Paper 348-2010

Performance Impact of Virtualization on SAS® Foundation

Ken Gahagan, SAS Institute Inc., Cary, NC

ABSTRACT

Virtualization is a hot topic in the industry. Potential benefits of virtualization include increased utilization of hardware, reduced power and cooling costs, reduced management costs, increased manageability of the computing environment, and reduced capital expenditures. This paper compares and contrasts the relative performance impact of architectural decisions and empowers the reader to make informed decisions with respect to running SAS® in a virtualized environment.

INTRODUCTION

The bulk of SAS offerings are designed to support a multi-user installation. None of these offerings require that they be hosted in a virtualized environment. However, the advent of more powerful commodity hardware, virtualization technologies becoming common in data centers, and cloud computing has lead to a number of customers choosing to deploy SAS in a virtual environment. In recognition of this fact, this paper outlines a number of factors that influence the performance of SAS Foundation in a virtualized environment.

This paper provides an overview of virtualization, the advertised benefits, the architectural requirements, and the impact of those architectural requirements on SAS performance. One goal of this paper is to allow the reader who might not be an IT professional to understand the driving factors behind virtualization so that they can have an informed discussion with their IT staff about architectural decisions that impact the performance of SAS Foundation, including virtualization.

WHY VIRTUALIZE?

As the cost of computing hardware dropped over the years it became easier to justify the purchase of computing infrastructure dedicated to departmental or even personal needs. Over the years this has lead to a situation where it is not uncommon to have many servers throughout the data center all of which must be managed with operating system patches and managed with respect to hardware failures. All of them consume energy and add to the cooling requirements in the data center. Many of these servers also run at a fraction of their total compute capacity.

As data centers are exhausting floor space, cooling capacity, and power capacity, data center managers are looking for ways to consolidate underutilized compute resources to reduce costs associated with power and cooling as well as to free data center rack space. Virtualization is an approach that allows the consolidation of compute workloads and preserves the ability to independently manage the various virtual machines within the environment.

Virtualization brings a higher degree of manageability to the data center in addition to addressing floor space, power, and cooling challenges. For example, all major players in the virtualization space offer functionality that enable the migration of workloads to different physical servers within a cluster. This enables high availability even during scheduled hardware maintenance. If compute demands of a given virtual machine (VM) increase, the virtualization layer can migrate less active virtual machines to less used utilized hardware. This migration frees resources to better meet the needs of the virtual machine that needs more compute resources. Assuming capacity in the physical cluster of hosts, far less time is required to provision an additional virtual machine than it is to go through the procurement and provisioning process to make a new physical server available to the organization. This enables the IT organization to be more responsive to the needs of the business.

ARCHITECTURAL REQUIREMENTS TO SUPPORT SOME VIRTUALIZATION BENEFITS

Storage that is accessible to all the physical hosts in a compute cluster is an enabling requirement for many of the benefits of virtualization, such as the ability to move a virtual machine from one host to another, to provide high-availability, and dynamic workload management. This storage can be either on NFS, iSCSI, or SAN devices. The only requirement is that the device be accessible to all physical hosts in the cluster.

VIRTUALIZATION IS NOT MAGIC

It probably goes without saying, but virtualization does not create more compute resources than what exists on a physical machine. In fact, the virtualization software consumes some resources such as CPU cycles and memory, thereby actually reducing the absolute capacity of a physical server by a small amount. Depending upon the workload placed on the server, this reduction in resources might not even be noticed by users of the virtual servers. If the workload on the physical server is capable of scaling up to exceed the capacity of the physical server hosting the virtual machines, then it is probable that the users of the virtual resources will notice a decline in performance due to the load on the server.

Virtualization provides an efficient way to timeshare the physical resources between virtual hosts, but the bottom line is that the physical server has limited capacity.

SAS FOUNDATION PERFORMANCE

SAS has certainly grown beyond its roots in statistical analysis, but the strength of SAS in this area remains a strong component of the SAS value proposition. In many respects, the larger the data volume, the higher the value of any analysis performed upon the data. Over the years, the volumes of data available to be analyzed has grown significantly. As a result, the demands upon SAS processing capabilities have grown. SAS has addressed these demands in a variety of ways, but one way that is particularly significant in a virtualized environment is the use of multi-threading to enhance the performance of SAS. When SAS executes a multi-threaded section of code, the performance of the code is proportional to the number of threads that can be executed concurrently. This means that multi-threaded code generally performs better when more CPUs are available to service SAS.

Multi-threading enhances performance, but the total performance enhancement realized is subject to the rate that data can be made available to the process. This means that SAS performance will generally vary based on the performance of the I/O subsystem(s) configured for use by SAS.

The amount of memory that is available to SAS as well as the CPU clock speed also have a direct impact on the performance of SAS. This is not unique to SAS performance but is mentioned here in the interest of completeness. It is correct to conclude that the factors that influence SAS performance are the same factors that impact the performance of any system that must process large volumes of data. These factors are of concern regardless of the use of virtualization.

SAS SOLUTIONS ARE DESIGNED FOR MULTI-USER SCENARIOS

SAS solutions are designed to meet the needs of multi-user organizations. This is true for combinations of traditional SAS technologies (for example, SAS Foundation, the SAS Workspace Server, and the SAS Object Spawner) as well as for SAS business and data integration solutions (for example, SAS BI Server, SAS Enterprise BI Server, SAS Data Integration Server, and SAS Enterprise Data Integration Server) and the entire suite of industry solutions (for example, SAS Financial Management and SAS Risk Management for Banking. In many respects, the architecture of the SAS solutions addresses some of the issues that have lead to server sprawl by centralizing the bulk of SAS processing on a server instead of requiring that each user have an installation of SAS on their workstation or a server dedicated to their needs.

In environments where SAS processing places a steady load on the physical host throughout the day, or even if SAS processing spikes at certain times of the day, the best performance is realized by running SAS on an operating system that is installed on *bare metal* with an I/O configuration suitable to the volume of data being processed, adequate RAM, and multiple processors.

With this background information, we now turn to an investigation of the impact of virtualization and the related configuration choices on the performance of SAS Foundation.

THE METHODOLOGY

To measure the impact of various architectural decisions upon SAS performance, I needed a benchmark process that exercised a variety of operations that are commonly used in SAS processing. To that end I leveraged a benchmark process that is used internally at SAS, which operates on census-like data. The steps in the job are as follows:

- build a data set of 1 million rows of data
- create formats
- sort
- summarize the data on various fields
- write the data set out to a text file
- create a new data set by reading in the text file
- execute PROC DATASETS to create a number of indices
- execute PROC OLAP
- execute DATA _NULL_ with 1000^3 iterations

Samples of the code used for each of these steps can be found in Appendix A.

The benchmark was executed in various scenarios:

- bare metal / local storage.
- bare metal / NFS or CIFS storage.
- bare metal / iSCSI storage (1 Gb LAN).
- bare metal / iSCSI storage (10 Gb LAN).
- virtualized / local storage single VM.
- virtualized / iSCSI storage (1 Gb LAN) single VM.
- virtualized / iSCSI storage (1 Gb LAN) multi-VM.
- virtualized / iSCSI storage (10 Gb LAN) single VM.
- virtualized / iSCSI storage (10 Gb LAN) multi-VM.
- The VM configurations were executed with 2, and 4 virtual CPUs per VM.
- The VMs were consistently given 4 GB of RAM.

The benchmark executed with 6 iterations. The initial iteration was recorded but not factored into the overall execution time. Iterations 2–6 were averaged together to get to the overall average execution time.

The benchmark was executed on various hardware platforms:

- SUN X2200 (2 x AMD 2220 (2.6 GHz) dual-core processors / 8 GB RAM)
- HP DL360 G6 (1 x Intel X5560 2.6 GHz quad-core processors / 16 GB RAM)
- HP BL465c G6 (2x AMD 2435 2.6 GHz 6 core processors / 32 GB RAM)
- HP BL 460c G6 (2x Intel X5570 2.93 GHz quad-core processors / 48 GB RAM)

Various virtualization technologies were evaluated. The performance of the virtualization technologies is very similar and given that the focus of this paper is not to recommend a specific virtualization product, the performance of the various virtualization technologies is not contrasted with each other. The results demonstrate that while there is a performance cost for virtualization, that cost is small compared to the other choices that can be made related to which processor is chosen, storage topology, and the configuration of the VMs with respect to workload and available physical resources.

SUMMARY OF RESULTS

The results offered here are a summary of running SAS only on the most modern hardware. The results were similar across the various hardware platforms, accounting for deltas based on processor type and speed, storage configuration, network speed (for iSCSI scenarios), and allocation of physical resources across the VMs that were executing concurrently.

The short answer to the question of how virtualization impacts SAS performance is "it depends". The answer seems to depend less upon the question of whether virtualization is in the mix than it depends upon other environmental choices.

Table 1: Iteration Completion Times In Seconds by Scenario

Scenario	1	2	3	4	5	6	7	8	9	10
BL460c G6 (Intel X5570 / 48 GB RAM)	77.68	120.13	164.88	93.97	92.52	92.40	197.25	215.98	112.24	93.14
BL465c G6 (AMD 2435 / 32 GB RAM)	121.95	172.43	194.68	136.28	162.82	143.40	239.34	317.92	182.69	123.90
Percentage Differences	-57%	-44%	-18%	-45%	-76%	-55%	-21%	-47%	-63%	-33%

Table 2: Iteration Completion Time Multipliers with Bare Metal Intel Install Normalized to 1

Scenario	1	2	3	4	5	6	7	8	9	10
BL460c G6 (Intel X5570 / 48 GB RAM)	1.00	1.55	2.12	1.21	1.19	1.19	2.54	2.78	1.44	1.20
BL465c G6 (AMD 2435 / 32 GB RAM)	1.57	2.22	2.51	1.75	2.10	1.85	3.08	4.09	2.35	1.59

Scenario descriptions are as follows:

1-bare metal – local storage
2-bare metal install - work I/O local - Data I/O on NFS or CIFS
3-bare metal install - work and data I/O on NFS or CIFS
4-bare metal install - work and data I/O on iSCSI
5-VM - OS work and data I/O local SINGLE VM - 4vCPUs / 4 GB RAM
6-VM - OS work and data I/O on iSCSI SINGLE VM - 4vCPUs / 4 GB RAM
7-VM - OS work and data I/O on iSCSI (1Gb/s) 4 VMs 2vCPU / 4GB RAM each
8-VM - OS work and data I/O on iSCSI (1 Gb/s) 4 VMs 4vCPU / 4GB RAM each
9-VM - OS work and data I/O on iSCSI (10 Gb/s) 4 VMs 2vCPU / 4GB RAM each
10-VM - OS work and data I/O on iSCSI (10 Gb/s) 4 VMs 4vCPU / 4 GB RAM each

Table 2 shows that it is possible to execute four VMs, each with four virtual CPUs and four GB RAM with only a 20% performance delta from a single bare metal install using local storage. The local storage in this case is configured with an HP E400 SAS controller backed by 6 Gb/s SAS drives using a RAID 5 configuration. Table 2 also indicates that it is possible to deploy on both bare metal and virtual machines with much higher impact to performance based on other architectural choices. It can also be seen that the four concurrent benchmark streams executing in virtual machines in scenario 10 completed in less time than a single iteration of the benchmark stream executing directly on the server with different processors.

DISCLAIMER

By its very nature all performance results are specific to the hardware environment, configuration, and workload used to measure performance. Your results *will* vary based on hardware selection, configuration, and workload. The results herein are a reflection of the specific hardware tested and under the tested configurations. Any number of different choices would result in different results.

The results presented here are believed by the author to be valid generalizations and useful to present the following generalized recommendations. They should be considered holistically, not be construed to be a benchmark of any specific component within the testing scenario.

RECOMMENDATIONS

The following recommendations are offered to achieve optimal performance for SAS Foundation in any environment (virtual or bare metal):

- The new generation of Intel processors known as Nahalem seem to offer advantageous performance for the workloads typical of SAS Foundation. This benefit can be realized in bare metal installations and when hosting virtual machines.
- Favor remote block storage (SAN) over NFS, CIFS, (NAS) or local storage. Clearly, the performance of your storage appliance can have an impact on the ultimate choice of remote storage versus local storage.
- If using local storage, favor RAID 5 or 6 over RAID 1 or 0. (RAID 0 offers striping and performance advantage over RAID 1, but limits recovery options should there be a drive failure.) Tune and configure your storage appliance according to best practices provided by the manufacturer.
- Optimize connectivity to remote storage. Higher bandwidth HBAs for SAN connectivity and higher bandwidth network connections for iSCSI should offer better performance overall than lower bandwidth connections. This should be especially true as data volumes grow.

The following recommendations are offered to achieve optimal performance for SAS Foundation in virtual environments:

- Do not over allocate memory. The resulting swapping tends to negatively impact the performance of all VMs.
- Be very cautious of the degree of over allocation of CPU resources. If enough VMs spike in utilization such that the VMs cannot be efficiently scheduled on the physical CPUs, then performance of the VMs will suffer.
- When allocating virtual CPUs, ensure that you allocate sufficient virtual CPUs to each VM hosting SAS so that multi-threaded operations can spawn enough threads to achieve the intended performance gain.
- Determination of the optimal balance of virtual CPU allocation is best accomplished by trial and error. Fortunately, the nature of virtualization makes this task straight-forward.

CONCLUSION

To get optimal performance from SAS there are two critical considerations. The I/O subsystem that hosts the data must be able to feed data to the SAS process rapidly enough to minimize I/O wait states for the process. A sub-optimal I/O subsystem is more noticeable with faster processors. Additionally, there are many sections of SAS code that are engineered to be multi-threaded. These sections of code will perform better when they are able to spawn sufficient threads to handle the load presented for processing. These considerations are true regardless of the presence of virtualization technologies.

The impact of virtualization is not easily summarized due to the fact that there are generally a number of other changes within the environment that commonly are made concurrent with the decision to implement virtualization. As

demonstrated in the testing that supports the conclusions of this paper, it is possible to accomplish four times the work with only a 20% performance impact. This paper has also demonstrated that it is possible to achieve much less satisfactory performance for the same workload when different configuration choices are made. All in all it should be possible to achieve satisfactory performance of SAS Foundation in a virtualized environment if an organization is willing to make the required investments in the supporting IT infrastructure and ensure that the physical server hosting the SAS processes is not over committed in terms of memory or concurrent demand on the physical CPUs.

It should be noted that SAS software does not assert any requirements for virtualization and while it might be possible to achieve satisfactory performance from SAS Foundation executing in a virtual machine, the best performance of SAS on any given hardware configuration should be expected from SAS Foundation executing in an operating system installed directly on the physical server.

ACKNOWLEDGMENTS

This paper would not have been possible without the contributions of the following:

- Tim Braam wrote the initial benchmarking program that served as the basis of my testing.
- Daniel Zuniga gave me a copy of the benchmarking program.
- Tom Keefer provided a more comprehensive benchmarking suite as well as the benefit of his years of experience analyzing SAS performance in a variety of deployment architectures.
- Darrin Kerchner was a sounding board and offered his insights and expertise based on years of experience with one of the virtualization technologies used in this testing.
- Many members of the SAS Information Technology Services team provided support in terms of network configuration, hardware setup and configuration, and SAN configuration in support of these testing efforts.

APPENDIX A

CODE TO BUILD THE INITIAL DATASET:

```
do i = 1 to &numobs ;
  ind1=ceil(&usehs*ranuni(0)) ;
  set ftd.IEDITMAST(rename=(hs=hs10 sitc=sitc5 enduse=enduse5 naics=naics6))
point=ind1 ;
   hs2=substr(hs10,1,2) ;
   hs4=substr(hs10,1,4) ;
   hs6=substr(hs10,1,6) ;
   sitc4=substr(sitc5,1,4) ;
   sitc3=substr(sitc5,1,3) ;
   sitc2=substr(sitc5,1,2) ;
   sitc1=substr(sitc5,1,1) ;
   enduse1=substr(enduse5,1,1);
   naics5=substr(naics6,1,5) ;
   naics4=substr(naics6,1,4) ;
   naics3=substr(naics6,1,3) ;
   naics2=substr(naics6,1,2) ;
   ind2=ceil(&usecountry*ranuni(0)) ;
   country=countrycodearray(ind2);
   cntry_recode=countryrecodearray(ind2) ;
   cntry_name=countryarray(ind2) ;
   district=put(ceil(4*ranuni(0)),z2.) ;
   ind3=ceil(&usecity*ranuni(0)) ;
   city=citycodearray(ind3) ;
   city_name=cityalphaarray(ind3) ;
   statmoyr=put(today()-(floor(365*ranuni(0))), monyy5.);
   do j=1 to &numvvar ;
      exp=ceil(6*ranuni(0));
      numvars(j)=int(10**exp*(ranuni(0)));
   end ;
   QTY1=put(ceil(12*ranuni(0)),z2.);
   QTY2=put(ceil(50*ranuni(0)),z3.);
output ftd.ftd ;
```

```
end ;
stop;
run ;
```

CODE TO BUILD FORMATS:

```
/* create format for HS10 values */
proc sql noprint ;
  create table hsfmt as
    select '$HSLong' as fmtname, 'C' as type, HS as start, Lalpha as label,
           48 as max
    from ftd.IEDITMAST ;
quit ;
proc format cntlin=hsfmt lib=work;
run ;
```

CODE TO SORT THE DATA:

```
proc sort data=ftd.ftd out=ftd0(compress=yes);
   by HS10 HS6 HS4 HS2 country district QTY1 QTY2 statmoyr ;
run;
```

CODE TO SUMMARIZE THE DATA:

```
proc summary data = ftd.ftd noprint nway ;
   class hs10 hs6 hs4 hs2  country district qty1 qty2 statmoyr ;
   var _NUMERIC_ ;
   output out=hsis(compress=yes drop = _type_ _freq_) sum= ;
run ;
proc summary data =hsis noprint nway ;
   class hs10 qty1 qty2 statmoyr ;
   var _numeric_ ;
   output out=h10statmoyr (drop=_type_ _freq_ compress=yes) sum=;
run ;
proc summary data = ftd.ftd noprint nway;
   class naics6 naics5 naics4 naics3 naics2 country qty1 qty2 ;
   var _NUMERIC_ ;
   output out=hsnaics(compress=yes drop = _type_ _freq_) sum= ;
run ;
proc summary data = ftd.ftd noprint nway;
   class sitc5 sitc4 sitc3 sitc2 sitc1 country qty1 qty2 ;
   var _NUMERIC_ ;
   output out=hssitc(compress=yes drop = _type_ _freq_) sum= ;
run ;
proc summary data = ftd.ftd noprint nway;
   class  statmoyr ;
   var _NUMERIC_ ;
   output out=hsmoyr(compress=yes drop = _type_ _freq_) sum= ;
run ;
```

CODE TO WRITE DATA OUT TO A FILE

```
data _NULL_ ;
   file "test.txt" lrecl=%eval(&numvvar*8 + 32) ;
   set ftd.ftd;
   put hs10 $10. cntry_name $15. district $2. qty1 $2. qty2 $3. statmoyr
       var1-var&numvvar;
run;
```

CODE TO READ THE FILE INTO A DATASET

```
data ftd_one;
   infile "test.txt" lrecl=%eval(&numvvar*8 + 32) truncover ;
   input hs10 $10. cntry_name $15. district $2. qty1 $2. qty2$3.
         statmoyr monyy5. var1-var&numvvar;
   format hs10 $hslong48. statmoyr mmddyy10. ;
run;
```

CODE TO DELETE THE FILE:

```
%if %index(&SYSSCP,WIN) %then %do ;
options noxwait noxsync ;
data _NULL_ ;
   x "del test.txt" ;
run ;
%end ;
%else %do ;
data _NULL_ ;
   call system("rm test.txt") ;
run ;
%end ;
```

CODE TO CREATE INDICES:

```
proc datasets library=ftd memtype=data nolist;
   modify ftd;
   index create HS10;
   index create HS6;
   index create HS4;
   index create HS2;
   index create country;
   index create district;
run;
quit;
```

CODE FOR THE CPU INTENSIVE OPERATION:

```
data _NULL_ ;
   do j=1 to 1000 ;
      do k=1 to 1000 ;
        do m=1 to 1000 ;
        end ;
      end;
   end ;
 run ;
```

CONTACT INFORMATION

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

Ken Gahagan

SAS Campus Drive
SAS Institute Inc.

E-mail: Ken.Gahagan@sas.com