Paper 4632-2020

Accelerating SAS Using High-Performing File Systems on Amazon Web Services

Darryl S. Osborne, Amazon Web Services

ABSTRACT

When running efficient, high-performing SAS® applications in the cloud is critical to your business, deploying the right storage architecture could be the difference between success and failure. However, running a high-performing parallel, distributed file system is difficult and often very expensive to setup, run, and maintain. What if you could offload this to someone else, someone with expertise and with virtually unlimited resources? You can! This paper guides you through selecting Amazon FSx for Lustre when running SAS Grid on AWS. It also discusses the different types of cloud servers available on AWS and which are best suited to access these high-performing file systems. This in-depth analysis considers the needs of the different SAS tiers and gives recommendations based on compute, memory, and network performance configurations, plus tips and tricks to help you get the most out of your investment.

INTRODUCTION

SAS Grid is a highly available, fast processing analytics platform that offers centralized management that balances workloads across different compute nodes. This application suite is capable of data management, visual analytics, governance and security, forecasting and text mining, statistical analysis, and environment management. It is designed to access and process large amounts of data efficiently by distributing tasks across different compute nodes managed by SAS Grid Manager. This accelerates job processing by distributing tasks across a pool of shared resources to execute multiple jobs and tasks in parallel. Many organizations are using SAS Grid to analyze their data and make critical business decisions. They also want to spend less time deploying, managing, and troubleshooting infrastructure, and more time focusing on their core business. Many of them are now making decisions to move from their on-premises data centers to the cloud.

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 175 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster. These customers want to spend less time deploying, managing, and troubleshooting infrastructure, and more time focusing on their core business.

Once the decision has been made to run SAS applications on AWS, further decisions are needed to design, deploy, and maintain a well architected environment. There are two decisions every SAS Grid customer needs to make when deciding how to architect for AWS. First is which file system to use to share permanent files between SAS Grid compute nodes. Second is which Amazon Elastic Compute Cloud (EC2) instance type is optimized for the selected shared file system while also satisfying SAS Grid compute node specifications and requirements. Getting these two decisions right is critical to your business and could be the difference between success or failure.

SHARED FILE SYSTEMS

SAS Grid is a highly available, fast processing analytics platform offering centralized job and task management. SAS Grid Manager distributes compute tasks among different compute nodes, allowing multiple jobs and tasks to run in parallel. It requires a shared file system to permanently store files shared by compute nodes. It is recommended that this shared file system be a parallel file system accessible from multiple Amazon EC2 instances and achieve millions of IOPS and gigabytes per second throughput with consistent sub-millisecond latencies.

AMAZON FSx FOR LUSTRE

Because you're not in the business of running large scale, cloud optimized parallel file systems, we recommend using Amazon FSx for Lustre for SAS Grid. This allows you to focus more on running your business and the SAS application and less on managing a high performing shared file system.

Amazon FSx if a fully managed file storage service that delivers third-party file systems so you no longer have to worry about the design, deployment, ongoing administration and overall complexity of running these file systems. There are multiple third-party file system types within Amazon FSx but we prefer and recommend using Amazon FSx for Lustre when running SAS Grid on AWS.

The open source Lustre file system is designed for applications that require fast storage – where you want your storage to keep up with your compute. Lustre was built to quickly and cost effectively process the fastest-growing data sets in the world, and it's the most widely used file system for the 500 fastest computers in the world. It provides sub-millisecond latencies, up to hundreds of gigabytes per second of throughput, and millions of IOPS.

Now as a fully managed service, Amazon FSx enables you to use Lustre file systems for any workload where storage speed matters. It eliminates the traditional complexity of setting up and managing Lustre file systems, allowing you to spin up a high-performance file system in minutes. It also provides multiple deployment options to optimize cost.

FSx for Lustre integrates with other AWS services like Amazon Virtual Private Cloud (VPC), AWS Key Management Service (KMS), AWS Parallel Cluster, Amazon Elastic Kubernetes Service (EKS), Amazon SageMaker, Amazon CloudWatch, AWS CloudTrail, and Amazon Simple Storage Service (S3). For those of you not familiar with S3, it is an object storage service that offers industry-leading scalability, data availability, security, and performance. It provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. It is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world. When linked to an S3 bucket, FSx for Lustre transparently presents S3 objects as files. FSx for Lustre tracks changes and enables you to write changed and new data on the file system back to your S3 bucket at any time. SAS Grid customers using FSx for Lustre can benefit from this integration in two ways. First it provides a simple mechanism to archive files to S3. Second it provides a simple and fast mechanism to duplicate file systems. I'll share more information on how to leverage these benefits later in the whitepaper.

Amazon FSx for Lustre is POSIX-compliant, so you can use your current Linux-based applications without having to make any changes. FSx for Lustre provides a native file system interface and works as any file system does with your Linux operating system. It also provides read-after-write consistency and supports file locking. You can control access to your FSx for Lustre file systems with POSIX permissions and Amazon Virtual Private

Cloud (VPC) rules. You can access your file systems from Amazon EC2 instances, and from on-premises computers using AWS Direct Connect or AWS VPN.

Amazon FSx for Lustre automatically encrypts your data at-rest and in-transit. If you are subject to regulatory compliance, FSx for Lustre is PCI-DSS, ISO, SOC, GDPR compliant, and is HIPAA eligible. You can also control network access via Amazon VPC Security Group rules.

Amazon FSx for Lustre delivers the performance to satisfy a wide variety of high-performance workloads. The Lustre file system is optimized for data processing, with sub-millisecond latencies and throughput that scales to hundreds of gigabytes per second.

Amazon FSx for Lustre offers a choice between scratch and persistent file systems for short-term and longer-term data processing. Scratch file systems are ideal for temporary storage and shorter-term processing of data. Data is not replicated and does not persist if a file server fails. Persistent file systems are ideal for longer-term storage and workloads. In persistent file systems, data is replicated, and file servers are replaced if they fail. Because of this higher data durability and higher availability of persistent file systems, we recommend persistent file system for all type of data for SAS Grid, including SASDATA, SASWORK, and UTILLOC.

Table 1 compares persistent and scratch file systems.

	Persistent	Scratch 2
API Name	PERSISTENT_1	SCRATCH_2
Availability and Durability	Metadata and storage servers automatically replaced on failure. Storage replicated within the same Availability Zone (AZ).	Only metadata servers automatically replaced on failure. Storage is not replicated.
Aggregated Throughput (Per TiB of Storage Capacity)	50 MB/s per TiB 100 MB/s per TiB 200 MB/s per TiB	200 MB/s per TiB Burst to 1,200 MB/s per TiB
IOPS	Millions	Millions
Latency	sub-millisecond	sub-millisecond
Workload Types	Jobs that are sensitive to file system failures and need data to be persistent	Jobs that can be re-run on file system failures
Encryption at Rest	AWS managed CMKs, or Customer Managed CMKs	AWS managed CMKs
Encryption in Transit	Yes, when accessed from supported EC2 instances in these regions	Yes, when accessed from supported EC2 instances in these regions
Storage Allocation	1.2 TiB, 2.4 TiB with increments of 2.4 TiB	1.2 TiB, 2.4 TiB with increments of 2.4 TiB

Table 1 Amazon FSx for Lustre Deployment Options

We recommend using persistent file systems for all SAS Grid libraries (SASDATA, SASWORK, UTILLOC). The high availability of these file systems, along with gigabytes per second throughput, millions of IOPS, sub-millisecond latencies, and encryption of data at rest and in transit aligns with the characteristics needed for SAS applications.

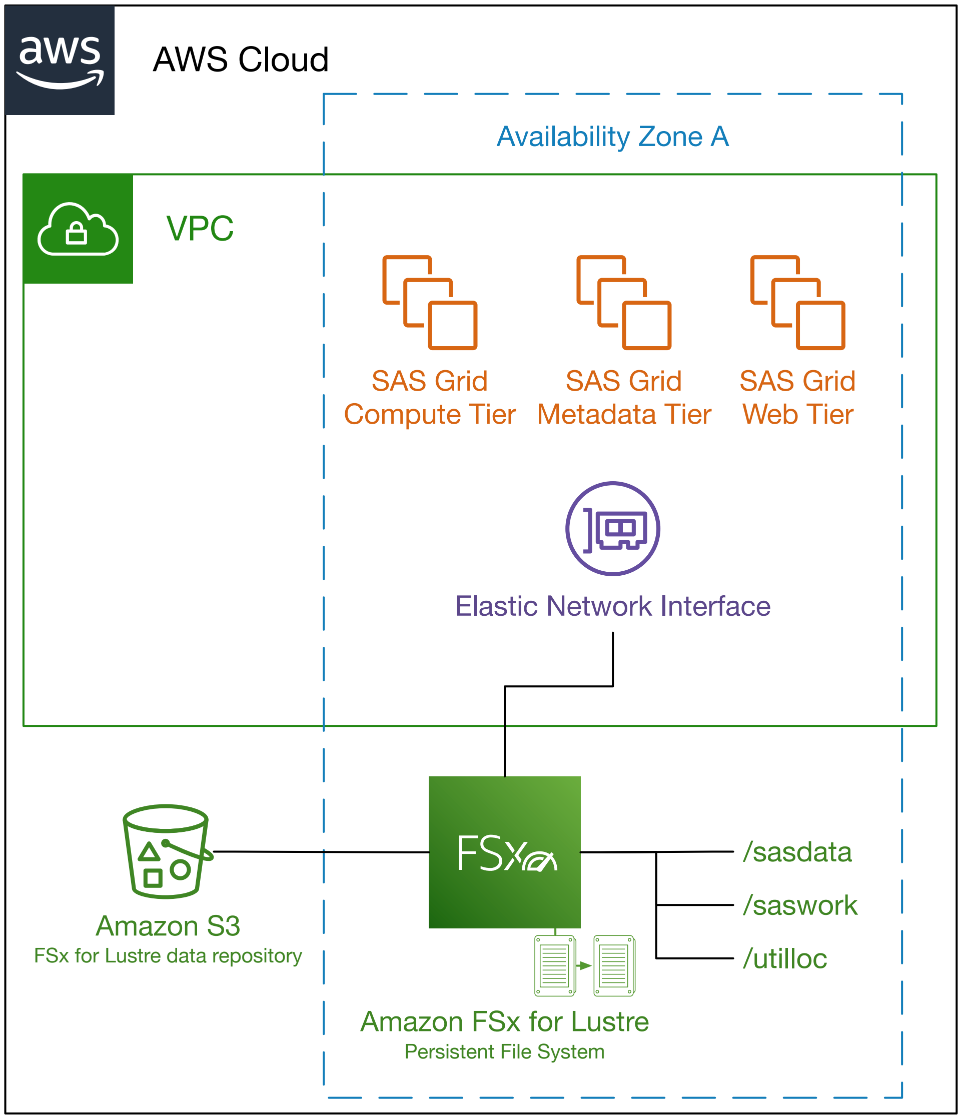

Figure 1 Sample FSx for Lustre Architecture Diagram for SAS Grid

Every AWS account has a default soft limit 100 file systems per region. This limit can be increased to thousands by contacting AWS Support. There is also a default soft limit on the total storage capacity of all file systems in a region. Depending on the region, this default soft limit could be 25.2 TiB or 100.8 TiB. This too can be increased to petabytes by contacting AWS Support.

We recommend selecting the file system’s storage capacity based on the greater of these two attributes; either the total storage capacity needed for your file system or the total throughput needed for your workload based on 50, 100, or 200 MB/s per TiB of the total storage capacity needed for your file system.

Getting Started with Amazon FSx for Lustre and SAS Grid

It takes only minutes to get started using FSx for Lustre with SAS Grid. Below are the steps to get started.

1. Open the Amazon FSx console – https://console.aws.amazon.com/fsx/.
2. Choose *Create file system*.
3. Choose *FSx for Lustre*, then choose *Next*.
4. Provide a descriptive name for the file system, e.g. *SAS Grid*.
5. For *Deployment type*, choose *Persistent*.
6. Provide a storage capacity in TiB.
7. Choose the *Per unit storage throughput*, either 50, 100, or 200 MB/s per TiB of storage capacity. If migrating for an existing file system, review your file systems performance metrics to help determine the level of throughput you should select.
8. Provide the VPC and VPC security group information.
9. Choose an AWS Key Management Service (KMS) encryption key to encrypt the data at rest.
10. For *Data repository integration*, choose *Amazon S3* and provide the name of an S3 bucket (with optional import prefix) that will be linked to the file system.
11. For *Export prefix*, choose *The same prefix that you imported from (replace existing objects with updated ones*).
12. Choose *Next*.
13. Review the attributes of the file system and select *Create file system*.

Within ~5 minutes the file system will be available and its status will change from creating to available. Creation time is roughly the same regardless of the file system size (1.2 TiB, 100.8 TiB, etc.), as Lustre resources are created in parallel. If an Amazon S3 data repository is linked to the file system, S3 objects' names and prefixes will be visible as files and directories. The number of objects in the linked S3 data repository may increase creation time. You now have the world's most popular high-performance parallel file system, capable of driving hundreds of gigabytes per second, millions of IOPS, with sub-millisecond latencies.

Amazon FSx for Lustre supports access from the 2.10 versions of the Lustre client. This client is available for multiple Linux distributions, including Amazon Linux, Amazon Linux 2, CentOS and Red Hat 7.5, 7.6, 7.7, and newer, SUSE Linux 12 SP3, and Ubuntu 16.04 and 18.04. Red Hat 8.0 is not currently supported. For the latest information on supported Linux distributions and Lustre client install instructions, please refer to the Installing the Lustre Client section of the Amazon FSx for Lustre User Guide.

Both SAS and AWS recommend mounting Lustre with the `flock` mount option. Below is a sample mount command and mount output for FSx for Lustre:

```
$ sudo mount -t lustre -o noatime,flock fs-0123456789abcd.fsx.us-west-
2.amazonaws.com@tcp:/za3atbmv /fsx

$ mount -t lustre
172.31.41.37@tcp:/za3atbmv on /fsx type lustre
```

```
(rw,noatime,seclabel,flock,lazystatfs)
```

Optional configuration and advanced features

As previously mentioned, the native integration between Amazon FSx for Lustre and Amazon S3 introduces exciting features and capabilities. When linked to an S3 bucket, FSx for Lustre transparently presents S3 objects as files and directories. FSx for Lustre tracks changes and enables you to write new and changed data on the file system back to your S3 bucket at any time. This provides a simple mechanism – data repository tasks – to backup or archive SASDATA to S3. Data repository tasks optimize data and metadata transfers between your FSx for Lustre file system and its data repository on S3. These tasks transfer file data, symbolic links (symlinks), and Portable Operating System Interface (POSIX) metadata, including ownership, permissions, and timestamps. When you export a file or directory, your file system exports only data files and metadata that were created or modified since the last export.

Because FSx for Lustre stores exported files as native objects in S3, you can use advanced features of S3 to manage this data. Consider enabling S3 Cross-Region replication (CRR) or Same-Region replication (SRR) to copy objects across S3 buckets in different or the same region. We recommend enabling object versioning on the S3 bucket so multiple versions of files are maintained in S3. This enables you to recover previous versions of files on demand using standard S3 utilities. We also recommend setting up S3 lifecycle policies that transitions files to the S3 Glacier cold storage class or deletes them with an expiration policy.

You can start an export task from the Amazon FSx console or using the FSx API. Depending on your recovery point objectives (RPO) and recovery time objectives (RTO), consider developing a solution that schedules the export data repository task on run frequently. You can automate this by using standard time-based job schedulers like cron or an Amazon CloudWatch scheduled event running an AWS Lambda function. You can find an example of this solution on the Amazon FSx for Lustre tutorial github repo.

You can create a new file system using the data and metadata archived to S3. Create a new file system and choose the same S3 bucket as the data repository. File metadata is imported into the new file system during creation and FSx for Lustre will lazy load data from S3 upon first access. This allows you to easily create replacement or duplicate file systems based on the data and metadata stored in S3.

The configuration of Amazon FSx for Lustre is optimized for workloads where speed matters, such as machine learning, high performance computing (HPC), video processing, and financial modeling. These default settings provide optimized performance for the majority of users but Lustre provides advanced configuration settings to further optimize it based on your requirements.

All file data in Lustre is stored on disks called object storage targets (OSTs). All file metadata (including file names, timestamps, permissions, etc) is stored on disks called metadata targets (MDTs). Amazon FSx for Lustre file systems are composed of a single MDT and multiple OSTs, each of which is built on SSD storage. Each OST is approximately 1.17 TiB in size. FSx for Lustre automatically spreads your file data across OSTs that make up your file system to balance storage capacity with throughput and IOPS load. To view the listing and storage usage of the MDT and OSTs that make up your file system, run the following command from a client that has the file system mounted:

```
lfs df -h mount/path
```

The output of this command looks like this but the size of the MDT and the number of OSTs will vary depending on the storage capacity of your file system.

```
UUID                           bytes        Used   Available Use% Mounted on
mountname-MDT0000_UUID         68.7G        5.4M       68.7G   0% /fsx[MDT:0]
mountname-OST0000_UUID          1.1T        4.5M        1.1T   0% /fsx[OST:0]
mountname-OST0001_UUID          1.1T        4.5M        1.1T   0% /fsx[OST:1]

filesystem_summary:             2.2T        9.0M        2.2T   0% /fsx
```

Output 1. Output from a Lustre lfs Command

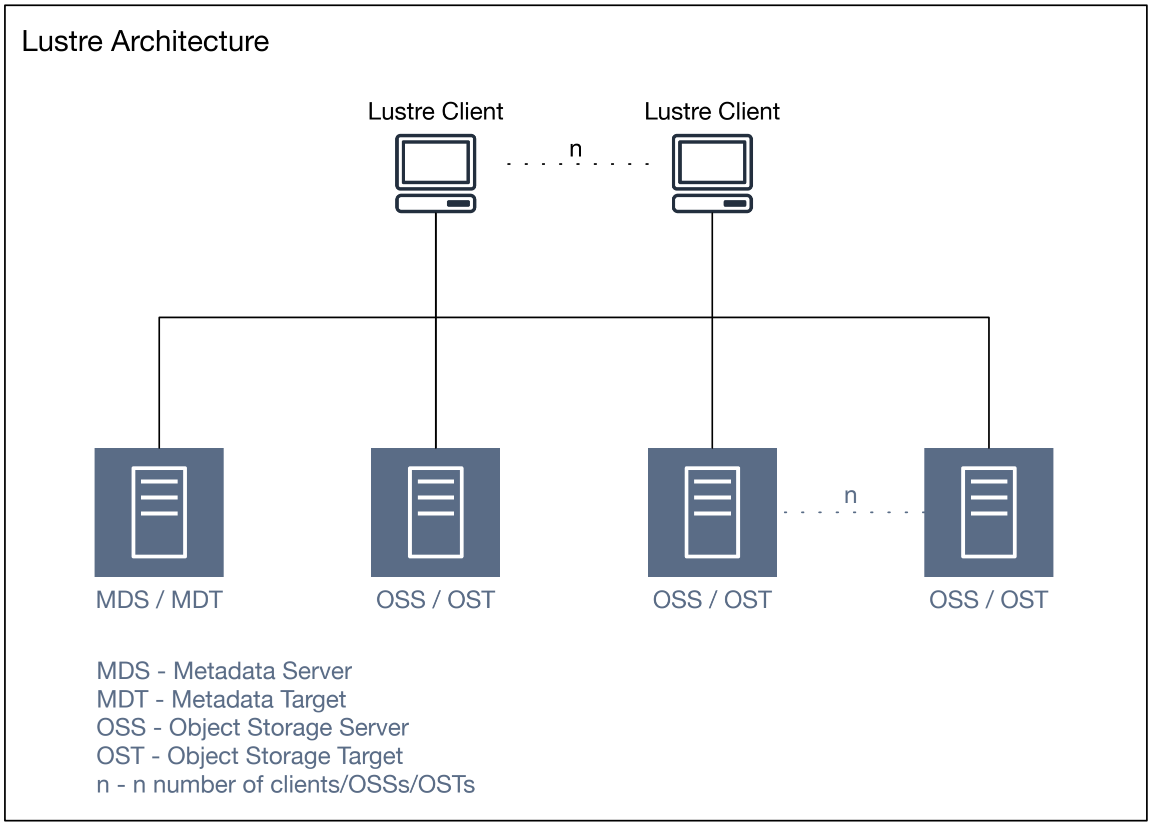

Figure 2 Simple Lustre Architecture Diagram

You can configure how files are striped across OSTs. When a file is striped across multiple OSTs, read or write requests to the file are spread across those OSTs, increasing the aggregate throughput or IOPS. By default, each file is stored on a single disk. A file's striping parameters are set when the file is first created and are inherited from the directory. You configure Lustre to spread files across multiple OSTs by setting the stripe configuration at the directory. Spreading larger files over more OSTs will improve read performance. Based on your access patterns, consider testing different directory stripe configurations to optimize performance based on specific IO patterns. You may also consider using a progressive file layout at the directory level to achieve good performance without having to know or understanding the IO pattern. For more information on Lustre progressive file layouts, please refer to progressive file layouts section on the Lustre wiki.

Heavy utilized SAS Grid environments may benefit from using separate FSx for Lustre persistent file systems for SASDATA, SASWORK, and UTILLOC. An alternative architecture may offload temporary non-shared file system workloads (SASWORK and UTILLOC) to instance store volumes or Amazon Elastic Block Store (EBS) volumes attached to the SAS Grid and compute nodes. This alternative architecture is not generally recommended and may add cost and complexity.

SAS GRID AND COMPUTE NODES

SAS Grid and compute nodes have the following file system performance and memory recommendations, all based on the number of physical cores available to the node.

- Memory: minimum of 8 GB per physical core
- File system performance: SASWORK – minimum 125 MB/s per physical core

  UTILLOC – minimum 125 MB/s per physical core

  SASDATA – minimum 75-100 MB/s per physical core

  Overall – minimum 100-125 MB/s per physical core

Because our file system recommendation for all SAS Grid libraries is FSx for Lustre, which is a network file system accessible to compute nodes over the network, we must help identify Amazon EC2 instance types for SAS Grid and compute nodes that have enough network performance that meet or exceed these recommendations. With over 260 different Amazon EC2 instance types to choose from, you may be looking for guidance on which instance families best align with these recommendations. Aligning these recommendations with EC2 resource characteristics is critical in selecting the right EC2 instance family or type. To help you with this selection, I ran a series of highly parallel network throughput tests from individual EC2 instances against a 100.8 TiB FSx for Lustre file system, which had an aggregate throughput capacity of 19.688 GB/s. IOR was used to generate write activity to the file system using parallel threads and direct I/O, bypassing I/O buffers. These tests were run in multiple AWS regions: N. Virginia (us-east-1); Ohio (us-east-2); Oregon (us-west-2); and Ireland (eu-west-1); using multiple Amazon EC2 instance families (c5, c5n, i3, i3en, m5, m5a, m5ad, m5n, m5dn, r5, r5a, r5ad, r5n, and r5dn). Not all these instance families are available in these 4 regions. The tests ran for 3 hours for each instance and the DataWriteBytes metric of the file system was recorded every 1 minute. Only 1 instance was accessing the file system at a time and the p99.9 results were captured. The metrics were consistent across all 4 regions and the results for the Oregon (us-west-2) region are available in table 3 below.

This script ran on every EC2 instance type listed in table 2:

```
module load mpi/openmpi-x86_64
mkdir -p /mnt/fsx/data
job_name=$(echo $(uuidgen)| grep -o ".\{6\}$")
sudo bash -c 'echo 3 > /proc/sys/vm/drop_caches'
mpirun -n 84 --oversubscribe ior --posix.odirect -Y -u -t 1m -b 1g -s 1024
-v -w -i 128 -F -o /mnt/fsx/data/${job_name}.txt
```

Table 2 shows the results of the IOR test to help identify which instance family meets the SAS Grid and compute node network performance and memory recommendations.

API Name	Physical Cores	Variable Network Performance Peak (MB/s)	Variable Network Performance Peak Duration (seconds)	Consistent Network Performance (MB/s)	Network Performance per Physical Core (MB/s)	Memory per Physical Core (GiB)
c5.large	1	1227.02091	300	92.81645	92.81645	4
c5.xlarge	2	1248.37167	600	155.01449	77.50724	4
c5.2xlarge	4	1251.85643	1200	310.09888	77.52472	4
c5.4xlarge	8	1251.75507	2400	620.22571	77.52821	4
c5.9xlarge	18	Not applicable	Not applicable	1502.32629	83.46257	4
c5.12xlarge	24	Not applicable	Not applicable	1502.05017	62.58542	4
c5.18xlarge	36	Not applicable	Not applicable	3020.19248	83.89424	4
c5.24xlarge	48	Not applicable	Not applicable	2994.46742	62.38474	4
c5n.large	1	1427.74807	1200	372.80372	372.80372	5.25
c5n.xlarge	2	2676.75238	1200	623.42212	311.71106	5.25
c5n.2xlarge	4	3168.92250	2400	1246.71842	311.67960	5.25
c5n.4xlarge	8	3168.97143	3300	1902.08191	237.76024	5.25
c5n.9xlarge	18	Not applicable	Not applicable	6338.39376	352.13299	5.25
c5n.18xlarge	36	Not applicable	Not applicable	11951.43157	331.98421	5.25
i3.large	1	1148.26587	300	92.83393	92.83393	15.25
i3.xlarge	2	1255.04236	600	154.82225	77.41112	15.25
i3.2xlarge	4	1258.54635	1200	309.67945	77.41986	15.25
i3.4xlarge	8	1258.56558	2400	622.72657	77.84082	15.25
i3.8xlarge	16	Not applicable	Not applicable	1516.82810	94.80176	15.25
i3.16xlarge	32	Not applicable	Not applicable	3071.22318	95.97572	15.25
i3en.large	1	1745.51204	600	260.43133	260.43133	16
i3en.xlarge	2	2557.30035	600	520.92382	260.46191	16
i3en.2xlarge	4	3169.26853	1200	1041.96997	260.49249	16
i3en.3xlarge	6	3169.09027	2400	1552.11093	258.68516	16
i3en.6xlarge	12	Not applicable	Not applicable	3169.19863	264.09989	16

API Name	Physical Cores	Variable Network Performance Peak (MB/s)	Variable Network Performance Peak Duration (seconds)	Consistent Network Performance (MB/s)	Network Performance per Physical Core (MB/s)	Memory per Physical Core (GiB)
i3en.12xlarge	24	Not applicable	Not applicable	6338.55104	264.10629	16
i3en.24xlarge	48	Not applicable	Not applicable	12020.12029	250.41917	16
m5.large	1	1225.68223	600	92.83393	92.83393	8
m5.xlarge	2	1249.12315	600	155.01449	77.50724	8
m5.2xlarge	4	1251.94732	1200	310.11286	77.52821	8
m5.4xlarge	8	1251.99800	2400	620.24319	77.53040	8
m5.8xlarge	16	Not applicable	Not applicable	1251.94382	78.24649	8
m5.12xlarge	24	Not applicable	Not applicable	1502.32979	62.59707	8
m5.16xlarge	32	Not applicable	Not applicable	2503.76531	78.24267	8
m5.24xlarge	48	Not applicable	Not applicable	2995.60687	62.40848	8
m5a.large	1	1219.16533	300	92.81645	92.81645	8
m5a.xlarge	2	1248.13225	600	155.01449	77.50724	8
m5a.2xlarge	4	1251.86867	1200	310.09888	77.52472	8
m5a.4xlarge	8	1251.77255	2400	620.18727	77.52341	8
m5a.8xlarge	16	1252.05217	3900	938.91592	58.68225	8
m5a.12xlarge	24	Not applicable	Not applicable	1251.90887	52.16287	8
m5a.16xlarge	32	Not applicable	Not applicable	1502.38222	46.94944	8
m5a.24xlarge	48	Not applicable	Not applicable	2503.96104	52.16586	8
m5ad.large	1	1228.86117	300	92.81645	92.81645	8
m5ad.xlarge	2	1248.96761	600	155.03196	77.51598	8
m5ad.2xlarge	4	1251.95780	1200	310.11111	77.52778	8
m5ad.4xlarge	8	1252.01547	2400	620.21523	77.52690	8
m5ad.12xlarge	24	Not applicable	Not applicable	1251.76906	52.15704	8
m5ad.24xlarge	48	Not applicable	Not applicable	2503.99774	52.16662	8
m5n.large	1	1774.61002	600	260.43133	260.43133	8

API Name	Physical Cores	Variable Network Performance Peak (MB/s)	Variable Network Performance Peak Duration (seconds)	Consistent Network Performance (MB/s)	Network Performance per Physical Core (MB/s)	Memory per Physical Core (GiB)
m5n.xlarge	2	2965.57565	600	510.44680	255.22340	8
m5n.2xlarge	4	3169.12872	1200	1014.85729	253.71432	8
m5n.4xlarge	8	3169.09377	2400	2056.20511	257.02564	8
m5n.8xlarge	16	Not applicable	Not applicable	4225.55157	264.09697	8
m5n.12xlarge	24	Not applicable	Not applicable	6338.58949	264.10790	8
m5n.16xlarge	32	Not applicable	Not applicable	8451.36003	264.10500	8
m5n.24xlarge	48	Not applicable	Not applicable	12114.19853	252.37914	8
m5dn.large	1	1915.05280	600	260.44531	260.44531	8
m5dn.xlarge	2	3109.39484	600	510.44680	255.22340	8
m5dn.2xlarge	4	3169.15668	1200	1014.14775	253.53694	8
m5dn.4xlarge	8	3169.06406	2400	2055.89053	256.98632	8
m5dn.8xlarge	16	Not applicable	Not applicable	4225.61797	264.10112	8
m5dn.12xlarge	24	Not applicable	Not applicable	6338.34482	264.09770	8
m5dn.16xlarge	32	Not applicable	Not applicable	8451.13459	264.09796	8
m5dn.24xlarge	48	Not applicable	Not applicable	12232.71208	254.84817	8
r5.large	1	1216.58234	300	92.83393	92.83393	16
r5.xlarge	2	1245.36575	600	155.03196	77.51598	16
r5.2xlarge	4	1251.77255	1200	310.06392	77.51598	16
r5.4xlarge	8	1251.77255	2400	620.16629	77.52079	16
r5.8xlarge	16	Not applicable	Not applicable	1252.01722	78.25108	16
r5.12xlarge	24	Not applicable	Not applicable	1501.98026	62.58251	16
r5.16xlarge	32	Not applicable	Not applicable	2503.66394	78.23950	16
r5.24xlarge	48	Not applicable	Not applicable	3004.34500	62.59052	16
r5a.large	1	984.12353	300	92.81645	92.81645	16
r5a.xlarge	2	1249.65443	600	155.00924	77.50462	16

API Name	Physical Cores	Variable Network Performance Peak (MB/s)	Variable Network Performance Peak Duration (seconds)	Consistent Network Performance (MB/s)	Network Performance per Physical Core (MB/s)	Memory per Physical Core (GiB)
r5a.2xlarge	4	1251.91236	1200	310.07616	77.51904	16
r5a.4xlarge	8	1251.85469	2400	620.18202	77.52275	16
r5a.8xlarge	16	1251.97178	3900	939.02078	58.68880	16
r5a.12xlarge	24	Not applicable	Not applicable	1251.85819	52.16076	16
r5a.16xlarge	32	Not applicable	Not applicable	1502.41367	46.95043	16
r5a.24xlarge	48	Not applicable	Not applicable	2503.83871	52.16331	16
r5ad.large	1	1218.97135	300	92.81645	92.81645	16
r5ad.xlarge	2	1248.88198	600	155.03196	77.51598	16
r5ad.2xlarge	4	1251.85294	1200	310.08140	77.52035	16
r5ad.4xlarge	8	1251.89314	2400	620.21523	77.52690	16
r5ad.12xlarge	24	Not applicable	Not applicable	1251.87741	52.16156	16
r5ad.24xlarge	48	Not applicable	Not applicable	2504.12182	52.16920	16
r5n.large	1	1789.06464	600	260.44880	260.44880	16
r5n.xlarge	2	2972.05760	600	510.42932	255.21466	16
r5n.2xlarge	4	3169.16367	1200	1014.30155	253.57539	16
r5n.4xlarge	8	3168.98542	2400	2055.92549	256.99069	16
r5n.8xlarge	16	Not applicable	Not applicable	4225.70885	264.10680	16
r5n.12xlarge	24	Not applicable	Not applicable	6338.69435	264.11226	16
r5n.16xlarge	32	Not applicable	Not applicable	8451.33032	264.10407	16
r5n.24xlarge	48	Not applicable	Not applicable	12183.11444	253.81488	16
r5dn.large	1	1844.38576	600	260.43133	260.43133	16
r5dn.xlarge	2	3111.51297	600	510.44680	255.22340	16
r5dn.2xlarge	4	3169.23358	1200	1014.02542	253.50636	16
r5dn.4xlarge	8	3169.12697	2400	2057.23271	257.15409	16
r5dn.8xlarge	16	Not applicable	Not applicable	4225.70885	264.10680	16

API Name	Physical Cores	Variable Network Performance Peak (MB/s)	Variable Network Performance Peak Duration (seconds)	Consistent Network Performance (MB/s)	Network Performance per Physical Core (MB/s)	Memory per Physical Core (GiB)
r5dn.12xlarge	24	Not applicable	Not applicable	6338.32385	264.09683	16
r5dn.16xlarge	32	Not applicable	Not applicable	8451.31984	264.10374	16
r5dn.24xlarge	48	Not applicable	Not applicable	12100.28567	252.08928	16

Table 2 IOR Test Results

Table 3 shows the EC2 instance families that meet the minimum network performance and memory recommendations.

Instance Family	Meets Minimum Network Performance Recommendation	Meets Minimum Memory Recommendation
c5	✗	✗
c5n	✓	✗
i3	✗	✓
i3en	✓	✓
m5	✗	✓
m5a	✗	✓
m5ad	✗	✓
m5n	✓	✓
m5dn	✓	✓
r5	✗	✓
r5a	✗	✓
r5ad	✗	✓
r5n	✓	✓
r5dn	✓	✓

Table 3 EC2 Instance Family SAS Grid and Compute Minimum Recommendations

The *i3en*, *m5n*, *m5dn*, *r5n*, and *r5dn* EC2 instance families meet or exceed the minimum network performance and memory recommendations. The m5n and m5dn instance families have 8,358,070,954.67 bytes of memory per physical core. If measuring memory using base 2 (2^{10} = 1024), the m5n and m5dn have 7.78406 GiB per physical core, just under the minimum recommendation. If measuring memory using base 10 (10^3 = 1000), the m5n and m5dn have 8.35807 GB per physical core, just above the minimum recommendation. The i3 instance family is just shy of meeting the minimum network performance recommendation, but based on your needs this instance family may meet your requirements. The i3, i3en, m5dn, and r5dn instance families include instance store volumes that provide temporary block-level storage located on disks that are attached to the underlying host. These instances have a higher EC2 compute price when compared to instances without instance store volumes. Because we recommend using Amazon FSx for Lustre for hosting SAS Grid libraries, selecting instances with instance store volumes at a higher price may not be warranted. To achieve the highest levels of throughput to an FSx for Lustre persistent file systems, you may need to use multiple threads per EC2 instance. We have to work within the laws of physics. FSx for Lustre is a high-performance file system accessed over the network. There is added latency when accessing storage over the network compared to local disks like instance store volumes. If you have a large number of single-threaded jobs, you may consider using i3en instances to offload your temporary scratch space - /SASWORK and /UTILLOC to instance store volumes attached to these instances.

To compare throughput between the i3en, m5d, m5dn, r5n, and r5dn instance families, I ran similar multi-threaded IOR write tests on these instances to an FSx for Lustre persistent file system that had a storage capacity of 100.8 TiB with an aggregate throughput capacity of 19.688 GB/s. For those instance families that have instance store volumes (i3en, m5dn, r5dn), I ran the same IOR write test against these instance store volumes. Instance types with multiple NVMe disks were mounted as RAID0. All instance store volumes were formatted using ext4.

Table 4 below shows the results of running the IOR write tests against these instance families.

Instance Families	Variable Network Performance per Physical Core (MB/s)	Consistent Network Performance per Physical Core (MB/s)	Instance Store Volumes - NVMe (GiB)	Instance Store Volume Performance per Physical Core (MB/s)
i3en	528.18 - 1745.51	259.81	1250 - 60000	154.26
m5n	396.18 - 1774.61	258.89	Not available	Not available
m5dn	396.13 - 1915.05	259.17	75 - 3600	62.48
r5n	396.12 - 1789.06	259.05	Not available	Not available
r5dn	396.14 - 1844.39	258.84	75 - 3600	62.22

Table 4 IOR write test results between FSx for Lustre and instance store volumes

M5n and r5n instances are a good blend of price and performance. We recommend m5n instances for general SAS Grid compute nodes but if your workload is memory bound, consider using r5n instances which provide double the memory per physical core at a slightly higher price.

To test the throughput of a few m5n and r5n instance types, I ran the `rhel_iotest.sh` script which is available from the SAS Technical Support Samples Tools (SASTSST) Repository - http://support.sas.com/kb/59/680.html. All instances had an FSx for Lustre persistent file system mounted with the default mount options and the file system path used the default Lustre stripe set. The file system had a storage capacity of 100.8 TiB with an aggregate throughput capacity of 19.688 GB/s.

Table 5 below shows the results of running rhel_iotest.sh against a few m5n and r5n instances.

	Variable Network Performance Peak per Physical Core	
Instance Type	Read (MB/s)	Write (MB/s)
m5n.large	850.20	357.07
m5n.xlarge	519.46	386.25
m5n.2xlarge	283.01	446.84
m5n.4xlarge	202.89	376.57
m5n.8xlarge	154.98	297.71
r5n.large	906.88	429.93
r5n.xlarge	488.36	455.76
r5n.2xlarge	256.96	471.65
r5n.4xlarge	203.31	390.03
r5n.8xlarge	149.63	299.45

Table 5 rhel_iotest.sh results

To take advantage of the elasticity, scalability, and flexibility of the cloud, we recommend spreading the SAS Grid and compute workload over a larger number of smaller instances versus using a smaller number of larger instances.

CONCLUSION

Because you're not in the business of running large scale, cloud optimized parallel file systems, we recommend using Amazon FSx for Lustre persistent file systems for all SAS Grid storage, including SASDATA, SASWORK, and UTILLOC. This allows you to focus more on running your business and the SAS Grid application and less on managing a high performing file system. Your goal when selecting a file system for your SAS deployment is to make sure you get consistent low latencies, high throughput, and millions of IOPS so your SAS jobs complete within the expected timeframe. While there are other storage options for running SAS Grid on AWS – like Amazon Elastic File Systems (EFS), Do-It-Yourself (DIY) file systems using Amazon EC2 with instance store volumes or Amazon Elastic Block Store (EBS), or even 3rd party storage solutions – these offerings add cost and complexity and may impact performance, availability, and data durability. We recommend Amazon FSx for Lustre for its ease of use, quick deployment, performance, simplicity, availability, and durability. We also recommend using m5n and r5n Amazon EC2 instance families for SAS Grid compute nodes when accessing SAS Grid libraries hosted on Amazon FSx for Lustre.

REFERENCES

Amazon Web Services. “AWS Service Level Agreements.” Accessed February 27, 2020. https://aws.amazon.com/legal/service-level-agreements/.

Amazon Web Services. “Amazon EC2 Instance Types.” Accessed February 27, 2020. https://aws.amazon.com/ec2/instance-types/.

Amazon Web Services. “Object Versioning.” Accessed February 27, 2020. https://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectVersioning.html.

Amazon Web Services. “Object Lifecycle Management.” Accessed February 27, 2020. https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html.

Amazon Web Services. “Installing the Lustre Client.” Accessed February 27, 2020. https://docs.aws.amazon.com/fsx/latest/LustreGuide/install-lustre-client.html.

Amazon Web Services. “Amazon FSx for Lustre Tutorial.” Accessed February 27, 2020. https://github.com/aws-samples/amazon-fsx-tutorial.

Amazon Web Services. “Amazon FSx for Lustre Quotas.” Accessed February 27, 2020. https://docs.aws.amazon.com/fsx/latest/LustreGuide/limits.html.

Lustre.org. “Progressive File Layouts.” Accessed February 27, 2020. http://wiki.lustre.org/Progressive_File_Layouts.

SAS. “Usage Note 59680: Testing throughput for your SAS® 9 File Systems: The rhel_iotest.sh script.” Accessed February 27, 2020. http://support.sas.com/kb/59/680.html.

RECOMMENDED READING

- *Amazon Elastic Compute Cloud User Guide for Linux Instances*
- *Amazon FSx for Lustre User Guide*
- *Amazon FSx for Lustre Tutorial on Github*
- *Lustre.org Wiki*

CONTACT INFORMATION

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

Darryl S. Osborne
Amazon Web Services
darrylo@amazon.com
https://aws.amazon.com