NFS Storage Solutions on AWS with Alces Flight Compute

How to deploy, configure and start computing

Alces Flight - 2016.2

Vaughan Jones - Alces Flight Ltd
May 2016

Revision number	Release date	Release notes	Author(s)
1.0	May 2016	Initial release	Vaughan Jones

CONTENTS

Overview

The following paper details some of the available network filesystem (NFS) storage solutions for use with an Alces Flight Compute environment on Amazon Web Services (AWS) public cloud.

Each dedicated compute host in an Alces Flight Compute environment is configured with a small amount of local scratch disk backed by AWS Elastic Block Storage (EBS) - this is often sufficient for applications to write data to, before being moved to a local persistent file storage - such as a shared cluster NFS storage solution - or remote object storage - for example Amazon Simple Storage Service (S3).

Intended audience

This paper is primarily aimed at AWS account administrators or advanced users wishing to set up an NFS storage server for existing or new Alces Flight Compute instances or your own research compute instances. This white paper is not a detailed configuration guide, but does provide the different options when deploying an NFS storage solution available on AWS.

The following topics are covered in this white paper;

- Overview of network file systems (NFS) and Alces Flight Compute
- Available configuration options when deploying an NFS storage solution on AWS
- Expected performance for deployed NFS storage solutions
- Overview of deployment and configuration details
- Expected costs when deploying an NFS storage solution

NFS

The simple setup and ease of maintenance makes NFS an often ideal candidate for smaller compute environments where scalability and overall performance is less of a concern.

AWS provides a range of options for creating an NFS storage solution for your Alces Flight Compute environment - depending on the mixture of performance, reliability and cost best suited to your applications, workload and compute environment.

We recommend deploying an NFS storage solution when;

- Total storage capacity requirements are less than 48TiB
- Total throughput performance requirements are less than 1GB/sec read/write
- Less than 32 clients (compute nodes, cluster login nodes) exist within your Alces Flight Compute environment

Alces Flight Compute

The Alces Flight software appliances manage the deployment and configuration of your AWS instances - automatically configuring each deployed host in the environment. By pre-configuring your compute environments, core hours can be better spent doing research rather than configuration.

The Alces Flight Compute appliance for Linux is a scalable, self-configuring cloud-ready High Performance Computing environment - built on traditional, well-known Linux tools and utilities. The Alces Flight Compute appliance is a multi-functional AMI; acting as either a cluster master node - hosting cluster scheduler services and providing shared user data and applications, or as a cluster compute host to run your batch workloads on.

An Alces Flight compute environment provides many useful and popular features, including;

- **Batch scheduler installation and configuration;** Open Grid Scheduler is currently available, which is automatically configured including queue setup with known-working settings. Other popular cluster schedulers will be made available with later releases of the Alces Flight Compute appliance including SLURM, Torque and OpenLava.
- **Automatic compute host scaling**; Alces Flight compute environments dynamically grow and shrink in size based on user demand
- **Shared user data**; each cluster is configured with a shared home directory - exported to each of the nodes via NFS. Additional storage spaces can also be configured from external sources including parallel filesystems and additional block storage volumes.
- **Alces Gridware**; the Alces Gridware utility is a tool that makes the installation of Linux applications, accelerated libraries and compilers simple and effortless. Gridware packages are served through the Linux *modules* environment. The Alces Gridware repository contains over 850 packages spread over a wide range of research use types including bioinformatics tools, chemistry applications, engineering applications and much more.
- **Storage management utilities**; also included is the `alces storage` utility - providing seamless use and access of multiple storage tiers including local POSIX storage and remote object storage targets, allowing you to easily manage your data across multiple tiers of storage without having to learn different tools and APIs.

Many other tools and utilities are included with an Alces Flight compute environment to help you with your research and environment deployment.

NFS Storage Options on AWS

The following section details the available options when deploying an NFS storage solution for use with your Alces Flight Compute environment, including storage performance and cost.

Instance-store backed storage

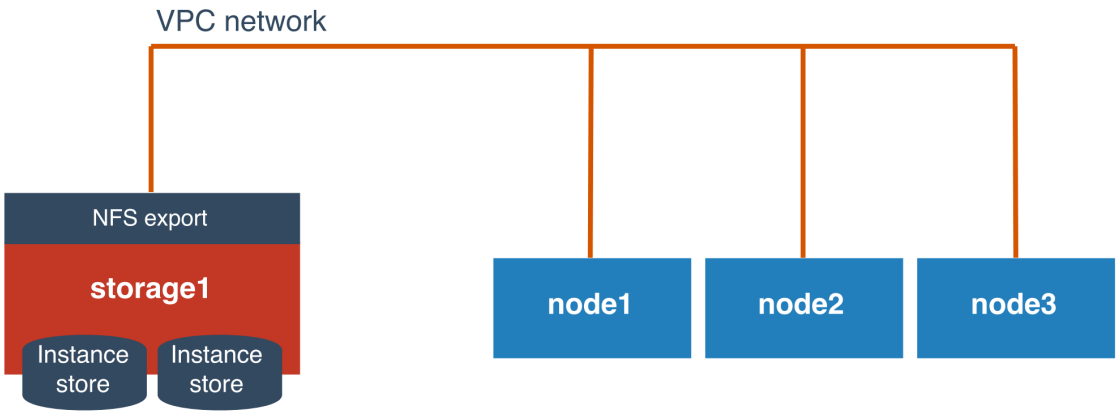
Amazon AWS offers a wide variety of instance types - including storage optimised instance types. These instance types include a number of attached *instance-store* disks - these are ephemeral disks local to the server your instance is running on.

Instances using instance-store disks should not be considered for permanent or even semi-permanent data storage unless precautions have been taken to regularly back-up or archive data stored on the NFS storage server.

Several instance types are available, catering for different requirements including the storage capacity and performance required. The following table details each of the instance types with instance-store disks available (excluding SSD instance types) -

Instance type	Number of CPUs	Memory (GB)	Raw storage capacity
`d2.xlarge`	4	30.5	6TB (3 x 2TB HDD)
`d2.2xlarge`	8	61	12TB (6 x 2TB HDD)
`d2.4xlarge`	16	122	24TB (12 x 2TB HDD)
`d2.8xlarge`	36	244	48TB (24 x 2TB HDD)

Architecture



When using instance-store disk instance types for an NFS storage solution - the disks are provided with the instance, making them immediately available for use, local to the NFS storage server as

opposed to an EBS storage solution where the storage may be in a different availability zone/data centre.

Instance-store disk enabled instance types will often see significantly higher total throughput performance than an EBS backed NFS storage solution as a result of the disks being local to the instance, allowing data to be served to clients using all available network bandwidth.
Instance-store disks provided with the instance are grouped into a single volume using LVM to create a single large pool of data ready for NFS export.

Advantages

There are several advantages to deploying an instance-store enabled NFS storage solution, including;

- **Total throughput performance increase**; each of the provided disks are local to the deployed instance, resulting in an often significantly faster storage solution. Instance-store disks are also not subject to network contention as an EBS backed NFS storage solution would be.
- **Consistent performance**; due to the lack of network contention when using local disks, the performance is generally consistent.
- **Lower running cost**; instance-store backed NFS storage solutions typically incur a lower overall running cost, this is detailed further in the following section

Expected costs

The following section details the expected costs of deploying various NFS storage server types using the storage optimised instance types. The following prices are for *dedicated* instance types. For short-term solutions, Spot instances can be deployed with a significantly lower running cost. Optionally - for longer-term fixed solutions, Reserved instance types can also be deployed.

Instance type	Total storage available (TB)	Cost per hour (USD)	Cost per month (USD)	Total cost (1 year)	Total cost (3 years)
d2.xlarge	6	$0.74	$536	$6,435	$19,305
d2.2xlarge	12	$1.47	$1,073	$12,870	$38,610
d2.4xlarge	24	$2.94	$2,145	$25,740	$77,221
d2.8xlarge	48	$5.88	$4,290	$51,481	$154,442

The following table details the pricing options (correct as of May 2016) when deploying a Reserved instance type (1-year minimum purchase).

Instance type	Total storage available (TB)	Total cost - 1-year term (USD)	Total cost - 3-year term (USD)
d2.xlarge	6	$3,278	$6,713
d2.2xlarge	12	$6,556	$13,426
d2.4xlarge	24	$13,112	$26,852
d2.8xlarge	48	$26,224	$53,704

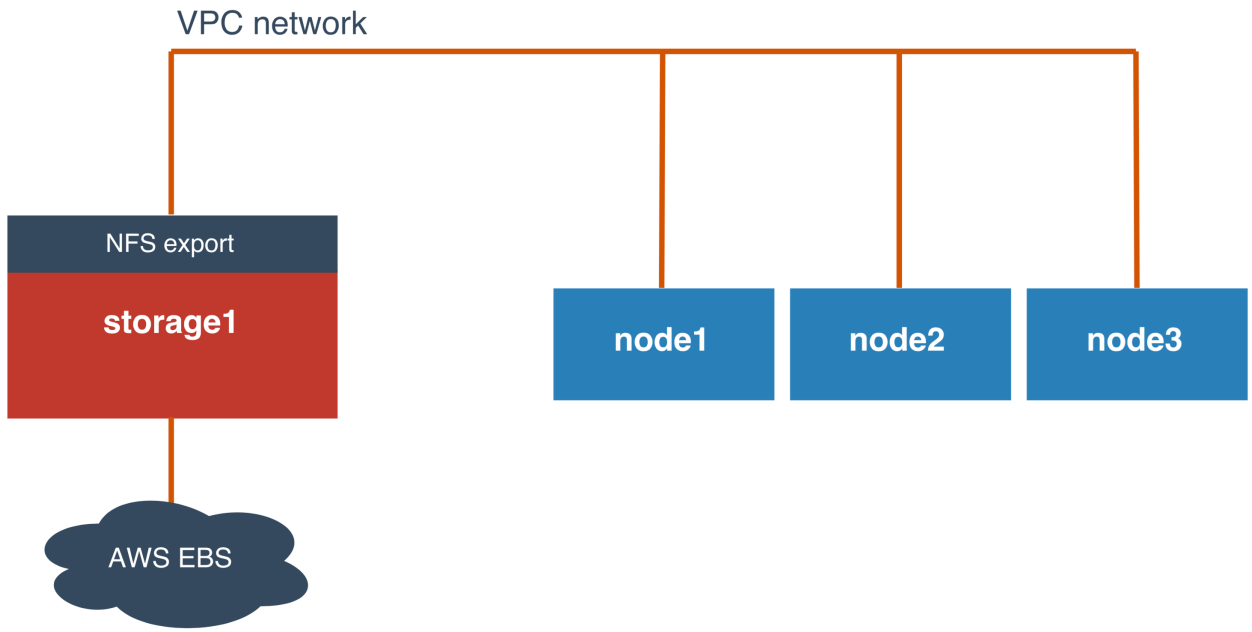
EBS backed storage

Another option when deploying an NFS storage server is to use a regular instance type (i.e non instance-store instance types) together with EBS storage volumes. AWS EBS provides block storage volumes up to 16TiB per volume and attaches them to an instance, which can then be formatted for use.

EBS volumes have built-in resiliency as opposed to the storage-optimised instance types which have no built-in resiliency, but are not as redundant as S3 storage which self-replicates across multiple availability zones.

EBS volumes will typically see considerably less total throughput performance than a storage-optimised NFS storage solution.

Architecture



When deploying an NFS storage solution utilising AWS EBS, volumes are attached to the NFS storage server as a device. Either a single large volume (up to 16TiB) or multiple smaller volumes (grouped into a single pool using LVM) can be deployed depending on your storage capacity requirements.

Each attached EBS storage volume should be deployed using the *storage optimised (st1)* volume type - the available volume types are:

- General purpose (gp2); general purpose volumes are SSD volume types, typically better suited to database type workloads
- Provisioned IOPS (io1); provisioned IOPS volumes are SSD volume types, also typically better suited to high I/O/database type workloads where a specific IOPS is required
- Throughput optimised (st1); throughput optimised use regular spinning disks - st1 volumes have a lower IOPS, but are better suited to larger read/write requirements
- Cold storage (sc1); cold storage is a lower class tier of storage, offering EBS-style storage at a reduced performance and cost

Advantages

When deploying an NFS storage server solution backed with AWS EBS storage volumes, you can expect to see the following advantages;

- **Increased resiliency**; EBS volumes provide increased resiliency over storage-optimised instance types, and self-replicate within an availability zone
- **Easily increase storage capacity**; due to the flexibility and scalability of EBS - it is simple to expand or increase the storage capacity available by simply creating additional EBS block storage volumes
- **Suitable for longer term solutions**; due to the increased resiliency of AWS EBS - NFS storage servers can be deployed as a long-term solution for your data storage requirements

Expected costs

The following section details the expected costs of deploying various NFS storage server types using regular instance types with attached EBS volumes. The following prices are for *dedicated* instance types. For short-term solutions, Spot instances can be deployed with a significantly lower running cost. Optionally - for longer-term fixed solutions, Reserved instance types can also be deployed.

Each instance type is shown with a varying amount of EBS storage attached:

Instance type	EBS storage	Total cost per hour (USD)	Total cost per month (USD)	Total cost - 1 year (USD)	Total cost - 3 years (USD)
c4.large	1TB	$0.19	$136.82	$1,641.87	$4,925.61
	5TB	$0.46	$336.82	$4,041.87	$12,125.61
	10TB	$0.80	$586.82	$7,041.87	$21,125.61
	25TB	$1.83	$1,336.82	$16,041.87	$48,125.61
c4.xlarge	1TB	$0.31	$223.64	$2,683.74	$8,051.21
	5TB	$0.58	$423.64	$5,083.74	$15,251.21
	10TB	$0.92	$673.64	$8,083.74	$24,251.21
	25TB	$1.95	$1,423.64	$17,083.74	$51,251.21
c4.2xlarge	1TB	$0.55	$398.02	$4,776.23	$14,328.69
	5TB	$0.82	$598.02	$7,176.23	$21,528.69
	10TB	$1.16	$848.02	$10,176.23	$30,528.69
	25TB	$2.19	$1,598.02	$19,176.23	$57,528.69
c4.4xlarge	1TB	$1.02	$745.31	$8,943.71	$26,831.12
	5TB	$1.30	$945.31	$11,343.71	$34,031.12
	10TB	$1.64	$1,195.31	$14,343.71	$43,031.12
	25TB	$2.67	$1,945.31	$23,343.71	$70,031.12
c4.8xlarge	1TB	$1.97	$1,440.62	$17,287.41	$51,862.23
	5TB	$2.25	$1,640.62	$19,687.41	$59,062.23
	10TB	$2.59	$1,890.62	$22,687.41	$68,062.23
	25TB	$3.62	$2,640.62	$31,687.41	$95,062.23

Architecting an NFS storage solution

An NFS Storage Server can be deployed in multiple ways for multiple use-cases; it is important to identify how you wish to use your NFS storage server before deploying. An NFS Storage Server can be deployed in some of the following ways;

- One or multiple NFS Storage Servers per Alces Flight Compute environment

  An NFS Storage Server can be deployed as a fixed storage solution for a single Alces Flight Compute environment by deploying into a dedicated cluster network and allowing the storage server to join the cluster ring. This is particularly useful for semi-persistent or persistent Alces Flight Compute environments.

- One or multiple NFS Storage Servers per AWS account shared amongst multiple ephemeral Alces Flight Compute environments

  An NFS Storage Server can be deployed into a separate AWS Virtual Private Cloud (VPC) network and mount to multiple Alces Flight Compute environments. This is particularly useful for users or accounts where Alces Flight Compute environments are typically ephemeral - storage can easily be mounted on each newly deployed cluster if required.

The following section covers how to deploy and configure an NFS Storage Server to an existing Alces Flight Compute environment for use as a semi-persistent storage server.

Deployment

Deployment can be performed manually through the available Elastic Cloud (EC2) tools - or automated using the example CloudFormation template included with this white paper.

The example CloudFormation templates includes the appropriate base configuration required to integrate the NFS Storage Server with your existing Alces Flight Compute environment - the example template includes the following:

- Creation of a single NFS Storage Server in the region you choose when you launch the CloudFormation stack. The NFS Storage Server instance type is automatically chosen based on the amount of storage you wish to configure and serve
- The NFS Storage Server is deployed to the existing Alces Flight Compute Virtual Private Cloud (VPC) network
- Storage is automatically provisioned and mounted to the created NFS Storage Server instance

The example CloudFormation templates can be found in the appendix of this paper.

Configuration

NFS Server configuration

Once the NFS Storage Server is active - appropriate steps should be taken to both configure the NFS Storage Server as well as mount the available storage to your existing Alces Flight Compute environment.

When configuring the NFS storage server, we recommend the following configuration steps:

- Install the NFS server utilities available through the package manager for your Linux distribution
- If working with multiple storage volume mounts - combine the mounts into a single logical volume or RAID group using available Linux storage management tools. This allows you to have a single pool of storage, optimising either for performance or increased resiliency and data protection
- Increase the number of concurrent connections available, this can be increased by editing the `RPCNFSD` setting

Compute host configuration

Using the Alces customisation tool, Alces Flight Compute hosts can easily be configured to interact with and mount the NFS storage server volumes. The customisation scripts should be set up prior to deploying your Alces Flight Compute environment in order to correctly function.

The Alces customisation tool enables you to easily configure existing and future hosts with any required customisations such as mounting the NFS storage server. Both existing hosts and future hosts that may join the cluster ring will perform the specified customisation scripts - allowing you to keep adding or removing compute hosts to your environment whilst mounting the NFS storage server with no manual intervention.

For more information on the Alces customisation tool for Amazon Web Services (AWS) environments, please visit the documentation:

http://docs.alces-flight.com/en/latest/customisation/customisation.html

A typical customisation script should;

- Check for the existence of the NFS storage server
- Create a suitable directory on each of the nodes to mount the storage volume to
- Add the storage volume mount entry to each nodes filesystem entries table
- Mount the external storage volume

Results

The following tables summarise the IOZone benchmark results performed on both NFS storage server types including read and write performance.

The following section details the performance results of different NFS storage server configurations together with an Alces Flight Compute environment. The Alces Flight Compute environment used with each NFS storage server is as follows:

- 1 x Alces Flight Compute login node (`t2.large` instance type)
- 8 x Alces Flight Compute dedicated compute hosts (`c4.8xlarge` instance type)
  - Each compute host placed into AWS Placement Group (default for all Alces Flight Compute environments launched through AWS Marketplace or the Alces Flight CloudFormation templates)
  - Each compute host uses the 10 gigabit Ethernet driver (default for all Alces Flight Compute hosts)

Each test uses the IOZone disk benchmarking application, distributed over the 8 compute hosts.

Each thread/node writes/re-writes then reads/re-reads a 120GB file per thread (double the nodes total memory capacity) to read/write a total of 960GB.

The following table shows the total sequential aggregate write throughput performance when writing from 8 clients in parallel using the IOZone application:

Instance type	Instance-store NFS solution (MB/sec)	EBS-backed; NFS solution (MB/sec)
xlarge instance type	147	90
2xlarge instance type	310	116
4xlarge instance type	599	260
8x/10xlarge instance type	1140	507

The following table shows the total sequential aggregate read throughput performance when reading from 8 clients in parallel using the IOZone application:

Instance type	Instance-store NFS solution (MB/sec)	EBS-backed; NFS solution (MB/sec)
xlarge instance type	138	88
2xlarge instance type	330	124
4xlarge instance type	606	420
8x/10xlarge instance type	1301	488

Which option is best for me?

Both NFS storage server solutions provide many advantages and disadvantages - across performance, resiliency and cost. These should all be considered when deploying your NFS storage server for research and production use.

An instance-store NFS storage server may be deployed when;

- Your environment is predominantly used for non-mission-critical research data purposes
- Your important data is archived to a more persistent storage platform such as Amazon S3
- You have considered the implications of storing your data on a single server with no high-availability features
- You require a higher performance throughput figure than an EBS backed NFS storage server

An EBS-backed NFS storage server may be deployed when;

- Data integrity and resiliency is of importance, EBS volumes provide built-in resiliency for added protection
- Your important data should still be archived to a more persistent storage platform such as Amazon S3. Additional resiliency steps can be taken to ensure data protection such as snapshotting your EBS volumes periodically

There is no correct option, each of the advantages and disadvantages of both solutions should be carefully considered prior to deployment in order to prevent potential data loss.

For more information on AWS EBS and Instance-store storage options, please visit the documentation:

https://aws.amazon.com/ebs/
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html
http://docs.alces-flight.com/en/latest/

Appendix

Test results

Instance-store backed NFS Server IOZone tests

`d2.xlarge` NFS server

The following benchmark results are performed using a `d2.xlarge` NFS server type, providing a total of 5.5TB storage available to NFS clients.

Write/re-write IOZone test

```
        Iozone: Performance Test of File I/O
                Version $Revision: 3.420 $
                Compiled for 64 bit mode.
                Build: linux-AMD64

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                     Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                     Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                     Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                     Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                     Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                     Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                     Vangel Bojaxhi, Ben England, Vikentsi Lapa.

        Run began: Wed May  4 12:38:24 2016

        Setting no_unlink
        Include fsync in write timing
        File size set to 125829120 KB
        Record Size 1024 KB
        Network distribution mode enabled.
        Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone.hosts
        Output is in Kbytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 Kbytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
        Throughput test with 8 processes
        Each process writes a 125829120 Kbyte file in 1024 Kbyte records

        Test running:
        Children see throughput for  8 initial writers  =  147717.73 KB/sec
        Min throughput per process                      =   14702.20 KB/sec
        Max throughput per process                      =   24095.58 KB/sec
        Avg throughput per process                      =   18464.72 KB/sec
        Min xfer                                        = 87612416.00 KB

        Test running:
        Children see throughput for  8 rewriters        =  151580.48 KB/sec
        Min throughput per process                      =   14162.02 KB/sec
        Max throughput per process                      =   25211.29 KB/sec
        Avg throughput per process                      =   18947.56 KB/sec
        Min xfer                                        = 71333888.00 KB


iozone test complete.
```

Read/re-read IOZone test

```
              Iozone: Performance Test of File I/O
                      Version $Revision: 3.420 $
                      Compiled for 64 bit mode.
                      Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                           Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                           Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                           Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                           Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                           Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                           Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                           Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Wed May  4 16:18:20 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 readers          =  138103.15 KB/sec
              Min throughput per process                      =   13125.17 KB/sec
              Max throughput per process                      =   22726.41 KB/sec
              Avg throughput per process                      =   17262.89 KB/sec
              Min xfer                                        = 72670208.00 KB

              Test running:
              Children see throughput for 8 re-readers        =  118435.39 KB/sec
              Min throughput per process                      =   12943.23 KB/sec
              Max throughput per process                      =   16673.60 KB/sec
              Avg throughput per process                      =   14804.42 KB/sec
              Min xfer                                        = 97745920.00 KB



       iozone test complete.
```

`d2.2xlarge` NFS server

The following benchmark results are performed using a `d2.2xlarge` NFS server type, providing a total of 11TB storage available to NFS clients.

Write/re-write IOZone test

```
              Iozone: Performance Test of File I/O
                    Version $Revision: 3.420 $
                   Compiled for 64 bit mode.
                   Build: linux-AMD64

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                         Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                         Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                         Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                        Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                        Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                        Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                        Vangel Bojaxhi, Ben England, Vikentsi Lapa.

            Run began: Tue May  3 14:48:59 2016

            Setting no_unlink
            Include fsync in write timing
            File size set to 125829120 KB
            Record Size 1024 KB
            Network distribution mode enabled.
            Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone-hosts
            Output is in Kbytes/sec
            Time Resolution = 0.000001 seconds.
            Processor cache size set to 1024 Kbytes.
            Processor cache line size set to 32 bytes.
            File stride size set to 17 * record size.
            Throughput test with 8 processes
            Each process writes a 125829120 Kbyte file in 1024 Kbyte records

            Test running:
            Children see throughput for  8 initial writers       =  310143.49 KB/sec
            Min throughput per process                   =   26197.83 KB/sec
            Max throughput per process                   =   75674.14 KB/sec
            Avg throughput per process                   =   38767.94 KB/sec
            Min xfer                               = 48568320.00 KB

            Test running:
            Children see throughput for  8 rewriters      =  322064.49 KB/sec
            Min throughput per process                   =   26539.37 KB/sec
            Max throughput per process                   =   75545.23 KB/sec
            Avg throughput per process                   =   40258.06 KB/sec
            Min xfer                               = 49434624.00 KB


      iozone test complete.
```

Read/re-read IOZone test

```
              Iozone: Performance Test of File I/O
                      Version $Revision: 3.420 $
                      Compiled for 64 bit mode.
                      Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                           Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                           Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                           Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                           Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                           Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                           Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                           Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Wed May  4 16:18:20 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 readers          =  330245.65 KB/sec
              Min throughput per process                      =   24595.21 KB/sec
              Max throughput per process                      =   74133.11 KB/sec
              Avg throughput per process                      =   34934.41 KB/sec
              Min xfer                                        = 48565122.00 KB

              Test running:
              Children see throughput for 8 re-readers        =  352061.43 KB/sec
              Min throughput per process                     =   26539.11 KB/sec
              Max throughput per process                     =   76515.22 KB/sec
              Avg throughput per process                     =   40251.81 KB/sec
              Min xfer                                  = 49463625.00 KB



       iozone test complete.
```

d2.4xlarge NFS server

The following benchmark results are performed using a d2.4xlarge NFS server type, providing a total of 22TB storage available to NFS clients.

Write/re-write IOZone test

```
              Iozone: Performance Test of File I/O
                     Version $Revision: 3.420 $
                    Compiled for 64 bit mode.
                    Build: linux-AMD64

                  Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                               Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                               Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                               Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                               Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                              Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                              Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                               Vangel Bojaxhi, Ben England, Vikentsi Lapa.

                  Run began: Tue May  3 12:47:50 2016

                  Setting no_unlink
                  Include fsync in write timing
                  File size set to 125829120 KB
                  Record Size 1024 KB
                  Network distribution mode enabled.
                  Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone-hosts
                  Output is in Kbytes/sec
                  Time Resolution = 0.000001 seconds.
                  Processor cache size set to 1024 Kbytes.
                  Processor cache line size set to 32 bytes.
                  File stride size set to 17 * record size.
                  Throughput test with 8 processes
                  Each process writes a 125829120 Kbyte file in 1024 Kbyte records

                  Test running:
                  Children see throughput for  8 initial writers       =  599067.59 KB/sec
                  Min throughput per process                    =   66914.90 KB/sec
                  Max throughput per process                    =   79759.72 KB/sec
                  Avg throughput per process                    =   74883.45 KB/sec
                  Min xfer                                 = 106586112.00 KB

                  Test running:
                  Children see throughput for  8 rewriters       =  612371.56 KB/sec
                  Min throughput per process                    =   73611.84 KB/sec
                  Max throughput per process                    =   79403.56 KB/sec
                  Avg throughput per process                    =   76546.45 KB/sec
                  Min xfer                                 = 122145792.00 KB



       iozone test complete.
```

Read/re-read IOZone test

```
              Iozone: Performance Test of File I/O
                      Version $Revision: 3.420 $
                     Compiled for 64 bit mode.
                     Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                           Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                           Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                           Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                           Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                           Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                           Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                           Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Tue May  3 13:46:00 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone-hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 readers         =  606504.35 KB/sec
              Min throughput per process                     =   72462.73 KB/sec
              Max throughput per process                     =   82662.19 KB/sec
              Avg throughput per process                     =   75813.04 KB/sec
              Min xfer                                  = 110304256.00 KB

              Test running:
              Children see throughput for 8 re-readers       =  611952.79 KB/sec
              Min throughput per process                     =   74532.22 KB/sec
              Max throughput per process                     =   84253.62 KB/sec
              Avg throughput per process                     =   76494.10 KB/sec
              Min xfer                                  = 111311872.00 KB



       iozone test complete.
```

d2.8xlarge **NFS server**

The following benchmark results are performed using a d2.8xlarge NFS server type, providing a total of 44TB storage available to NFS clients.

Write/re-write IOZone test

```
            Iozone: Performance Test of File I/O
                    Version $Revision: 3.420 $
                   Compiled for 64 bit mode.
                   Build: linux-AMD64

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                         Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                         Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                         Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                         Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                        Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                        Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                        Vangel Bojaxhi, Ben England, Vikentsi Lapa.

            Run began: Tue May  3 11:49:29 2016

            Setting no_unlink
            Include fsync in write timing
            File size set to 125829120 KB
            Record Size 1024 KB
            Network distribution mode enabled.
            Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone-hosts
            Output is in Kbytes/sec
            Time Resolution = 0.000001 seconds.
            Processor cache size set to 1024 Kbytes.
            Processor cache line size set to 32 bytes.
            File stride size set to 17 * record size.
            Throughput test with 8 processes
            Each process writes a 125829120 Kbyte file in 1024 Kbyte records

            Test running:
            Children see throughput for  8 initial writers       = 1140811.62 KB/sec
            Min throughput per process                   =  127677.90 KB/sec
            Max throughput per process                   =  163313.20 KB/sec
            Avg throughput per process                   =  142601.45 KB/sec
            Min xfer                                = 99718144.00 KB

            Test running:
            Children see throughput for  8 rewriters       = 1191331.59 KB/sec
            Min throughput per process                   =  130097.68 KB/sec
            Max throughput per process                   =  163526.95 KB/sec
            Avg throughput per process                   =  148916.45 KB/sec
            Min xfer                                = 107812864.00 KB


       iozone test complete.
```

Read/re-read IOZone test

```
              Iozone: Performance Test of File I/O
                      Version $Revision: 3.420 $
                     Compiled for 64 bit mode.
                     Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                           Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                           Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                          Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                          Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                          Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                          Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                          Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Tue May  3 12:22:13 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone-hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 readers         = 1301490.27 KB/sec
              Min throughput per process                     =  130498.52 KB/sec
              Max throughput per process                     =  211150.80 KB/sec
              Avg throughput per process                     =  162686.28 KB/sec
              Min xfer                                  = 77768704.00 KB

              Test running:
              Children see throughput for 8 re-readers       = 1402937.38 KB/sec
              Min throughput per process                     =  149971.20 KB/sec
              Max throughput per process                     =  193203.47 KB/sec
              Avg throughput per process                     =  175367.17 KB/sec
              Min xfer                                  = 97675264.00 KB



       iozone test complete.
```

EBS backed NFS Server IOZone tests

`m4.xlarge` NFS server

The following benchmark results are performed using an `m4.xlarge` NFS server type together with EBS-backed storage providing a total of 2TB storage available to NFS clients.

Write/re-write IOZone test

```
              Iozone: Performance Test of File I/O
                     Version $Revision: 3.420 $
                     Compiled for 64 bit mode.
                     Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                           Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                           Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                           Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                           Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                           Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                           Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                           Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Thu May  5 18:39:22 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 writers          =   90782.33 KB/sec
              Min throughput per process                      =    9015.30 KB/sec
              Max throughput per process                      =   15058.61 KB/sec
              Avg throughput per process                      =   11347.79 KB/sec
              Min xfer                                        = 75331584.00 KB

              Test running:
              Children see throughput for 8 re-writers        =   90784.08 KB/sec
              Min throughput per process                      =    9079.63 KB/sec
              Max throughput per process                      =   15413.14 KB/sec
              Avg throughput per process                      =   11348.01 KB/sec
              Min xfer                                        = 74124288.00 KB


       iozone test complete.
```

Read/re-read IOZone test

```
              Iozone: Performance Test of File I/O
                      Version $Revision: 3.420 $
                      Compiled for 64 bit mode.
                      Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                      Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                      Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                      Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                      Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                      Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                      Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                      Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Fri May  6 11:39:28 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 readers         =  88966.68 KB/sec
              Min throughput per process                     =   8834.99 KB/sec
              Max throughput per process                     =   14757.43 KB/sec
              Avg throughput per process                     =   11120.83 KB/sec
              Min xfer                                  = 73824952.00 KB

              Test running:
              Children see throughput for 8 re-readers       =  88968.39 KB/sec
              Min throughput per process                     =   8898.03 KB/sec
              Max throughput per process                     =   15104.87 KB/sec
              Avg throughput per process                     =   11121.04 KB/sec
              Min xfer                                  = 72641802.00 KB



       iozone test complete.
```

`m4.2xlarge` NFS server

The following benchmark results are performed using an `m4.2xlarge` NFS server type together with EBS-backed storage providing a total of 5TB storage available to NFS clients.

Write/re-write IOZone test

```
            Iozone: Performance Test of File I/O
                    Version $Revision: 3.420 $
                    Compiled for 64 bit mode.
                    Build: linux-AMD64

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                    Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                    Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                    Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                    Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                    Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                    Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                    Vangel Bojaxhi, Ben England, Vikentsi Lapa.

            Run began: Fri May  6 07:15:49 2016

            Setting no_unlink
            Include fsync in write timing
            File size set to 125829120 KB
            Record Size 1024 KB
            Network distribution mode enabled.
            Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone.hosts
            Output is in Kbytes/sec
            Time Resolution = 0.000001 seconds.
            Processor cache size set to 1024 Kbytes.
            Processor cache line size set to 32 bytes.
            File stride size set to 17 * record size.
            Throughput test with 8 processes
            Each process writes a 125829120 Kbyte file in 1024 Kbyte records

            Test running:
            Children see throughput for  8 initial writers       =  116330.98 KB/sec
            Min throughput per process                   =   10925.37 KB/sec
            Max throughput per process                   =   19485.61 KB/sec
            Avg throughput per process                   =   14541.37 KB/sec
            Min xfer                                = 72830976.00 KB

            Test running:
            Children see throughput for  8 rewriters      =  125249.73 KB/sec
            Min throughput per process                   =   12337.31 KB/sec
            Max throughput per process                   =   20698.28 KB/sec
            Avg throughput per process                   =   15656.22 KB/sec
            Min xfer                                = 81739776.00 KB


       iozone test complete.
```

Read/re-read IOZone test

```
             Iozone: Performance Test of File I/O
                     Version $Revision: 3.420 $
                     Compiled for 64 bit mode.
                     Build: linux-AMD64

             Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                     Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                     Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                     Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                     Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                     Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                     Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                     Vangel Bojaxhi, Ben England, Vikentsi Lapa.

             Run began: Fri May  6 09:21:43 2016

             Setting no_unlink
             Include fsync in write timing
             File size set to 125829120 KB
       Record Size 1024 KB
             Network distribution mode enabled.
             Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
             Output is in Kbytes/sec
             Time Resolution = 0.000001 seconds.
             Processor cache size set to 1024 Kbytes.
             Processor cache line size set to 32 bytes.
             File stride size set to 17 * record size.
             Throughput test with 8 processes
             Each process writes a 125829120 Kbyte file in 1024 Kbyte records

             Test running:
             Children see throughput for  8 initial readers       =  124474.14 KB/sec
             Min throughput per process                    =   11690.14 KB/sec
             Max throughput per process                    =   20849.60 KB/sec
             Avg throughput per process                    =   15559.26 KB/sec
             Min xfer                                 = 77929144.00 KB

             Test running:
             Children see throughput for  8 re-readers     =  134017.21 KB/sec
             Min throughput per process                    =   13200.92 KB/sec
             Max throughput per process                    =   22147.15 KB/sec
             Avg throughput per process                    =   16752.15 KB/sec
             Min xfer                                 = 87461560.00 KB


       iozone test complete.
```

`m4.4xlarge` NFS server

The following benchmark results are performed using an `m4.4xlarge` NFS server type together with EBS-backed storage providing a total of 9.5TB storage available to NFS clients.

Write/re-write IOZone test

```
            Iozone: Performance Test of File I/O
                    Version $Revision: 3.420 $
                    Compiled for 64 bit mode.
                    Build: linux-AMD64

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                    Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                    Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                    Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                    Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                    Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                    Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                    Vangel Bojaxhi, Ben England, Vikentsi Lapa.

            Run began: Fri May  6 15:37:45 2016

            Setting no_unlink
            Include fsync in write timing
            File size set to 125829120 KB
            Record Size 1024 KB
            Network distribution mode enabled.
            Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone.hosts
            Output is in Kbytes/sec
            Time Resolution = 0.000001 seconds.
            Processor cache size set to 1024 Kbytes.
            Processor cache line size set to 32 bytes.
            File stride size set to 17 * record size.
            Throughput test with 8 processes
            Each process writes a 125829120 Kbyte file in 1024 Kbyte records

            Test running:
            Children see throughput for  8 initial writers       =  260693.92 KB/sec
            Min throughput per process                   =   18707.17 KB/sec
            Max throughput per process                   =  115765.63 KB/sec
            Avg throughput per process                   =   32586.74 KB/sec
            Min xfer                                = 25030656.00 KB

            Test running:
            Children see throughput for  8 rewriters       =  270876.66 KB/sec
            Min throughput per process                   =   21351.16 KB/sec
            Max throughput per process                   =  118152.34 KB/sec
            Avg throughput per process                   =   33859.58 KB/sec
            Min xfer                                = 31328256.00 KB



            iozone test complete.
```

Read/re-read IOZone test

```
            Iozone: Performance Test of File I/O
                    Version $Revision: 3.420 $
                    Compiled for 64 bit mode.
                    Build: linux-AMD64

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                    Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                    Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                    Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                    Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                    Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                    Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                    Vangel Bojaxhi, Ben England, Vikentsi Lapa.

            Run began: Fri May  6 17:16:08 2016

            Setting no_unlink
            Include fsync in write timing
            File size set to 125829120 KB
      Record Size 1024 KB
            Network distribution mode enabled.
            Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
            Output is in Kbytes/sec
            Time Resolution = 0.000001 seconds.
            Processor cache size set to 1024 Kbytes.
            Processor cache line size set to 32 bytes.
            File stride size set to 17 * record size.
            Throughput test with 8 processes
            Each process writes a 125829120 Kbyte file in 1024 Kbyte records

            Test running:
            Children see throughput for  8 readers         =  420912.50 KB/sec
            Min throughput per process                     =   45498.47 KB/sec
            Max throughput per process                     =   96257.82 KB/sec
            Avg throughput per process                     =   52614.06 KB/sec
            Min xfer                                  = 59476992.00 KB

            Test running:
            Children see throughput for 8 re-readers       =  569492.38 KB/sec
            Min throughput per process                     =   49359.10 KB/sec
            Max throughput per process                     =  100858.23 KB/sec
            Avg throughput per process                     =   71186.55 KB/sec
            Min xfer                                  = 61582336.00 KB



      iozone test complete.
```

`m4.10xlarge` NFS server

The following benchmark results are performed using an `m4.10xlarge` NFS server type together with EBS-backed storage providing a total of 14TB storage available to NFS clients.

Write/re-write IOZone test

```
            Iozone: Performance Test of File I/O
                    Version $Revision: 3.420 $
                    Compiled for 64 bit mode.
                    Build: linux-AMD64

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                    Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                    Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                    Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                    Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                    Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                    Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                    Vangel Bojaxhi, Ben England, Vikentsi Lapa.

            Run began: Mon May  9 08:50:24 2016

            Setting no_unlink
            Include fsync in write timing
            File size set to 125829120 KB
            Record Size 1024 KB
            Network distribution mode enabled.
            Command line used: iozone -w -e -s 120G -i 0 -r 1M -t 8 -+m /home/alces/
iozone.hosts
            Output is in Kbytes/sec
            Time Resolution = 0.000001 seconds.
            Processor cache size set to 1024 Kbytes.
            Processor cache line size set to 32 bytes.
            File stride size set to 17 * record size.
            Throughput test with 8 processes
            Each process writes a 125829120 Kbyte file in 1024 Kbyte records

            Test running:
            Children see throughput for  8 initial writers       =  507382.46 KB/sec
            Min throughput per process                   =   56653.40 KB/sec
            Max throughput per process                   =   69806.99 KB/sec
            Avg throughput per process                   =   63422.81 KB/sec
            Min xfer                               = 104564736.00 KB

            Test running:
            Children see throughput for  8 rewriters      =  526611.62 KB/sec
            Min throughput per process                   =   59031.40 KB/sec
            Max throughput per process                   =   70774.79 KB/sec
            Avg throughput per process                   =   65826.45 KB/sec
            Min xfer                               = 105134080.00 KB


       iozone test complete.
```

Read/re-read IOZone test

```
              Iozone: Performance Test of File I/O
                      Version $Revision: 3.420 $
                      Compiled for 64 bit mode.
                      Build: linux-AMD64

              Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                      Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                      Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                      Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                      Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                      Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                      Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                      Vangel Bojaxhi, Ben England, Vikentsi Lapa.

              Run began: Mon May  9 09:53:53 2016

              Setting no_unlink
              Include fsync in write timing
              File size set to 125829120 KB
              Record Size 1024 KB
              Network distribution mode enabled.
              Command line used: iozone -w -e -s 120G -i 1 -r 1M -t 8 -+m /home/alces/
iozone.hosts
              Output is in Kbytes/sec
              Time Resolution = 0.000001 seconds.
              Processor cache size set to 1024 Kbytes.
              Processor cache line size set to 32 bytes.
              File stride size set to 17 * record size.
              Throughput test with 8 processes
              Each process writes a 125829120 Kbyte file in 1024 Kbyte records

              Test running:
              Children see throughput for  8 readers         =  488387.16 KB/sec
              Min throughput per process                     =   59843.45 KB/sec
              Max throughput per process                     =   62238.11 KB/sec
              Avg throughput per process                     =   61048.40 KB/sec
              Min xfer                                  = 120988672.00 KB

              Test running:
              Children see throughput for 8 re-readers       =  487483.99 KB/sec
              Min throughput per process                     =   59636.79 KB/sec
              Max throughput per process                     =   62291.44 KB/sec
              Avg throughput per process                     =   60935.50 KB/sec
              Min xfer                                  = 120468480.00 KB



       iozone test complete.
```

Appendix

CloudFormation template

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Launch an NFS server to your existing environment",
    "Parameters": {
        "SUBNETID": {
            "Description": "Subnet ID",
            "Type": "AWS::EC2::Subnet::Id"
        },
        "SECURITYGROUP": {
            "Description": "Security Group ID",
            "Type": "AWS::EC2::SecurityGroup::Id"
        },
        "CLUSTERNAME": {
            "Description": "Cluster name",
            "Type": "String"
        },
        "LOGINIP": {
            "Description": "Environment login node internal IP",
            "Type": "String"
        },
        "STORAGETYPE": {
            "Description": "Select the type of storage server to deploy. This defines the number of
cores and memory available - affecting the read/write performance of your storage",
            "Type": "String",
            "Default": "Medium-5TB",
            "AllowedValues": [
                "Small-2TB",
                "Medium-5TB",
                "Large-10TB",
               "XLarge-15TB"
            ]
        },
        "USERNAME": {
            "Description": "Enter the name of your environments cluster administrator user",
            "Type": "String",
            "Default": "alces"
        },
        "PLACEMENTGROUP": {
            "Description": "Enter the name of your existing clusters placement group - if your
cluster is not part of one, select None",
            "Type": "String",
            "Default": "None"
        },
        "SPOTPRICE": {
            "Description": "Enter your spot price in USD per-hour. If you do not wish to use spot,
please enter None",
            "Type": "String",
            "Default": "1"
        },
        "KEYPAIR": {
            "Description": "Administrator keypair",
            "Type": "AWS::EC2::KeyPair::KeyName"
        }
    },
    "Mappings": {
        "StorageSize": {
            "Small-2TB": {
                "volumesize": "2000"
            },
            "Medium-5TB": {
                "volumesize": "5000"
            },
            "Large-10TB": {
                "volumesize": "10000"
            },
            "XLarge-15TB": {
                "volumesize": "15000"
            }
        },
        "InstanceFlavour": {
            "Small-2TB": {
```

```
                "instancetype": "m4.xlarge"
            },
            "Medium-5TB": {
                "instancetype": "m4.2xlarge"
            },
            "Large-10TB": {
                "instancetype": "m4.4xlarge"
            },
            "XLarge-15TB": {
                "instancetype": "m4.10xlarge"
            }
        }
    },
    "Conditions": {
        "UsePlacementGroup": {
            "Fn::Equals": [
                {
                    "Ref": "PLACEMENTGROUP"
                },
                "None"
            ]
        },
        "UseSpot": {
            "Fn::Not": [
                {
                    "Fn::Equals": [
                        {
                            "Ref": "SPOTPRICE"
                        },
                        "None"
                    ]
                }
            ]
        }
    },
    "Resources": {
        "StorageConfig": {
            "Type": "AWS::AutoScaling::LaunchConfiguration",
            "Properties": {
                "AssociatePublicIpAddress": "True",
                "KeyName": {
                    "Ref": "KEYPAIR"
                },
                "ImageId": "ami-b9ff71ca",
                "InstanceType": {
                    "Fn::FindInMap": [
                        "InstanceFlavour",
                        {
                            "Ref": "STORAGETYPE"
                        },
                        "instancetype"
                    ]
                },
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "VolumeSize": "100"
                        }
                    },
                    {
                        "DeviceName": "/dev/xvdb",
                        "Ebs": {
                            "VolumeType": "st1",
                            "VolumeSize": {
                                "Fn::FindInMap": [
                                    "StorageSize",
                                    {
                                        "Ref": "STORAGETYPE"
                                    },
                                    "volumesize"
                                ]
                            }
                        }
                    }
                ],
                "SpotPrice": {
                    "Fn::If": [
                        "UseSpot",
```

```
                    {
                        "Ref": "SPOTPRICE"
                    },
                    {
                        "Ref": "AWS::NoValue"
                    }
                ]
            },
            "SecurityGroups": [
                {
                    "Ref": "SECURITYGROUP"
                }
            ],
            "UserData": {
                "Fn::Base64": {
                    "Fn::Join": [
                        "",
                        [
                            "#cloud-config",
                            "\n",
                            "system_info:",
                            "\n",
                            "  default_user:",
                            "\n",
                            "    name: ",
                            {
                                "Ref": "USERNAME"
                            },
                            "\n",
                            "hostname: storage1",
                            "\n",
                            "write_files:",
                            "\n",
                            "- content: |",
                            "\n",
                            "    cluster:",
                            "\n",
                            "      uuid: '11111111-2222-3333-444444444444'",
                            "\n",
                            "      token: '1A0a1aaAA1aAAA/aaa1aAA=='",
                            "\n",
                            "      name: ",
                            {
                                "Ref": "CLUSTERNAME"
                            },
                            "\n",
                            "      role: 'slave'",
                            "\n",
                            "      master: ",
                            {
                                "Ref": "LOGINIP"
                            },
                            "\n",
                            "  owner: root:root",
                            "\n",
                            "  path: /opt/clusterware/etc/config.yml",
                            "\n",
                            "  permissions: '0640'",
                            "\n"
                        ]
                    ]
                }
            }
        }
    },
    "Storage": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "Properties": {
            "DesiredCapacity": "1",
            "LaunchConfigurationName": {
                "Ref": "StorageConfig"
            },
            "PlacementGroup": {
                "Fn::If": [
                    "UsePlacementGroup",
                    {
                        "Ref": "AWS::NoValue"
                    },
                    {
```

```
                            "Ref": "PLACEMENTGROUP"
                        }
                    ]
                },
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": {
                            "Fn::Join": [
                                "",
                                [
                                    {
                                        "Ref": "CLUSTERNAME"
                                    },
                                    "-storage"
                                ]
                            ]
                        },
                        "PropagateAtLaunch": "true"
                    }
                ],
                "MinSize": "1",
                "MaxSize": "100",
                "VPCZoneIdentifier": [
                    {
                        "Ref": "SUBNETID"
                    }
                ]
            }
        }
    }
}
```