests to determine which sampling rate offers the best balance of time savings and good performance. ■



Got a great tip for developers or DBAs? A new method, some great advice, or excellent syntax?

Write an article for the ISUG Technical Journal!

Becoming an author for the Journal helps build your resume, establish your reputation, and share your great concepts with fellow ISUG members.

For more information, check out the *ISUG Technical Journal* Online to see sample articles and guidelines for authors. Then contact Journal Director Anthony Mandic at anthony@isug.com or Managing Editor Mary Freeman at freemancomm@yahoo.com to discuss your article concept.

www.isug.com