TECHNICAL PAPER

Performance and Tuning Considerations on Oracle Cloud Infrastructure with SAS® 9.4 Using IBM Spectrum Scale™

Last update: July 2020

Contents

Introduction

Shared file systems are a necessity when you implement SAS Grid with multiple SAS 9.4 compute nodes in an Oracle Cloud Infrastructure environment. This paper presents recent testing and research performed by SAS and Oracle using SAS 9.4 with IBM Spectrum Scale as the shared file system. It covers a range of topics, from performance implications to best practices and tunings for ideal Oracle Cloud Infrastructure configurations. This paper is the basis for an informal, round-table discussion that will provide some guidance, solicit customer experience feedback, and create group discussion about trends that will be helpful to SAS, the Oracle Cloud Infrastructure High-Performance Computing (HPC) team, and the IBM Spectrum Scale team.

This paper presents test results for the SAS iotest.sh script and SAS mixed analytics workload using Oracle Cloud Infrastructure Compute instances and the Oracle Cloud Infrastructure Block Volumes service. The testing was conducted using SAS 9.4 and the IBM Spectrum Scale 5.0.3.2 file system. The configuration included three VM.DenseIO2.24 instances for the compute nodes and four BM.Standard2.52 instances for the IBM Spectrum Scale Network Shared Disk (NSD) nodes.

Before you dive deep into the test setup, read the following short summary of the test results. For more details, please refer to Test Results.

Summary: The Oracle Cloud Infrastructure resources, combined with the IBM Spectrum Scale shared file system, provided the performance and I/O throughput that is required for a performant SAS Grid deployment. The results from the SAS mixed analytics workload demonstrate the performance benefits of Oracle Cloud block volume storage for high I/O throughput and of IBM Spectrum Scale for building a shared file system.

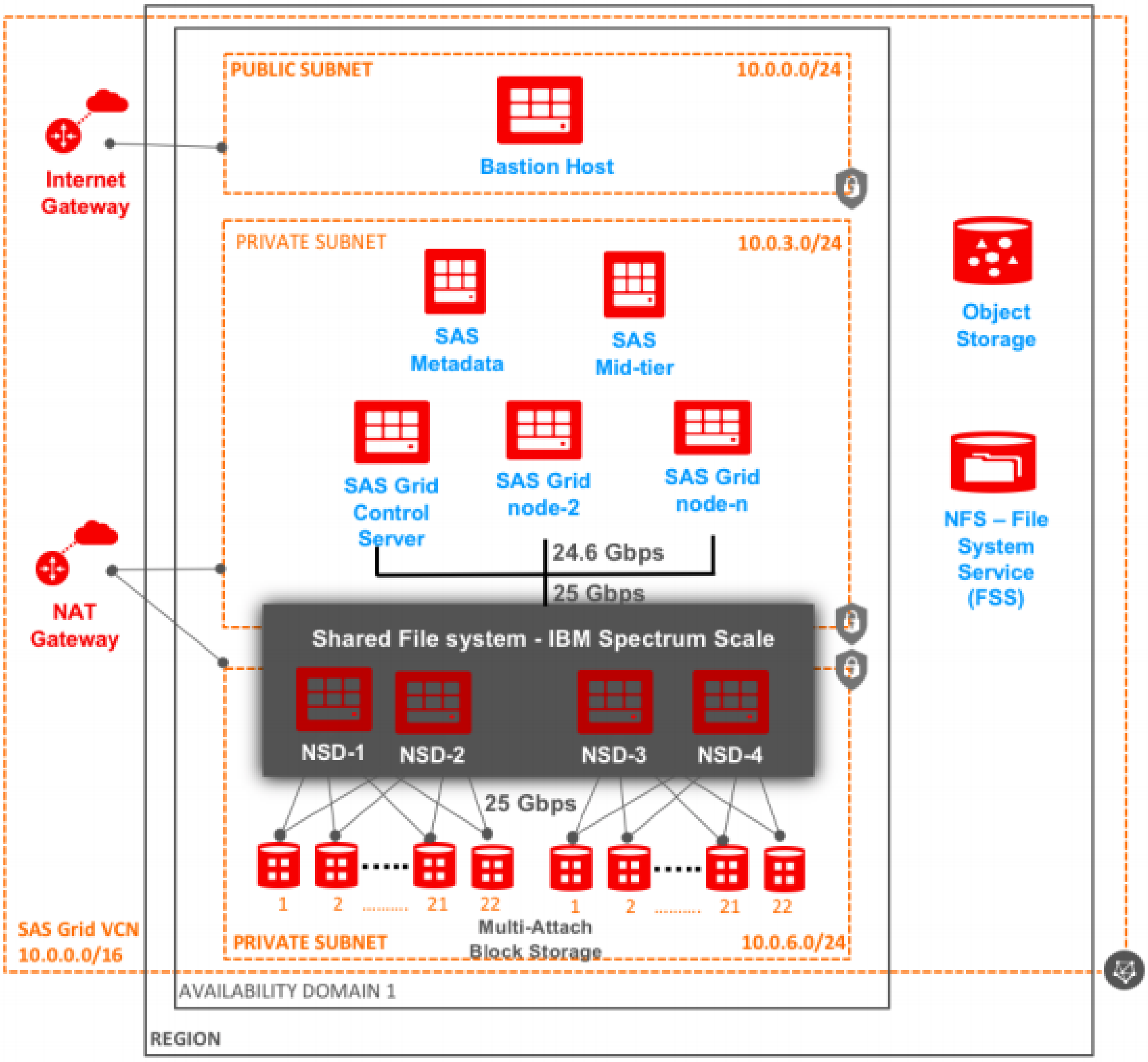
Note: The decision to use the Oracle Cloud Infrastructure implementation was made in the last quarter of 2019. Because Oracle Cloud Infrastructure Compute and Block Volumes offerings are constantly changing, you should understand the rationale that was used in the selection process. Consider what was done for these specific results as a point-in-time design. Future improvements in Oracle Cloud Infrastructure offerings might change the selections that were made here.

Test Environment

Oracle Cloud Infrastructure Networking

Testing was performed in Oracle Cloud Infrastructure using a virtual cloud network (VCN). A VCN provides a logically isolated virtual network in the cloud. Essential components within the VCN include a public subnet, internet gateway, network address translation (NAT) gateway, security list, route table, and multiple private subnets. Figure 1 shows (in Oracle Cloud notation) the system components that were used in the testing environment.

SAS Grid Architecture



Notes:

1. Object Storage – is used to store SAS Depot binaries.
2. OCI FSS – an NFS file system is used for SASHOME and SASCFG for grid nodes. Alternatively, you can also use Spectrum Scale for them.

Figure 1. *Oracle Cloud Infrastructure VCN Configuration for SAS Testing*

The test setup was initially deployed using Oracle Cloud Infrastructure QuickStart Terraform templates that were developed by the Oracle Cloud Infrastructure HPC team. The following templates are available:

- Deploy SAS Grid

- Deploy IBM Spectrum Scale

The compute nodes (grid control server and grid nodes) used VM.DenseIO2.24 virtual machine instances with a network bandwidth of 24.6 Gbit/sec. The storage nodes (NSDs) used bare metal instances of BM.Standard2.52 with a network bandwidth of 2x25 Gbit/sec. Oracle Cloud Infrastructure provides a service-level agreement (SLA) for network throughput between instances in the same availability domain in a VCN. You might think of this configuration as a measurement of LAN performance. This SLA applies only to bare metal instances.

The VCN had a public subnet to run a bastion host on and an internet gateway to route traffic for the internet through. In this type of setup, the outbound traffic in private subnets is routed through the NAT gateway, which provides isolation from public internet traffic while allowing outbound traffic from the private subnet.

The VCN also had two private subnets. One private subnet was used to run the Spectrum Scale NSD daemon network between the compute nodes (primary NIC) and storage nodes (secondary NIC). The storage nodes were also on the second private subnet to separate traffic going to block storage using the primary NIC of the node.

Security groups are also a part of the network design and are important for privacy and security. They act as firewalls and can restrict outbound and inbound traffic. For test purposes, the security groups were minimally configured. Security setup is not covered in this paper.

Test Bed: SAS Mixed Analytics Workload

The test bed was a scaled workload of computation and IO-oriented tests to measure concurrent, mixed job performance. Of interest was whether the compute instances that used IBM Spectrum Scale as the clustered file system (CFS) would yield benefits for SAS large-block sequential IO patterns.

The actual workload consisted of 19 individual SAS tests: 10 computation, 2 memory, and 7 IO-intensive tests. Each test consisted of multiple steps. Some tests relied on existing data stores, and other tests (primarily, computation tests) relied on generated data. The tests were chosen as a matrix of long-running and short-running tests, ranging in duration from approximately five minutes to one hour and 20 minutes. Actual test times vary by hardwareprovisioning differences. In some instances, the same test (running against replicated data streams) was run concurrently and back-to-back (or back-to-back) in a serial fashion to achieve an average of *20 simultaneous streams of heavy IO, computation (fed by significant IO in many cases), and memory stress. In all, to achieve the 20- concurrent test matrix, 77 tests were launched.

Data I/O Throughput

The I/O tests submitted as input an aggregate of approximately 300 GB of data. The computation tests input more than 120 GB of data for a single instance of each SAS mixed analytics workload within 20 simultaneous tests on each node. Much more data is generated as a result of test-step activity and threaded kernel procedures such as the SAS SORT routines. For example, PROC SORT makes three copies of the incoming file to be sorted. As stated, some of the same tests were run concurrently using different data, and some of the same tests were run back-to-back to produce a total average of 20 tests running concurrently. A concurrent running of tests raises the total I/O throughput of the workload significantly.

In Figure 2, the aggregate SASDATA I/O bandwidth quickly exceeds 4GB/sec and achieves a peak of about 5GB/sec with the workload. This is a good, average "SAS shop" throughput characteristic for a three-node cluster. Note that no SASWORK I/O is reflected in Figure 2.

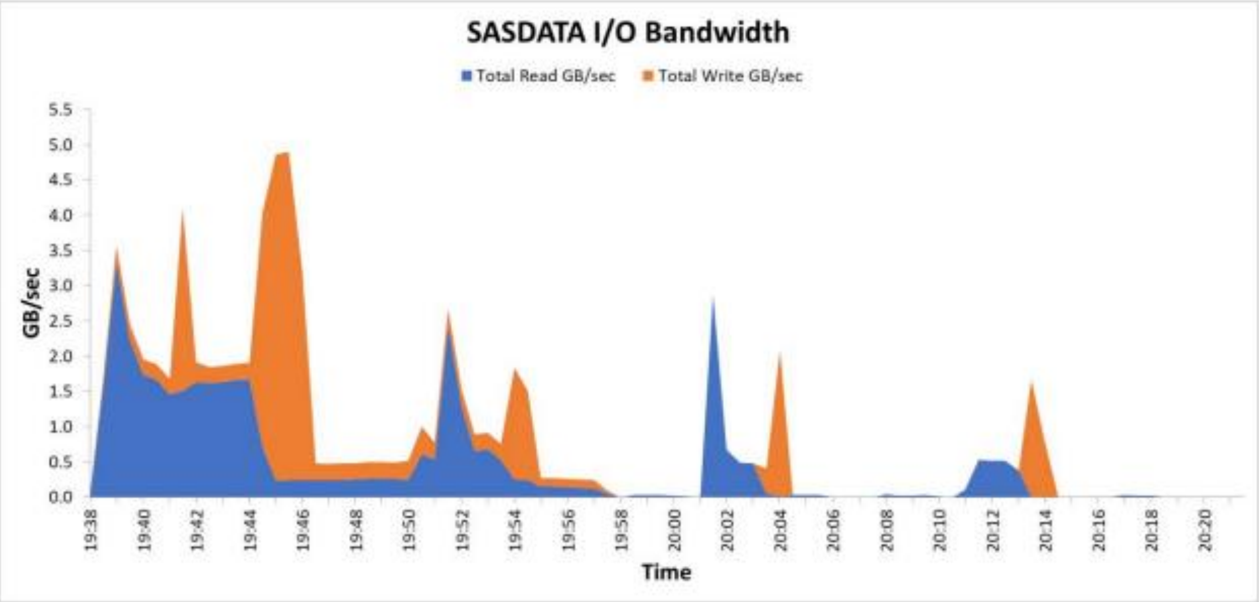


Figure 2. *I/O Bandwidth of SASDATA in Oracle Cloud Infrastructure*

Figure 3. shows the SASWORK I/O Bandwidth

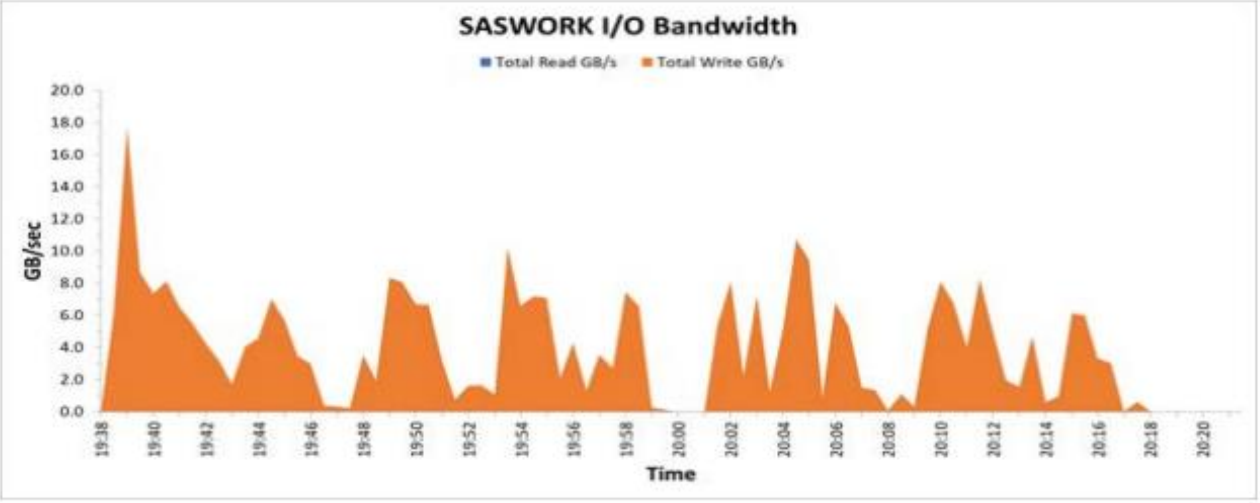


Figure 3. *I/O Bandwidth of SASWORK in Oracle Cloud Infrastructure*

File System I/O Throughput

The I/O throughput of the SAS file systems was tested using the SAS iotest.sh script for CentOS. The script uses Linux dd commands to measure the I/O behavior of the system under defined loads. The script was easy to use for launching individual or multiple concurrent I/O tests to flood the file system and for determining the file system's raw performance.

The script creates files and writes them to the file system that is being tested. It then reads them back to test both the write and read performance of the file system. The primary output from the script is the megabytes written per second (Write MB/sec) and the megabytes read per second (Read MB/sec). These metrics can help analyze and tune I/O for file systems that support SAS.

A good performance target for a typical system is for throughput from the file system to match at least 75 MB/sec per processor core for small to average systems and upward of 100 for larger, more heavily used systems. A minimum throughput rate of 100 MB/sec should be provided for any SAS file system.

The three primary file systems (SASDATA, SASWORK, UTILLOC) for SAS were tested for I/O throughput. For details about test results, see Test Results.

SAS File Systems

Three primary file systems were involved in the testing:

- SASDATA: SAS permanent data file space

- SASWORK: SAS working data file space

- UTILLOC: SAS utility data file space

The SASDATA file system resided on IBM Spectrum Scale. The SASWORK and UTILLOC file systems resided on CentOS Linux XFS.

The XFS file systems were configured on each compute node using local NVMe SSD storage with multiple drives. Local NVMe SSD storage is uniquely able to survive reboots. Data is never lost when a compute instance is rebooted or stopped.

For this workload's code set, data, result space, working space, and utility space, the following space allocations were made:

- SASDATA: 30.8 TB of block volumes for balanced performance storage

- SASWORK: 25.6 TB of local NVMe SSD storage per grid compute node

- UTILLOC: Combined with the SASWORK space

These specifications provide the general size of the workload's on-storage footprint. It is important to note that throughput, not capacity, is the key factor in configuring storage to enhance SAS performance.

With regard to the enormous amount of file space (30.8 TB) reserved for SASDATA, the input data for tests is only a few TBs in size. However, the overprovisioning of storage is required in order to achieve better throughput. To achieve a sustained 320 MB/sec per block volume, we created volumes whose size was greater than 700 GB. For more information about this strategy, see the Block Volumes documentation. In November 2019, after testing, Oracle released block volumes with higher throughput of 480 MB/sec for 800 GB or higher volume sizes.

Hardware

This test bed was run against three compute nodes using the SAS mixed analytics workload with 20 simultaneous tests. The storage nodes were configured as four IBM Spectrum Scale NSD nodes. All compute nodes were identical in configuration, and all storage nodes were identical in configuration.

Compute Nodes

Oracle Cloud Infrastructure offers a variety of compute shapes to choose from for SAS grid nodes. For the remainder of this paper, SAS Grid nodes are referred to as compute nodes.

The VM.DenseIO2.24 shape was chosen for compute nodes. The selection process included consideration of the number of CPU cores, memory per core, onboard disk drives, network speed, maximum throughput, and cost per hour.

Here are the specifications for the compute nodes:

- **Count:** 3 nodes
- **Host:** VM.DenseIO2.24 shape instance
- **Kernel:** Linux 3.10.0-957.27.2.el7.x86_64 (CentOS 7.6.1810 (Core)
- **CPU:** 24 cores, Genuine Intel® Xeon® CPU 2.0 GHz Platinum 8167M
- **Memory:** 320 GB
- **Disks:** 1 x 47 GB block volume virtual disk for OS and system usage, 4 x 6.4 TB local NVMe SSD onboard, and striped together
- **Network:** 24.6 Gbps
- **Maximum throughput:** 3075 MB/sec from system NIC to block volume storage

Two metrics worth noting are the 24-core system and the 3075 MB/sec throughput, which suggest a throughput-percore ratio of 3075/24=128.12 MB/sec/core. SAS applications can operate at higher ratios typically at or above 125 MB/sec/core.

Host Tuning

Host tuning is based on the Red Hat Linux throughput-performance profile with modifications that are performed and saved as sas-performance. For information about the actual modifications and other details, see the Appendix.

Additional Settings Used

The four NVMe disks were formatted and striped as an XFS file system and used for SASWORK and UTILLOC. The file systems were created as a logical RAID0 volume that consists of four LUNs that are striped at a 64 KB-block size to match the SAS BUFSIZE that is used by most SAS applications. For more details about creating the XFS file system and provisioning the directories see the Appendix.

Storage Nodes

Oracle Cloud Infrastructure offers a variety of compute shapes from which to choose for storage nodes and IBM Spectrum Scale NSD nodes. For the remainder of this paper, the IBM Spectrum Scale NSD nodes are referred as storage nodes.

The BM.Standard2.52 shape was chosen for storage nodes. The selection process started with trying to match the network speed and throughput performance of the compute node, which used the VM.DenseIO2.24 shape. The selection process was further narrowed by considering the number of CPU cores, memory per core, and cost per hour. Note that a storage node with a large amount of memory enables a larger IBM Spectrum Scale pagepool size setting. A sufficient pagepool size can improve the effective throughput of the system, but an excessive pagepool size can have a negative performance impact.

Here are the specifications for the storage nodes:

- **Count:** 4 nodes
- **Host:** BM.Standard2.52 shape instance
- **Kernel:** Linux 3.10.0-957.27.2.el7.x86_64 (CentOS 7.6.1810 (Core)
- **CPU:** 52 cores, Genuine Intel® Xeon® CPU 2.0 GHz Platinum 8167M
- **Memory**: 768 GiB
- **Disks:** 1 x 47 GB block volume virtual disk for OS and system usage
- **NFS:** SASHOME and SASCFG were NFS mounted. Data was stored in Oracle Cloud Infrastructure File System service
- **Network**: 2 x 25 Gbps
- **Maximum throughput**: 3125 MB/sec from system NIC to block volume storage

Host Tuning

Host tuning is identical to the compute node, based on the Red Hat Linux throughput-performance profile with modifications performed and saved as “sas-performance”. tuned-adm profile configuration file. For information about the actual modifications and other details, see the Appendix.

Additional Settings Used

Storage is configured as IBM Spectrum Scale NSD nodes. For more information about IBM Spectrum Scale and its settings, see the Appendix.

Oracle Cloud Infrastructure Storage

Oracle Cloud Infrastructure provides several types of storage. Block volume storage was chosen because of its highly available, consistent, and low-latency attributes. Block volume storage offers three performance tiers to select.

- Higher performance elastic performance
- Balanced elastic performance
- Lower cost elastic performance

A balanced elastic performance storage tier was chosen for its large-block I/O capabilities (480 MB/sec for 1 TB or larger volume size). A higher performance elastic performance tier can also be selected for SAS. It delivers 480 MB/sec for an 800 GB or larger volume size.

In general, SAS Foundation requires large-block sequential I/O for the best performance. We concluded that throughput was more important than IOPS for our workload. A lower cost tier was not considered because it was not designed for large-block I/O. For more information, see the Block Volumes documentation.

The compute nodes and storage nodes were selected and matched for throughput at 3125 MB/sec per node (25Gbps network bandwidth). Therefore, each storage node required 10 or 11 volumes of 700 GB (gives 336 MB/sec throughput), which collectively deliver 3360/3696 MB/s (slightly higher than 3125 MB/s) to supply the necessary I/O for the 3125 MB/sec maximum network throughput of the storage nodes.

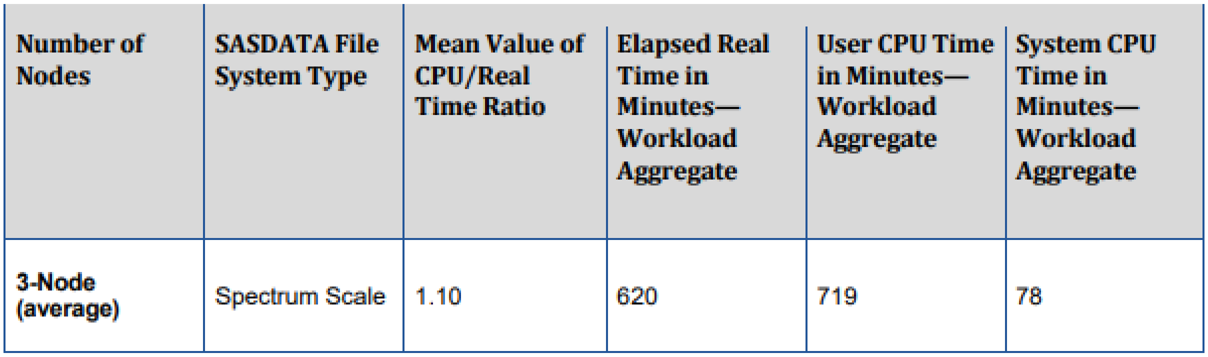
Spectrum Scale was configured for high availability (HA) by configuring a pair of storage nodes to act as primary and secondary for each other. Oracle Cloud Infrastructure allows volume to be attached to multiple instances in sharable read/write mode, similar to storage area network (SAN).Therefore, if one storage node in the HA pair fails, the other storage node has read/write access to 10 or 11 volumes that were accessed by the failed node. Each pair of storage 9 nodes was provisioned with 22 x 700 GB of block volumes for a total of 15.4 TiB for two nodes. The four storage nodes had a total capacity of 30.8 TiB.

Test Results

SAS Mixed Analytics Workload

The SAS mixed analytics workload was run in a “quiet” setting. This means that there was no competing activity on the server or on storage. Multiple runs were performed in order to standardize the results.

Table 1 shows the per-node average performance of a three-node run with SASDATA on IBM Spectrum Scale. The SASWORK and UTILLOC file systems resided on local NVMe SSDs for all the runs. This table shows a frequency mean value of the CPU/Real Time ratio, which is summed from the 77 submitted tests. It shows the summed User CPU Time and the summed System CPU Time in minutes.



Number of Nodes	SASDATA File System Type	Mean Value of CPU/Real Time Ratio	Elapsed Real Time in Minutes—Workload Aggregate	User CPU Time in Minutes—Workload Aggregate	System CPU Time in Minutes—Workload Aggregate
3-Node (average)	Spectrum Scale	1.10	620	719	78

Table 1. Performance

The third column shows the ratio of total CPU Time (User + System CPU) to total Real Time. If the ratio is less than 1, then the CPU is spending time waiting for the availability of resources, usually for I/O. The Oracle Cloud Infrastructure test configurations delivered an excellent ratio of 1.10 of Real Time to CPU Time. A logical question is “How can I get above a ratio of 1.0?” Because some SAS procedures are threaded, you can actually use more CPU cycles than wall-clock time (Real Time).

The fourth column shows the total elapsed Real Time in minutes, which is summed from each of the jobs in the workload. The three-node run with SASDATA on IBM Spectrum Scale executes the aggregate run time of the workload in 620 minutes and in 719 minutes of Real Time per node, respectively.

The primary takeaway from these tests is that the three-node Oracle Cloud Infrastructure configurations using IBM Spectrum Scale easily provided enough throughput to fully exploit this host environment. This configuration could meet the accelerated and scaled throughput demand while providing a very healthy CPU/Real Time ratio per node.

File System I/O Throughput Test Results

SASDATA

The SAS iotest.sh script was concurrently run on all three compute nodes, each with 24 concurrent write, then read, iterations (-i) against the SASDATA IBM Spectrum Scale file system that was mounted at /gpfs/fs1. The block size (-s) of 64K was chosen to match the SAS buffer size. The number of blocks (-b) was set to 5259264, which means that each thread writes a file of 328 GB. This size is larger than the 320 GB of RAM on the machine that was used in this test.

```
iotest.sh -i 24 -t /gpfs/fs1/test3 -b 5259264 -s 64
```

Oracle Cloud Infrastructure delivered an average I/O throughput rate of 115 MB/s for read and 98 MB/s for write. For detailed information about the test results, see the Appendix.

SASWORK and UTILLOC

The SAS iotest.sh script was run separately on each compute node, each with 24 concurrent write, then read, iterations (-i) against the SASWORK file system mounted at /sas/SASWORK. The block size (-s) of 64K was chosen to match the SAS buffer size. The number of blocks (b) was set to 3355438, which means that each thread writes a file of 209GB.

```
iotest.sh -i 24 -t /sas/SASWORK/test1 -b 3355438 -s 64
```

Oracle Cloud Infrastructure delivered an average I/O throughput rate of 320 MB/s for read and 194 MB/s for write. For detailed information about the test results, see the Appendix.

The same results apply to the UTILLOC file system because it was combined with SASWORK.

Conclusion

The Oracle Cloud Infrastructure VM.DenseIO2.24 instance, combined with the IBM Spectrum Scale shared file system, provided the performance and I/O throughput that is required for a performant SAS Grid deployment. The BM.Standard2.52 server instance was used for the storage nodes and proved to be a good choice because it matched the performance metrics of the compute nodes for throughput, network bandwidth, and memory.

The SAS mixed analytics workload results demonstrate the performance benefits of Oracle Cloud block volume storage for high I/O throughput and of using it to build a shared file system using IBM Spectrum Scale.

The guidelines listed in this paper are beneficial and are therefore recommended. Your experience might require additional guidance by Oracle Cloud and SAS engineers, depending on your workload characteristics.

A final note: The workload for this testing attempts to model a mixed representation of what is a typical SAS computing environment. Because your results might vary from those that are presented in this paper, it is crucial that you plan and perform your own tests to confirm the viability of this solution for your particular needs. Oracle Cloud Infrastructure might be able to satisfy a business need for your company in this domain of high-performance analytics. Be sure to consider all factors that are related to performance and cost so that you can set the proper expectations.

Acknowledgments

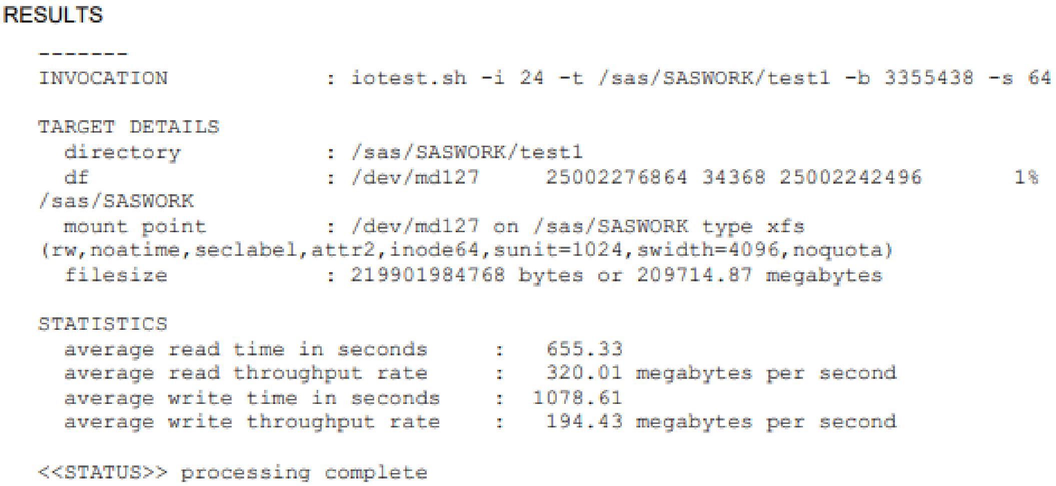
This project was a collaboration between Oracle and SAS. Special thanks go to the SAS Performance team (Margaret Crevar, Jim Kuell), the SAS Partners team (Richard O’Brien), and the Oracle ISV Partners team (Jesse Adelson) for support during this project.

Appendix

Raw Test Results from iotest.sh

SASWORK and UTILLOC File System

```
[root@grid-1 ~]# /home/opc/sas/iotest.sh -i 24 -t /sas/SASWORK/test1 -b 3355438 -s 64 | tee
saswork_iotest.sh_24_result
```

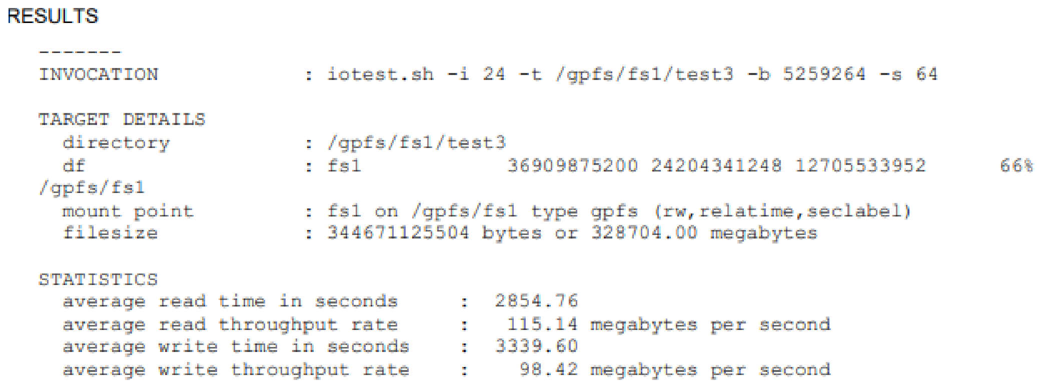


```
RESULTS
  -------
  INVOCATION             : iotest.sh -i 24 -t /sas/SASWORK/test1 -b 3355438 -s 64

  TARGET DETAILS
    directory            : /sas/SASWORK/test1
    df                   : /dev/md127     25002276864 34368 25002242496       1%
  /sas/SASWORK
    mount point          : /dev/md127 on /sas/SASWORK type xfs
  (rw,noatime,seclabel,attr2,inode64,sunit=1024,swidth=4096,noquota)
    filesize             : 219901984768 bytes or 209714.87 megabytes

  STATISTICS
    average read time in seconds      :   655.33
    average read throughput rate      :   320.01 megabytes per second
    average write time in seconds     :  1078.61
    average write throughput rate     :   194.43 megabytes per second

  <<STATUS>> processing complete
```

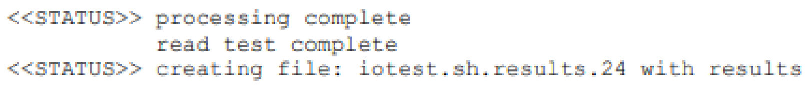
SAS Data File System

```
[root@grid-1 ~]# for i in {1..3};do ssh grid-${i} "mkdir -p /gpfs/fs1/test${i}" ; ssh grid-${i} "/home/opc/sas/iotest.sh -
```

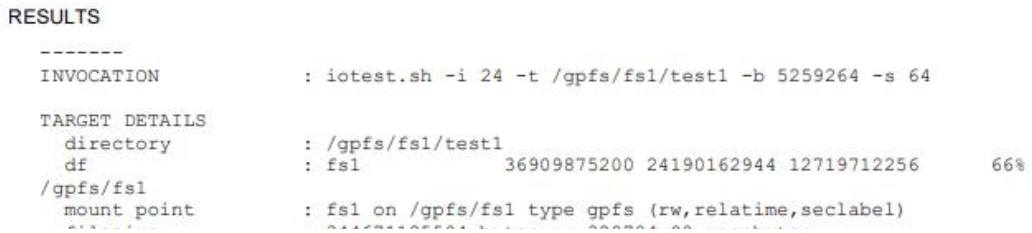
i 24 -t /gpfs/fs1/test${i} -b 5259264 -s 64 " & done



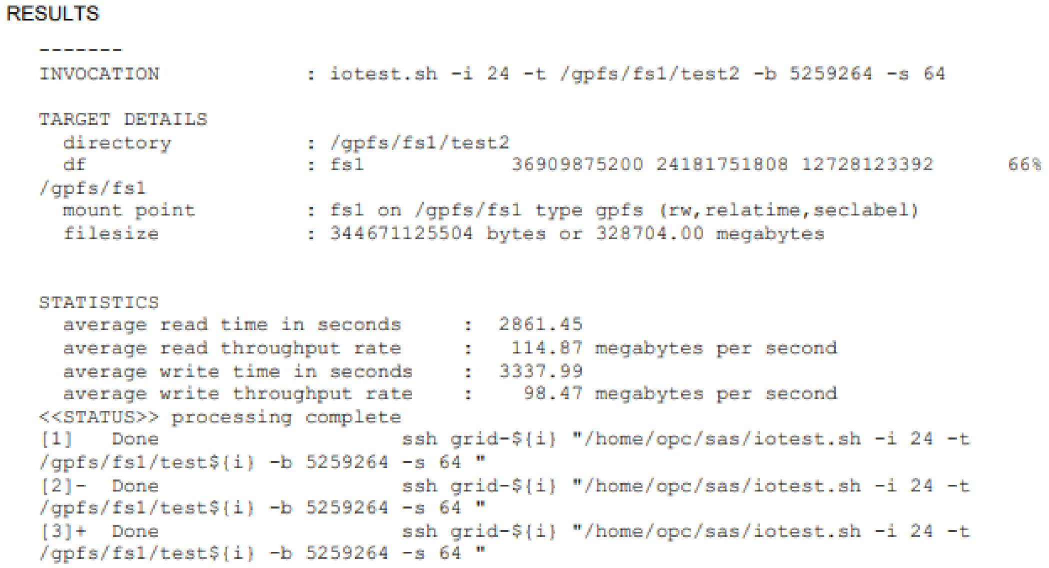
```
RESULTS

  -------
  INVOCATION             : iotest.sh -i 24 -t /gpfs/fs1/test3 -b 5259264 -s 64

  TARGET DETAILS
    directory            : /gpfs/fs1/test3
    df                   : fs1           36909875200 24204341248 12705533952      66%
  /gpfs/fs1
    mount point          : fs1 on /gpfs/fs1 type gpfs (rw,relatime,seclabel)
    filesize             : 344671125504 bytes or 328704.00 megabytes

  STATISTICS
    average read time in seconds     :  2854.76
    average read throughput rate     :   115.14 megabytes per second
    average write time in seconds    :  3339.60
    average write throughput rate    :    98.42 megabytes per second
```



```
<<STATUS>> processing complete
           read test complete
<<STATUS>> creating file: iotest.sh.results.24 with results
```



```
RESULTS

  -------
  INVOCATION             : iotest.sh -i 24 -t /gpfs/fs1/test1 -b 5259264 -s 64

  TARGET DETAILS
    directory            : /gpfs/fs1/test1
    df                   : fs1           36909875200 24190162944 12719712256      66%
  /gpfs/fs1
    mount point          : fs1 on /gpfs/fs1 type gpfs (rw,relatime,seclabel)
```



```
RESULTS

  -------
  INVOCATION             : iotest.sh -i 24 -t /gpfs/fs1/test2 -b 5259264 -s 64

  TARGET DETAILS
    directory            : /gpfs/fs1/test2
    df                   : fs1           36909875200 24181751808 12728123392      66%
  /gpfs/fs1
    mount point          : fs1 on /gpfs/fs1 type gpfs (rw,relatime,seclabel)
    filesize             : 344671125504 bytes or 328704.00 megabytes


  STATISTICS
    average read time in seconds     :  2861.45
    average read throughput rate     :   114.87 megabytes per second
    average write time in seconds    :  3337.99
    average write throughput rate    :    98.47 megabytes per second
  <<STATUS>> processing complete
  [1]   Done                    ssh grid-${i} "/home/opc/sas/iotest.sh -i 24 -t
  /gpfs/fs1/test${i} -b 5259264 -s 64 "
  [2]-  Done                    ssh grid-${i} "/home/opc/sas/iotest.sh -i 24 -t
  /gpfs/fs1/test${i} -b 5259264 -s 64 "
  [3]+  Done                    ssh grid-${i} "/home/opc/sas/iotest.sh -i 24 -t
  /gpfs/fs1/test${i} -b 5259264 -s 64 "
```

IBM Spectrum Scale

IBM Spectrum Scale is a powerful shared file system that provides world-class reliability, scalability, and availability for cloud, big data analytics, and high-performance computing environments. IBM Spectrum Scale simplifies data 14 management with integrated tools that are designed to help manage from gigabytes to petabytes of data and thousands to billions of files. Some key features of IBM Spectrum Scale follow:

- Unified block, file, and object storage

- Massively parallel data access for high performance

- True software-defined storage deployment using either cloud storage on-premises commodity hardware or the powerful Elastic Storage Server grid architecture

- Integrated user interface with the entire IBM Spectrum storage family for simplified administration

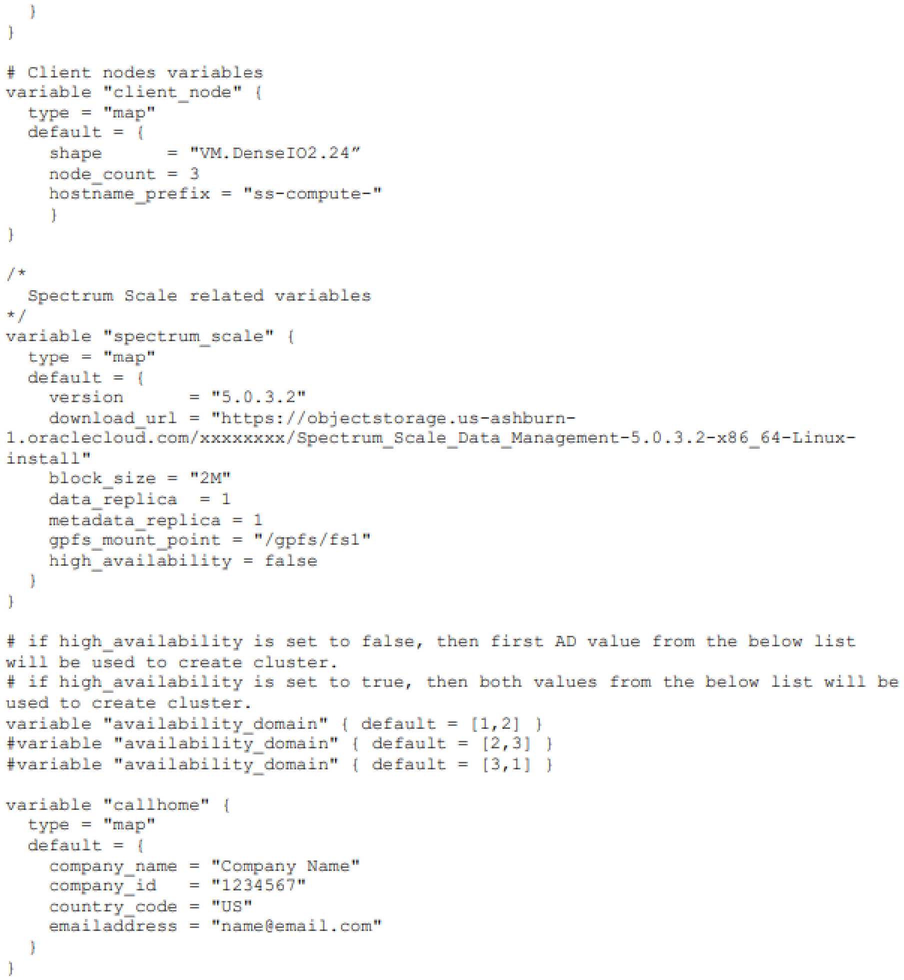
- Comprehensive data life cycle management tools, including transparent policy-driven data migration, highspeed metadata scanning, and native data compression and encryption

The SAS performance team described IBM Spectrum Scale as “a mature and scalable clustering product that has been tested and proven with SAS workloads to have specific advantages compared to competing products.”

Terraform Template Used to Deploy IBM Spectrum Scale

The Oracle Cloud Infrastructure Terraform template that was used to create the VCN, security list, internet gateway, NAT gateway, bastion host, and other components of the test environment is located at https://github.com/oracle-quickstart/oci-ibm-spectrum-scale/tree/master/network_shared_disk_server_model.

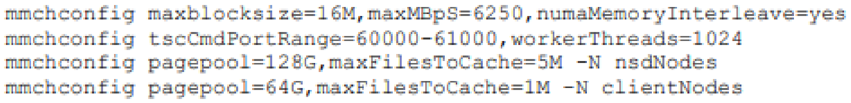
The Spectrum Scale cluster was built with the following parameters:



```
    }
  }

  # Client nodes variables
  variable "client_node" {
    type = "map"
    default = {
      shape      = "VM.DenseIO2.24”
      node_count = 3
      hostname_prefix = "ss-compute-"
      }
  }

  /*
    Spectrum Scale related variables
  */
  variable "spectrum_scale" {
    type = "map"
    default = {
      version      = "5.0.3.2"
      download_url = "https://objectstorage.us-ashburn-
1.oraclecloud.com/xxxxxxxx/Spectrum_Scale_Data_Management-5.0.3.2-x86_64-Linux-
install"
      block_size = "2M"
      data_replica  = 1
      metadata_replica = 1
      gpfs_mount_point = "/gpfs/fs1"
      high_availability = false
    }
  }

  # if high_availability is set to false, then first AD value from the below list
  will be used to create cluster.
  # if high_availability is set to true, then both values from the below list will be
  used to create cluster.
  variable "availability_domain" { default = [1,2] }
  #variable "availability_domain" { default = [2,3] }
  #variable "availability_domain" { default = [3,1] }

  variable "callhome" {
    type = "map"
    default = {
      company_name = "Company Name"
      company_id   = "1234567"
      country_code = "US"
      emailaddress = "name@email.com"
    }
  }
```

IBM Spectrum Scale Tunable Parameters

The IBM Spectrum Scale clustered parallel file system has many tunable parameters. Typically, only a few of these are changed to improve performance for SAS. The following commands were used to configure the file system for Oracle Cloud Infrastructure testing:

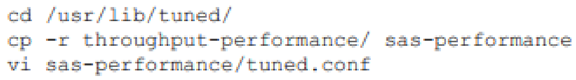


```
mmchconfig maxblocksize=16M,maxMBpS=6250,numaMemoryInterleave=yes
mmchconfig tscCmdPortRange=60000-61000,workerThreads=1024
mmchconfig pagepool=128G,maxFilesToCache=5M -N nsdNodes
mmchconfig pagepool=64G,maxFilesToCache=1M -N clientNodes
```

CentOS Tuning of the Oracle Cloud Infrastructure Instances

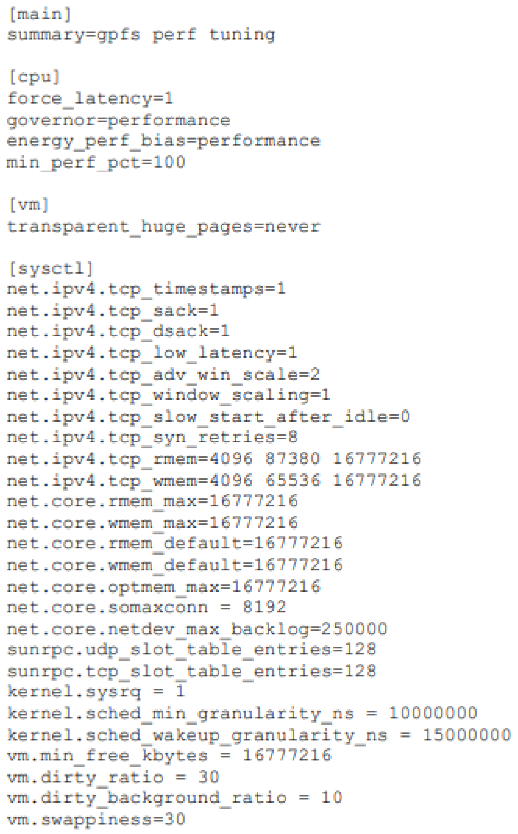
The following settings used for the BM.Standard2.52 storage nodes and the VM.Standard2.24 compute nodes are based on the CentOS Enterprise Linux 7.6 OS. Any other version of the OS might require different settings. For more information about configuring CentOS and Red Hat Linux systems for SAS workloads, see Optimizing SAS on Red Hat Enterprise Linux (RHEL) 6 & 7.

Here are the steps for tuning the Oracle Cloud Infrastructure instances:
1. Navigate to the tuned directory, and create a sas-performance profile as a copy of the throughputperformance profile. Edit its tuned.conf file.



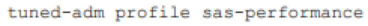
```
cd /usr/lib/tuned/
cp -r throughput-performance/ sas-performance
vi sas-performance/tuned.conf
```

2. Edit the tuned.conf file to appear as follows:



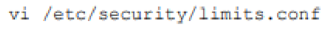
```
[main]
summary=gpfs perf tuning

[cpu]
force_latency=1
governor=performance
energy_perf_bias=performance
min_perf_pct=100

[vm]
transparent_huge_pages=never

[sysctl]
net.ipv4.tcp_timestamps=1
net.ipv4.tcp_sack=1
net.ipv4.tcp_dsack=1
net.ipv4.tcp_low_latency=1
net.ipv4.tcp_adv_win_scale=2
net.ipv4.tcp_window_scaling=1
net.ipv4.tcp_slow_start_after_idle=0
net.ipv4.tcp_syn_retries=8
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.optmem_max=16777216
net.core.somaxconn = 8192
net.core.netdev_max_backlog=250000
sunrpc.udp_slot_table_entries=128
sunrpc.tcp_slot_table_entries=128
kernel.sysrq = 1
kernel.sched_min_granularity_ns = 10000000
kernel.sched_wakeup_granularity_ns = 15000000
vm.min_free_kbytes = 16777216
vm.dirty_ratio = 30
vm.dirty_background_ratio = 10
vm.swappiness=30
```

3. Set the new tuned profile as the active profile.



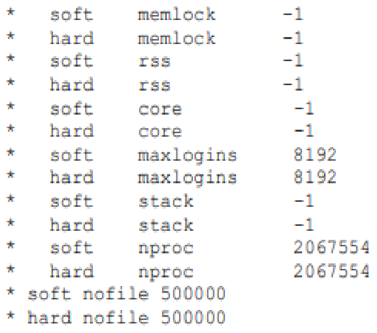
```
tuned-adm profile sas-performance
```

4. Update the limits.conf file:



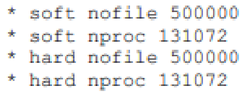
```
vi /etc/security/limits.conf
```

5. On the storage nodes, add the following two lines at the end of the file:



```
*   soft    memlock      -1
*   hard    memlock      -1
*   soft    rss          -1
*   hard    rss          -1
*   soft    core          -1
*   hard    core          -1
*   soft    maxlogins     8192
*   hard    maxlogins     8192
*   soft    stack         -1
*   hard    stack         -1
*   soft    nproc         2067554
*   hard    nproc         2067554
* soft nofile 500000
* hard nofile 500000
```

6. On the compute nodes, add the following two lines at the end of the file:



```
 * soft nofile 500000
 * soft nproc 131072
 * hard nofile 500000
 * hard nproc 131072
```

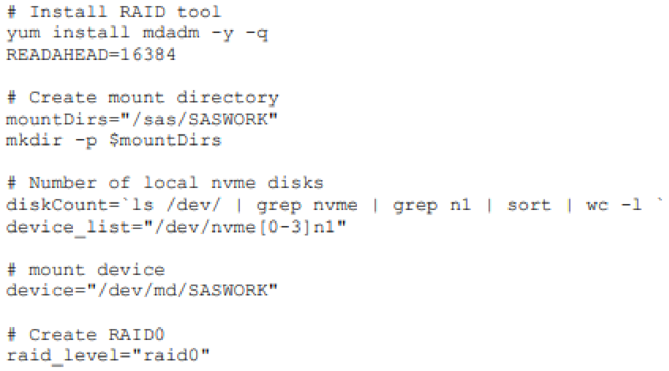
7. Log off and then back on, and verify changes with ulimit -a.

Provisioning the SASWORK Storage

The SAS compute nodes are VM.DenseIO2.24 compute shapes. Each node includes four 6.4 TiB NVMe disk drives that are used as a combined SASWORK and UTILLOC directory. The NVMe drives are a persistent (not an ephemeral) disk, so that data is not lost if the instance is rebooted.

To use these local NVMe persistent drives to their best advantage, the following commands were executed.

Because use cases and environments vary, consider these commands as models for similar tasks when creat local file systems to use with SAS. In this specific case, the four drives were striped as a RAID0 system (default chunksize is 64 KB) and a 16 MB readahead.



```
# Install RAID tool
yum install mdadm -y -q
READAHEAD=16384

# Create mount directory
mountDirs="/sas/SASWORK"
mkdir -p $mountDirs

# Number of local nvme disks
diskCount=`ls /dev/ | grep nvme | grep n1 | sort | wc -l `
device_list="/dev/nvme[0-3]n1"

# mount device
device="/dev/md/SASWORK"

# Create RAID0
raid_level="raid0"
```

```
echo -e "RAID level of $raid_level for $diskCount disk."
echo "DEVICE $device_list" >  /etc/mdadm.conf
echo "ARRAY ${device} devices=$device_list" >> /etc/mdadm.conf
mdadm -C ${device} --level=$raid_level --raid-devices=$dcount $device_list

#set readahead for RAID volumes - /dev/md/SASWORK
blockdev --setra $READAHEAD $device

# Create XFS file system
mkfs.xfs -b size=4096 $device

# Mount the SAS WORK file system
mount -t xfs -o noatime $device $mountDirs

# Add to fstab
UUID=$(lsblk -no UUID $device)
echo "UUID=$UUID   $mountDirs    xfs
defaults,noatime,_netdev,nofail,discard,barrier=0 0 1" | sudo tee -a /etc/fstab
```

Release Information Content Version: 1.0 July 2020

Trademarks and Patents SAS Institute Inc. SAS Campus Drive, Cary, North Carolina 27513

To contact your local SAS office, please visit: sas.com/offices