Raising the bar on Business Intelligence:

SAS9® and HP Integrity® Servers

sas. | hp invent

Executive summary

For more than 20 years, HP and SAS Institute have joined efforts to deliver intelligence to customers from all industries with outstanding performance and precision. With the introduction of the Intel Itanium based server family, HP Integrity, HP today provides the 64 bit compute platform that makes the difference when turning raw data into intelligence. Optimized for the HP Integrity server platform, SAS Institute launched its SAS9 Enterprise Intelligence Platform capable of delivering strategic intelligence beyond Business Intelligence. This paper provides insights into system architectures and performance metrics, giving readers an understanding of the behavior of SAS9 on HP Integrity servers.

The tests described throughout this paper demonstrate HP/SAS performance for the major data handling steps as they are used in typical intelligence solutions. Starting with the loading of data into a central data store, we create summaries in a second step that provide results for standard reports. In order to provide the level of interactivity expected by customers today, we generate multidimensional data in the third step. For the fourth test, we implemented a whole SAS Enterprise Intelligence Platform as it would run at a customer site and tested its capabilities to handle typical BI data.

When loading the data, we demonstrate that SAS9 is capable of truly utilizing all resources provided by the HP Integrity server. SAS9 immediately benefits from additional IO bandwidth provided to it, showing true scalability and effective resource utilization by cutting the load time by half. In the areas of data aggregation, we experience a breakthrough performance due to the 64 bit address spaces used by SAS and provided by HP, enabling the system to cut down the aggregation step by 45%. Companies can run similar data preparation jobs in 45% of the time they needed before.

The fourth test demonstrates the many options customers will have when implementing SAS9 on HP Integrity servers and the benefits they gain by using this distributed intelligence platform. We demonstrate that the tested platform could build a multidimensional data source based on 250 million records in only 32 minutes, a staggering reduction from the 47 hours the same data source would need to be built on previous configurations. The results provide a validation for scalability and true workload balancing when HP Integrity servers run SAS9.

Advances in hardware and software continue to improve SAS application performance. This paper reviews improvements in the SAS9 Enterprise Intelligence Platform and how they are complemented by the HP Integrity server. Four user scenarios were selected to represent the different areas of SAS application behavior, with appropriate data and SAS application code to support each area of testing. A detailed review of the configuration provides the foundation for understanding the importance of platform choice for SAS applications. Details of the performance improvements, as well as important architectural considerations, are noted throughout the paper. Charts and graphs are used to depict the performance improvements in each area, with detailed explanations of the significance to the SAS Enterprise Intelligence Platform environment. We conclude with a review of the results from an administrative perspective and a discussion of the HP-SAS relationship.

This paper is intended for SAS administrators, technical evaluators of data warehousing architectures, and those who are responsible for designing hardware and software architectures for enterprise data warehousing solutions.

Defining performance

In order to understand the content of this paper, it is important to understand the context and approach which was taken. Performance may mean different things to different people. To an analyst, performance may be defined as the ability to get their data analyzed and reports generated in a given timeframe – say, before lunch. To a system administrator, it may mean that the system is fully utilized during normal operation. Thus, the context and definition of performance becomes important in gaining an understanding of the analysis of this and other performance data.

One of the most common definitions of performance is throughput. This is often referenced in the amount of work done per unit time – for example, the analysis of 100 MB of data in a 3 hour period. This definition contains little if any reference to the amount of resources needed by the server to accomplish this work. This is the definition most often used by system analysts and other users of various application software, and will typically be given in real (wall clock) time.

Another definition, less common, is utilization. This refers to how much capacity of a resource is used to perform a given workload. This provides more detail about the server resources, such as CPU, IO, and memory, which are needed to complete the requested workload. This definition may contain several distinct areas, such as CPU utilization, and is of interest to system administrators and capacity planners. This definition may contain several distinct areas, such as CPU utilization, when describing the characteristics of a particular application workload. Note that high utilization is not necessarily a problem, if other performance goals (such as throughput) are attained.

For the purposes of this paper, we will most often reference throughput as our metric of interest, since that is the area of most interest to application users. We will also provide some comparisons of server utilizations where appropriate to highlight the Itanium architecture and the beneficial impact it has on SAS.

Description of tests

For this project, four sets of tests were selected. These groups of tests were chosen to represent different types of SAS applications. All tests were submitted in batch mode, allowing for a more accurate representation of execution (real) time for each of the test groups. Scenario one deals with standard SAS procedures loading data of varying sizes into SAS data sets. Scenario two extends scenario one by adding the scalability of partitioned datasets. In Scenario three we have created a number of multidimensional databases (SAS/MDDB), simulating the data preparation of a typical reporting environment. The final scenario four focuses on the new SAS9 Enterprise Intelligence Platform.

The following subsections provide a detailed description of the tests conducted for each scenario:

Scenario1 – Loading data into Base SAS

Scenario 1 deals with the initial process used to create any data warehouse or data mart. In this scenario we load test data into an empty table and measure the time it takes to load different volumes of data. After data is loaded into data sets, we create 6 indexes on the tables and measure the time it takes to create these indexes.

Once the data has been loaded in the data set and indexes have been created, we sort the data to prepare it for the aggregation. After sorting the data, we create a summary on top of the data simulating the creation of materialized views on the data. The summaries store aggregations of the data along 9 attributes (class variables).

In this scenario we used the Base SAS engine so all of the data was stored in individual SAS data sets. The datasets contain 25 million, 50 million, 100 million, 200 million and 250 million records. With this sequence of data sets, we demonstrate the scalability of a single data set.

Dataset	Records
Ftd_25	25,000,000
ftd_50	50,000,000
Ftd_100	100,000,000
Ftd_150	150,000,000
Ftd_200	200,000,000
Ftd_250	250,000,000

Scenario 2 – Base SAS procedures with SAS scalable performance data engine

In this scenario we used the same data flow as in Scenario 1, but we used the SAS Scalable Performance Data Engine technology. This engine partitions data sets across multiple file systems available to the HP Integrity server.

We iterated the tests for 50 million to 250 million records on 2, 3 and 4 file systems to demonstrate the impact of using scalable storage instead of a single data set.

For this scenario we created empty data sets with indexes defined on the same columns as in Scenario 1. We loaded the data into these template data sets, but this time we did not sort the data sets before creating the aggregations, as the SAS Scalable Performance Data Engine implicitly sorts data. The engine creates all 6 indexes in parallel as data is loaded into the partitioned data sets. The aggregations we create are the same ones as in Scenario 1.

Scenario 3 – SAS/MDDB build and query tests

Scenario 3 focuses on creating multidimensional databases. SAS/MDDB is a legacy product from SAS 8 that continues to exist in SAS9 for backwards compatibility. In this test we used the data created in Scenarios 1 and 2 to create MDDBs. The test demonstrates the differences in runtime when using SAS data sets in comparison to portioned SAS data sets generated by the SAS Scalable Performance Data Engine. The data for the tests still contains 50 million to 250 million records.

Scenario 4 – SAS9 Enterprise Intelligence platform

Scenario 4 is the most complex one, as it involves the whole SAS Enterprise Intelligence Platform. We use a metadata server running on 4 processors of the test system that handles all of the metadata about the environment. There is a workspace server using 6 processors dealing with all the data transformation work and an OLAP server using 6 processors. The workspace server is used to build cubes based on the data created in Scenarios 1 and 2. The OLAP server is used to answer queries coming into the OLAP server from client applications. In Scenario 4 we used the new SAS9 OLAP Server to create new cubes with the same structure as the MDDBs in order to compare SAS9 OLAP Server with SAS/MDDB.

For all testing, log and list files were collected for analysis. In addition, a performance monitor (HP Measureware) was running during the tests to collect system data. Although all of the test servers were on a network, analysis showed that network traffic during the testing was negligible.

System configuration

Here are the hardware and software configurations used during the testing process:

Server configuration
ctcipf1:
HP Integrity rx8620 Itanium 2 server
16 processors, 1.5 GHz 6 MB L3 cache
64 GB memory, 25% buffer cache (fixed)
Three internal 146GB 10K RPM disks
Four 2GB Fibre Channel Adapters
HP-UX 11i V2 Enterprise OE, March 2004 Patch Bundle
SAS 9.1.2 Foundation Release

Processor sets were used to optimize the rx8620 architecture in Scenario 3 and 4 testing. Through the use of processor sets, we were able to configure the rx8620 as follows:

Processor Set	Number of CPUs	Load
PSET 0	6	Default pset and Workspace Server
PSET 1	4	Metadata Server
PSET 2	6	OLAP Server

By isolating these SAS application servers into separate logical partitions, we were able to utilize two important attributes for configuring the rx8620 with processor sets. They are:

- Improved memory latency for small processor sets
- Improved cache and TLB (translation lookaside buffer) efficiency that results from isolation of SAS application servers

The following kernel tunable was configured:

STREAMPIPES=1

Disk configuration
In order to meet enterprise-class storage requirements, an EVA5000 disk array with four 2GB FC-AL adapters was used for all testing. The EVA5000 is a 'virtual' storage array, meaning that the storage array creates high performance 'pools' of storage behind the controllers, This results in improved utilization of capacity, simplified management, and high performance.

Whenever possible, multiple file systems were used to take advantage of the performance of the EVA5000. File systems were created using the HP Logical Volume Manager (LVM) and were 400-700GB in size.

The file systems were configured as shown:

Mount point	Size (MB)	Comments
/data1	700GB	Single controller (cntl 1)
/data2	700GB	Single controller (cntl 2)
/data3	700GB	Single controller (cntl 3)
/data4	700GB	Single controller (cntl 4)
/work	400GB	Load-balanced across four FC controllers

SAS configuration
The SAS9 software was installed into the /sas directory on the root volume group. All SAS options and configuration variables were left at the installed defaults.

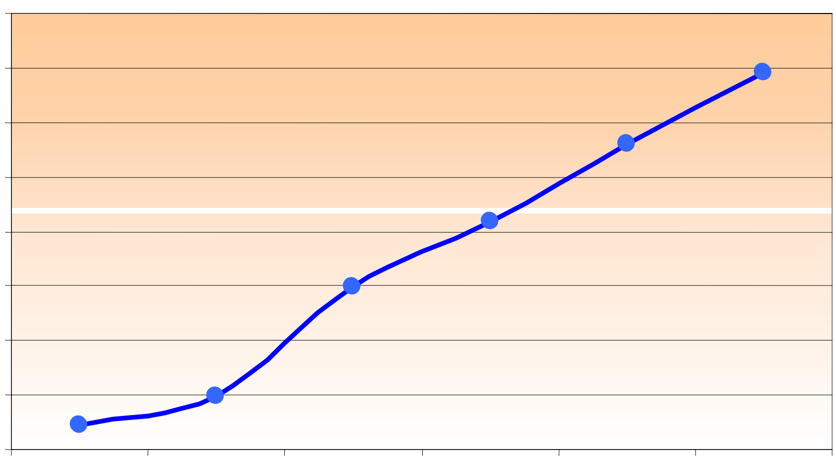
Results and findings

The aim of the first scenario is to show how the load time evolves as more and more data is loaded into SAS data sets. The following table lists the run times for each of the data sets we created in this scenario.

Dataset	Records	Load time	Index time	Sort time	Overall
Ftd_25	25,000,000	136.69 sec	315.43 sec	433.94 sec	886.06 sec
Ftd_50	50,000,000	458.06	744.50	763.68	1966.24
Ftd_100	100,000,000	887.03	3581.14	1507.07	5975.24
Ftd_150	150,000,000	1293.82	4843.22	2244.26	8381.30
Ftd_200	200,000,000	1810.48	5833.37	3575.83	11219.68
Ftd_250	250,000,000	2224.93	7321.95	4299.63	13846.51

If we create a graph that displays the overall runtime depending on the number of records loaded into the data set, the scalability of the SAS data set becomes visible. The jump between 50 million and 100 million records is due to the caching of data inside the HP Integrity server. Below 100 million records, all data can be handled in memory but as we cross the memory limit, the indexing algorithm uses utility files on disk that slows down overall performance.

Loading Data – Overall Run Time

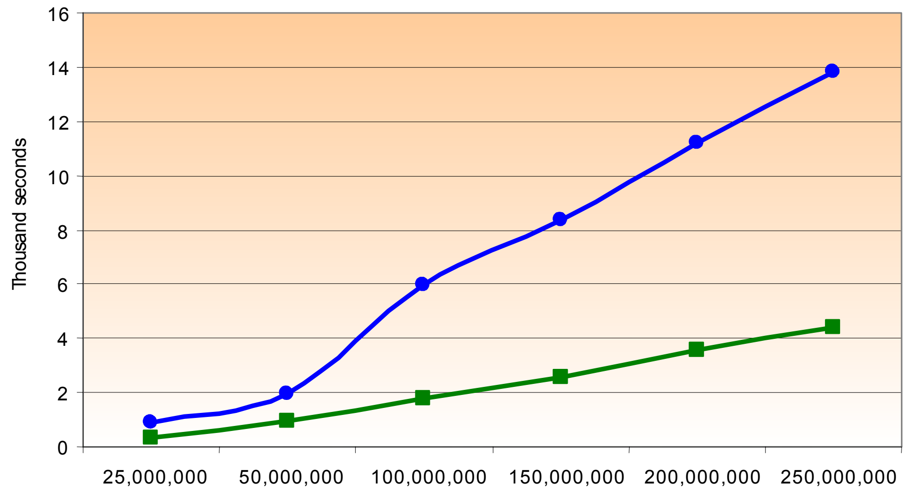


Now let's take a look at Scenario 2 and check how the SAS Scalable Performance Data Server technology works. This time we load the same data as in Scenario 1 but as described above, the indexes get built while the data is loaded and the sorting is done implicitly. The SAS Scalable Performance Data Engine uses two file systems to partition the data.

Dataset	Records	SPD (2 file systems)
Ftd_25	25,000,000	311.81 sec
Ftd_50	50,000,000	958.13 sec
Ftd_100	100,000,000	1781.74 sec
Ftd_150	150,000,000	2540.76 sec
Ftd_200	200,000,000	3543.02 sec
Ftd_250	250,000,000	4421.42 sec

It is worth mentioning that these times *are less than half* of the times used by the base engine. In order to make the difference more obvious, the following chart displays the times of the SAS Scalable Performance Data Engine in direct comparison to the Base engine.

SPD (2 File Systems) versus Base Engine on HP Integrity server



This graph shows the benefit we gain from using two file systems instead of a single file system. The activities of creating indexes and sorting data benefit from the availability of two separate IO paths and the ability to spawn multiple threads that work in parallel. The task at hand is now divided into sub-tasks and conquered in parallel using the excellent I/O performance provided by the HP Integrity server. As an example, the SAS Scalable Performance Data Server technology will generate the indexes defined on the tables in parallel while the data is being loaded into the library.

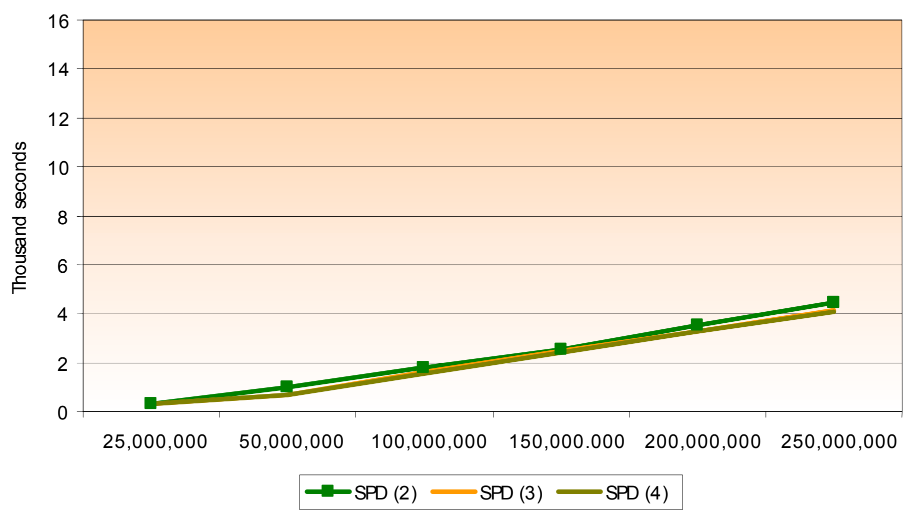
Looking at the server logs, it becomes obvious that this is still not the optimum.

The entire system (SAS Software running on HP Integrity server) is still bound to IO performance. The CPUs are not yet busy but the two channels used in this test reach their bandwidth limit. In order to overcome this limit, we added additional file systems. The following table lists the data loading performance as the number of file system increases.

Dataset	Records	SPD (2)	SPD (3)	SPD (4)
Ftd_25	25,000,000	311.81 sec	303.85 sec	306.78 sec
Ftd_50	50,000,000	958.13 sec	683.05 sec	702.23 sec
Ftd_100	100,000,000	1781.74 sec	1589.37 sec	1559.87 sec
Ftd_150	150,000,000	2540.76 sec	2445.39 sec	2406.85 sec
Ftd_200	200,000,000	3543.02 sec	3289.48 sec	3303.68 sec
Ftd_250	250,000,000	4421.42 sec	4125.30 sec	4102.44 sec

The numbers already provide a hint that the system did hit another boundary: when reading the data into the SAS Scalable Performance Data Engine, the file system we were reading from would impose the speed limit. We simply cannot read the data fast enough from a single file system to effectively use the additional file systems. So in this case the loading would not run faster with additional file systems because the system would not be able to obtain the data fast enough.

Additional Filesystems used by SPD on HP Integrity server



Even though we could not further decrease the loading time by adding more disks to the system, it still makes sense to load the data into more partitions, as we can use these partitions to boost overall performance in later steps. All procedures that will read data from these partitioned data sets will benefit from a much higher bandwidth, as the next step in the test clearly shows.

In this next step we summarized the data that we loaded before. The summary does not summarize all the data but a large subset of the data (for a given value of one class variable). In Scenario1 we summarized the data stored using Base SAS:

Dataset	Records	Base

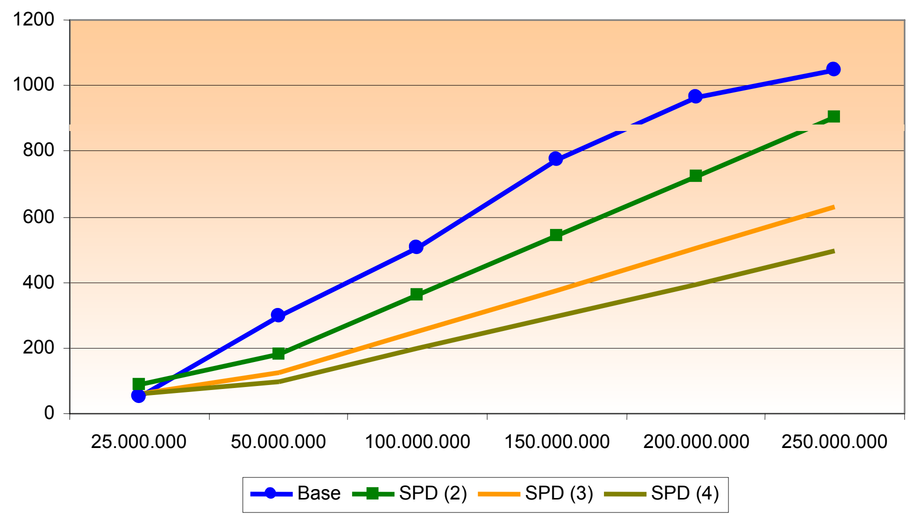
Ftd_25	25,000,000	49.63 sec
Ftd_50	50,000,000	297.33 sec
Ftd_100	100,000,000	503.00 sec
Ftd_150	150,000,000	774.25 sec
Ftd_200	200,000,000	965.70 sec
Ftd_250	250,000,000	1048.27 sec

In Scenario 2 we summarized the data using SAS Scalable Performance Data Engine:

Dataset	Records	SPD (2)	SPD (3)	SPD (4)
Ftd_25	25,000,000	89.44 sec	62.45 sec	61.67 sec
Ftd_50	50,000,000	178.59 sec	124.64 sec	98.48 sec
Ftd_100	100,000,000	360.73 sec	251.27 sec	197.58 sec
Ftd_150	150,000,000	541.43 sec	377.56 sec	295.08 sec
Ftd_200	200,000,000	722.90 sec	503.64 sec	395.63 sec
Ftd_250	250,000,000	904.72 sec	630.20 sec	493.93 sec

If we take these numbers to draw a graph, the impact of partitioned data sets becomes obvious:

PROC summary on HP Integrity server

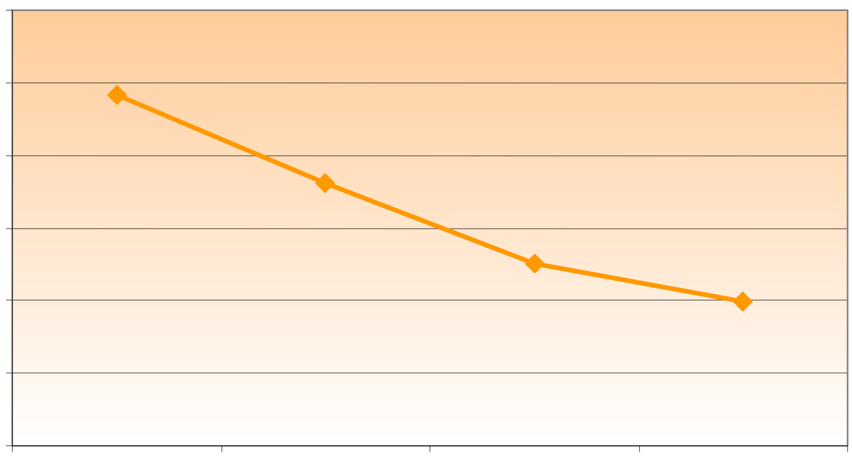


The graph shows that it there is no performance advantage to using partitioned data sets for small amounts of data. In the case of 25 million records, the base engine is faster than any of the partitioned datasets. But as soon as the data reaches a significant size, partitioning makes a significant difference. This fact is especially important if a Business Intelligence environment is driven by SAS Scalable Performance Data Server. Every single report requires summarized data every time it runs. In an environment with large user communities reporting on large volumes of data, the summary step becomes very important for the overall performance of the whole system.

By partitioning the data we were able to reduce the time of the summary step by more than 45%. As summarization is the major activity in Business Intelligence environments, the benefit of running SAS Scalable Performance Data technology on multiple HP file systems makes a noticeable difference for end users accessing business reports.

The following chart makes the benefit gained from more partitions obvious. The graph shows how the time to calculate an aggregation decreases as the number of partitions used increases.

SAS9 and HP Integrity server – Impact of Multiple Partitions for Aggregation



SAS9 and the HP Integrity server working together cause the performance increase. SAS9 does not run in a single thread doing all the work in sequence, but it spawns off a series of threads that do work in parallel. HP Integrity server handles these threads and distributes them across its processors. The HP Integrity server architecture ensures the efficient use of threads through the HPUX operating system, which is specifically designed and tuned for maximizing performance of multi-threaded applications.

SAS also does not need to store data in a single big ‘chunk’, but it can partition data across multiple file systems in order to increase bandwidth to and from those file systems. The HP Integrity server that was used for the test was equipped with 4 controllers connecting the system to an EVA storage device. To validate this, we used up to four file systems, each one stored on a RAID1array and connected to the server via an individual controller channel. Looking at the peak bandwidth being used during the tests, it becomes clear how the data throughput increases as more and more file systems are used.

Dataset	Records	Base MB/s	SPD (2) MB/s	SPD (3) MB/s	SPD (4) MB/s
Ftd_25	25,000,000	84,173	92,467	99,123	107,418
Ftd_50	50,000,000	138,445	107,110	114,278	144,077
Ftd_100	100,000,000	139,571	146,538	176,918	264,589
Ftd_150	150,000,000	145,203	156,459	216,474	263,373
Ftd_200	200,000,000	119,500	155,139	237,981	259,604
Ftd_250	250,000,000	136,621	155,243	252,334	299,421

Up to this point we tested the load speed and the summary speed of SAS9. We saw a dramatic performance increase when using partitioned data sets and we also observed very good summary performance. A typical warehouse environment also contains a number of multidimensional data sources besides the main data warehouse and some aggregation tables (materialized views). Therefore, we created a number of OLAP data sources in the next test phase.

Based on each one of the datasets created earlier, we created a multidimensional database. As the data sets scale from 25 million to 250 million records, this test provides insights about the build times of OLAP cubes when the input data increases. This test demonstrates the capabilities of a 64bit operating system to handle such large volumes of information.

In the first test sequence we used SAS9 in its compatibility mode. In this mode, customers can use all of the functionality known from SAS8 without having to change any code. Starting with SAS9, a new procedure has been introduced that generates cubes for the new multithreaded OLAP Server: PROC OLAP. In order to measure the performance of PROC OLAP, we generated a “base line” using the old PROC MDDB that was used in SAS/MDDB server in SAS8.

PROC MDDB generates a file that contains a multidimensional database. In this test, the resulting MDDB would have 4 dimensions (16073, 1763, 6565, and 1180 leaf members) and 40 measures. The NWAY aggregation (base aggregation) contains 220,411,933 cells.

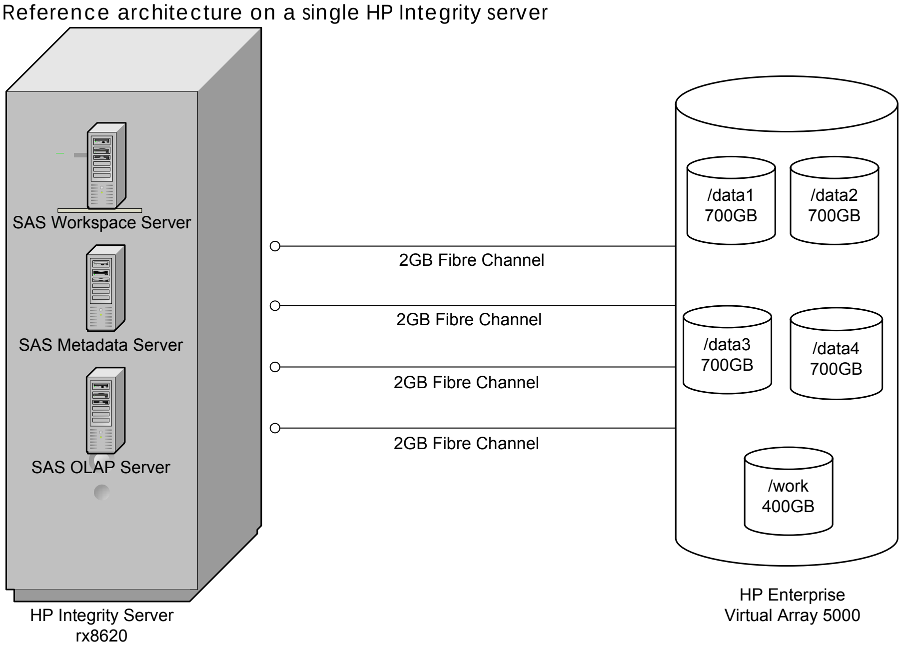
MDDB	Records	Built Time	Built Time(seconds)
Ftd_25	25,000,000	Not tested	Not tested
Ftd_50	50,000,000	1:43:03.96	6,183.96
Ftd_100	100,000,000	5:14:09.78	18,849.78
Ftd_150	150,000,000	13:26:13.87	48,373.87
Ftd_200	200,000,000	28:46:40.72	103,600.72
Ftd_250	250,000,000	47:14:30.01	170,070.01

As you can see, SAS/MDDB generates quite a lot of work. PROC MDDB is a single threaded procedure that reads in the detailed data, summarizes the data and generates the MDDB file. Only the summary object can benefit from the new SAS9 architecture as it can use multiple threads to divide the table and summarize multiple parts in parallel (a strategy commonly known as divide and conquer). It is worth mentioning that some of these MDDBs were built with more than a 100GB of detailed data (ftd_200, ftd_250) without partitioning the cube or other types of optimizations. HP’s 64 bit architecture enables this direct approach to the build process as it can directly access a very large address space.

Besides PROC MDDB, SAS9 also offers users PROC OLAP, a new procedure that makes use of the multithreaded kernel in SAS9, parallel IO and the new SAS OLAP Server. In order to use this new procedure, the SAS9 Enterprise Intelligence Platform needs the setup that is the foundation for all SAS9 solutions. This platform consists of a number of SAS servers with different purposes that establish a distributed environment for transforming, storing, and reporting on data. For our scenario we need a SAS Metadata Server that handles all of the metadata used throughout the entire platform. We need a SAS Workspace Server to actually do the work of building the cubes, and finally we need the SAS OLAP Server to surface the OLAP cubes.

For this test we implemented a reference architecture on a single HP Integrity server. HP provides a concept for virtualization of hardware resources that lets system administrators model any kind of virtual server inside a physical server, by assigning processor sets and memory sections to individual threads. This provides for isolation of the different SAS application servers as well as improved memory latency for the smaller processor sets. For our tests we defined three processor sets and let SAS handle the memory on its own. We assigned the SAS Metadata Server to a set of 4 processors, the SAS Workspace Server to a set of 6 processors and the SAS OLAP Server to another set of 6 processors. With this approach we have three SAS servers running in a single HP Integrity server without interfering with each other. The SAS processes believe that they are running on their own equipment. The HP Integrity server is attached to an EVA storage device via four fiber optical data links (2Gb/s).

Reference architecture on a single HP Integrity server



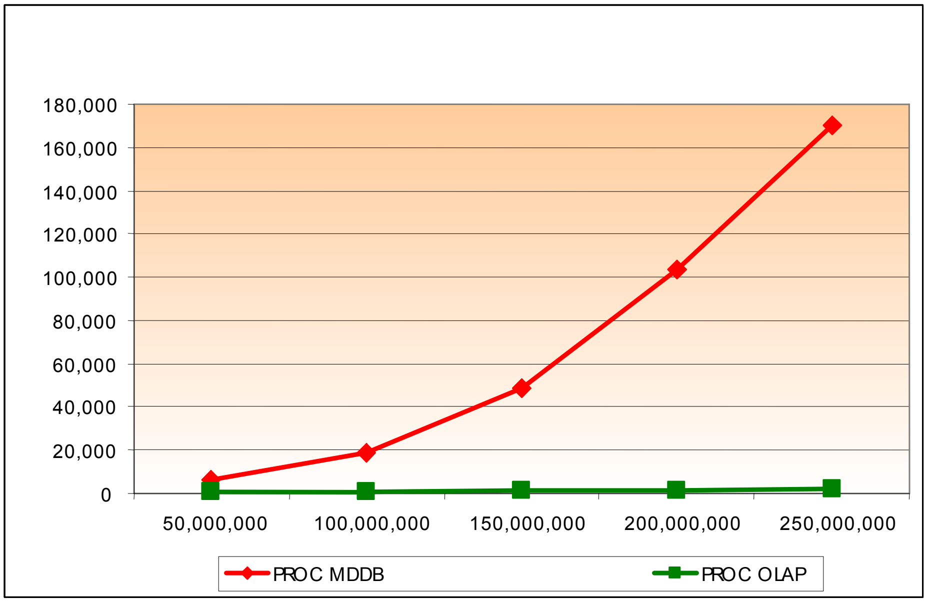
In this environment we used PROC OLAP to create exactly the same cubes as the ones we had built earlier using PROC MDDB. Again we used four dimensions and forty measures, but this time we did not write the cube data to a single file but used SAS9 OLAP server's capability to use parallel IO. We stored the cube data on three separate file systems and used one file system as the work area for the cube build process.

In this architecture we ran five PROC OLAP processes to build the five cubes based on the data sets we used throughout the entire test. The following table provides the details about how long PROC OLAP needed to build exactly the same cubes as PROC MDDB above.

MDDB	Records	Build Time	Build Time(seconds)
Ftd_25	25,000,000	Not tested	Not tested
Ftd_50	50,000,000	6:56.21	416.21
Ftd_100	100,000,000	12:18.17	738.17
Ftd_150	150,000,000	19:33.08	1173.08
Ftd_200	200,000,000	26:22.41	1582.41
Ftd_250	250,000,000	32:29.75	1949.75

The following chart compares the PROC MDDB build times with the PROC OLAP build times for the same cubes:

PROC OLAP versus PROC MDDB



First of all, it becomes immediately obvious that PROC OLAP runs an order of magnitude faster than PROC MDDB. We have cut down a 47 hour job to 32 minutes! There is one more major difference between the procedures: while build times for PROC MDDB seem to evolve exponentially, the PROC OLAP plot seems to be linear. In addition, data collected from Measureware indicates that the processor sets for the Metadata server and OLAP server were lightly loaded during the PROC OLAP testing, suggesting additional headroom for scalability of the SAS application in those partitions.

Looking at the hardware usage during this test, it turned out that our 6-processor workspace server processor set did not have too much to do while building the cubes in sequence. Even though we were using four file systems, the IO was still the major bottleneck for the whole process. We decided to exploit the full threading capability of SAS9 by building the cubes in parallel. If you look at the SAS9 Enterprise Intelligence Platform, there is no reason why you should do your work in sequence. If there is any work you can do in parallel, the platform will handle it! The threaded kernel will take care of the workload and balance it across the processors. So we decided to actually go for the test and build the five cubes in parallel on the 6-processor workspace server.

OLAP Cubes	Cubes	Build Time	Build Time (seconds)
Sequential build	5 cubes	1:37:39.62	5859.62
Parallel build	5 cubes	43:32.74	2612.74

These tables and graphs clearly show the benefit of combining a powerful hardware with an intelligent software application to not only get better performance but also to get a completely different behavior that would be impossible by just tuning one part of the platform. These tests clearly demonstrate the benefits of the SAS9 application environment on HP Integrity servers. Simply put, the HP Integrity server is an ideal platform for the SAS9 application environment!

The HP Integrity server/HP-UX advantage

The HP partitioning continuum provides flexibility in configuring SAS solutions, allowing different servers on different partitions and even different operating environments, if desired. For example, a 32 processor HP Integrity Superdome could easily be configured as four 'virtual' servers, with a 4-processor SAS Metadata Server, an 8-processor OLAP server, a 8-processor SAS Stored Process server, and a 12-processor SAS Workspace Server. These four servers, although they are all within a single physical server, are 4 unique operating environments, each with their own individual configurations – they can even be different Operating Environments! And, by using resource utilization management software such as HP Workload Manager, system resources such as processing capacity and disk utilization can be modified dynamically across partitions, providing unmatched flexibility.

HP offers more than just hardware. Built for the future and tested in today's most demanding enterprise environments, the HP Integrity server line is designed to deliver the performance, availability, scalability, flexibility, security and manageability needed for applications ranging from mission-critical to compute-intensive. In addition, HP delivers complete solutions with industry-leading services, support, consulting, education and financing.

Based on the revolutionary Intel® Itanium® 2 processor co-developed by HP and Intel, HP Integrity servers reduce platform costs, enable higher performance and scalability, and enhance the SAS application environment. As part of HP's Adaptive Enterprise, HP offers customers the industry's broadest solution of scalable servers—all with a common architecture. Industry-standard processors in the Integrity server allow HP and SAS customers to reap numerous performance and cost benefits, including higher flexibility and scalable solutions.

The HP difference

Since 1986, HP has worked with SAS in development of the SAS application environment to provide our joint customers with a robust, manageable platform. We have continually worked together to extend and improve the SAS application environment through the use of HP optimizing compiler technology as well as integration with management and high availability software. In early 1999, HP and SAS partnered together in the development of SAS on early Itanium prototypes, which have evolved into the HP Integrity family of servers. SAS was an initial development partner with HP optimizing compiler technologies, and provided valuable insight and feedback in the development of the HP optimizing compiler for HP-UX 11i. The result is a tuned, optimized compiler which delivers outstanding performance for enterprise SAS applications. Also, SAS was one of the first partners to embrace the Integrity server family. Building experience with the HP Integrity server platform through the years, SAS has developed a knowledge base around the Integrity design center to optimize and exploit the Integrity architecture. Finally, our long-standing development partnership is complemented by our business relationship, with a team of dedicated professionals working together to insure the best customer experience for SAS applications on HP servers.

Conclusions

The scalability demonstrated in this paper shows how scalable hardware that provides flexible configuration of processors and memory as well as scalability of storage can leverage the SAS9 Enterprise Intelligence Platform to deliver true enterprise scalability as it is needed for modern Business Intelligence Platforms.

We have demonstrated how SAS9 and HP Integrity server can provide the intelligence platform needed today in order to handle an ever increasing number of users accessing an ever increasing wealth of intelligence in their daily business. Workloads can be divided and conquered among several parallel threads, enabling enterprises to handle their ever increasing workload within a stable and predictable time window.

References

HP StorageWorks Enterprise Virtual Array 5000
http://h18006.www1.hp.com/products/storageworks/enterprise/index.html

HP Integrity rx7620-16 server summary
http://www.hp.com/products1/servers/integrity/mid_range/rx7620/index.html

SAS OLAP Server technologies
http://www.sas.com/technologies/dw/storage/mddb

SAS9 Architecture Overview
http://support.sas.com/software/index.htm

For more information

Visit www.hp.com/go/sas

John Hall, Senior Systems Engineer
Hewlett Packard Company, Cary, NC
(919) 677-3606 j.hall@hp.com

Ingo Brenckmann, Technology Strategist
SAS Institute EMEA. Heidelberg, Germany
Ingo.Brenckmann@eur.sas.com

5983-0190EN, 12/2004