Paper 263-2009

Very Large Dataset Performance as Measured Across Multiple SAS 9.1.3 Data Storage Options.

Don Kros, Humana, Inc. Louisville, KY

ABSTRACT

SAS® 9.1.3 provides two basic library setup options, Base and Scalable Performance Data Engine. Our purpose is to compare and contrast performance statistics via FULLSTIMER across multiple scenarios against a dataset of 500 million records. These scenarios include enabling a mixture of Threading, CPU Counts, Index Creation, and Engine Type across BASE and SPDE Libraries while performing a PROC SUMMARY. Our platform is an IBM P-590 with 16 dual-core 3.4 GHz processors, 70 gigabytes of RAM, and 17 terabytes of disk space on a RAID 5 SAN with IBM DS8000 disks.

INTRODUCTION

When faced with the challenge of providing a succinct data layer that has the capability in both speed and accuracy to meet the needs of a highly analytical user community, the method of storing data resulting in the fastest and most accurate retrieval of the data is one of the most important factors in the success of the storage solution. The difficulty of meeting this goal is compounded when the data in question happens to be several SAS datasets with between 550 million and 700 million records and between 75 and 186 columns. At this point the speed and accuracy of the data solution becomes not important, but becomes paramount to the success of each and every analytical study dependent on the solution.

This paper will focus on simple concepts and measurements used in comparing and contrasting performance statistics via FULLSTIMER across multiple data storage scenarios of SAS datasets consisting of 500 million records to determine an approach that allows for the most efficient and most accurate retrieval of data.

This investigation is designed in such a way that BASE and SPDE tables can be tested in an independent environment measuring Extraction, Sorting, a Statistical test, and a Summarization procedure while compensating for possible external confounds. Our methodology achieved this by performing the procedures discussed above on a separate mount point within our dedicated SAN. Although not realistic in the business world, this controlled environment provided a review of the test scenarios in such a way that they would be unaffected by daily activity within the SAS system.

The findings from these tests demonstrate the advantages that can result from careful research, planning and making full use of all pertinent available data storage options within the SAS system. We also demonstrate the importance of initial design and development of architecting a storage solution intended to house very large data sets. We have learned that it is important to explore all options and implement storage combinations depending on the problem needing to be solved.

PROBLEM DEFINITION

To be competitive in the ever changing, high-demand world of data analytics for Healthcare, one

of the more overlooked facets of an analytical process is the ability to quickly and accurately extract necessary atomic level data for use in evaluation and risk mitigation research. When the ability to quickly, easily and accurately extract large scale data for these purposes works towards the analyst's advantage, the mechanism by which the data are stored goes largely unnoticed. However, data loses its value quickly if the extraction process becomes time consuming or is overly burdened with multiple combinations and joins resulting in a disproportionate amount of time spent gathering data rather than interpretation. This ultimately results in lost opportunity for business related analytics. This simple fact demonstrates why the correct development of data architecture consistent with the intended purpose of data usage should not be overlooked.

SOLUTION

The purpose of this investigation is to review the use of BASE and SPDE dataset, as well as indexing applications to create an environment designed for optimum performance of the datasets for the analysts use. As data grows in volume and complexity, so do opportunities to explore alternative methods related to storage and retrieval methodologies. These solutions, if properly implemented, can make the process of extracting and manipulating the necessary data an afterthought to the analysts. Although impossible to anticipate all analytical data needs early in the design process, the creation of a flexible data model allowing for growth, for transparent changes to be made within the storage solution does not need to be a complicated process. It needs to be noted however that not all available solutions are interchangeable. Careful planning and research of which data space is necessary for either 'read only' or space designed for analyst use dictates the methodology used to create each storage solution. Outlining these needs requires careful planning on the part of the Data Architect as well as ongoing involvement from the user community. It is also noteworthy that results of tests will be impacted and can vary significantly depending on the type of hardware used.

TESTS FOR VALIDATION

A series of procedures were designed to test the hypothesis that there are significant differences in the time involved for extraction, manipulation and summarization of very large data sets depending on the method of storage and indexing applied. For each scenario the tests performed involved an Extraction using a Data Step to subset data from the main table:

```
DATA DATACOPY.sort (compress=yes partsize=512);
SET DATACOPY.DATA(KEEP=AGE NTWK_ALLOWED_CHRG_AMT MEDCLM_KEY PRIMARY_DIAG_CD
FIN_PROD_CD_MNR MNR_CAT_CD MNR_CD PROCESS_DATE);
WHERE PROCESS_DATE BETWEEN '01JAN2006'd AND '31DEC2007'd;
RUN;
```

Next, a Sort procedure was performed on the subset. The Sorted Data was then run through an ANOVA statistical test:

```
PROC SORT
DATA=DATACOPY.SORT
OUT=DATACOPY.ANOVA (compress=yes partsize=512);
BY FIN_PROD_CD_MNR MNR_CAT_CD MNR_CD;
RUN;


PROC ANOVA DATA=DATACOPY.ANOVA;
CLASS PRIMARY_DIAG_CD;
MODEL AGE = PRIMARY_DIAG_CD ;
RUN;
QUIT;
```

Finally the original data was summarized using a PROC Summary statement.

```
PROC SUMMARY NWAY MISSING DATA = DATACOPY.DATA;
CLASS MNR_CAT_CD MNR_CD MNR_CD SERV_TYPE_CD;
VAR NTWK_ALLOWED_CHRG_AMT MNR_UNIT_CNT;
WHERE PROCESS_DATE BETWEEN '01JAN2006'd AND '31DEC2007'd
AND SERV_TYPE_CD IN ('99201','99202','99203','99204','99205','99211','99213',
'99213','99214','99215');
OUTPUT OUT = WORK.SUMMARY (drop = _freq_ _type_) sum=;
RUN;
```

The dataset used for this series of tests consisted of 519,306,041 rows and 34 columns. This data was created such that the BASE table was identical to the data in the SPDE table. Control totals and row count validation were performed to ensure the tables were identical in content. These tables were then validated against the original source to ensure consistency which resulted in the conclusion that the tables were identical representations of the source data.

Specific methods of Indexing were performed on the datasets prior to each iteration of running the code. The indexing procedures involved code execution with no indexing, indexing on a Primary Key, and using multiple indexes within the datasets. Prior to each test, the previous indexing was removed so each new test started with an un-indexed dataset. The indexing mythologies were:

Base Indexing Methodology:

```
proc datasets library=libname;
modify table_name;
index create element(‘s);
run;
```

SPDE Indexing Methodology:

```
proc datasets library= libname;
modify table_name (ASYNCINDEX=yes);
index create element(‘s);
RUN;
```

Analyses of the key measurements of interest provided by the FULLSTIMER option on the resulting log output were:

1. Real Time - the amount of time spent to process the SAS job. (Real time is also referred to as elapsed time.)
2. System Time - the CPU time spent to perform system overhead tasks on behalf of the SAS process.
3. User Time - the CPU time spent to execute your SAS code.

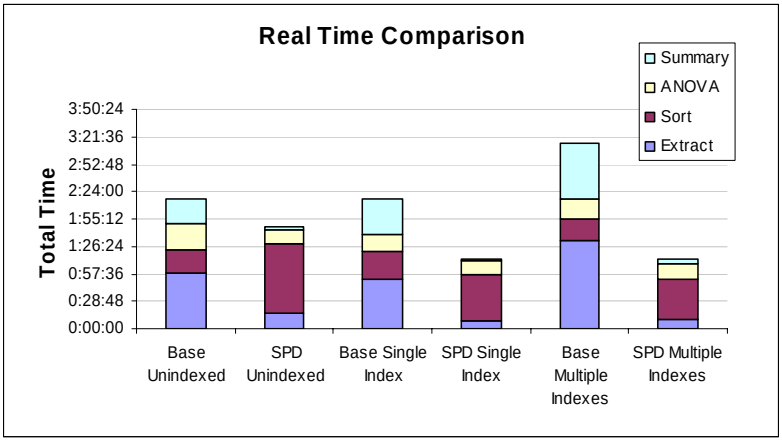
In order to make a valid comparison only time measurement obtained from the Extraction, Sorting, ANOVA, and Summary are compared. Indexing time, because not all measurements from each test involved the creation of one or more indexes, is not counted in the comparative measurements. However it should be noted that when reviewing the total time (see attachment A) the inclusion of indexing time did not alter the results. Similarly, as an attempt to remove confounds caused by normal maintenance and back-up processes of the SAS system, each test was performed five times varying from early morning to overnight. The resulting FULLSTIMER statistics were compiled and an average was calculated to create the comparison discussed below. A full list of the average times can be found at the end of this document.

Finally, during tests involving single or multiple indexes, prior to each new step, the indexes were created on the resulting dataset. For example, if the test series involved extraction, sorting, and the ANOVA, Prior to sorting, the dataset resulting from the extraction is indexed.

This detail is carried throughout the tests to ensure that each resulting dataset maintains the same indexing criteria as the original dataset.

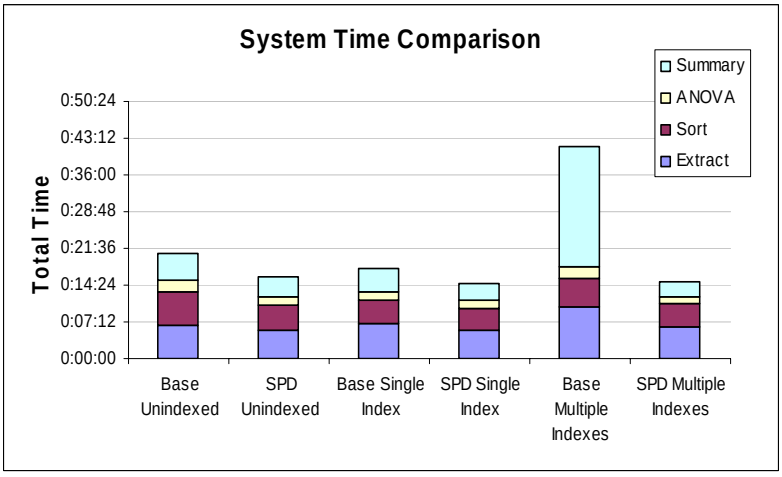
Result Set One

Real Time Comparison – During this series of tests, the most surprising result was the overall maximum and minimum times to perform the given tests were obtained when multiple indexes were used. The maximum time resulted from the dataset on SAS Base while the minimum time resulted from the SPDE table. What was expected was that the BASE table with no index would result in the longest real time while the multiple indexed SPDE table would have resulted in the shortest time. Also interesting is that as indexes were introduced to the BASE table, the average real time increased for the overall process with multiple indexing but largely stayed the same between no indexing and a single index. The same result was observed for BASE data for the ANOVA calculation. Not surprising during this measurement is that as indexes were introduced to the SPDE table, the real time decreased. Again, times spent on the indexing process are not considered in these measurements.



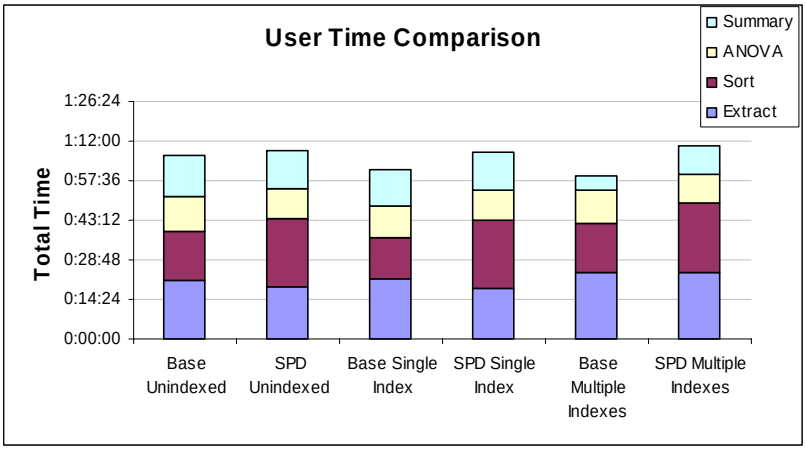
Result Set Two

System Time Comparison – System time comparison resulted in similar findings as the Real time comparison. Again, The BASE table with multiple indexes performed unexpectedly. The hypothesis was that indexing would greatly improve the behavior of the Summary Procedure. As depicted in the System Time Comparison Graph, the Summary procedure accounted for more than half of the time spent performing the designed set of tasks. There are slight variations in the remaining tests, however none compare with the variation in the Base table with multiple indexes Summary step. Unlike the Real Time comparisons, each addition of an indexing process resulted in an almost insignificant difference for the SPDE datasets.



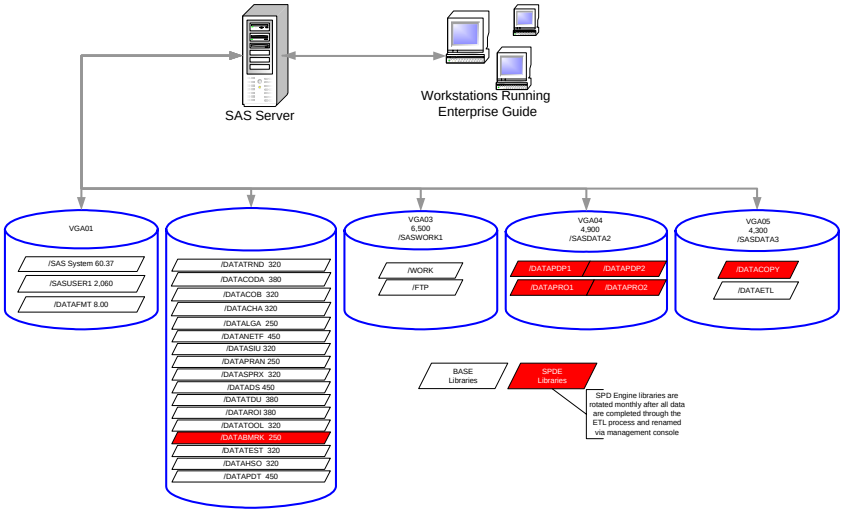
Result Set Three

User Time Comparison – As expected based on the hardware configuration, there is very little difference in the User Time Comparison as our server, even when at user capacity, does not suffer from any I/O bottleneck due to the amount of RAM. If a conclusion can be drawn from User Time, it is that depending on the Engine and the Indexing Scheme, different procedures within the testing parameters varied slightly in the time they took to execute. Though not conclusive in such a controlled setting, further investigation in real world scenarios, more than likely, would result in a much different result.



A REAL WORLD APPLICATION

After experiencing massive growth in as little as three years, it became increasingly evident that it was important to investigate alternative methods of storage and retrieval of data for Analytical Purposes. Research has pointed us to a methodology allowing for growth while controlling access to data and data sharing. We have created a back end storage solution which takes all the above research into consideration. Several of our mount points consist of both BASE and SPDE engines. As a rule, SPDE data are used as a 'Read Only' Production or Benchmarking data while all other mount points are created with BASE systems. Users and User Groups enjoy segmented data mounts in BASE and create an environment where different groups of analysts can collaborate and share data all the while, transactional data from a variety of sources can be continually updated in the SPDE environments by the ETL team to keep the most up to date data available in a controlled production environment. In short, what we have created is, to the analyst, a small system allowing them the access to their own USER library, access to their particular analytical group's library and access to the production data.



Interesting statistics that we have realized in the real world since adopting this architecture;

- Table scans on over 600 million rows happen in less than 12 seconds.
- Atomic level searches in the SPDE data environment have improved by over 300% as measured by User Time.
- SAS Data Steps and PROC SQL are equally as efficient
- Parts of the ETL process have decreased by 200% due to the use of asynchronous creation of indexes within the SPDE datasets.

The advantages other than mentioned above are too numerous to mention here. As a real world conclusion, we can state that with over 20 billion lines of data within our SAS system, we are experiencing no data latency. This statement would be impossible to make without adopting the architectural adjustments based on the research undertaken resulting in this paper.

CONCLUSION

In a very controlled environment, it is clear that use of the Scalable Performance Data Engine in conjunction with a specifically designed indexing schema, does result in much faster real time data acquisition than do other methodologies. This result was demonstrated in both Real Time and System Time. Surprisingly, in Base Tables of this size, a single index produced an unexpected result. Single indexing resulted in significantly less time in processing than did multiple indexing in both Real Time and System Time.

ACKNOWLEDGMENTS

I wish to acknowledge the Humana SAS Users group and Humana SAS Steering Committee for constantly pushing for new and better ways to make massive data more accessible. Without the business push, innovation would have no reason to take place.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Don Kros
Enterprise:	Humana, Inc.
Address:	500 West Main Street
City, State ZIP:	Louisville, KY 40202
Work Phone:	502.476.1134
Fax:	
E-mail:	DKros@Humana.com
Web:	www.Humana.com

Attachment A

Time Summaries

SPD Unindexed	Index	Extract	Index	Sort	Index	ANOVA	Summary	**Total**	Total+Index
real time		0:15:41		1:13:29		0:14:23	0:03:09	**1:46:42**	
user cpu time		0:18:53		0:24:56		0:10:54	0:14:00	**1:08:42**	
system cpu time		0:05:43		0:04:52		0:01:36	0:03:48	**0:15:59**	

Base Unindexed	Index	Extract	Index	Sort	Index	ANOVA	Summary	**Total**	Total+Index
real time		0:58:08		0:24:56		0:26:51	0:26:27	**2:16:22**	
user cpu time		0:21:09		0:18:13		0:12:25	0:15:03	**1:06:50**	
system cpu time		0:06:33		0:06:36		0:02:20	0:05:14	**0:20:43**	

SPD Single Index	Index	Extract	Index	Sort	Index	ANOVA	Summary	**Total**	Total+Index
real time	0:17:39	0:07:24	0:10:51	0:50:06	0:08:38	0:13:12	0:03:07	**1:13:49**	1:50:57
user cpu time	0:25:09	0:18:32	0:11:51	0:24:52	0:12:25	0:10:31	0:13:51	**1:07:47**	1:57:11
system cpu time	0:05:07	0:05:27	0:01:52	0:04:31	0:02:03	0:01:22	0:03:21	**0:14:42**	0:23:44

Base Single Index	Index	Extract	Index	Sort	Index	ANOVA	Summary	**Total**	Total+Index
real time	0:39:02	0:51:48	0:15:25	0:28:48	0:26:48	0:18:39	0:37:34	**2:16:49**	3:38:04
user cpu time	0:22:43	0:21:42	0:08:32	0:15:11	0:09:09	0:11:14	0:13:45	**1:01:53**	1:42:17
system cpu time	0:06:56	0:06:44	0:02:11	0:04:43	0:02:22	0:01:47	0:04:26	**0:17:40**	0:29:08

SPD Multiple Indexes	Index	Extract	Index	Sort	Index	ANOVA	Summary	**Total**	Total+Index
real time	0:19:17	0:09:17	0:11:34	0:43:22	0:09:21	0:15:20	0:04:17	**1:12:16**	1:52:28
user cpu time	1:06:35	0:24:14	0:52:05	0:25:02	0:25:20	0:10:35	0:10:31	**1:10:23**	3:34:23
system cpu time	0:08:10	0:06:06	0:03:38	0:04:35	0:02:48	0:01:23	0:03:08	**0:15:13**	0:29:49

Base Multiple Indexes	Index	Extract	Index	Sort	Index	ANOVA	Summary	**Total**	Total+Index
real time	1:49:39	1:33:15	0:44:43	0:21:30	0:34:27	0:21:28	0:58:48	**3:15:01**	6:23:50
user cpu time	1:13:17	0:24:07	0:39:24	0:17:49	0:21:58	0:12:20	0:05:13	**0:59:29**	3:14:08
system cpu time	0:17:47	0:10:13	0:09:05	0:05:27	0:05:13	0:02:14	0:23:41	**0:41:35**	1:13:40

* Data displayed represents the mathematical average of the five independent executions of the tests described in this paper