Paper 289-2014

SAS® Grid Manager, SAS® Visual Analytics, and SAS® High-Performance Analytics: Sharing Hardware and More

Ken Gahagan, SAS Institute Inc., Cary, NC

ABSTRACT

There are exciting new capabilities available from SAS® High-Performance Analytics and SAS® Visual Analytics. Current customers seek a deployment strategy that enables gradual migration to the new technologies. Such a strategy would mitigate the need for "rip and replace" and would enable resource utilization to evolve along a continuum rather than partitioning resources, which would result in underused computing or storage hardware. New customers who deploy a combination of SAS® Grid Manager, SAS High-Performance Analytics, and SAS Visual Analytics seek to reduce the cost of computing resources and reduce data duplication and data movement by deploying these solutions on the same pool of hardware. When sharing hardware, it is important to implement resource management in order to help guarantee that resources are available for critical applications and processes. This paper discusses various methods for managing hardware resources in a multi-application environment. Specific strategies are suggested, along with implementation suggestions.

INTRODUCTION

SAS Grid Manager provides robust scheduling and workload management capabilities, support for high availability, as well as capabilities for customers to construct parallel job flows distributing work across the grid to reduce execution time for these jobs. SAS Visual Analytics and SAS High-Performance Analytics are engineered to deliver the information that you need as quickly as possible. Functionality from each of these products can place extreme demands on the underlying hardware. The simple approach of dedicating hardware to each function could lead to increased hardware costs and frequently underutilized hardware, increased costs related to managing the environment, and data replication. Sharing hardware can mitigate some of these concerns but can lead to other considerations. Customers also desire a clearly understandable adoption path that builds upon their current investment in SAS Grid Manager—particularly as customers begin to leverage SAS high-performance analytical procedures from within SAS Grid Manager jobs.

This paper shares the results of SAS internal proof points running all three solutions within the same hardware environment. This paper will

- demonstrate the need for monitoring and capacity planning and offer some suggestions in this area
- provide high-level descriptions of the workloads used in the internal proof and stress exercises
- document findings of various approaches to managing workload
- provide references for system administrators responsible for such an environment

This paper is not a tutorial on Linux system administration nor is it a “how-to” reference for Linux control groups. Control groups are presented as an option for managing workload, but this paper is not an in-depth reference. The reader is referred to excellent documentation provided by Red Hat should capabilities be needed beyond those offered by the SAS High-Performance Management Console and other SAS configuration options.

GOALS

- Deploy SAS Grid Manager, SAS Visual Analytics, and SAS High-Performance Analytics on shared hardware.
- Manage workloads in such a way that
  - All three workloads can execute concurrently.
  - Response times for SAS Visual Analytics are protected and the risk of spikes in response times is mitigated.
  - SAS High-Performance Analytics procedures are given higher priority than SAS Grid Manager jobs.
  - SAS Grid Manager jobs get the least amount of resources when SAS Visual Analytics or SAS High-Performance Analytics workloads are executing—yet are not unduly constrained when they are the only workload running in the environment.
- Understand the behavior of each workload independently.
- Understand how behavior changes when workloads are executed concurrently.
- Evaluate the use of NICE as a means of prioritizing workloads, and then document the results and limits.

- Evaluate the use of Linux Control Groups (CGROUPs) as a means of prioritizing the utilization of compute resources among the workloads, and then document the results and limits.
- Share findings and recommendations.

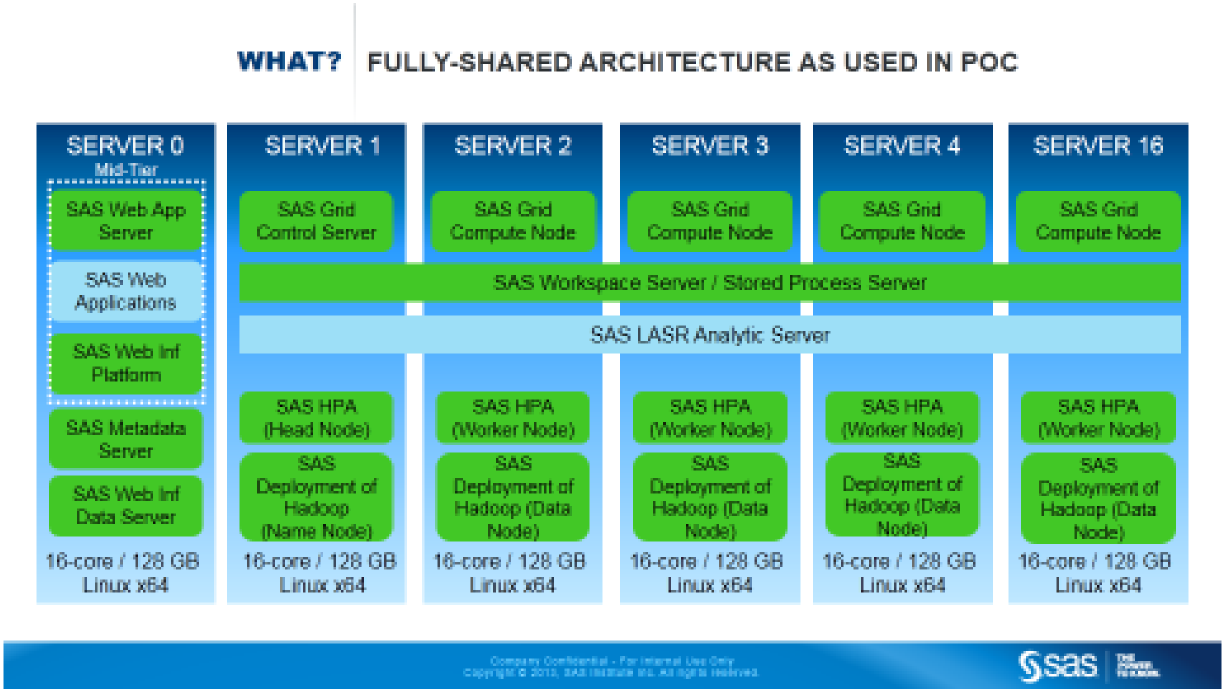
TO SHARE OR NOT TO SHARE—IS THAT THE QUESTION?

There are many “how-to” papers that begin with a statement similar to “Sure you can do this—with caveats—but should you…?” It is tempting to begin this paper the same way. This paper covers a number of caveats, but frequently the larger picture drives an implementation choice. This paper begins with the assumption that sharing the hardware is a requirement, and then explores various configuration choices, tradeoffs, and the impacts of those choices. The good news is that by design all three technologies are scale-out technologies so, in general, when an on-going demand for resources exceeds the environmental capacity, the answer is to grow the environment by adding more nodes. This is true of each product that is implemented stand-alone as well. Sharing hardware offers the possibility of lowering the total cost of ownership by sharing resources to increase overall utilization throughout the business day, rather than sizing three discrete environments to meet their respective peak demands, which might be underutilized during part of the business day. For example, SAS Visual Analytics might consume more resources during business hours and SAS Grid Manager might consume more resources during non-business hours. Data mining activities during the day might leverage SAS high-performance analytical procedures and those same procedures might be utilized by scheduled SAS jobs that are launched by SAS Grid Manager during off-peak hours. Sharing hardware means that the overall utilization is higher than having dedicated environments with low utilization during part of a 24-hour period. This approach also facilitates sharing of a single SAS Metadata Server and mid-tier components.

METHODOLOGY

Figure 1 provides an overview of the hardware utilization for the SAS proof of concept (POC) and stress test. SAS Grid Manager requires a shared file system and that requirement does not change with the addition of SAS Visual Analytics and/or SAS High-Performance Analytics.

- SERVER0 hosts the SAS Metadata Server and the mid-tier components.
- SERVER1–SERVER16 hosts SAS Grid Manager jobs. This is also where the SAS high-performance analytical procedures execute as well as the hardware resources that host the SAS LASR Analytic Server that provides the analytics engine for SAS Visual Analytics.
- Shared storage is not depicted in the following diagram, but shared storage was used for GRIDWORK, SASDATA, and SASWORK.


WHAT? FULLY-SHARED ARCHITECTURE AS USED IN POC

SERVER 0 Mid-Tier: SAS Web App Server; SAS Web Applications; SAS Web Inf Platform; SAS Metadata Server; SAS Web Inf Data Server; 16-core / 128 GB Linux x64

SERVER 1: SAS Grid Control Server; SAS HPA (Head Node); SAS Deployment of Hadoop (Name Node); 16-core / 128 GB Linux x64

SERVER 2: SAS Grid Compute Node; SAS HPA (Worker Node); SAS Deployment of Hadoop (Data Node); 16-core / 128 GB Linux x64

SERVER 3: SAS Grid Compute Node; SAS HPA (Worker Node); SAS Deployment of Hadoop (Data Node); 16-core / 128 GB Linux x64

SERVER 4: SAS Grid Compute Node; SAS HPA (Worker Node); SAS Deployment of Hadoop (Data Node); 16-core / 128 GB Linux x64

SERVER 16: SAS Grid Compute Node; SAS HPA (Worker Node); SAS Deployment of Hadoop (Data Node); 16-core / 128 GB Linux x64

Servers 1–16: SAS Workspace Server / Stored Process Server; SAS LASR Analytic Server

Company Confidential - For Internal Use Only
Copyright © 2013, SAS Institute Inc. All rights reserved.

Figure 1: Logical Deployment Topology for Stress Testing

BASELINE EACH WORKLOAD INDEPENDENTLY

There is a cart and horse problem for an effort such as this…. Does one start with a workload and then size an environment to the workload, or do constraints mandate the available hardware and then, in order to be successful, workloads must be sized such that the various workloads have any chance of co-existing on the hardware and achieving reasonable results in terms of performance? For this exercise, the available hardware was the constraint. It is important to state the obvious at the outset: If the overall workload executing within the environment at a given time exceeds the capabilities of the hardware, then execution times will lengthen. We see this reality play out not only in computing but also on our roadways. Commutes at rush hour take longer. In the traffic world, throughput management strategies such as HOV lanes allow one class of traveler a more timely commute—but a dedicated HOV lane does nothing for all the other vehicles on the road beyond removing some vehicles from the non-HOV lanes. Thankfully resource management in the Linux world is more adaptive. When prioritized workloads are present, they benefit from preferential treatment; but, when only non-priority workloads are present, all resources are available to those workloads unless or until a priority workload is launched.

POC WORKLOAD DESCRIPTIONS

SAS Grid Manager Workload

In this POC, the SAS Grid Manager workload is constructed using a DATA step, PROC FORMAT, PROC SUMMARY, and PROC DATASETS to create indices and input and output of delimited files. These capabilities are frequently used and are particularly useful for ETL and information exchange processes. Each compute node has 128GB of RAM, which provides a large file cache. Therefore, when run in isolation this workload tends to be write heavy with respect to storage. Based on experimentation, it was determined that running 32 concurrent jobs across the 16 nodes was sufficient to saturate the bandwidth available to the iSCSI SAN that hosts the shared file systems. Scheduling more than 32 jobs did not make sense because additional jobs caused elongated execution times due to IO contention among the various jobs that comprise this workload alone. Each SAS job in this workload is constrained to MEMSIZE of 2G. SAS indicated in the logs for these jobs that performance could be enhanced by increasing MEMSIZE or SORTSIZE to reduce writes to disk. In the interest of managing memory and demonstrating management of IO bandwidth for this paper, I did nothing based on this information. This highlights the fact that workload management and capacity planning are exercises in tradeoffs. Your choices will vary based on the needs of your organization. Figure 2 shows the average CPU utilization across the grid nodes for a 2-hour baseline of this workload.

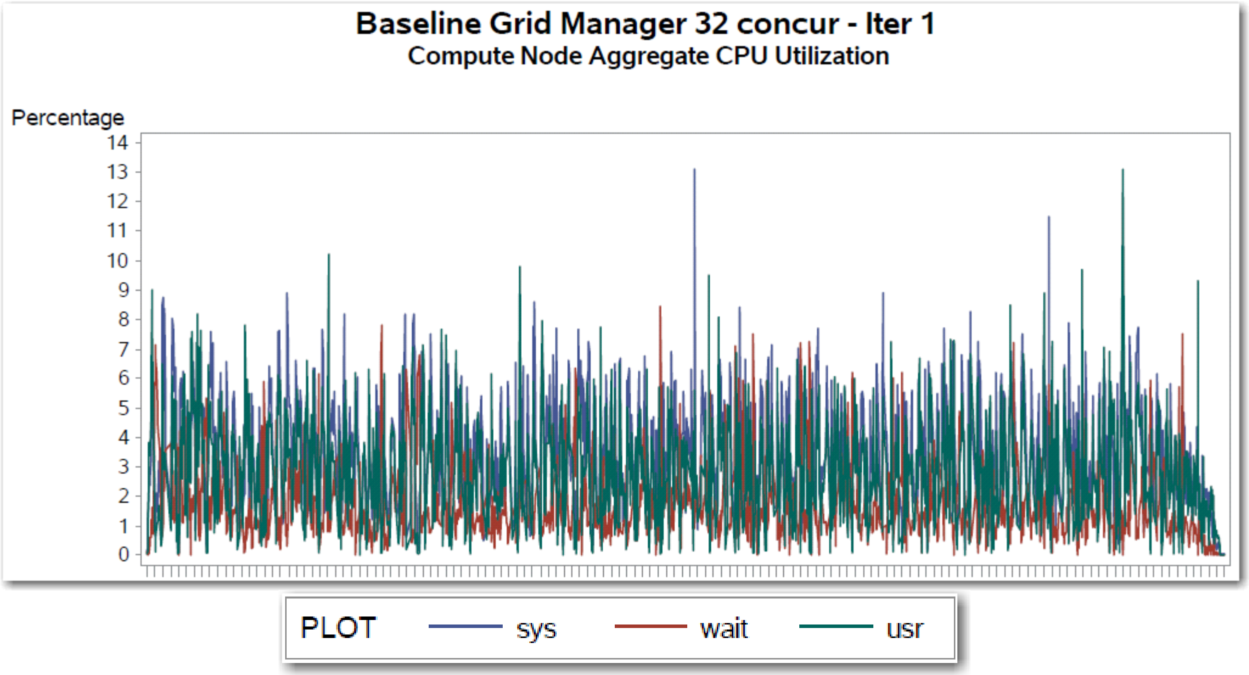


Figure 2: CPU Utilization for SAS Grid Manager Baseline

Figure 3 depicts the disk write activity in KB/s for the SAS Grid Manager baseline alone.

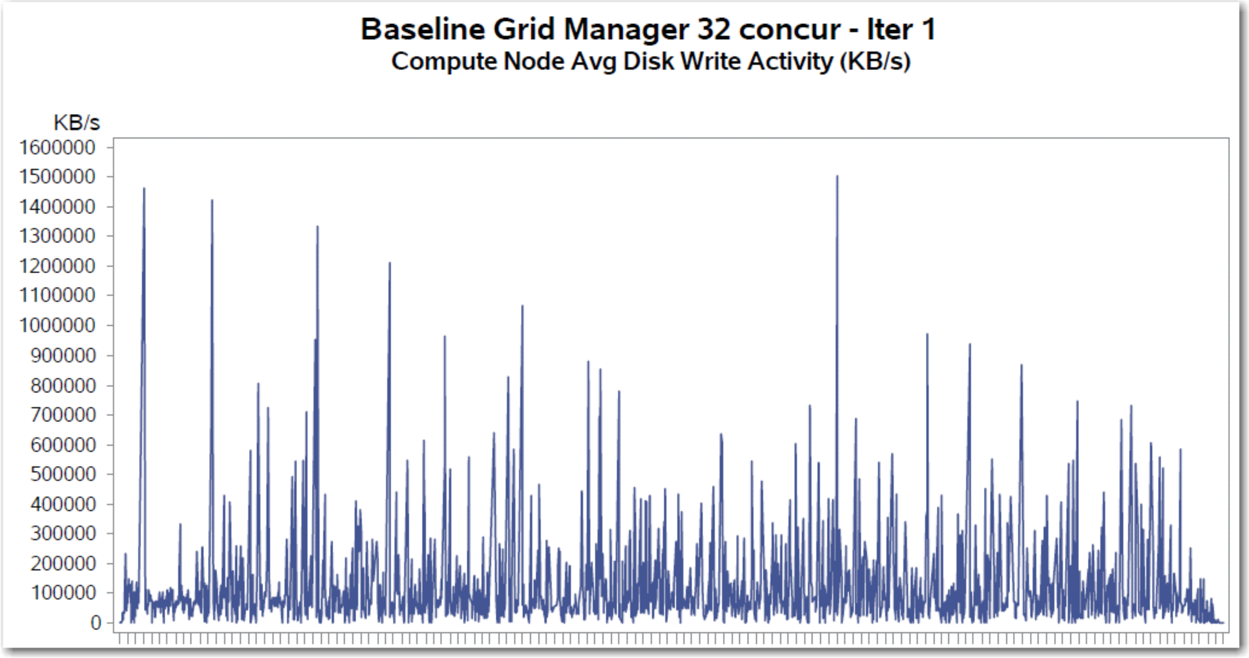


Figure 3: SAS Grid Manager Baseline—Disk Write Activity

Figure 4 depicts the disk read activity that is associated with the SAS Grid Manager baseline.

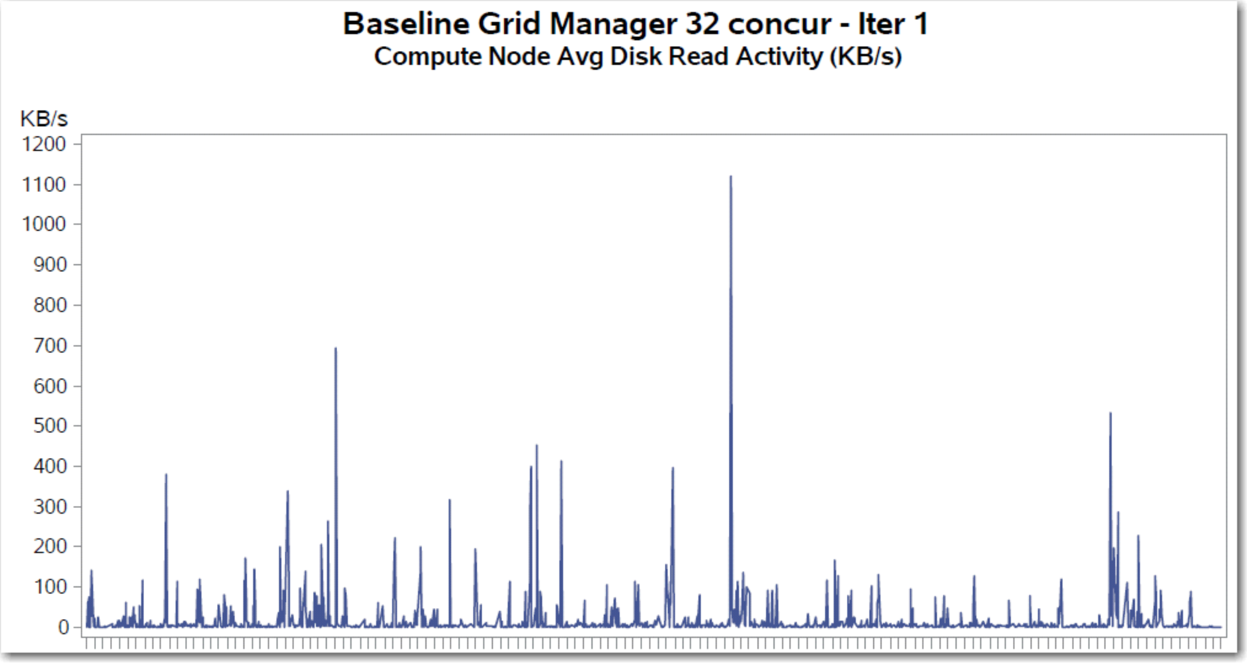


Figure 4: SAS Grid Manager Baseline—Disk Read Activity

SAS high-performance analytical procedures Workload

This workload consisted of multiple job streams each submitting a High-Performance PROC one after the other. The procedures used in this workload were HPNEURAL, HPLOGISTIC, HPSAMPLE, HPMDB, HPREDUCE, and HPTREE. The performance statement in all invocations indicated that the procedures should use all nodes. Based on average CPU utilization of the nodes when running this workload, five concurrent job streams were selected as a workload that was demanding enough that the CPU utilization would need to be prioritized between SAS high-performance analytical procedures and SAS Visual Analytics, and IO bandwidth would need to be prioritized between the SAS Grid Manager jobs and the SAS High-Performance Analytics workload. Figure 5 depicts the CPU utilization of the SAS high-performance analytical workload baseline.

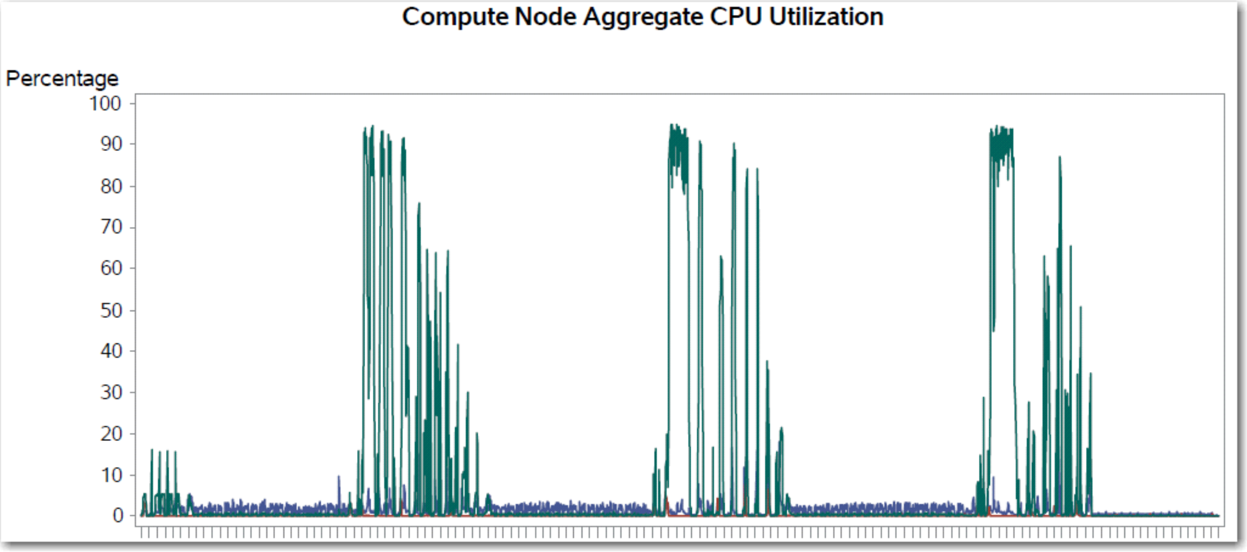


Figure 5: SAS high-performance analytical procedures Workload CPU Utilization

Figure 6 depicts the memory utilization for the SAS high-performance analytical procedure workload that was used in the stress test.

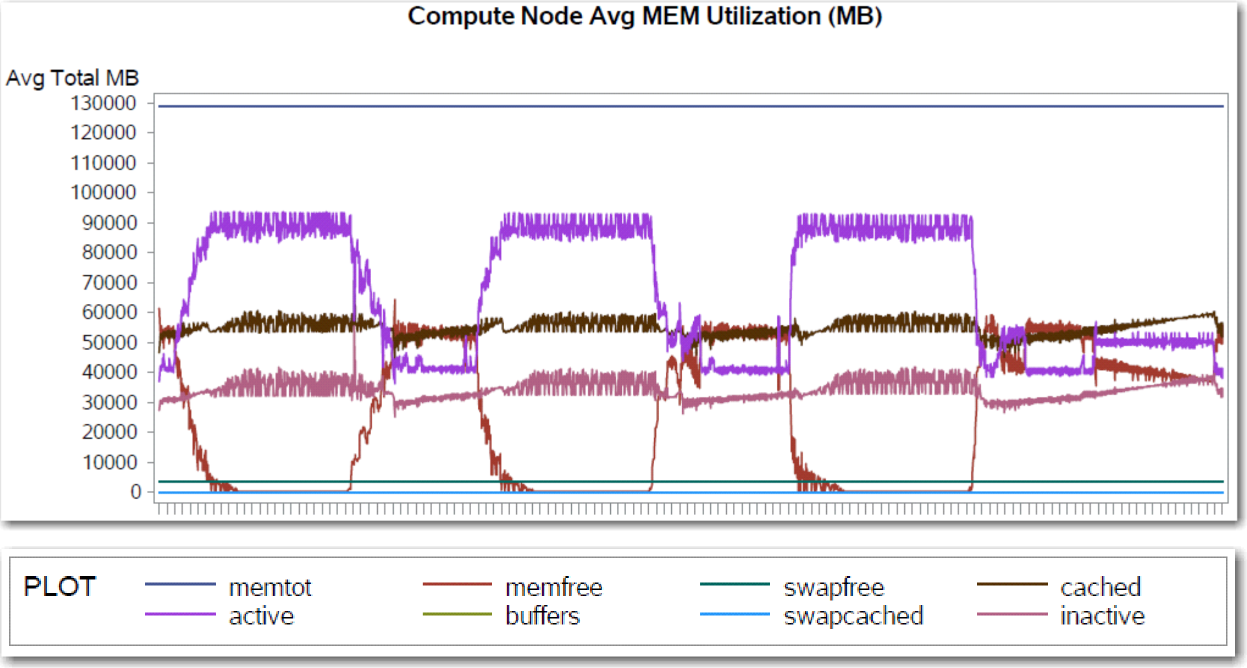


Figure 6: SAS high-performance analytical procedure Workload Memory Utilization

Figure 7 depicts the disk write activity for the SAS high-performance analytical procedure workload that was used in the stress test.

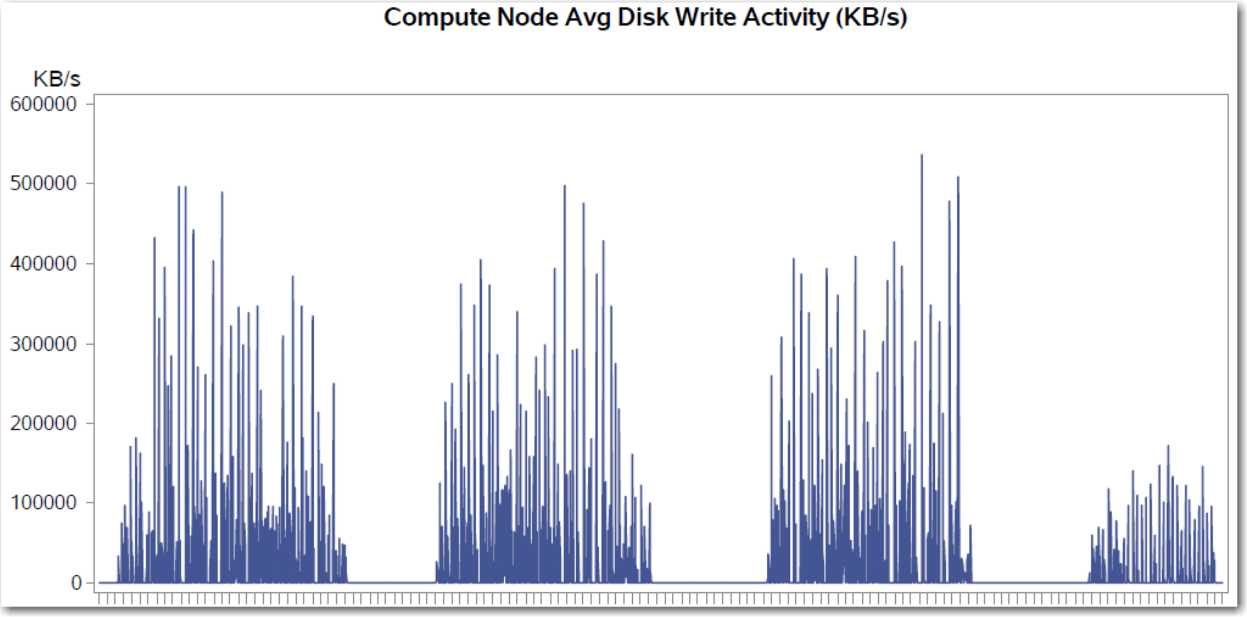


Figure 7: SAS high-performance analytical procedures Workload Disk Write Activity

SAS Visual Analytics Workload

This workload simulates a number of users of the SAS Visual Analytic Explorer using the web interface to visualize data. The size of the underlying data and the number of users was selected at a point that placed significant demand on the CPUs but still allowed some headroom for processing SAS High-Performance Analytics and SAS Grid Manager workloads. The SAS Visual Analytics workload simulates 54 concurrent users creating a variety of reports including correlation reports, line charts, and bar charts; and includes some forecasting. The source file is ~38 GB on disk and consists of more than 139 million observations. The CPU utilization for a 2-hour execution of this scenario follows:

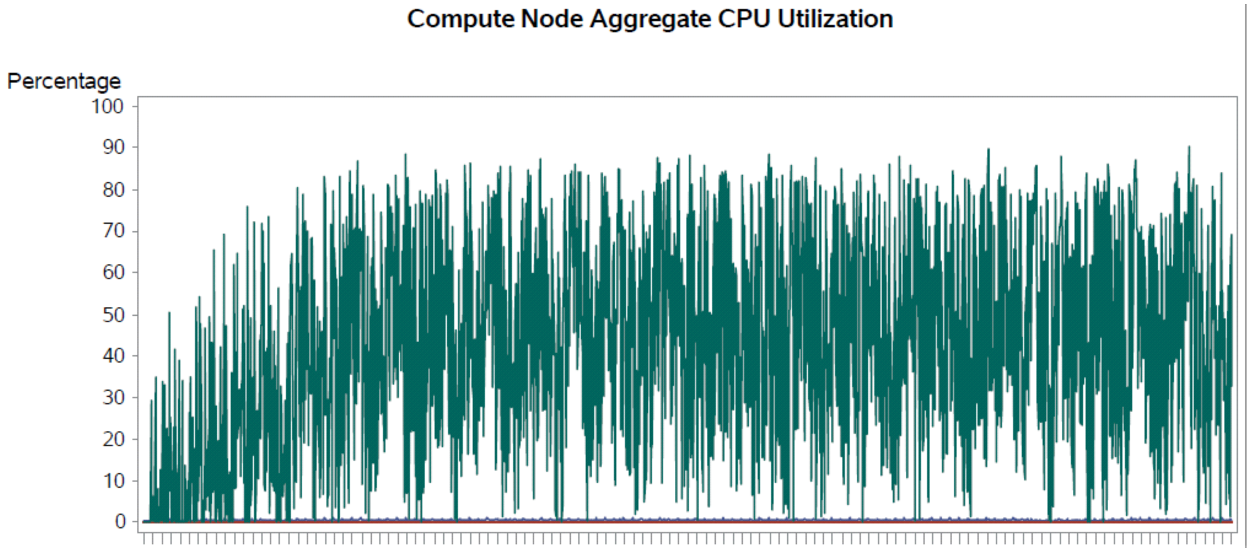


Figure 8: SAS Visual Analytics Workload CPU Utilization

Figure 9 depicts the memory utilization of the SAS Visual Analytics workload baseline.

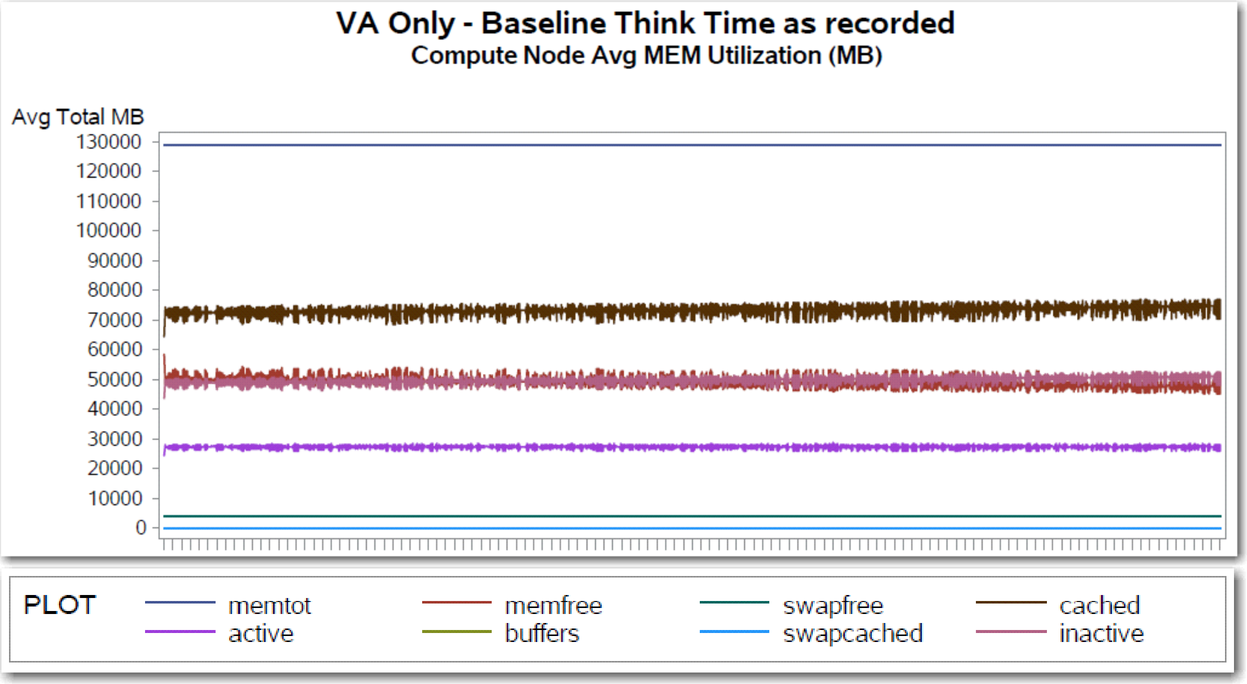


Figure 9: SAS Visual Analytics Workload Memory Utilization

Figure 10 depicts the SAS Visual Analytics transaction baseline response times.

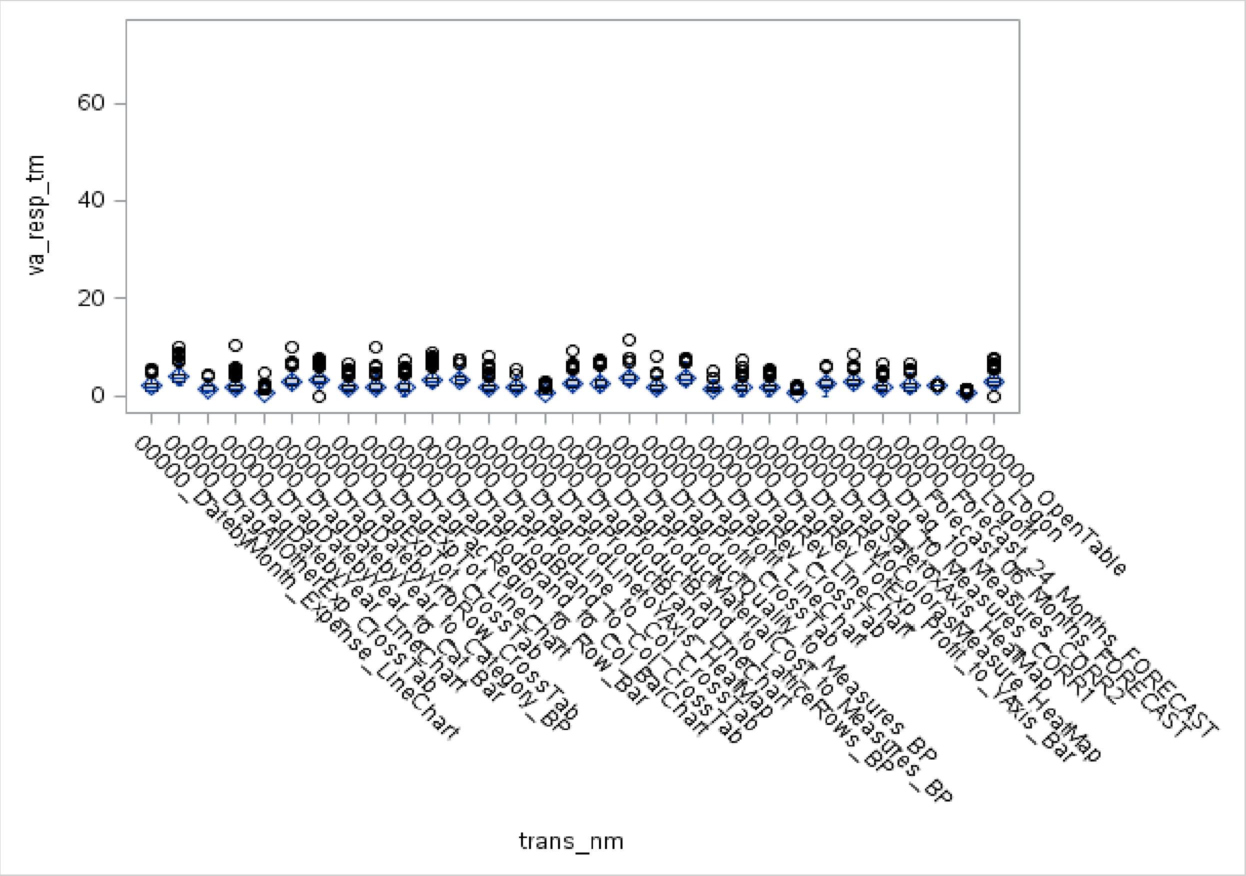


Figure 10: SAS Visual Analytics Workload Transaction Response Time Baseline

MIXING WORKLOADS

Because the LASR server that backs SAS Visual Analytics is an in-memory server, there was no IO attributable to this scenario.

Given this limited amount of information describing the resource utilization of each workload independently, we can see that our primary resource management needs are going to be

- moderating CPU utilization between SAS Visual Analytics and SAS High-Performance Analytics with the goal being to protect SAS Visual Analytics response times
- moderating IO utilization between SAS High-Performance Analytics and SAS Grid Manager

MIXING SAS VISUAL ANALYTICS AND SAS HIGH-PERFORMANCE ANALYTICS

Figure 12 depicts the transaction response times for the SAS Visual Analytics workload when running the SAS Visual Analytics and SAS High-Performance Analytics workloads concurrently without any resource management.

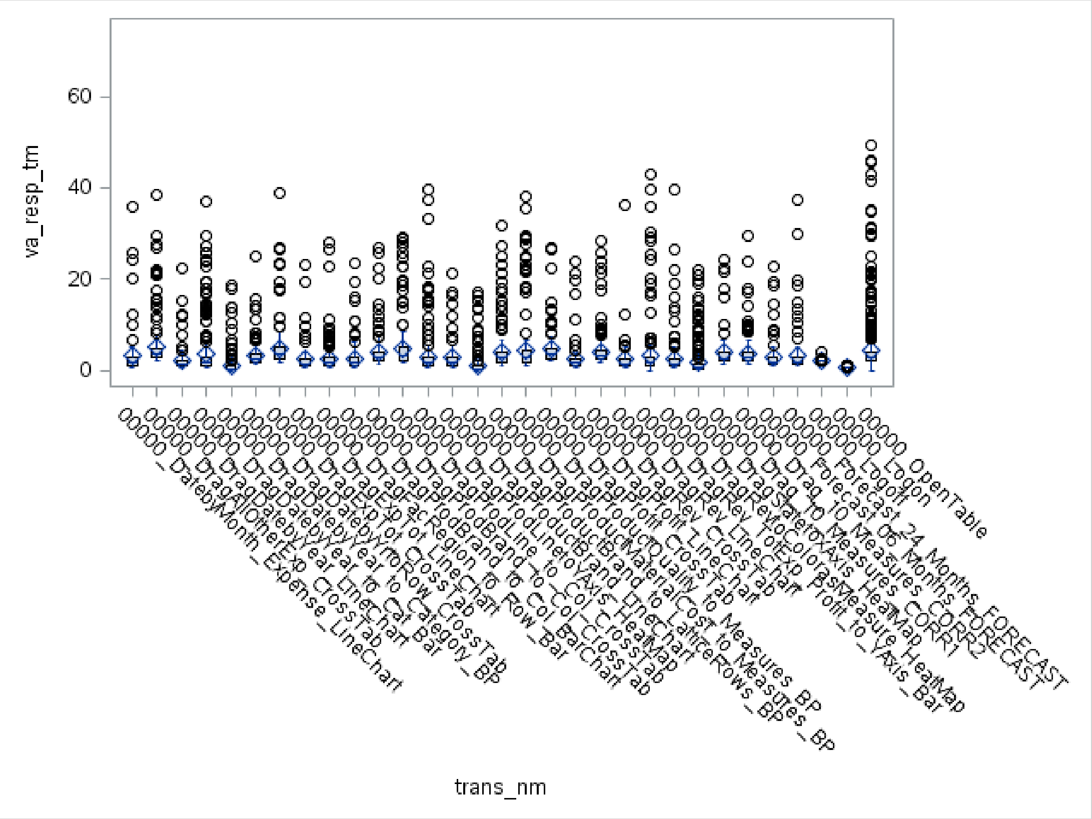


Figure 11: SAS Visual Analytics Response Times—SAS Visual Analytics and SAS High-Performance Analytics Concurrent—No Workload Management

Our maximum response time has gone from 12 seconds (baseline) to 50 seconds with no workload management. Figure 12 shows an overlay of response times and aggregate CPU utilization across the grid nodes.

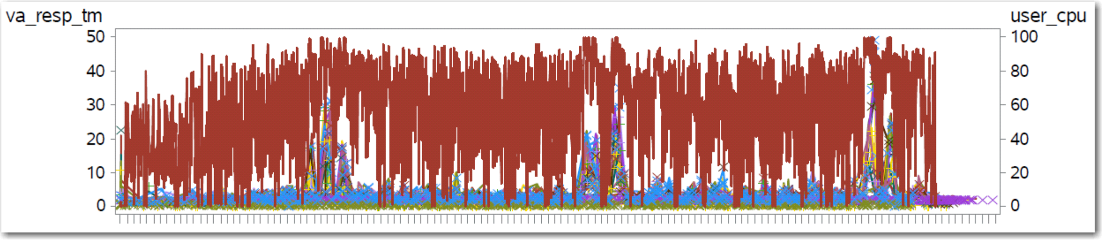


Figure 12: SAS Visual Analytics Response Times CPU Utilization Overlay SAS Visual Analytics and SAS High-Performance Analytics—No Resource Management

The spikes in response times are highly correlated to periods of high CPU utilization. When the SAS High-Performance Analytics workloads were "NICEed" to a value of 8, the spikes in response time were reduced but not as far as I would have liked. Using a NICE value of 12 for the SAS High-Performance Analytics workload, however, resulted in much better maximum response times:

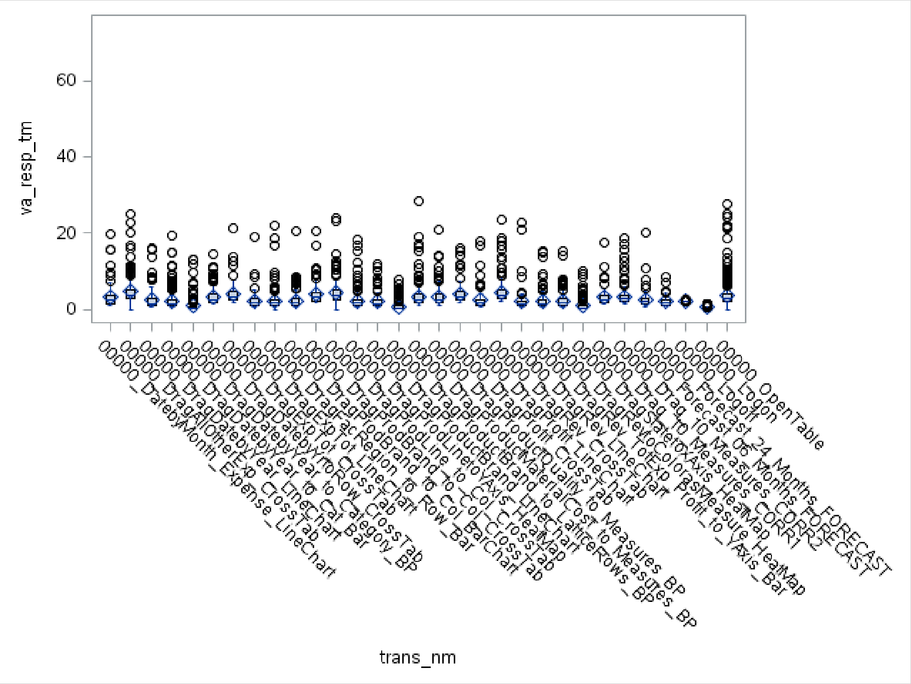


Figure 13: SAS Visual Analytics Response Times Running SAS Visual Analytics and SAS High-Performance Analytics with SAS High-Performance Analytics "NICEed" to 12

Using CGROUPS to manage CPU resources (85% to SAS Visual Analytics and 15% to SAS High-Performance Analytics) provided even better results as seen in Figure 14.

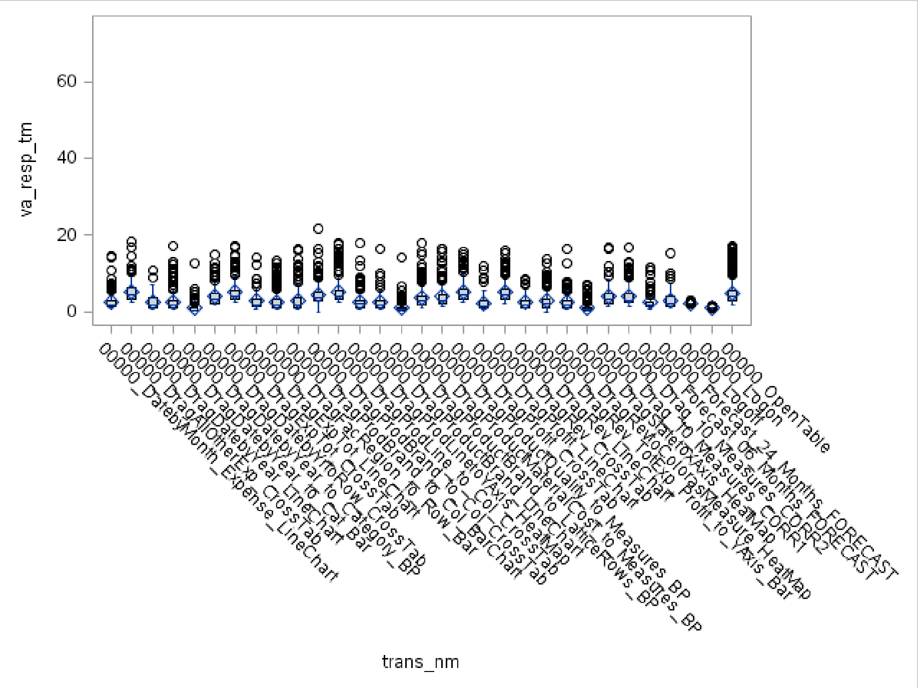


Figure 14: SAS Visual Analytics Response Times Running SAS Visual Analytics and SAS High-Performance Analytics CGROUP Management 85% CPU to SAS Visual Analytics / 15% CPU to SAS High-Performance Analytics

MIXING SAS HIGH-PERFORMANCE ANALYTICS AND SAS GRID MANAGER WORKLOADS

Figure 15 depicts the SAS High-Performance Analytics workload baseline execution times (SAS High-Performance Analytics running alone in the environment).

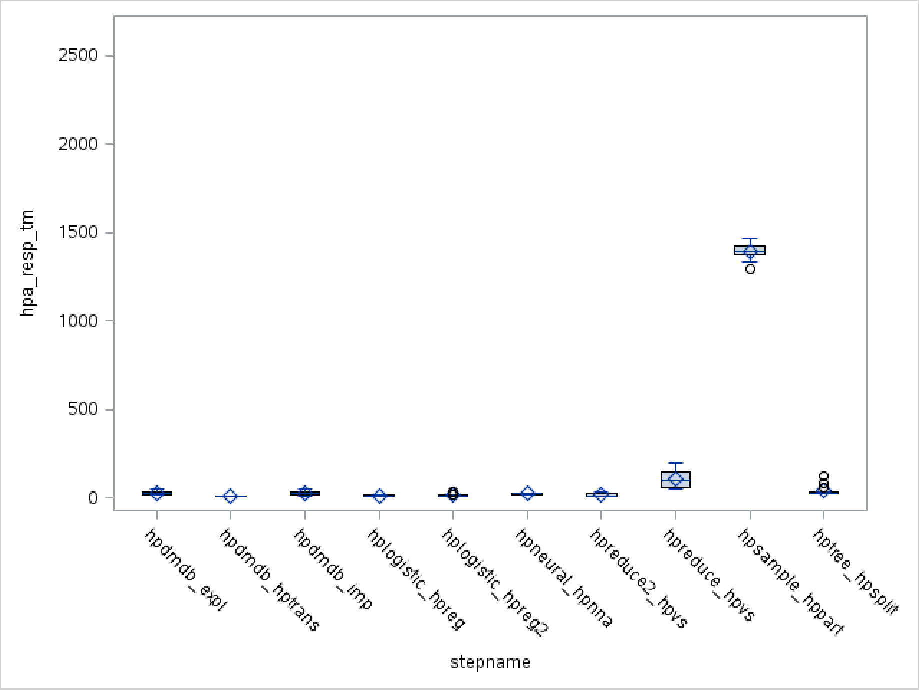


Figure 15: SAS High-Performance Analytics Procedure Baseline Execution Times

The default configuration for SAS Grid Manager is to “NICE” jobs submitted to the NORMAL queue to a value of 20. On Linux, this results in jobs running with a NICE value of 19. Figure 16 depicts the response times for the HPA PROCs when the HPA and SAS Grid Manager workloads are run together with no additional resource management.

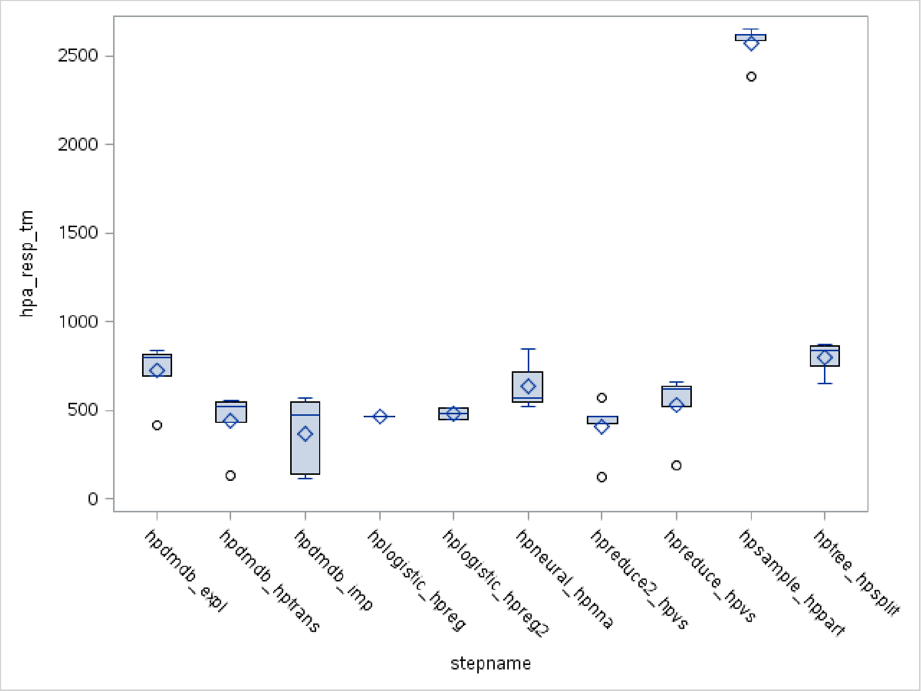


Figure 16: SAS High-Performance Analytics Procedure Execution Times When Run with SAS Grid Manager

Workload (No resource management other than the default NICE value of the normal queue.)

Figure 17 shows that when run with CGROUPs managing blkio with at 10:1 weighting in favor of SAS High-Performance Analytics, the SAS High-Performance Analytics execution times come back in line:

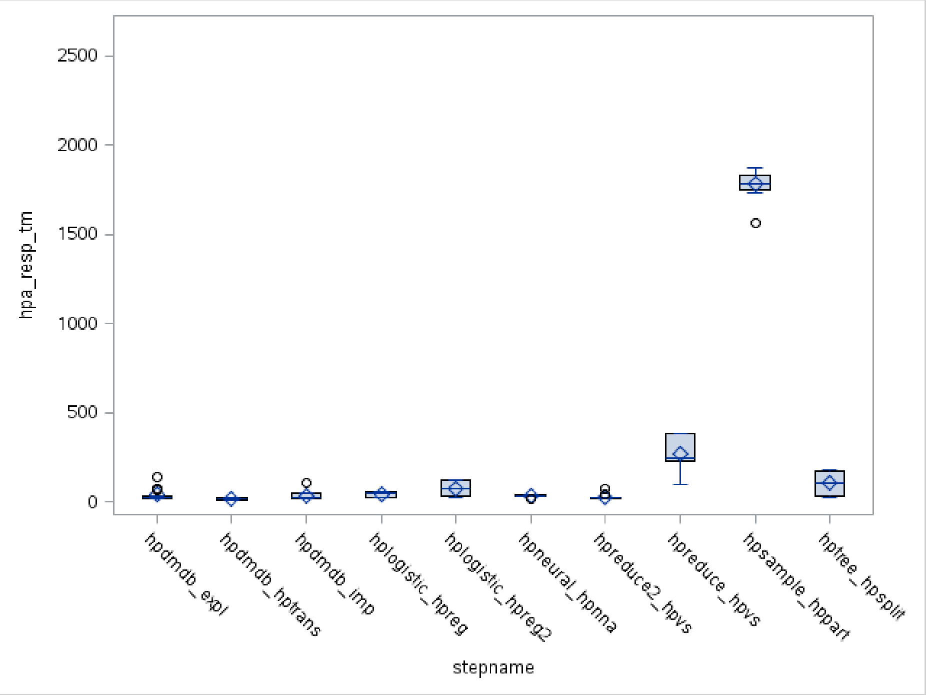


Figure 17: SAS High-Performance Analytics Procedure Execution Times When Run with SAS Grid Manager Workload and CGROUP Control

MIXING SAS VISUAL ANALYTICS, SAS HIGH-PERFORMANCE ANALYTICS PROCEDURES, AND SAS GRID MANAGER WORKLOADS

When running SAS Visual Analytics and SAS High-Performance Analytic PROCs the primary point of contention is CPU utilization. When running SAS Grid Manager and SAS High-Performance Analytics prodecure workloads, the primary point of contention is block IO resources. Managing any two of these workloads is essentially a one-dimensional problem. For moderating CPU utilization between SAS Visual Analytics and SAS High-Performance Analytics, NICE values might be sufficient to meet your needs. If, however, you are going to run all three workloads, then the management capabilities of CGROUPs will be required. In the final design, I chose a CGROUP configuration as follows:

CGROUP NAME	CPU.SHARES	BLKIO.WEIGHT
CGLASR	980	n/a
CGHPA	10	1000
CGGRID	10	100

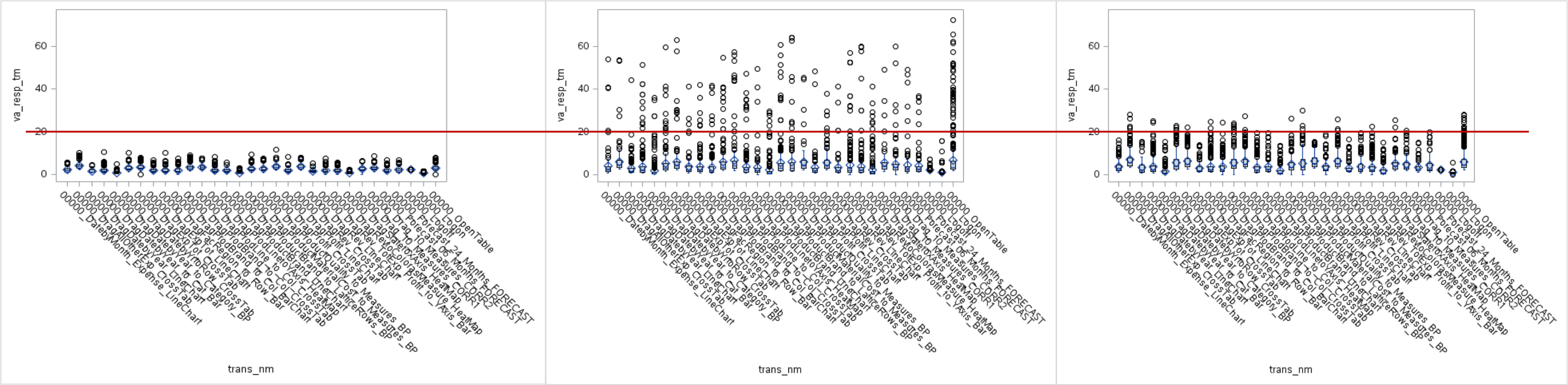
CGROUPs also allow management of memory. However, there is a concern currently with the fact that Linux provides only two options when a CGROUP is going to exceed the amount of memory for which it is entitled. The process can “freeze” until more memory becomes available in the CGROUP, or the process can be terminated. Neither of these options is a good option for SAS products at this time, and SAS has engaged the Linux community in hopes of a third option being enabled that will allow applications to receive an out-of-memory indication and react accordingly such as returning a message to the user rather than simply freezing or failing. For now, SAS strongly recommends that CGROUP memory management not be used with CGROUPs that manage SAS processes.

The full listing of the /etc/cgconfig.conf file from the POC is located in Appendix A of this paper.

SAS Visual Analytics

Figure 18 depicts the response times for SAS Visual Analytics baseline when run in conjunction with SAS High-Performance Analytics and SAS Grid Manager with default resource management, and when run with the CGROUP definition in Appendix A. The red line across the graphs is simply a visual aid to facilitate comparison of response times across the graphs.

Figures 18: SAS Visual Analytics Response Times Baseline, Default Resource Management, and CGROUP Management



VA Transaction Response times—VA running in isolation.

VA Transaction Response times – VA running with HPA and Grid jobs—No resource management.

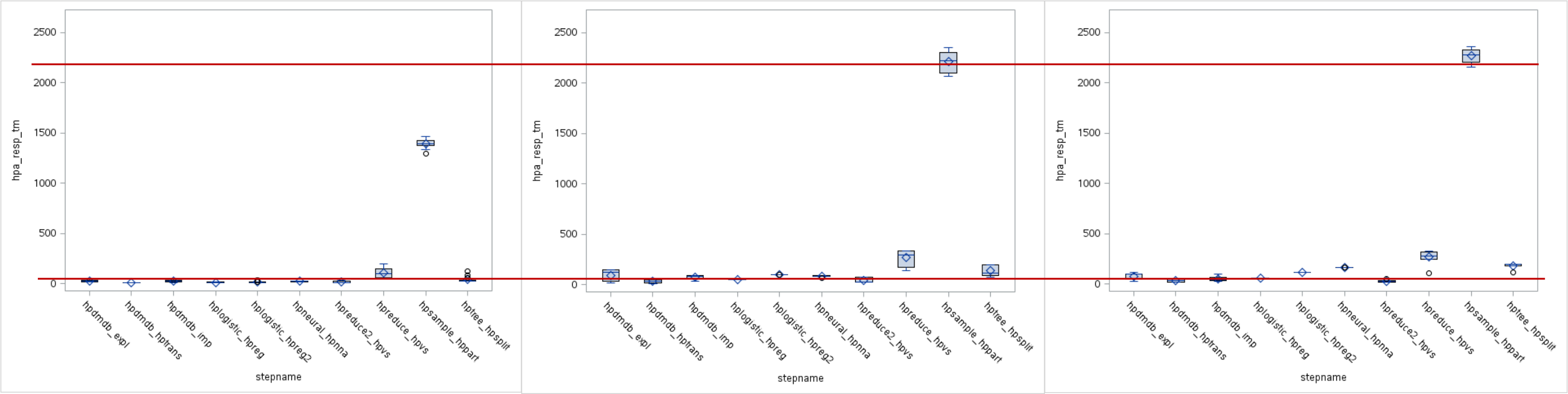
VA Transaction Response times—VA running with HPA and Grid jobs—CGROUP management.

The graphs provide visual reinforcement of the ability to manage response times as well as to reinforce the fact that for the amount of hardware in the environment if the desire is to bring overall SAS Visual Analytics response times down further it might be required that some workload be removed from the environment—or that more compute resources be added to the environment. The average response times in any of the scenarios is still quite good—on the order of a few seconds to compute and render the reports on a 139+ million row table. However, the spikes in response times can lead to an unsatisfactory user experience. Each organization has a different tolerance for variable response times and different budgets to provide consistency. The value equation will differ from one organization to the next.

SAS High-Performance Analytics

Figure 19 depicts the execution times for the SAS High-Performance Analytic PROCs (baseline, when run in conjunction with SAS Visual Analytics and SAS Grid Manager with default resource management and with the CGROUP definition in Appendix A). The red lines across the graphs are a visual aid to facilitate comparison of execution times across the graphs.

Figure 19:SAS High-Performance Analytics PROC Execution Times Baseline, Default Resource Management, and CGROUP Management



HPA PROC execution time summary—HPA PROCs running in isolation.

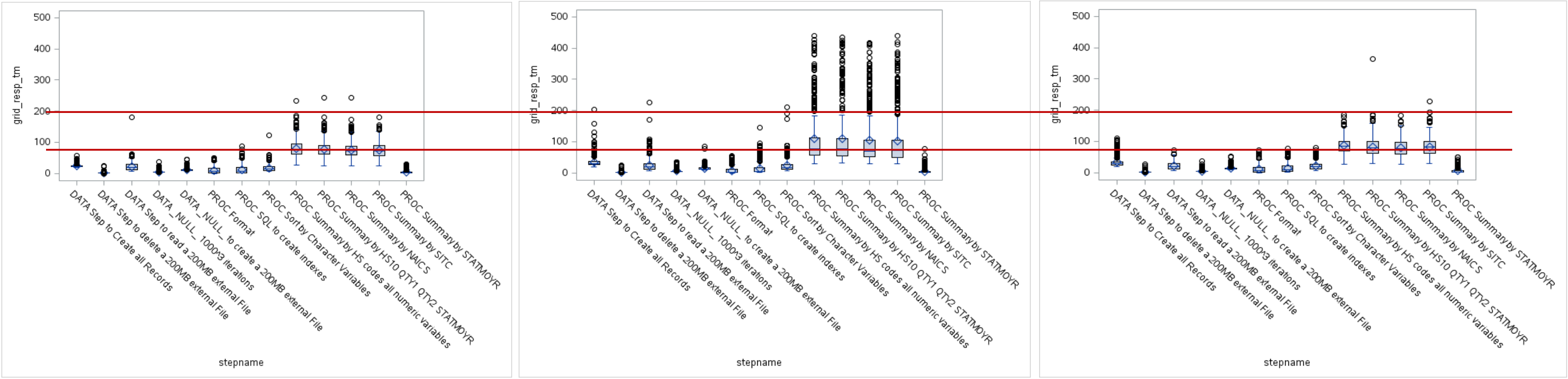
HPA PROC execution time summary—VA running with HPA and Grid jobs—No resource management.

HPA PROC execution time summary—VA running with HPA and Grid jobs—CGROUP management.

SAS Grid Manager

Figure 20 depicts the execution times for the steps in the SAS Grid Manager jobs (baseline, when run in conjunction with SAS Visual Analytics and SAS High-Performance Analytics with default resource management, and with the CGROUP definition in Appendix A). The red lines are a visual aid to facilitate comparison of execution times across the graphs.

Figure 20: SAS Grid Manager Step Execution Times Baseline, Default Resource Management, and CGROUP Management



SAS Grid job step execution time summary—SAS Grid Jobs running in isolation.

SAS Grid job step execution time summary—VA running with HPA and Grid jobs—No resource management.

SAS Grid job step execution time summary—VA running with HPA and Grid jobs—CGROUP management.

HOW TO ENABLE RESOURCE MANAGEMENT

TO USE CGROUPS WITH SAS GRID MANAGER

To place SAS Grid Manager jobs into a control group, modify the <config root>/Lev1/SASApp/GridServer/grid_usermods.cfg file similar to the highlighted line below:

```
# /----------------------------------------------------------------------------\
# |                                                                            |
# | This file sets additional properties used by the sasgrid script.           |
# | Currently you can define the following variables:                          |
# |                                                                            |
# |   SASUSERCONNECTOPTS - additional options to specify on the SAS command    |
# |                        line when the SAS/CONNECT server is started         |
# |   SASWARNINGSNOERROR - if set to anything, will change a return code of 1  |
# |                        (indicating SAS completed with warnings) to a       |
# |                        return code of 0 so jobs appear to complete OK      |
# |   SASSTAGECMDOPTS    - additional options to specify on the command        |
# |                        used to stage files                                 |
# |                                                                            |
# \----------------------------------------------------------------------------/

# SASUSERCONNECTOPTS=???
SASWARNINGSNOERROR=1
# SASSTAGECMDOPTS=???
cgclassify -g cpu,blkio:cgGRID $$
```

USE CGROUPS WITH SAS HIGH-PERFORMANCE ANALYTICS PROCEDURES

To place SAS High-Performance Analytics procedures in a CGROUP, modify the …/TKGrid/tkmpinodelib.sh script with a line similar to the one highlighted below:

```
#! /bin/sh
if [ -n "$TKMPI_UMASK" ]; then umask "$TKMPI_UMASK"; fi
if [ -n "$TKMPI_CORESIZE" ]; then
  ulimit -Sc $TKMPI_CORESIZE
else
  ulimit -Sc 0
fi
if [ -n "$TKMPI_ULIMIT" ]; then
  ulimit $TKMPI_ULIMIT
fi
cgclassify -g cpu,blkio:cgHPA $$
$TKMPI_DIR/bin/timed_run $TKMPI_MAXRUNTIME $TKMPI_NODELIB $*
```

USE CGROUPS WITH SAS VISUAL ANALTYICS

Both the SAS LASR Analytic Server procedures and the SAS High-Performance Analytics procedures are built on and require an installation of the SAS High-Performance Node Installation. To place the SAS LASR Analytic Server procedures in a different CGROUP than the SAS High-Performance Analytics procedures, install two copies of the SAS High-Performance Node Installation. This will result in two /path/to/TKGrid directories with two …/TKGrid/tkmpinodelib.sh scripts. Modify the tkmpinodelib.sh script that is used by SAS Visual Analytics in the same way as shown in the previous code—changing only the CGROUP:

```
#! /bin/sh

if [ -n "$TKMPI_UMASK" ]; then umask "$TKMPI_UMASK"; fi

if [ -n "$TKMPI_CORESIZE" ]; then

  ulimit -Sc $TKMPI_CORESIZE

else

  ulimit -Sc 0

fi

if [ -n "$TKMPI_ULIMIT" ]; then

  ulimit $TKMPI_ULIMIT

fi

cgclassify -g cpu:cgLASR $$

$TKMPI_DIR/bin/timed_run $TKMPI_MAXRUNTIME $TKMPI_NODELIB $*
```

NICE SAS HIGH-PERFORMANCE ANALYTICS PROCEDURES

To “NICE” the execution of SAS High-Performance Analytics procedures modify the …/TKGrid/tkmpirsh.sh script with a line similar to the one highlighted below:

```
#! /bin/sh

#

# TKGRID Configuration file.


export TKMPI_DIR=/lab/stress/TKGrid_HPA/TKGrid

export MPI_DIR=/lab/stress/TKGrid_HPA/TKGrid/mpich2-install

export HADOOP_HOME=/home/hadoop/hadoop-0.23.1

export TKMPI_NODELIB=$TKMPI_DIR/bin/tkmpinodelib

export HYDRA_LAUNCHER_EXTRA_ARGS="-q -o StrictHostKeyChecking=no -o
PasswordAuthentication=no "

export TKMPI_MACHINELIST=/lab/stress/TKGrid_HPA/TKGrid/grid.hosts

export TKMPI_MPIRUN=$TKMPI_DIR/bin/mpirun.sh

export TKMPI_MPDTRACE=$TKMPI_DIR/bin/listmachines.sh

export TKMPI_MAXRUNTIME=7200

export TKMPI_UMASK=027

export TKMPI_NICE=12

export LANG=POSIX

ulimit -Sc 0

if [ "$1" = "-" -o "$1" = "-np" -a "$2" = "1" ]; then
```

```
      if [ "$1" = "-" ]; then
```

NETWORK TUNING

The mix of SAS Visual Analytics, SAS High-Performance Analytics, and SAS Grid Manager workloads within the same environment collocates workloads that requires balanced tuning to achieve low network latency as well as to enable large throughput without placing undue burden on the CPU. Interrupt coalescence as well as Linux kernel tuning can benefit in these areas. The following configuration is presented as one that improved performance in the experiments supporting this paper. They are not presented as an optimal configuration for all customers.

General Linux tuning:

```
sysctl -w net.ipv4.tcp_timestamps=0
sysctl -w net.ipv4.tcp_sack=0
sysctl -w net.core.netdev_max_backlog=250000
sysctl -w net.core.rmem_max=16777216
sysctl -w net.core.wmem_max=16777216
sysctl -w net.core.rmem_default=16777216
sysctl -w net.core.wmem_default=16777216
sysctl -w net.core.optmem_max=16777216
sysctl -w net.ipv4.tcp_mem="16777216 16777216 16777216"
sysctl -w net.ipv4.tcp_rmem="4096 87380 16777216"
sysctl -w net.ipv4.tcp_wmem="4096 65536 16777216"
sysctl -w net.ipv4.tcp_low_latency=1
```

Ethernet configuration:

```
> ethtool -c eth0
Coalesce parameters for eth0:
Adaptive RX: off  TX: off
stats-block-usecs: 0
sample-interval: 0
pkt-rate-low: 0
pkt-rate-high: 0

rx-usecs: 1
rx-frames: 0
rx-usecs-irq: 0
rx-frames-irq: 0

tx-usecs: 0
tx-frames: 0
tx-usecs-irq: 0
tx-frames-irq: 256

rx-usecs-low: 0
rx-frame-low: 0
tx-usecs-low: 0
tx-frame-low: 0

rx-usecs-high: 0
rx-frame-high: 0
tx-usecs-high: 0
tx-frame-high: 0
```

APPENDIX A: CONTENT OF THE FINAL /ETC/CGCONFIG.CONF USED IN THESE EXPERIMENTS

NOTE: This example is provided for illustrative purposes only, and is not intended to imply that this configuration will meet the needs of any specific SAS implementation.

```
mount {
      cpu=/cgroup/cpu;
```

```
              memory=/cgroup/memory;
              blkio=/cgroup/blkio;
       }

       group cgGRID {
              perm {
                     task{
                            uid=sas;
                            gid=sas;
                     } admin {
                            uid=root;
                            gid=root;
                     }
              }
              cpu {
                     cpu.shares=5;
              }
              blkio {
                     blkio.weight=100;
              }
       }

       group cgHPA {
              perm {
                     task {
                            uid=sas;
                            gid=sas;
                     } admin {
                            uid=root;
                            gid=root;
                     }
              }
              cpu {
                     cpu.shares=5;
              }
              blkio {
                     blkio.weight=1000;
              }
       }

       group cgLASR {
              perm {
                     task {
                            uid=sas;
                            gid=sas;
                     } admin {
                            uid=root;
                            gid=root;
                     }
              }
              cpu {
                     cpu.shares=500;
              }
       }
```

APPENDIX B: HARDWARE USED IN POC

16 Compute Nodes

- Dell M620
- 2 x Intel(R) Xeon(R) CPU E5-2680 @ 2.70GHz
- 128 GB RAM
- 2 x 10 Gb NICs (one connected to “the network” and one connected to the iSCSI network)

SAS Metadata and Mid-Tier Server

- HP DL380 G7
- 2 x Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
- 24 GB RAM
- 2 x 10 Gb NICs (one connected to “the network” and one connected to the iSCSI network)

iSCSI SAN

12 x HP P4500 G2

- Each node with 12 x 15k RPM 300GB LFF SAS Disk drives
- 2 x 10 Gb NICs

iSCSI network utilized Jumbo Frames (MTU = 9000) and all switches were configured cut-through.

CONCLUSION

SAS Grid Manager, SAS High-Performance Analytics, and SAS Visual Analytics can successfully share hardware. For satisfactory performance it is necessary to

- prioritize the requirements related to performance
- understand the demand placed on the environment
- size the environment based on the total workload
- continually monitor and adjust the configuration and/or capacity based on changes in the environment

There are two primary options for controlling the relative priority of work within the environment – the use of NICE values and the use of CGROUPs. The use of NICE values is straight-forward but does not offer the same level of control as is offered by CGROUPs. Customers should select the option that best meets their business needs both in terms of response times and effort required to configure and manage the environment going forward.

RECOMMENDED READING

- SAS® High-Performance Computing Management Console: User's Guide
- Red Hat Enterprise Linux 6.5 Resource Management Guide

CONTACT INFORMATION

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

Ken Gahagan
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Ken.Gahagan@sas.com
http://www.sas.com