CONSOLIDATE SAS® 9.4 WORKLOADS WITH INTEL® XEON® PROCESSOR E7 V3 FAMILY AND INTEL® SSD DATA CENTER FAMILY

SIMPLIFY, SAVE, & DO MORE

Modernize your data center by consolidating the SAS® workloads of your legacy servers onto next-generation Intel® Xeon® processor-powered servers

Is an outdated data center holding back your business? It might be time to modernize and simplify with new systems that can do the work of many older servers. This can translate to savings in space and operating costs while leaving headroom for future growth.

For businesses running SAS® Analytics software, this means virtualizing bare-metal legacy servers onto a newer Intel® Xeon® processor E7 v3 platform, such as the new Dell PowerEdge R930. In the Principled Technologies data center, we found that a server powered by Intel Xeon processors E5-2680 v2 with Intel Solid-State Drives (SSD) DC S3700 Series did the SAS work of nearly four legacy servers, and a newer PowerEdge R930 powered by Intel Xeon processors E7-8890 v3 did the SAS work of 12 legacy servers.[1] With 12 VMs, the server powered by Intel Xeon processors E7 v3 with Intel SSD DC P3700 Series for PCIe® delivered nearly 14 times the relative performance of the legacy server. What's more, each VM on the server powered by Intel Xeon processors E7 v3 completed the SAS multi-user scenario workload an average of over 25 minutes sooner, and completed nearly 110 more jobs per hour on average than the bare-metal

[1] For more information on the SAS 9.4 workload and jobs, see The SAS workload section on page 3 and Appendix A.

APRIL 2015
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by SAS

legacy server did. In addition, some performance capacity remained when running 12 virtual instances simultaneously on the newer server, potentially leaving room for more virtual SAS 9.4 instances.

SAVING WITH MODERNIZATION

One approach to the modernization of data centers combines consolidating resources through virtualization with upgrading to new hardware technology. These improvements can lead to reduced complexities throughout the technology stack, and potentially reduce operational expenses (OPEX). Specifically, consolidating physical hardware in the data center typically provides these advantages:

- Reduced power and cooling requirements
- Smaller physical footprints, such as reduced number of racks and servers, which reduces space-related costs
- Lower management costs due to fewer physical servers to manage
- Fewer resources wasted as a result of underutilized hardware

Newer systems generally support, and will have, substantially more RAM and storage, sometimes in the form of faster SSDs. These technologies are ideal for in-memory data processing and analytics, such as SAS 9.4. The additional RAM and high-performance SSDs work with Intel Xeon processor E5 v2 and E7 v3 technology, and are designed by Intel to deliver fast performance for analytics workloads while increasing data center efficiencies. Moving from multiple legacy servers onto newer servers powered by Intel Xeon processors E7-8890 v3 and Intel SSDs DC P3700 running SAS 9.4, can help in consolidation efforts, contributing to your modernization initiatives. In our tests, which we describe further below, we found that one newer four-socket Dell PowerEdge R930 server powered by Intel Xeon processors E7-8890 v3 could replace 12 legacy servers running SAS workloads (see Figure 1).

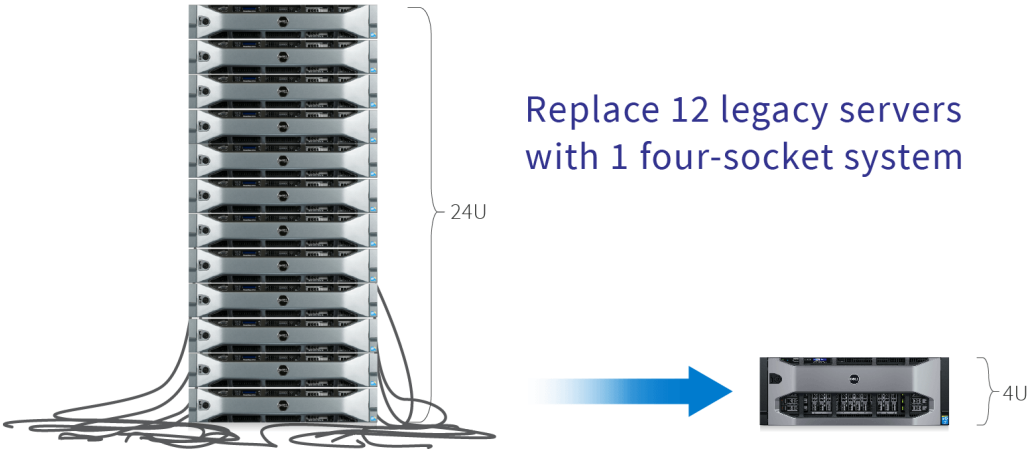


Figure 1: Consolidating legacy servers is a key to modernization.

THE RIGHT COMBINATION WITH INTEL + SAS

Deployments of SAS 9.4, an environment designed by SAS for business and advanced complex data analytics, can benefit from servers powered by the Intel Xeon processor E7 v3 family as processing, data storage, and analysis requirements increase over time. The updated Intel technology in four-socket servers can make them well suited to support large-volume, complex data analytics software such as those from SAS. The Dell PowerEdge R930, which contains four Intel Xeon processor E7 v3 processors, provides up to 72 cores and 144 threads and supports up to 6TB of DDR4 memory to improve performance and help speed up your SAS workloads. Newer servers can also support increased RAM and faster SSD storage. In our testing, we took advantage of these RAM and SSD improvements. The servers powered by Intel Xeon processors E5-2680 v2 had 256 GB RAM with DC S3700 SATA SSDs and the servers powered by Intel Xeon processors E7-8890 v3 had 1,024 GB RAM with DC P3700 PCIe SSDs.

For the modern data center, servers based on the Intel Xeon processor E7 v3 family also offer a number of enhanced reliability, availability, and serviceability (RAS) and resource management features that build on previous processor releases. With Intel Run Sure technology, these processors add new RAS features such as eMCA Gen 1, MCA Recovery – Execution Path, MCA I/O, and PCIe Live Error Recovery to help keep SAS workloads up and running. For more information on components used in our testing, see Appendix A. For detailed hardware configuration information, see Appendix B.

The SAS workload

For our hands-on testing, SAS created a multiuser workload to simulate the workload of a typical SAS foundation environment consisting of 25 different data analysis tasks. The test scenario was a self-contained SAS 9.4 workload that simulated 8 to 16 simultaneous users performing computation-focused jobs, as well as I/O-focused jobs. The data analysis jobs varied from small and lightweight (e.g., simple reporting), to more complex analytics (e.g., regression), to heavy data manipulations (e.g., joins, sorts).

For more information on SAS 9.4 and the SAS workload, see Appendix A. Appendix C shows how we performed our tests. See Appendix D and Appendix E for configuration files and test scripts.

WHAT WE FOUND

More jobs in less time and more headroom with 12 VMs

Virtualizing meant more work in less physical space—up to 12 times the amount of jobs ran simultaneously on one four-socket server powered by Intel Xeon processors E7-8890 v3 as compared to the legacy server.

Running 12 virtual SAS 9.4 instances on the server powered by Intel Xeon processors E7-8890 v3 and PCIe SSDs provided a number of benefits. First, we found that simply virtualizing the SAS workloads increased the amount of jobs that both the two-socket server powered by the Intel Xeon processors E5-2680 v2 and the four-socket server powered by the Intel Xeon processors E7-8890 v3 could perform simultaneously. Simply put, virtualizing meant more work in less physical space, which creates an opportunity to consolidate legacy two-socket servers in your data center.

Second, the two-socket server powered by Intel Xeon processors E5-2680 v2 provided up to nearly four times the relative performance of the two-socket legacy server, while the four-socket Intel Xeon processor E7-8890 v3 server provided nearly 14 times the relative performance. This performance calculation uses the average time to complete the workload to compare the work of each VM. Figure 2 shows the relative performance for our three solutions.

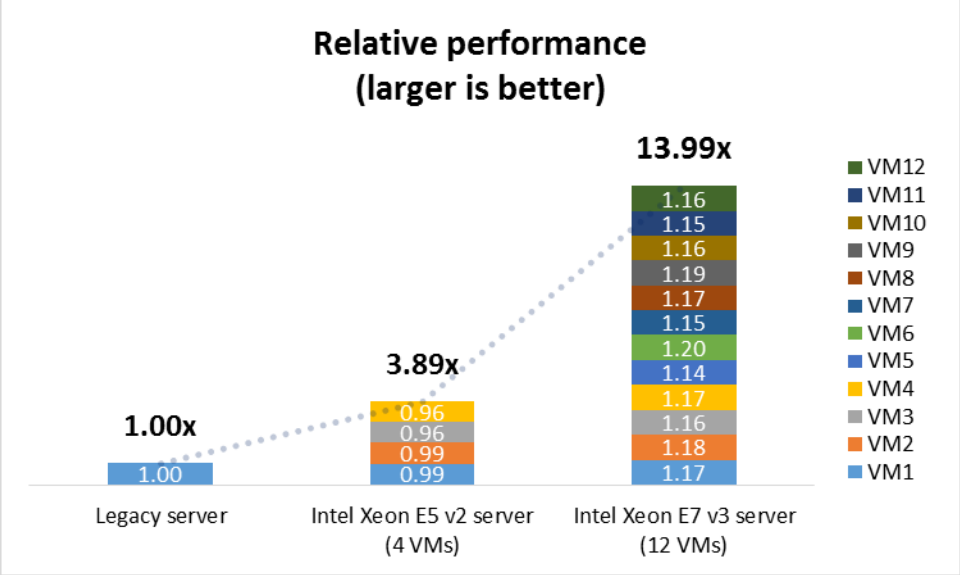


Figure 2: The relative performance of each solution.

The four-socket server with Intel Xeon processors E7-8890 v3 completed the SAS workload on average **14 percent faster** than the legacy server did.

Keeping those two previous points in mind, the four-socket server powered by Intel Xeon processors E7-8890 v3 provided nearly 14 times the relative performance of the legacy server in 14.2 percent less time—completing the SAS workload over 25 minutes faster. The shorter average time to complete the workload was driven by the four Intel Xeon processors E7-8890 v3 processors and the improved I/O performance offered by the Intel DC P3700 PCIe SSDs. Improved technology simply meant less time to complete the work. Figure 3 shows the average time each solution took to complete the SAS multi-user scenario workload.

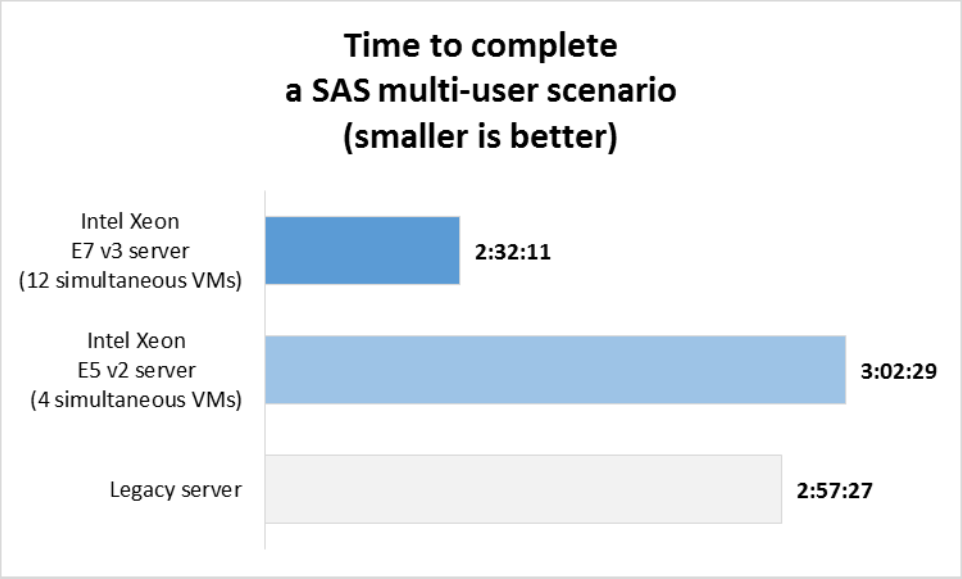


Figure 3: The average time it took each solution to complete one SAS multi-user scenario in hours:minutes:seconds.

Examining the time savings another way, the server powered by Intel Xeon processors E7-8890 v3 completed an average of over 118 jobs in an hour, which is nearly 13 times the number of jobs per hour that the legacy server performed. As shown in Figure 4, the server powered by Intel Xeon processors E5-2680 v2 completed an average of over 32 jobs per hour, or 2.77 times more jobs per hour than the legacy server did. More SAS jobs per hour means your business can analyze more quickly.

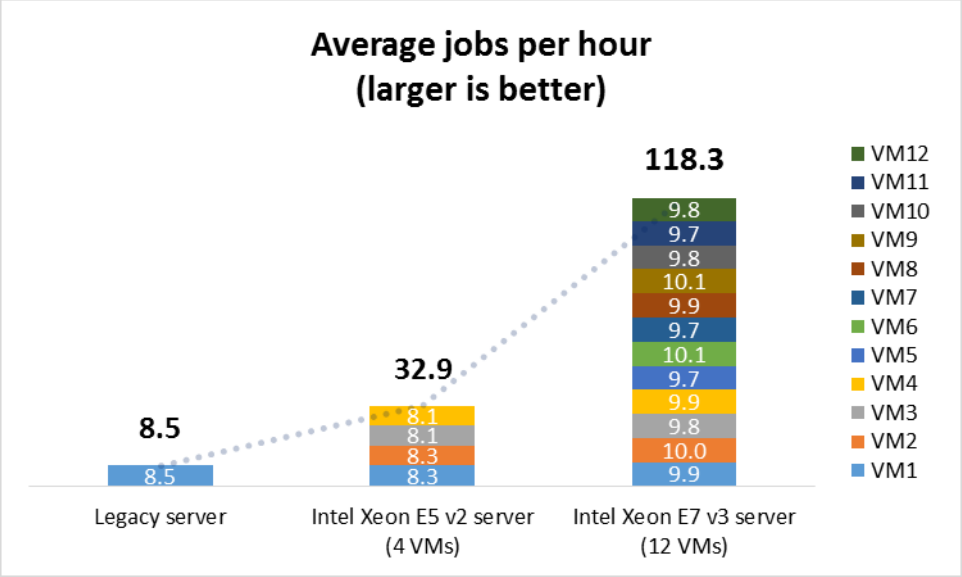


Figure 4: The average number of jobs per hour for each solution.

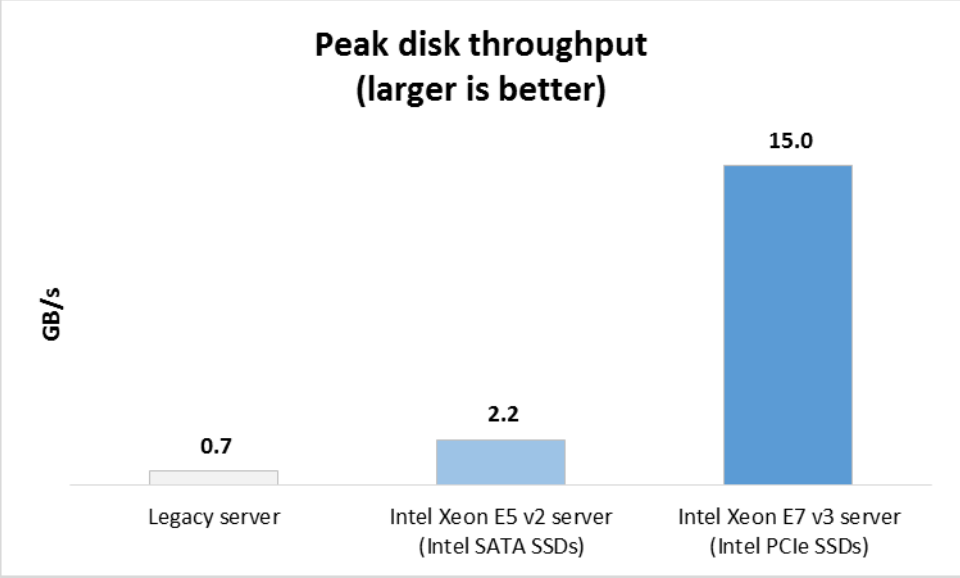
In addition to the performance and time benefits, we saw that the four-socket server powered by Intel Xeon processors E7-8890 v3 had system processing power remaining even when it was running 12 virtual SAS 9.4 instances. This headroom means that for some periods during the test scenario, the 12 VMs did not fully saturate the processors, storage, or memory. With the available headroom, one could increase the number of VMs on the server for more SAS 9.4 instances running additional analytics, though we did not do so.

Even with our 12 VMs on the server powered by Intel Xeon processors E7-8890 v3 with Intel SSDs DC P3700, we had headroom remaining at points in the test, which could translate to a potentially greater number of VMs in your environment.

SUPPORTING PROCESSOR PERFORMANCE WITH INTEL SSD STORAGE

Running SAS 9.4 software on a server without improved storage performance could place limitations on the workloads, leave the processor underutilized, and limit the speed of your SAS jobs. Combining the Intel Xeon processors E5 v2 and E7 v3 with the Intel DC S3700 SATA and DC P3700 PCIe SSDs respectively, allowed us to increase the workload density and get more from our storage subsystem. A higher workload density meant the newer servers completed more jobs simultaneously and averaged more jobs per hour. Using the Intel DC S3700 SATA and DC P3700 PCIe SSDs improved the outputs from the disk subsystem. Figure 5 shows the peak disk throughput that each solution achieved.

Figure 5: The peak disk throughput in GB per second for each solution.



Your business could see a number of potential benefits by upgrading your legacy servers with hard-disk drives (HDDs) to newer servers with SATA and PCIe SSDs:

- Get SAS data sooner and have more time to analyze the data
- Save on management time by having fewer servers to maintain
- Reduce waste by utilizing processor and memory resources more effectively

CONCLUSION

A key to modernizing your data center is to consolidate your legacy workloads through virtualization, which can help reduce complexity for your business. Fewer servers require fewer physical resources, such as power, cabling, and switches, and reduce the burden on IT for ongoing management tasks such as updates. In addition, integrating newer hardware technology into your data center can provide new features that strengthen your infrastructure, such as RAS features on the processor and disk performance improvements. Finally, using SAS 9.4 ensures that you have the latest features and toolsets that SAS can offer.

Compared to a legacy server, we found that a modern four-socket server powered by Intel Xeon processors E7-8890 v3 with Intel SSD DC P3700 Series provided 12 times the amount of SAS work, nearly 14 times the relative performance, and a shorter average time to complete the SAS workload. Running 12 virtual SAS instances also left capacity on the server for additional work. Consolidating your SAS workloads from legacy servers onto servers powered by Intel Xeon processors E7 v3 and SAS 9.4 can provide your business with the latest hardware and software features, reduce complexity in your data center, and potentially reduce costs for your business.

APPENDIX A – ABOUT THE COMPONENTS

About the new Intel Xeon processor E7 v3 family

Intel designed the new Intel Xeon processor E7 v3 family to support mission-critical, high-performance workloads by featuring up to 18 cores and 36 threads per socket to provide significant jumps in performance from previous releases. The Intel Xeon processor E7 v3 family supports up to 1.5TB DDR4 memory per socket, supports up to 24 DDR4 DIMMs per socket, and supports up to 2,133MHz DDR4 speeds to improve performance and increase scalability.

The Intel Xeon processor E7 v3 family supports all the previous reliability, availability, and serviceability features of previous processor releases to support critical workloads. With Intel® Run Sure technology, these processors add new RAS features, including eMCA Gen 1, MCA Recovery – Execution Path, MCA IO, and PCIe Live Error Recovery. For more information about the Intel Xeon processor E7 v3 product family, visit www.intel.com.

About the new Intel SSD DC P3700 Series

According to Intel, "[t]he Intel® Solid-State Drive Data Center Family for PCIe brings extreme data throughput directly to Intel Xeon processors with up to six times faster data transfer speed than 6 Gbps SAS/SATA SSDs. The performance of a single drive from the Intel SSD Data Center Family for PCIe, specifically the Intel Solid-State Drive Data Center P3700 Series (460K IOPS), can replace the performance of seven SATA SSDs aggregated through a host bus adapter (HBA) (approximately 500K IOPS)." For more information on the Intel SSD PC P3700 Series, visit www.intel.com/content/www/us/en/solid-state-drives/solid-state-drives-dc-p3700-series.html.

About SAS 9.4

SAS 9.4, the latest release of SAS®9 architecture, uses multicore technologies[2] to deliver processing capabilities through in-database and in-memory analytics. According to SAS, this results "in greater insights more quickly from big data and streaming data." SAS upgraded the architecture with features that meet the needs of traditional on-site SAS deployments and of private and public cloud deployments:

- Choose from many deployment options, including hosted and managed cloud options with SAS Solutions OnDemand
- Use potentially highly scalable environments for testing and development with six new products
- Integrate SAS into your business processes with new APIs, including mobile delivery options for popular smartphones and tablets
- Get monitoring and management capabilities with the new Web-based SAS Environment Manager
- Monitor data according to your schedule (daily, weekly, or monthly) with the new SAS Environment Manager
- Receive alerts, notifications, and data from your servers in the customizable SAS Environment Manager dashboard

[2] Multi-core/threading technologies were not fully leveraged with the workload used in this study, but can be with newer SAS 9.4 capabilities such as In-memory and high-performance analytics.

The second maintenance release, SAS® 9.4 TS1M2 (9.4M2), offers new features and enhancements for products such as SAS/STAT®, SAS® Enterprise Miner™, and SAS/ETS®. For more information on SAS 9.4, visit sas.com/en_us/software/sas9.html.

About the SAS workload

Test details

Each test scenario consisted of a set of 25 jobs running in a multiuser fashion to simulate a typical SAS batch and SAS enterprise user environment where 8 to 16 simultaneous jobs were executing at any one time (except during initial startup and during the ramp down period at the end). In the scenario-controlled scripts, jobs were launched over a set interval to simulate a multiuser environment where users came and went from the system. To help simulate a typical work environment during the test, some periods were heavier than others were. Some periods were batch-only and other periods were batch and interactive users. This simulated a real-world environment where there are peaks and valleys in system resource use.

The jobs used a mix of CPU, memory, and I/O-intensive workloads designed to simulate a larger work-group of 8 to 16 simultaneous sessions. During the test, 14 different programs were executed (25 different jobs were run during the test, some more than once). We used different input data for all jobs. SAS procedures used during execution included: LOGISTIC, REG, GLM, SORT, MEANS, MIXED, SUMMARY, DATA steps, and SQL.

Data

A single copy of the workload includes the following details:

- Individual file sizes up to 30 GB
- File types: text, sas7bdat, binary
- Row counts up to 76 million
- Variable and or column counts up to 297
- Binary file usage with EBCDIC and binary data from a mainframe (MVS)
- Approximately 130 GB of input data was used in each 8-to-16-simultaneous-user scenario that we executed

Input and output volumes were at least three to four times the input size (SAS programs typically read and write data depending on function).

APPENDIX B – SYSTEM CONFIGURATION INFORMATION

Figure 6 provides detailed configuration information for the test systems.

System	Dell PowerEdge R710	Dell PowerEdge R720	Dell PowerEdge R930
General			
Number of processor packages	2	2	4
Number of cores per processor	4	10	18
Number of hardware threads per core	2	2	2
System power management policy	Maximum performance	Performance	Performance
Power supplies			
Total number	2	2	4
Vendor and model number	Dell A870P-00	Dell E1100E-S0	Dell E1100E-S0
Wattage of each (W)	870	1,100	1,100
Cooling fans			
Total number	5	6	6
Vendor and model number	Dell PFC0612DE	San Ace® 60 9GA0612P1K641	Nidec® UltraFlo® V12C12BS1M3
Dimensions (h × w) of each	3″ × 3″	3″ × 3″	6″ × 6″
Volts	12	12	12
Amps	1.68	0.95	2.31
CPU			
Vendor	Intel	Intel	Intel
Name	Xeon	Xeon	Xeon
Model number	X5570	E5-2680 v2	E7-8890 v3
Socket type	LGA1366	LGA2011	LGA2011
Core frequency (GHz)	2.93	2.8	2.5
Bus frequency	6.4 GT/s	8 GT/s	9.6 GT/s
L1 cache	4 × 32 KB Data 4 × 32 KB Instr.	10 × 32 KB Data 10 × 32 KB Instr.	18 × 32 KB Data 18 × 32 KB Instr.
L2 cache	4 × 256 KB	10 × 256 KB	18 × 256 KB
L3 cache	8 MB	25 MB	45 MB
Platform			
Vendor and model number	Dell PowerEdge R710	Dell PowerEdge R720	Dell PowerEdge R930
Motherboard model number	0YDJK3	0M1GCRX04	0Y4CNCA01
BIOS name and version	6.4.0	2.5.2	1.0.2 [MRC_096]
BIOS settings	Default	Default	Default

System	Dell PowerEdge R710	Dell PowerEdge R720	Dell PowerEdge R930
Memory module(s)			
Total RAM in system (GB)	48	256	1,024
Vendor and model number	Samsung® M393B1K70DH0-YH9	Hynix HMT42GR7MFR4C-PB	Samsung M386A4G40DM0-CPB
Type	PC3L-10600R	PC3-12800R	PC4-17000R
Speed (MHz)	1,333	1,600	2,133
Speed running in the system (MHz)	1,333	1,600	1,600
Timing/Latency (tCL-tRCD-tRP-tRASmin)	9-9-9-36	11-11-11-35	15-15-15-36
Size (GB)	8	16	32
Number of RAM module(s)	6	16	32
Chip organization	Double Sided	Double Sided	Double Sided
Rank	Dual	Dual	Quad
Operating system			
Name	CentOS 6.6	CentOS 6.6	CentOS 6.6
Kernel Version	2.6.32-504.16.2.el6.x86_64	2.6.32-504.16.2.el6.x86_64	2.6.32-504.16.2.el6.x86_64
File system	ext4 (OS and SAS program files) xfs (SAS DATA/OUTPUT) xfs (SAS WORK/UTILLOC)	ext4 (OS and SAS program files) xfs (SAS DATA/OUTPUT) xfs (SAS WORK/UTILLOC)	ext4 (OS and SAS program files) xfs (SAS DATA/OUTPUT) xfs (SAS WORK/UTILLOC)
RAID controller			
Vendor and model number	Dell PERC 6/i	Dell PERC H710P Mini	Dell PERC H730P
Firmware version	6.3.3.0002	21.3.1-0004	25.2.2-0004
Driver version	06.803.01.00-rh1 (megaraid_sas)	06.803.01.00-rh1 (megaraid_sas)	06.803.01.00-rh1 (megaraid_sas)
Cache size (MB)	256	1,024	2,048
Solid-state drives			
Vendor and model number	N/A	Intel SSD DC S3700	Intel SSD DC P3700
Number of drives	N/A	8	8
Size	N/A	800 GB	2.0 TB
Type	N/A	2.5in SATA 6Gb/s	1/2 Height PCIe 3.0
Hard drives			
Vendor and model number	8 × Seagate® ST9300605SS	2 × Seagate ST300MM0006, 4 × Toshiba® MBF2600RC	2 × Toshiba AL13SEB900, 8 × Toshiba MBF2600RC
Number of drives	8	6	10
Size (GB)	300	300, 600	900, 600
RPM	15K	10K	10K
Type	SAS 6Gb/s	SAS 6Gb/s	SAS 6Gb/s

System	Dell PowerEdge R710	Dell PowerEdge R720	Dell PowerEdge R930
Ethernet adapters			
Vendor and model number	Broadcom NetXtreme® II 4P	Intel 2P X540/2P I350	Intel 2P X540/2P I350
Type	rNDC	rNDC	rNDC
Driver	bnx2 v2.2.4	Intel 5.0.5-k	Intel 5.0.5-k

Figure 6: System configuration information for the test systems.

APPENDIX C – HOW WE TESTED

Installing the CentOS 6.6 64-bit operating system

1. Insert the CentOS 6.6 installation DVD, and boot from it.
2. On the Welcome to CentOS 6! screen, select Install or upgrade an existing system, and press Enter.
3. On the Disc Found screen, select Skip, and press Enter.
4. On the CentOS 6 screen, click Next.
5. On the installation-selection screen, keep the default, and click Next.
6. One the keyboard-selection screen, keep the default, and click Next.
7. On the storage-selection screen, click Basic Storage Devices, and click Next.
8. On the Storage Device Warning pop-up screen, click Yes, discard any data.
9. On the Hostname screen, enter the server's name, and click Configure Network.
10. On the Network Connections pop-up screen, click Add.
11. On the Choose a Connection Type, select Wired, and click Create.
12. On the Editing Wired Connection pop-up, select the IPv4 Settings tab, change Method to Manual, click Add, enter the interface's IP address, netmask, and gateway, and click Apply.
13. Close the Network Connections pop-up screen.
14. Click next on the Hostname screen.
15. On the time-zone screen, click Next.
16. On the administrator-password screen, enter the Root Password (twice), and click Next.
17. On the Which type of installation would you like screen, click both Replace Existing Linux Systems(s), and click Next.
18. On the Format Warnings pop-up screen, click Format.
19. On the Writing storage configuration to disk pop-up screen, click Write changes to disk.
20. On the boot-loader selection screen, click Next.
21. On the software-selection screen, click Basic Server (R710, R720 VM, R930 VM) or Virtual Host (R720, R930), and click Next.
22. On the Congratulations screen, click Reboot.

Configuring the Dell R710 (bare-metal)

Run the following commands for each process.

Installing updates and additional packages

```
yum install -y epel-release
yum update -y
yum install -y chrony nmon time xfsprogs tuned numactl acpid cpuspeed wget vim
nfs-utils openssh-clients man unzip smartmontools numactl ipmitool OpenIPMI
sysstat
```

Disabling SELINUX and the firewall

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
```

```
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

Editing /etc/fstab

Change the following:

```
/dev/mapper/vg_r710-lv_root   /   ext4   defaults   1 1
```

To the following:

```
/dev/mapper/vg_r710-lv_root   /   ext4   defaults,nobarrier,noatime,nodiratime   1 1
```

Synching the time

```
service ntpd stop
chkconfig ntpd off
sed -i '/server .*/d' /etc/chrony.conf
echo 'server time.ptnet.principledtech.com iburst prefer' >> /etc/chrony.conf
service chronyd restart
chkconfig chronyd on
```

Creating and formatting volumes (assume RAID10)

```
VG_NAME=vg_r710
DISKS=8
STRIPE=256
lvcreate ${VG_NAME} -L 350G -C y -n lv_sasdata
lvcreate ${VG_NAME} -L 250G -n lv_saswork

mkfs.xfs -d su=${STRIPE}k,sw=$(($DISKS / 2)) /dev/${VG_NAME}/lv_sasdata
mkfs.xfs -d su=${STRIPE}k,sw=$(($DISKS / 2)) /dev/${VG_NAME}/lv_saswork

mkdir -p /data
echo -e "/dev/${VG_NAME}/lv_sasdata   /data   xfs
defaults,nobarrier,noatime,nodiratime,inode64   0 0" >> /etc/fstab
mount -v /data

mkdir -p /data/saswork
echo "/dev/${VG_NAME}/lv_saswork   /data/saswork   xfs
defaults,nobarrier,noatime,nodiratime,inode64   0 0" >> /etc/fstab
mount -v /data/saswork
```

Creating and applying a tuned profile

```
cp -r /etc/tune-profiles/enterprise-storage /etc/tune-profiles/sas
sed -i 's/set_transparent_hugepages.*always/set_transparent_hugepages never/'
/etc/tune-profiles/sas/ktune.sh
sed -i 's/multiply_disk_readahead.*4/multiply_disk_readahead 64/' /etc/tune-
profiles/sas/ktune.sh

tuned-adm profile sas
reboot
```

Creating users

```
groupadd -g 500 sas
useradd -u 500 -g 500 sasdemo
useradd -u 400 -g 500 sas
passwd sasdemo
passwd sas
echo 'export PATH=$PATH:/opt/SAS/SASFoundation/9.4' >> /home/sasdemo/.bashrc
echo 'export ASUITE=/data/asuite' >> /home/sasdemo/.bashrc
echo 'export PATH=$PATH:/opt/SAS/SASFoundation/9.4' >> /root/.bashrc
echo 'export ASUITE=/data/asuite' >> /root/.bashrc
```

Extracting SAS workload files

```
cd /
tar -zxvf opt_SAS.tar.gz
tar -zxvf data.tar.gz
```

Configure the R720 (Virtual Host)

Run the following commands for each process.

Installing updates and additional packages

```
yum install -y epel-release
yum update -y
yum install -y chrony nmon time xfsprogs tuned numactl acpid cpuspeed wget vim
nfs-utils openssh-clients man unzip smartmontools numactl ipmitool OpenIPMI
sysstat
```

Disabling SELINUX and the firewall

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

Editing /etc/fstab

1. Change the following:

```
/dev/mapper/vg_r720-lv_root   /   ext4   defaults   1 1
```

To the following:

```
/dev/mapper/vg_r720-lv_root   /   ext4   defaults,nobarrier,noatime,nodiratime   1 1
```

2. Add the following:

```
hugetlbfs   /dev/hugepages   hugetlbfs   mode=1770,gid=107   0 0
```

Editing /etc/libvirt/qemu.conf

```
hugetlbfs_mount = "/dev/hugepages"
```

Editing /etc/sysctl.conf

```
vm.hugetlb_shm_group = 107
vm.nr_hugepages = 122912
```

Synching the time

```
service ntpd stop
chkconfig ntpd off
sed -i '/server .*/d' /etc/chrony.conf
echo 'server time.ptnet.principledtech.com iburst prefer' >> /etc/chrony.conf
service chronyd restart
chkconfig chronyd on
```

Creating and formatting volumes (assume RAID10)

```
STRIPE=256
VMCOUNT=4
POOL_NAME[0]=vmpool
POOL_DEVS[0]="/dev/sdb"
POOL_DISKS[0]=4
POOL_HWRAID[0]=true
POOL_VOLNAME[0]=os
POOL_VOLSIZE[0]=80

POOL_NAME[1]=datapool
POOL_DEVS[1]="/dev/sdc"
POOL_DISKS[1]=8
POOL_HWRAID[1]=true
POOL_VOLNAME[1]=data
POOL_VOLSIZE[1]=600

for (( i=0; i<${#POOL_NAME[@]}; i++ ));
do
  NAME=${POOL_NAME[$i]}
  DEVS=${POOL_DEVS[$i]}
  DISKS=${POOL_DISKS[$i]}
  HWRAID=${POOL_HWRAID[$i]}
  VOLNAME=${POOL_VOLNAME[$i]}
  VOLSIZE=${POOL_VOLSIZE[$i]}

  vgremove -f $NAME
  pvremove $DEVS
  for dev in $DEVS; do
    dd if=/dev/zero of=${dev} bs=1M count=10 oflag=direct
  done

  if $HWRAID; then
    pvcreate --dataalignment $((${DISKS} / 2 * ${STRIPE}))K $DEVS
  else
    pvcreate $DEVS
  fi

  vgcreate -s 1g $NAME $DEVS

  for vm in `seq 1 $VMCOUNT`;
  do
    if $HWRAID; then
```

```
      lvcreate -n ${VOLNAME}${vm} -l ${VOLSIZE} -C y $NAME
    else
      if [ ${DISKS} -lt 4 ]; then
        lvcreate --type raid1 -m 1 -n ${VOLNAME}${vm} -l ${VOLSIZE} -C y $NAME
      else
        lvcreate --type raid10 -i $((${DISKS} / 2)) -I ${STRIPE} -m 1 -n
${VOLNAME}${vm} -l ${VOLSIZE} -C y $NAME
      fi
    fi
  done

  virsh pool-define-as --name $NAME --type logical --target /dev/$NAME
  virsh pool-start --pool $NAME
  virsh pool-autostart --pool $NAME
  sync
done
```

Creating and applying a tuned profile

```
tuned-adm profile virtual-host
```

Configure the R930 (Virtual Host)

Installing updates and additional packages

```
yum install -y epel-release
yum update -y
yum install -y chrony nmon time xfsprogs tuned numactl acpid cpuspeed wget vim
nfs-utils openssh-clients man unzip smartmontools numactl ipmitool OpenIPMI
sysstat
```

Disabling SELINUX and the firewall

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

Editing /etc/fstab

1. Change the following:

```
/dev/mapper/vg_r930-lv_root   /   ext4   defaults   1 1
```

To the following:

```
/dev/mapper/vg_r930-lv_root   /   ext4   defaults,nobarrier,noatime,nodiratime   1 1
```

2. Add the following:

```
hugetlbfs   /dev/hugepages   hugetlbfs   mode=1770,gid=107   0 0
```

Editing /etc/libvirt/qemu.conf

```
hugetlbfs_mount = "/dev/hugepages"
```

Editing /etc/sysctl.conf

```
vm.hugetlb_shm_group = 107
vm.nr_hugepages = 491616
```

Synching the time

```
service ntpd stop
chkconfig ntpd off
sed -i '/server .*/d' /etc/chrony.conf
echo 'server time.ptnet.principledtech.com iburst prefer' >> /etc/chrony.conf
service chronyd restart
chkconfig chronyd on
```

Creating and formatting volumes (assume RAID10)

```
STRIPE=256
VMCOUNT=12
POOL_NAME[0]=vmpool
POOL_DEVS[0]="/dev/sdb"
POOL_DISKS[0]=4
POOL_HWRAID[0]=true
POOL_VOLNAME[0]=os
POOL_VOLSIZE[0]=80

for (( i=0; i<${#POOL_NAME[@]}; i++ ));
do
  NAME=${POOL_NAME[$i]}
  DEVS=${POOL_DEVS[$i]}
  DISKS=${POOL_DISKS[$i]}
  HWRAID=${POOL_HWRAID[$i]}
  VOLNAME=${POOL_VOLNAME[$i]}
  VOLSIZE=${POOL_VOLSIZE[$i]}

  vgremove -f $NAME
  pvremove $DEVS
  for dev in $DEVS; do
    dd if=/dev/zero of=${dev} bs=1M count=10 oflag=direct
  done

  if $HWRAID; then
    pvcreate --dataalignment $((${DISKS} / 2 * ${STRIPE}))K $DEVS
  else
    pvcreate $DEVS
  fi

  vgcreate -s 1g $NAME $DEVS

  for vm in `seq 1 $VMCOUNT`;
  do
    if $HWRAID; then
      lvcreate -n ${VOLNAME}${vm} -l ${VOLSIZE} -C y $NAME
    else
      if [ ${DISKS} -lt 4 ]; then
```

```
        lvcreate --type raid1 -m 1 -n ${VOLNAME}${vm} -l ${VOLSIZE} -C y $NAME
      else
        lvcreate --type raid10 -i $((${DISKS} / 2)) -I ${STRIPE} -m 1 -n
${VOLNAME}${vm} -l ${VOLSIZE} -C y $NAME
      fi
    fi
  done

  virsh pool-define-as --name $NAME --type logical --target /dev/$NAME
  virsh pool-start --pool $NAME
  virsh pool-autostart --pool $NAME
  sync
done
```

Creating and formatting volumes (using LVM RAID1)

```
vgcreate -s 1g datapool /dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1
/dev/nvme4n1 /dev/nvme5n1 /dev/nvme6n1 /dev/nvme7n1

lvcreate --type raid1 -m 1 -n data1 -l 600 -C y datapool /dev/nvme0n1
/dev/nvme1n1
lvcreate --type raid1 -m 1 -n data2 -l 600 -C y datapool /dev/nvme2n1
/dev/nvme3n1
lvcreate --type raid1 -m 1 -n data3 -l 600 -C y datapool /dev/nvme4n1
/dev/nvme5n1
lvcreate --type raid1 -m 1 -n data4 -l 600 -C y datapool /dev/nvme6n1
/dev/nvme7n1
lvcreate --type raid1 -m 1 -n data5 -l 600 -C y datapool /dev/nvme1n1
/dev/nvme0n1
lvcreate --type raid1 -m 1 -n data6 -l 600 -C y datapool /dev/nvme3n1
/dev/nvme2n1
lvcreate --type raid1 -m 1 -n data7 -l 600 -C y datapool /dev/nvme5n1
/dev/nvme4n1
lvcreate --type raid1 -m 1 -n data8 -l 600 -C y datapool /dev/nvme7n1
/dev/nvme6n1
lvcreate --type raid1 -m 1 -n data9 -l 600 -C y datapool /dev/nvme0n1
/dev/nvme1n1
lvcreate --type raid1 -m 1 -n data10 -l 600 -C y datapool /dev/nvme2n1
/dev/nvme3n1
lvcreate --type raid1 -m 1 -n data11 -l 600 -C y datapool /dev/nvme4n1
/dev/nvme5n1
lvcreate --type raid1 -m 1 -n data12 -l 600 -C y datapool /dev/nvme6n1
/dev/nvme7n1
```

Configure the R720 Virtual Machines

Installing updates and additional packages

```
yum install -y epel-release
yum update -y
yum install -y chrony nmon time xfsprogs tuned acpid wget vim nfs-utils openssh-
clients man unzip sysstat numactl
```

```
ln -s /dev/null /etc/udev/rules.d/75-persistent-net-generator.rules
rm -f /etc/udev/rules.d/70-persistent-net.rules
```

Disabling SELINUX and the firewall

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

Editing /etc/fstab

Change the following:

```
/dev/mapper/vg_r720vm1-lv_root / ext4 defaults 1 1
```

To the following:

```
/dev/mapper/vg_r720vm1-lv_root / ext4 defaults,nobarrier,noatime,nodiratime 1 1
```

Synching the time

```
service ntpd stop
chkconfig ntpd off
sed -i '/server .*/d' /etc/chrony.conf
echo 'server time.ptnet.principledtech.com iburst prefer' >> /etc/chrony.conf
service chronyd restart
chkconfig chronyd on
```

Creating users

```
groupadd -g 500 sas
useradd -u 500 -g 500 sasdemo
useradd -u 400 -g 500 sas
passwd sasdemo
passwd sas
echo 'export PATH=$PATH:/opt/SAS/SASFoundation/9.4' >> /home/sasdemo/.bashrc
echo 'export ASUITE=/data/asuite' >> /home/sasdemo/.bashrc
echo 'export PATH=$PATH:/opt/SAS/SASFoundation/9.4' >> /root/.bashrc
echo 'export ASUITE=/data/asuite' >> /root/.bashrc
```

Creating and applying a tuned profile

```
cp -r /etc/tune-profiles/enterprise-storage /etc/tune-profiles/sas
sed -i 's/set_transparent_hugepages.*always/set_transparent_hugepages never/'
/etc/tune-profiles/sas/ktune.sh
sed -i 's/multiply_disk_readahead.*4/multiply_disk_readahead 64/' /etc/tune-
profiles/sas/ktune.sh

tuned-adm profile sas
reboot
```

Creating and formatting volumes (assume RAID10)

```
VG_NAME=vg_sas
STRIPE=256
```

```
DATA_DISKS=8
WORK_DISKS=8
DATA_DEV=/dev/vdb
WORK_DEV=/dev/vdb

pvcreate --dataalignment $((${DATA_DISKS} / 2 * ${STRIPE}))K $DATA_DEV
pvcreate --dataalignment $((${WORK_DISKS} / 2 * ${STRIPE}))K $WORK_DEV

vgcreate -s ${STRIPE}m $VG_NAME $DATA_DEV
vgextend $VG_NAME $WORK_DEV

lvcreate ${VG_NAME} ${DATA_DEV} -L 350G -n lv_sasdata
lvcreate ${VG_NAME} ${WORK_DEV} -l 100%FREE -n lv_saswork

mkfs.xfs -d su=${STRIPE}k,sw=$((${DATA_DISKS} / 2)) /dev/${VG_NAME}/lv_sasdata
mkfs.xfs -d su=${STRIPE}k,sw=$((${WORK_DISKS} / 2)) /dev/${VG_NAME}/lv_saswork

mkdir -p /data
echo -e "/dev/${VG_NAME}/lv_sasdata  /data  xfs
defaults,nobarrier,noatime,nodiratime,inode64  0 0" >> /etc/fstab
mount -v /data

mkdir -p /data/saswork
echo "/dev/${VG_NAME}/lv_saswork  /data/saswork  xfs
defaults,nobarrier,noatime,nodiratime,inode64  0 0" >> /etc/fstab
mount -v /data/saswork
poweroff
```

Extracting SAS workload files

```
cd / ; tar -zxvf opt_SAS.tar.gz
cd / ; tar -zxvf data.tar.gz
```

Configure the R920 Virtual Machines

Installing updates and additional packages

```
yum install -y epel-release
yum update -y
yum install -y chrony nmon time xfsprogs tuned acpid wget vim nfs-utils openssh-
clients man unzip sysstat numactl

ln -s /dev/null /etc/udev/rules.d/75-persistent-net-generator.rules
rm -f /etc/udev/rules.d/70-persistent-net.rules
```

Disabling SELINUX and the firewall

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

Editing /etc/fstab

Change the following:

```
/dev/mapper/vg_r920vm1-lv_root / ext4 defaults 1 1
```

To the following:

```
/dev/mapper/vg_r920vm1-lv_root / ext4 defaults,nobarrier,noatime,nodiratime 1 1
```

Synching the time

```
service ntpd stop
chkconfig ntpd off
sed -i '/server .*/d' /etc/chrony.conf
echo 'server time.ptnet.principledtech.com iburst prefer' >> /etc/chrony.conf
service chronyd restart
chkconfig chronyd on
```

Creating users

```
groupadd -g 500 sas
useradd -u 500 -g 500 sasdemo
useradd -u 400 -g 500 sas
passwd sasdemo
passwd sas
echo 'export PATH=$PATH:/opt/SAS/SASFoundation/9.4' >> /home/sasdemo/.bashrc
echo 'export ASUITE=/data/asuite' >> /home/sasdemo/.bashrc
echo 'export PATH=$PATH:/opt/SAS/SASFoundation/9.4' >> /root/.bashrc
echo 'export ASUITE=/data/asuite' >> /root/.bashrc
```

Creating and applying a tuned profile

```
cp -r /etc/tune-profiles/enterprise-storage /etc/tune-profiles/sas
sed -i 's/set_transparent_hugepages.*always/set_transparent_hugepages never/'
/etc/tune-profiles/sas/ktune.sh
sed -i 's/multiply_disk_readahead.*4/multiply_disk_readahead 64/' /etc/tune-
profiles/sas/ktune.sh

tuned-adm profile sas
reboot
```

Creating and formatting volumes (assume no RAID)

```
VG_NAME=vg_sas
STRIPE=256
DATA_DISKS=2
WORK_DISKS=2
DATA_DEV=/dev/vdb
WORK_DEV=/dev/vdb

pvcreate $DATA_DEV
pvcreate $WORK_DEV

vgcreate -s ${STRIPE}m $VG_NAME $DATA_DEV
vgextend $VG_NAME $WORK_DEV
```

```
lvcreate ${VG_NAME} ${DATA_DEV} -L 350G -n lv_sasdata
lvcreate ${VG_NAME} ${WORK_DEV} -l 100%FREE -n lv_saswork

mkfs.xfs -d su=${STRIPE}k,sw=${DATA_DISKS} /dev/${VG_NAME}/lv_sasdata
mkfs.xfs -d su=${STRIPE}k,sw=${WORK_DISKS} /dev/${VG_NAME}/lv_saswork

mkdir -p /data
echo -e "/dev/${VG_NAME}/lv_sasdata  /data  xfs
defaults,nobarrier,noatime,nodiratime,inode64  0 0" >> /etc/fstab
mount -v /data

mkdir -p /data/saswork
echo "/dev/${VG_NAME}/lv_saswork  /data/saswork  xfs
defaults,nobarrier,noatime,nodiratime,inode64  0 0" >> /etc/fstab
mount -v /data/saswork
poweroff
```

Extracting SAS workload files

```
cd / ; tar -zxvf opt_SAS.tar.gz
cd / ; tar -zxvf data.tar.gz
```

APPENDIX D – CONFIGURATION FILES

R720-vm1.xml

```
<domain type='kvm' id='1'>
  <name>r720-vm1</name>
  <uuid>95e5115b-fe41-9a1a-4006-26d90eed832d</uuid>
  <memory unit='KiB'> 62914560</memory>
  <currentMemory unit='KiB'> 62914560</currentMemory>
  <memoryBacking>
    <hugepages/>
    <nosharepages/>
  </memoryBacking>
  <vcpu placement='static'
cpuset='0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38'>8</vcpu>
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
  <os>
    <type arch='x86_64' machine='rhel6.6.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='host-passthrough'>
  </cpu>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/vmpool/os1'/>
      <target dev='vda' bus='virtio'/>
      <alias name='virtio-disk0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/datapool/data1'/>
      <target dev='vdb' bus='virtio'/>
      <alias name='virtio-disk1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </disk>
    <disk type='block' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <target dev='hdc' bus='ide'/>
```

```
      <readonly/>
      <alias name='ide0-1-0'/>
      <address type='drive' controller='0' bus='1' target='0' unit='0'/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <alias name='usb0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x7'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <alias name='usb0'/>
      <master startport='0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'
multifunction='on'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <alias name='usb0'/>
      <master startport='2'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x1'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <alias name='usb0'/>
      <master startport='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x2'/>
    </controller>
    <controller type='ide' index='0'>
      <alias name='ide0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='bridge'>
      <mac address='52:54:00:0d:4a:29'/>
      <source bridge='br0'/>
      <target dev='vnet0'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/0'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/0'>
      <source path='/dev/pts/0'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <input type='tablet' bus='usb'>
      <alias name='input0'/>
    </input>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900' autoport='yes' listen='127.0.0.1'
keymap='en-us'>
```

```
      <listen type='address' address='127.0.0.1'/>
    </graphics>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```

R920-vm1.xml

```
<domain type='kvm'>
  <name>r930-vm1</name>
  <uuid>9e770a15-6fa8-41ed-466d-c3a31f3aaada</uuid>
  <memory unit='KiB'>83886080</memory>
  <currentMemory unit='KiB'>83886080</currentMemory>
  <memoryBacking>
    <hugepages/>
    <nosharepages/>
  </memoryBacking>
  <vcpu placement='static'
cpuset='0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60,64,68,72,76,80,84,88,92,96,1
00,104,108,112,116,120,124,128,132,136,140'>8</vcpu>
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
  <os>
    <type arch='x86_64' machine='rhel6.6.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='host-passthrough'>
  </cpu>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/vmpool/os1'/>
      <target dev='vda' bus='virtio'/>
```

```
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/datapool/data1'/>
      <target dev='vdb' bus='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </disk>
    <disk type='block' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <target dev='hdc' bus='ide'/>
      <readonly/>
      <address type='drive' controller='0' bus='1' target='0' unit='0'/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x7'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <master startport='0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'
multifunction='on'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <master startport='2'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x1'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <master startport='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x2'/>
    </controller>
    <controller type='ide' index='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='bridge'>
      <mac address='52:54:00:41:34:a3'/>
      <source bridge='br0'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <serial type='pty'>
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <input type='tablet' bus='usb'/>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes' keymap='en-us'/>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
```

```
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```

APPENDIX E – TEST SCRIPTS

run_sas_multi.sh

```
#!/bin/bash
STATS_INTERVAL=15
MAX_RUNTIME=7200

REMOTE_ASUITE=/data/asuite
REMOTE_SASPATH=/opt/SAS/SASFoundation/9.4

SSH_CMD="ssh -2"
MAX_SAMPLES=$((${MAX_RUNTIME} / ${STATS_INTERVAL}))
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")

if [ "$SAS_HOSTS" == "" ]; then
  echo Variable SAS_HOSTS is invalid...exiting
  exit
fi
echo "SAS Hosts: $SAS_HOSTS"

# Check if VM
VM_TEST=true
if [ "$VIRT_HOSTS" == "$SAS_HOSTS" ]; then
  VM_TEST=false
elif [ "$VIRT_HOSTS" == "" ]; then
  VM_TEST=false
  VIRT_HOSTS="$SAS_HOSTS"
fi

if $VM_TEST; then
  echo "Hypervisor Hosts: $VIRT_HOSTS"
fi

TEST_NAME=$(echo $VIRT_HOSTS | awk '{print $1}')
RESULTS_DIR="${TEST_NAME}_${TIMESTAMP}"
REMOTE_HOST=

echo "Results Directory: $RESULTS_DIR"

mkdir -p $RESULTS_DIR

# Cleanup and drop caches
echo Cleanup and drop caches
for REMOTE_HOST in $SAS_HOSTS;
do
  $SSH_CMD root@$REMOTE_HOST "export ASUITE=${REMOTE_ASUITE} ; pkill sas ; sync ;
sleep 1 ; \$ASUITE/bin/clean.sh ; rm -f \$ASUITE/logs/*.* ; sync ; sleep 1 ; echo
3 > /proc/sys/vm/drop_caches ; sync ; sleep 1 ; find /* > /dev/null" 1> /dev/null
&
done
wait
```

```
# Start performance monitoring on VIRT hosts
if $VM_TEST; then
  echo Start performance monitoring on VIRT hosts
  for REMOTE_HOST in $VIRT_HOSTS;
  do
    $SSH_CMD root@$REMOTE_HOST "pkill nmon ; rm -f /tmp/*.nmon ; nmon -F
/tmp/${REMOTE_HOST}.nmon -s${STATS_INTERVAL} -c${MAX_SAMPLES} -t -C sas"
    $SSH_CMD -f root@$REMOTE_HOST "pkill sar ; rm -f /tmp/*.sar ; sar -o
/tmp/${REMOTE_HOST}.sar ${STATS_INTERVAL} ${MAX_SAMPLES} > /dev/null 2>&1" 2>
/dev/null
  done
  sleep $((${STATS_INTERVAL} * 2))
fi

# Start performance monitoring on SAS hosts
echo Start performance monitoring on SAS hosts
for REMOTE_HOST in $SAS_HOSTS;
do
  $SSH_CMD root@$REMOTE_HOST "pkill nmon ; nmon -F
${REMOTE_ASUITE}/logs/${REMOTE_HOST}.nmon -s${STATS_INTERVAL} -c${MAX_SAMPLES} -t
-C sas"
  $SSH_CMD -f root@$REMOTE_HOST "pkill sar ; sar -o
${REMOTE_ASUITE}/logs/${REMOTE_HOST}.sar ${STATS_INTERVAL} ${MAX_SAMPLES} >
/dev/null 2>&1" 2> /dev/null
done

sleep $((${STATS_INTERVAL} * 2))

# Run SAS Demo
date +%s.%N > ${RESULTS_DIR}/start_time.txt
for REMOTE_HOST in $SAS_HOSTS;
do
  #/usr/bin/time -p $SSH_CMD sasdemo@$REMOTE_HOST "export ASUITE=${REMOTE_ASUITE}
; export PATH=\$PATH:$REMOTE_SASPATH ; echo Running on ${REMOTE_HOST}!
PATH=\$PATH 2> \$ASUITE/logs/time.log | tee \$ASUITE/logs/output.log ; sleep 5 ;
echo Done on ${REMOTE_HOST}!" 2> ${RESULTS_DIR}/${REMOTE_HOST}_time.log 1>
${RESULTS_DIR}/${REMOTE_HOST}_output.log &
  /usr/bin/time -p $SSH_CMD sasdemo@$REMOTE_HOST "/usr/bin/time -p
\$ASUITE/bin/mixed_8_to_16.sh 2> \$ASUITE/logs/time.log | tee
\$ASUITE/logs/output.log ; sync" 2> ${RESULTS_DIR}/${REMOTE_HOST}_time.log 1>
${RESULTS_DIR}/${REMOTE_HOST}_output.log &
done
sleep 1
echo "Running SAS Demo! (To see output run: tail -f ${RESULTS_DIR}/*_output.log)"
wait
date +%s.%N > ${RESULTS_DIR}/stop_time.txt
echo SAS Demo complete!
sync
echo "$(cat ${RESULTS_DIR}/stop_time.txt) - $(cat ${RESULTS_DIR}/start_time.txt)"
| bc > ${RESULTS_DIR}/test_time.txt
```

```
sleep $((${STATS_INTERVAL} * 2))

# Stop performance monitoring on SAS hosts
echo Stop performance monitoring on SAS hosts
for REMOTE_HOST in $SAS_HOSTS;
do
  $SSH_CMD root@$REMOTE_HOST "pkill -USR2 nmon ; killall -w sar ; sync"
done

# Stop performance monitoring on VIRT hosts
if $VM_TEST; then
  sleep $((${STATS_INTERVAL} * 2))
  echo Stop performance monitoring on VIRT hosts
  for REMOTE_HOST in $VIRT_HOSTS;
  do
    $SSH_CMD root@$REMOTE_HOST "pkill -USR2 nmon ; killall -w sar ; sync"
  done
fi

# Parse and transfer remote results
echo Parse and transfer remote results
for REMOTE_HOST in $SAS_HOSTS;
do
  $SSH_CMD root@$REMOTE_HOST "export ASUITE=${REMOTE_ASUITE} ; cd \$ASUITE/logs ;
grep -A100 \"NOTE: The SAS System used:\" *.log | grep -v \"The SAS System\" |
sed 's/\.log-//' > sas_system.log ; cat sas_system.log | awk '/real time/{print
\$1\",\"\$4}' | awk -F'[,:]'
'BEGIN{min_sum=0;sec_sum=0};/:/{printf(\"%s,%s\n\",\$1,(\$2*60)+\$3);min_sum+=\$2
;sec_sum+=\$3};!/:/{printf(\"%s,%s\n\",\$1,\$2);sec_sum+=\$2};END{printf(\"total_
real_time,%s\n\",min_sum*60+sec_sum)}' > real_time.csv"
  mkdir -p ${RESULTS_DIR}/${REMOTE_HOST}
  scp root@$REMOTE_HOST:${REMOTE_ASUITE}/logs/*.* ${RESULTS_DIR}/${REMOTE_HOST}
done
OLD_DIR=$(pwd)
cd ${RESULTS_DIR}
find */real_time.csv | awk -F'/' '{printf("jobname,%s,",$1)}END{print""}' >
real_time.csv
paste -d',' */real_time.csv >> real_time.csv
cd ${OLD_DIR}

# Transfer virthost stats
if $VM_TEST; then
  echo Transfer virthost stats
  for REMOTE_HOST in $VIRT_HOSTS;
  do
    scp root@$REMOTE_HOST:/tmp/*.nmon ${RESULTS_DIR}/
      scp root@$REMOTE_HOST:/tmp/*.sar ${RESULTS_DIR}/
      for SAS_HOST in $SAS_HOSTS;
      do
        $SSH_CMD root@$REMOTE_HOST "virsh dumpxml ${SAS_HOST}" >
${RESULTS_DIR}/${SAS_HOST}.xml
      done
```

```
  done
fi
```

mixed_8_to_16.sh

```
#!/bin/sh

echo "computation multi-user ~8-16 simultaneous session test..."
echo ""


cd $ASUITE

OPTIONS="-memsize 2048M -bufsize 256k -sortsize 256M -fullstimer -work
/data/saswork "
OPTIONS2="-memsize 4096M -bufsize 256k -sortsize 256M -fullstimer -work
/data/saswork "

date

echo "ramp up..."

sas $ASUITE/code/codegen_issue.sas  -sysparm 1      -log
$ASUITE/logs/codegen_a_1.log       -print $ASUITE/lst/codegen_a_1.lst
$OPTIONS &
sleep 30
sas $ASUITE/code/io1_ca.sas -sysparm ca1   -log $ASUITE/logs/io1_ca_a_1.log
-print $ASUITE/lst/io1_ca_a_1.lst           $OPTIONS &
sleep 30
sas $ASUITE/code/comp_test1.sas            -log $ASUITE/logs/comp_test1_a_1.log
-print $ASUITE/lst/comp_test1_a_1.lst        $OPTIONS &
sas $ASUITE/code/comp_test1.sas            -log $ASUITE/logs/comp_test1_a_2.log
-print $ASUITE/lst/comp_test1_a_2.lst        $OPTIONS &
sleep 30
sas $ASUITE/code/mixed4.sas  -sysparm 1    -log $ASUITE/logs/mixed4_a_1.log
-print $ASUITE/lst/mixed4_a_1.lst      $OPTIONS &
sleep 30
sas $ASUITE/code/dim.sas -sysparm 1  -log $ASUITE/logs/dim_a_1.log   -print
$ASUITE/lst/dim_a_1.lst    $OPTIONS &
sleep 30
sas $ASUITE/code/hist_clm.sas -sysparm 1   -log $ASUITE/logs/hist_clm_a_1.log
-print $ASUITE/lst/hist_clm_a_1.lst          $OPTIONS &
sleep 30
sas $ASUITE/code/comp_test2.sas            -log $ASUITE/logs/comp_test2_a_1.log
-print $ASUITE/lst/comp_test2_a_1.lst     $OPTIONS &
sas $ASUITE/code/comp_test2.sas            -log $ASUITE/logs/comp_test2_a_2.log
-print $ASUITE/lst/comp_test2_a_2.lst     $OPTIONS &
sleep 90
sas $ASUITE/code/customer1.sas  -sysparm 1    -log $ASUITE/logs/customer1_a_1.log
-print $ASUITE/lst/customer1_a_1.lst      $OPTIONS &
sleep 5
```

```
sas $ASUITE/code/gsort.sas -sysparm 1      -log $ASUITE/logs/gsort_a_1.log
-print $ASUITE/lst/gsort_a_1.lst         $OPTIONS &
sleep 60
sas $ASUITE/code/gsort.sas -sysparm 2      -log $ASUITE/logs/gsort_a_2.log
-print $ASUITE/lst/gsort_a_2.lst         $OPTIONS &
sleep 30
sas $ASUITE/code/customer1.sas  -sysparm 2     -log
$ASUITE/logs/customer1_a_2.log   -print $ASUITE/lst/customer1_a_2.lst
$OPTIONS &
sleep 30
sas $ASUITE/code/io2.sas  -sysparm large1  -log $ASUITE/logs/io2_a_1.log
-print $ASUITE/lst/io2_a_1.lst              $OPTIONS &
sleep 30
sas $ASUITE/code/ranrw.sas  -sysparm 1     -log $ASUITE/logs/ranrw_a_1.log
-print $ASUITE/lst/ranrw_a_1.lst            $OPTIONS &
sleep 60
sas $ASUITE/code/comp_test5.sas            -log $ASUITE/logs/comp_test5_a_1.log
-print $ASUITE/lst/comp_test5_a_1.lst     $OPTIONS &
sas $ASUITE/code/comp_test5.sas            -log $ASUITE/logs/comp_test5_a_2.log
-print $ASUITE/lst/comp_test5_a_2.lst     $OPTIONS &
sleep 60
sas $ASUITE/code/comp_test2.sas            -log $ASUITE/logs/comp_test2_a_3.log
-print $ASUITE/lst/comp_test2_a_3.lst     $OPTIONS &
sas $ASUITE/code/comp_test5.sas            -log $ASUITE/logs/comp_test5_a_3.log
-print $ASUITE/lst/comp_test5_a_3.lst     $OPTIONS &
sas $ASUITE/code/comp_test5.sas            -log $ASUITE/logs/comp_test5_a_4.log
-print $ASUITE/lst/comp_test5_a_4.lst     $OPTIONS &
sleep 60
sas $ASUITE/code/gsort.sas -sysparm 3      -log $ASUITE/logs/gsort_a_3.log
-print $ASUITE/lst/gsort_a_3.lst         $OPTIONS &
sas $ASUITE/code/ds.sas   -sysparm 1       -log $ASUITE/logs/ds_a_1.log
-print $ASUITE/lst/ds_a_1.lst     $OPTIONS &
sleep 120
sas $ASUITE/code/comp_test2.sas            -log $ASUITE/logs/comp_test2_a_4.log
-print $ASUITE/lst/comp_test2_a_4.lst     $OPTIONS &
sas $ASUITE/code/comp_test5.sas            -log $ASUITE/logs/comp_test5_a_5.log
-print $ASUITE/lst/comp_test5_a_5.lst     $OPTIONS &
sas $ASUITE/code/ds.sas   -sysparm 2       -log $ASUITE/logs/ds_a_2.log
-print $ASUITE/lst/ds_a_2.lst     $OPTIONS &
sleep 5

echo "All jobs are submitted!!!! "

wait

echo "All Done!!!!"

date

echo ""
echo "*************************************************************************"
echo "*** ERROR CHECK - if there are ERRORS between the stars!! Check run!!***"
```

```
echo "*************************************************************************"
echo "ERRORS:  "
grep "ERROR:" $ASUITE/logs/*.log
echo "*************************************************************************"
echo "*************************************************************************"
echo ""
```

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.