Scalable File Service Turbo

Best Practices

 Issue
 01

 Date
 2024-11-05





Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions

NUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road Qianzhong Avenue Gui'an New District Gui Zhou 550029 People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Migrating Data to SFS Turbo	1
1.1 Solution Overview	1
1.2 Migrating Data Using Direct Connect (rclone)	1
1.3 Cross-Server Migration (rclone)	3
1.4 Migrating Data Using Direct Connect (rsync)	7
1.5 Migrating Data Between File Systems	
1.6 Migrating Data from SFS Capacity-Oriented to SFS Turbo	11
2 Testing SFS Turbo Performance	14
3 Testing SFS Turbo Latency	22

Migrating Data to SFS Turbo

1.1 Solution Overview

By default, an SFS Turbo file system can only be accessed by Elastic Cloud Servers (ECSs) or Cloud Container Engine (CCE) containers in the same Virtual Private Cloud (VPC) as the file system. To access an SFS Turbo file system from an onpremises data center or a different VPC, you need to establish network connections using Direct Connect, Virtual Private Network (VPN), or VPC peering connections.

- To access from on premises or another cloud, use Direct Connect or VPN.
- To access from a different VPC using the same account in the same region, use VPC peering.
- To access using a different account in the same region, use VPC peering.
- To access from a different region, use Cloud Connect.

You can use an ECS that can access the Internet to migrate data to SFS Turbo.

- Mount the SFS Turbo file system to the ECS and migrate data from the local Network Attached Storage (NAS) to the SFS Turbo file system.
 - Migrating Data Using Direct Connect (rclone)
 - Migrating Data Using Direct Connect (rsync)
- If the connection cannot be established through file system mounting, migrate data through the ECS over the Internet.

Cross-Server Migration (rclone)

1.2 Migrating Data Using Direct Connect (rclone)

Solution Overview

You can migrate data from a local NAS to SFS Turbo using rclone over Direct Connect.

In this solution, a Linux ECS is created to connect the local NAS and SFS Turbo, and data is migrated to the cloud using this ECS.

You can also refer to this solution to migrate data from an on-cloud NAS to SFS Turbo. For details, see **Migrating Data from On-Cloud NAS to SFS Turbo**.

Notes and Constraints

- Only Linux ECSs can be used to migrate data.
- The UID and GID of your file will no longer be consistent after data migration.
- The file access modes will no longer be consistent after data migration.
- Incremental migration is supported, so you can only migrate the changed data.
- If data is written to the file system after you have run the rclone command to migrate data, data inconsistency may occur.

Prerequisites

- You have purchased and configured Direct Connect. For details, see *Direct Connect User Guide*.
- You have created a Linux ECS.
- You have created an SFS Turbo file system and have obtained the shared path of the file system.
- You have obtained the shared path of the local NAS.

Resource Planning

 Table 1-1 describes the resource planning in this solution.

Table	1-1	Resource	planning
		Resource	praiming

Resource	Example Configuration	Description
ECS	Specifications: 8 vCPUs 16 GB c7.2xlarge.2 OS: Linux Region: EU-Dublin VPC: VPC1	Ensure that the /mnt/src and /mnt/dst directories have been created.

Procedure

- **Step 1** Log in to the ECS console.
- **Step 2** Log in to the Linux ECS.
- **Step 3** Mount the local NAS to the ECS. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Shared path of the local NAS* /mnt/src
- **Step 4** Mount the SFS Turbo file system to the ECS. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Shared path of the file system* /mnt/dst

Step 5 Install rclone on the Linux ECS.

wget https://downloads.rclone.org/v1.53.4/rclone-v1.53.4-linux-amd64.zip --no-check-certificate unzip rclone-v1.53.4-linux-amd64.zip chmod 0755 ./rclone-*/rclone cp ./rclone-*/rclone /usr/bin/ rm -rf ./rclone-*

Step 6 Synchronize data to the SFS Turbo file system.

rclone copy /mnt/src /mnt/dst -P --transfers 32 --checkers 64 --links --create-empty-src-dirs

NOTE

The parameters are described as follows. Set **transfers** and **checkers** based on the system specifications.

- --transfers: number of files that can be transferred concurrently
- --checkers: number of local files that can be scanned concurrently
- -P: data copy progress
- --links: replicates the soft links from the source. They are saved as soft links in the destination.

--copy-links: replicates the content of files to which the soft links point. They are saved as files rather than soft links in the destination.

--create-empty-src-dirs: replicates the empty directories from the source to the destination.

After data synchronization is complete, go to the SFS Turbo file system to check whether data is migrated.

----End

Verification

- **Step 1** Log in to the Linux ECS.
- Step 2 Check the file synchronization results on the destination server.
- **Step 3** If the data volume is the same as that on the source server, data is migrated successfully.

----End

Migrating Data from On-Cloud NAS to SFS Turbo

To migrate data from an on-cloud NAS to your SFS Turbo file system, ensure that the NAS and SFS Turbo file system are in the same VPC, or you have established the network using Cloud Connect.

For details about how to configure Cloud Connect, see Cloud Connect User Guide.

1.3 Cross-Server Migration (rclone)

Solution Overview

You can use rclone to migrate data from a local NAS to SFS Turbo over the Internet or private network.

In this solution, to migrate data from the local NAS to the cloud, a Linux server is created both on the cloud and on-premises. Inbound and outbound traffic is allowed on port 22 of the two servers. The on-premises server is used to access the local NAS, and the ECS is used to access SFS Turbo.

You can also refer to this solution to migrate data from an on-cloud NAS to SFS Turbo over the Internet or private network.

Notes and Constraints

- Only Linux ECSs can be used to migrate data.
- The UID and GID of your file will no longer be consistent after data migration.
- The file access modes will no longer be consistent after data migration.
- Inbound and outbound traffic must be allowed on port 22.
- Incremental migration is supported, so you can only migrate the changed data.
- If data is written to the file system after you have run the rclone command to migrate data, data inconsistency may occur.

Prerequisites

- A Linux server has been created on the cloud and on-premises respectively.
- An elastic IP address (EIP) has been bound to the ECS to ensure that the two servers can communicate with each other.
- You have created an SFS Turbo file system and have obtained the shared path of the file system.
- You have obtained the shared path of the local NAS.

Resource Planning

Table 1-2 describes the resource planning in this solution.

Table	1-2	Resource	planning
-------	-----	----------	----------

Resource	Example Configuration	Description
ECS	Specifications: 8 vCPUs 16 GB c7.2xlarge.2	Ensure that the /mnt/dst directory has been created.
	OS: Linux	
	Region: EU-Dublin	
	VPC: VPC1	
	Enabled port: 22	
	EIP: xxx.xxx.xxx	

Procedure

Step 1 Log in to the ECS console.

Step 2 Log in to the on-premises server **client1** and run the following command to mount the local NAS:

mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Shared path of the local NAS /mnt/src

Step 3 Log in to the Linux ECS **client2** and run the following command to mount the SFS Turbo file system:

mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Shared path of the SFS Turbo file system* /mnt/dst

Step 4 Install rclone on client 1.

wget https://downloads.rclone.org/v1.53.4/rclone-v1.53.4-linux-amd64.zip --no-check-certificate unzip rclone-v1.53.4-linux-amd64.zip chmod 0755 ./rclone-*/rclone cp ./rclone-*/rclone /usr/bin/ rm -rf ./rclone-*

Step 5 Configure the environment on **client1**.

rclone config No remotes found - make a new one n) New remote s) Set configuration password q) Quit config n/s/q> n name> remote name (New name) Type of storage to configure. Enter a string value. Press Enter for the default (""). Choose a number from below, or type in your own value 24 / SSH/SFTP Connection \ "sftp" Storage> 24 (Select the SSH/SFTP number) SSH host to connect to Enter a string value. Press Enter for the default (""). Choose a number from below, or type in your own value 1 / Connect to example.com \ "example.com' host> ip address (IP address of client2) SSH username, leave blank for current username, root Enter a string value. Press Enter for the default (""). user> user name (Username of client2) SSH port, leave blank to use default (22) Enter a string value. Press Enter for the default (""). port> 22 SSH password, leave blank to use ssh-agent. y) Yes type in my own password g) Generate random password n) No leave this optional password blank y/g/n> y Enter the password: password: (Password for logging in to client2) Confirm the password: password: (Confirm the password) Path to PEM-encoded private key file, leave blank or set key-use-agent to use ssh-agent. Enter a string value. Press Enter for the default (""). key_file> (Press Enter) The passphrase to decrypt the PEM-encoded private key file. Only PEM encrypted key files (old OpenSSH format) are supported. Encrypted keys in the new OpenSSH format can't be used. y) Yes type in my own password g) Generate random password n) No leave this optional password blank y/g/n> n When set forces the usage of the ssh-agent. When key-file is also set, the ".pub" file of the specified key-file is read and only the associated key is requested from the ssh-agent. This allows to avoid 'Too many authentication failures for *username*' errors when the ssh-agent contains many keys. Enter a boolean value (true or false). Press Enter for the default ("false").

Enter a boolean value (true or false). Press Enter for the default ("false key use agent> (Press Enter)

Enable the use of the aes128-cbc cipher. This cipher is insecure and may allow plaintext data to be recovered by an attacker. Enter a boolean value (true or false). Press Enter for the default ("false"). Choose a number from below, or type in your own value 1 / Use default Cipher list. \ "false" 2 / Enables the use of the aes128-cbc cipher. \ "true" use_insecure_cipher> (Press Enter) Disable the execution of SSH commands to determine if remote file hashing is available. Leave blank or set to false to enable hashing (recommended), set to true to disable hashing. Enter a boolean value (true or false). Press Enter for the default ("false"). disable hashcheck> Edit advanced config? (y/n) y) Yes n) No y/n> n Remote config [remote name] type = sftp host=(*client2 ip*) user=(*client2 user name*) port = 22 pass = *** ENCRYPTED *** key_file_pass = *** ENCRYPTED *** y) Yes this is OK e) Edit this remote d) Delete this remote y/e/d> y Current remotes: Name Туре ==== ==== sftp remote_name e) Edit existing remote n) New remote

d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q

NOTE

Enter a public IP address for IP address of client2.

Step 6 View the rclone.conf file in /root/.config/rclone/rclone.conf.

cat /root/.config/rclone/rclone.conf
[remote_name]
type = sftp
host=(*client2 ip*)
user=(*client2 user name*)
port = 22
pass = ***
key_file_pass = ***

Step 7 Run the following command on client1 to synchronize data:

rclone copy /mnt/src remote_name./mnt/dst -P --transfers 32 --checkers 64

- Replace *remote_name* in the command with the actual remote name.
- The parameters are described as follows. Set **transfers** and **checkers** based on the system specifications.
 - transfers: number of files that can be transferred concurrently
 - checkers: number of local files that can be scanned concurrently
 - P: data copy progress

After data synchronization is complete, go to the SFS Turbo file system to check whether data is migrated.

----End

Verification

- **Step 1** Log in to the Linux ECS.
- Step 2 Check the file synchronization results on the destination server.
- **Step 3** If the data volume is the same as that on the source server, data is migrated successfully.

----End

1.4 Migrating Data Using Direct Connect (rsync)

Solution Overview

You can migrate data from a local NAS to SFS Turbo using rsync over Direct Connect.

In this solution, a Linux ECS is created to connect the local NAS and SFS Turbo, and data is migrated to the cloud using this ECS.

You can also refer to this solution to migrate data from an on-cloud NAS to SFS Turbo using the Internet. Ensure that the on-cloud NAS and SFS Turbo belong to the same VPC.

Notes and Constraints

- Special files, such as devices and linked files, can be migrated.
- Resumable data transfer is supported.
- Properties, such as permissions, time, soft and hard links, owner, and group, of the original files and directories can be retained after data migration.
- The rcp, rsh, and ssh tools are supported during file transfer.
- Incremental migration is supported, so you can only migrate the changed data.
- If there are multi-level directories or massive small files, you are recommended to use a multi-process script or rclone. This is because rsync has low efficiency in these scenarios.

Prerequisites

- You have purchased and configured Direct Connect. For details, see *Direct Connect User Guide*.
- You have created a Linux ECS.
- You have created an SFS Turbo file system and have obtained the shared path of the file system.
- You have obtained the shared path of the local NAS.
- A trust relationship has been established between the local NAS and the ECS.

Resource Planning

 Table 1-3 describes the resource planning in this solution.

Resource	Example Configuration	Description
ECS	Specifications: 8 vCPUs 16 GB c7.2xlarge.2 OS: Linux	Ensure that the /mnt/src and /mnt/dst directories have been created.
	Region: EU-Dublin VPC: VPC1	

Table '	1-3	Resource	planning
---------	-----	----------	----------

Procedure

- **Step 1** Log in to the Linux ECS.
- **Step 2** Mount the local NAS to the ECS. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Shared path of the local NAS* /mnt/src
- **Step 3** Mount the SFS Turbo file system to the ECS. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Shared path of the file system* /mnt/dst
- Step 4 Install rsync on the Linux ECS. yum install rsync

Figure 1-1 Installing rsync

[root@	<pre>~]# yum install rsync</pre>
Loaded plugins: fastes	stmirror
Determining fastest mi	irrors
epel/x86_64/metalink	
* base: mirrors.huawe	eicloud.com
<pre>* epel: mirrors.bfsu.</pre>	edu.cn
* extras: mirrors.ust	tc.edu.cn
* undatos, mirrors hu	uawaicloud com

NOTE

Ensure that rsync is installed on both the source and destination servers. Or, an error will be reported.

Step 5 Check the installation result and version of rsync.

rsync -version

Figure 1-2 Checking the installation result

[root@]# rsync -version
rsync version 3.1.2 protocol version 31
Copyright (C) 1996-2015 by Andrew Tridgell, Wayne Davison, and others.
Web site: http://rsync.samba.org/
Capabilities:
64-bit files, 64-bit inums, 64-bit timestamps, 64-bit long ints, socketpairs, hardlinks, symlinks, IPv6, batchfiles, inplace, append, ACLs, xattrs, iconv, symtimes, prealloc
rsync comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under certain conditions. See the GNU General Public Licence for details.
rsync is a file transfer program capable of efficient remote update via a fast differencing algorithm.
Usage: rsync [OPTION] SRC [SRC] DEST
or rsync [OPTION] SRC [SRC] [USER@]HOST:DEST
or rsync [0PTION] SRC [SRC] [USER@]H0ST::DEST
or rsync [0PTI0N] SRC [SRC] rsync://[USER@]H0ST[:P0RT]/DEST
or rsync [OPTION] [USER@]HOST:SRC [DEST]
or rsync [OPTION] [USER@]HOST::SRC [DEST]
or rsync [OPTION] rsync://[USER@]HOST[:PORT]/SRC [DEST]
The ':' usages connect via remote shell, while '::' & 'rsync://' usages connect
to an rsync daemon, and require SRC or DEST to start with a module name.

Step 6 Migrate data in the **/mnt/src** directory on the source server to the **/mnt/dst** directory (file system) on the destination server. rsync -avP /mnt/src /mnt/dst

J	.,				
103669.dat					
1,024	100%	38.46kB/s	0:00:00	(xfr#4080,	to-chk=295920/300001
10367.dat					
1,024	100%	37.04kB/s	0:00:00	(xfr#4081,	to-chk=295919/300001
103670.dat					
1,024	100%	37.04kB/s	0:00:00	(xfr#4082,	to-chk=295918/300001
103671.dat					
1,024	100%	35.71kB/s	0:00:00	(xtr#4083,	to-chk=295917/300001
1036/2.dat	1000		0 00 00	1 5	
1,024	100%	35./1KB/S	0:00:00	(XTT#4084,	to-cnk=295916/300001
1030/3.dat	1000	24 40LD/-	0.00.00	1	+
1,024	100%	34.48KB/S	0:00:00	(XTT#4085,	to-cnk=295915/300001
103074.0at	100%	24 40kp/a	0.00.00	(xfp#4006	to obk-205014/200001
1,024 102675 dat	100%	34,40KD/5	0:00:00	(XII#4000,	CO-CHK=295914/500001
1 020/2-Udl	100%	22 22kB/c	0.00.00	(xfr#1097	to-chk-205013/300001
1,024	100%	33-33KD/8	0.00:00	(X11#4007,	CO-CIIK-295915/500001

Figure 1-3 Data synchronization

----End

Verification

- **Step 1** Log in to the Linux ECS.
- Step 2 Check the file synchronization results on the destination server. cd /mnt/dst ls | wc -l
- **Step 3** If the data volume is the same as that on the source server, data is migrated successfully.

----End

1.5 Migrating Data Between File Systems

Solution Overview

You can migrate data from an SFS Capacity-Oriented file system to an SFS Turbo file system or the other way around.

In this solution, a Linux ECS is used to connect the SFS Capacity-Oriented file system and the SFS Turbo file system.

Notes and Constraints

- Only Linux ECSs can be used to migrate data.
- The Linux ECS, SFS Capacity-Oriented file system, and SFS Turbo file system must be in the same VPC.
- Incremental migration is supported, so you can only migrate the changed data.

Prerequisites

- You have created a Linux ECS.
- You have created an SFS Capacity-Oriented file system and an SFS Turbo file system and have obtained their addresses.

Procedure

- **Step 1** Log in to the ECS console.
- **Step 2** Log in to the Linux ECS.
- **Step 3** Mount either the SFS Capacity-Oriented or SFS Turbo file system to the ECS. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Address of the file system* /mnt/src
- **Step 4** Mount the other file system to the ECS. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Address of the other file system* /mnt/dst
- **Step 5** Download and install rclone. For the download address, see https://rclone.org/ downloads/.
- Step 6
 Synchronize data.

 rclone copy /mnt/src /mnt/dst -P --transfers 32 --checkers 64 --links --create-empty-src-dirs

The parameters are described as follows. Set **transfers** and **checkers** based on the system specifications.

- /mnt/src: source path
- /mnt/dst: destination path
- --transfers: number of files that can be transferred concurrently
- --checkers: number of local files that can be scanned concurrently
- -P: data copy progress
- --links: replicates the soft links from the source. They are saved as soft links in the destination.
- --copy-links: replicates the content of files to which the soft links point. They are saved as files rather than soft links in the destination.
- --create-empty-src-dirs: replicates the empty directories from the source to the destination.

After data synchronization is complete, go to the target file system to check whether data is migrated.

----End

Verification

- **Step 1** Log in to the Linux ECS.
- Step 2 Check the file synchronization results on the destination server. cd /mnt/dst ls | wc -l
- **Step 3** If the data volume is the same as that on the source server, data is migrated successfully.

----End

1.6 Migrating Data from SFS Capacity-Oriented to SFS Turbo

Solution Overview

You can migrate data from an SFS Capacity-Oriented file system to an SFS Turbo file system.

In this solution, a Linux ECS is used to connect the SFS Capacity-Oriented file system and the SFS Turbo file system.

Notes and Constraints

- Only Linux ECSs can be used to migrate data.
- The Linux ECS, SFS Capacity-Oriented file system, and SFS Turbo file system must be in the same VPC.
- Incremental migration is supported, so you can only migrate the changed data.

Prerequisites

- You have created a Linux ECS.
- You have created an SFS Capacity-Oriented file system and an SFS Turbo file system and have obtained their addresses.

Procedure

- **Step 1** Log in to the ECS console.
- **Step 2** Log in to the Linux ECS.
- **Step 3** Mount the SFS Capacity-Oriented file system, which is *file system 1* in this example.

mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Address of the Capacity-Oriented file system* /mnt/src

Step 4 Mount the SFS Turbo file system, which is *file system 2* in this example. mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp *Address of the SFS Turbo file system* /mnt/dst

Step 5 Install rclone on the Linux ECS.

```
wget https://downloads.rclone.org/v1.53.4/rclone-v1.53.4-linux-amd64.zip --no-check-certificate
unzip rclone-v1.53.4-linux-amd64.zip
chmod 0755 ./rclone-*/rclone
cp ./rclone-*/rclone /usr/bin/
rm -rf ./rclone-*
```

NOTE

rclone does not retain the file permissions or owner group information on the source. Use rsync if you have such requirements.

Step 6 Synchronize data to the SFS Turbo file system.

rclone copy /mnt/src /mnt/dst -P --transfers 32 --checkers 64 --links --create-empty-src-dirs

NOTE

The parameters are described as follows. Set **transfers** and **checkers** based on the system specifications.

- /mnt/src: source path
- /mnt/dst: destination path
- --transfers: number of files that can be transferred concurrently
- --checkers: number of local files that can be scanned concurrently
- -P: data copy progress
- --links: replicates the soft links from the source. They are saved as soft links in the destination.
- --copy-links: replicates the content of files to which the soft links point. They are saved as files rather than soft links in the destination.
- --create-empty-src-dirs: replicates the empty directories from the source to the destination.

After data synchronization is complete, go to the SFS Turbo file system to check whether data is migrated.

----End

Verification

- **Step 1** Log in to the Linux ECS.
- Step 2 Check the file synchronization results on the destination server.

If the data volume is the same as that on the source server, data is migrated successfully.

----End

2 Testing SFS Turbo Performance

Fio is an open-source I/O tester. You can use fio to test the throughput and IOPS of SFS Turbo file systems.

Prerequisites

Fio has been installed on the cloud server. It can be downloaded from **the official website** or from **GitHub**.

Note and Description

The test performance depends on the network bandwidth between the client and server, as well as the capacity of the file system.

Installing fio

The following uses a Linux CentOS system as an example:

- 1. Download fio. yum install fio
- 2. Install the libaio engine. yum install libaio-devel
- 3. Check the fio version. **fio --version**

File System Performance Data

Table 2-1 lists the performance metrics of SFS Turbo file systems including IOPS and throughput.

Para meter	20 MB/s/Ti B	40 MB/s/Ti B	125 MB/s/TiB	250 MB/s/TiB	500 MB/s/TiB	1,000 MB/s/TiB	
Max. capaci ty	1 PB	1 PB	1 PB	1 PB	1 PB	1 PB	
Max. IOPS	250,000	250,000	1 million	1 million	n 4 million 4 mil If you If you need a need higher higher IOPS, IOPS submit a subm service servit ticket. ticket The The maximum maxi IOPS is 10 IOPS million. millio		
Max. throu ghput	8 GB/s If you need a higher through put, submit a service ticket. The maximu m through put is 20 Gbit/s.	8 GB/s If you need a higher throughp ut, submit a service ticket . The maximu m throughp ut is 20 Gbit/s.	20 GB/s If you need a higher throughp ut, submit a service ticket . The maximum throughp ut is 100 Gbit/s.	20 GB/s If you need a higher throughpu t, submit a service ticket . The maximum throughpu t is 100 Gbit/s.	80 GB/s If you need a higher throughpu t, submit a service ticket . The maximum throughpu t is 200 Gbit/s.	80 GB/s If you need a higher throughpu t, submit a service ticket . The maximum throughpu t is 200 Gbit/s.	
Formu la used to calcul ate the IOPS	IOPS = Min. (250,000 , 600 x Capacity) Capacity unit: TB	IOPS = Min. (250,000, 1,200 x Capacity) Capacity unit: TB	IOPS = Min. (1,000,00 0, 6,000 x Capacity) Capacity unit: TB	IOPS = Min. (1,000,000 , 12,500 x Capacity) Capacity unit: TB	IOPS = Min. (4,000,000 , 25,000 x Capacity) Capacity unit: TB	IOPS = Min. (4,000,000 , 50,000 x Capacity) Capacity unit: TB	

 Table 2-1
 File system performance data

Common Test Configuration Example

NOTE

The following estimated values are obtained from the test on a single ECS. You are advised to use multiple ECSs to test the **SFS Turbo** performance.

In the following examples, the Performance type and cloud servers with the following specifications are used for illustration.

Specifications: General computing-plus | c3.xlarge.4 | 4 vCPUs | 16 GB

Image: CentOS 7.5 64-bit

fio command:

```
fio --randrepeat=1 --ioengine=libaio --name=test -output=output.log --
direct=1 --filename=/mnt/nfs/test_fio --bs=1M --iodepth=128 --
size=10240M --readwrite=rw --rwmixwrite=30 --fallocate=none
```

NOTE

Variable */mnt/nfs/test_fio* is the location of the file to be tested. The location must be specific to the file name, which is the **test_fio** file in the **/mnt/nfs** directory in this example. Set it based on the site requirements.

• fio result:

<pre>test: (group1d=0, jobs1): err= 0: pid=10110: Mon Jun 0 11:40:57 2020 read: 1095-7423. Bk+23.0H1ks: (30,4H9:2)[7167H18-271160mec) stat (usec): min=12.54, max=397477, avg=11156.18, stdev=3216.31 lat (usec): min=252, max=133338, avg=11166.32, stdev=12136.31 lat (usec): min=252, max=133338, avg=11166.32, stdev=12136.31 lat (usec): min=252, max=133338, avg=11166.32, stdev=12136.34 clat percentiles (usec): i. 0.0000+12.5, 5.000+112.540], 10.0000+12.430.35, 00.0000+12.430.45, 00.0000+12.44, 00.0000+12.430.45,</pre>	
<pre>pread: 10PS=7423, BW=23.4HTB/s (38.4HB/s)[716/M18/24716Bmsec) stat (nscc): min=245, max=133325, avg=11162.18, stdew=12136.31 lat (uscc): min=252, max=133325, avg=11162.18, stdew=12136.31 clat (uscc): min=252, max=133325, avg=11162.18, stdew=12136.34 clat percentiles (uscc): i 1.08th=[2245], 5.08th=[2540], 10.08th=[2671], 20.08th=[7932], i 78.08th=[113173], 00.08th=[1972], 90.08th=[2071], 20.08th=[7932], i 99.98th=[513173], 00.08th=[1972], 90.08th=[2071], 20.08th=[7932], i 99.98th=[513173], 00.08th=[1972], 90.08th=[2071], 20.08th=[2053], i 99.99th=[10842] bu (K12x3): min=16600, max=45560, per=100.08%, avg=29696.08, stdew=5544.46, samples=494 i min= 1150, max=11390, avg=7424.01, stdew=3359.60 clat (uscc): min=1480, max=495660, avg=14166.45, stdew=13373.72 lat (uscc): min=1487, max=148666, avg=14166.45, stdew=13373.72 lat (uscc): min=1447, max=148666, avg=14178.73, stdew=3359.60 clat (uscc): min=1447, max=148666, avg=14178.73, stdew=13373.74 clat percentiles (mscc): i 1.08th=[41, 5.08th=[41], 10.08th=[41], 20.86th=[51], i 38.08th=[51], 40.08th=[61], 99.98th=[67], 99.95th=[91], i 99.98th=[101], 00.08th=[241], 90.08th=[33], 95.60th=[42], i 99.98th=[101], 00.08th=[241], 90.08th=[33], 95.60th=[42], i 99.98th=[113], 00.08th=[241], 90.08th=[37], 99.95th=[91], i 99.98th=[122] bu (K12x3): min= 7144, max=19600, per=100.08%, avg=12730.90, stdew=2395.77, samples=494 iops</pre>	test: (groupid=0, jobs=1): err= 0: pid=10110: Mon Jun 8 11:40:57 2020
<pre>slat (usec): min=124, max=3y/47/, aug=1195.45, stdev=3347.48 clat (usec): min=255, max=133325, aug=11162.18, stdev=12136.31 lat (usec): min=252, max=133328, aug=11162.63, stdev=12136.34 clat percentiles (usec):</pre>	read: 10PS=7423, BW=28.001B/s (30.406/s)(716/01B/24716000sec)
<pre>clat (usec): min=252, max=133328, avg=11162.10, stdev=12136.31 lat (usec): min=252, max=133338, avg=11166.32, stdev=12136.34 clat percentiles (usec):</pre>	slat (nsec): min=1234, max=397477, avg=1145.45, stdev=3344.40
<pre>lat (usec): min=252, max=133388, avg=11166.32, stdev=12136.34 clat percentiles (usec): i 1.08th=t 22451, 5.08th=t 25401, 10.08th=t 26711, 20.08th=t 29001, i 30.08th=t 231301, 40.08th=t 34501, 50.08th=t 242331, 60.08th=t 29001, i 70.08th-t 131731, 80.08th=t 137921, 90.08th=t 20431, 95.08th=t 364391, i 99.08th=t (532161, 99.58th=t 600311, 99.99th=t (791601, 99.55th=t 854591, i 99.59th=t (50042) bw (K18xs): min=16600, max=45568, per=100.08te, avg=29696.08, stdev=5544.46, samples=494 iops : min=14680, max=452580, org=4013.59, stdev=3359.60 clat (usec): min=1487, max=148667, avg=1412.47160msec) slat (nsec): min=1487, max=148667, avg=14170.73, stdev=13373.72 lat (usec): min=1477, max=1486671, avg=14170.73, stdev=13373.74 clat percentiles (msec): i 1.08th=t 41, 5.08th=t 41, 10.080th=t 331, 95.08th=t 421, i 70.08th=t 131, 80.08th=t 241, 90.08th=t 331, 95.08th=t 421, i 99.08th=t 591, 99.59th=t 671, 99.98th=t 871, 99.95th=t 941, i 99.08th=t 591, 99.59th=t 671, 99.98th=t 871, 99.95th=t 941, i 99.08th=t 1223 bw (K18xs): min=7144, max=19600, per=100.08x, avg=12730.98, stdev=2395.77, samples=494 iops : min=1786, max=4900, avg=3182.78, stdev=538, 96, samples=494 lat (usec) : 100=1.62x, 250=0.042x, 100=0.042, 220=0.06x lat (msec) : 2=0.28x, 4=33.15x, 10=21.012x, 28=17.92x, 59=20.06x lat (msec) : 2=0.28x, 4=33.15x, 10=21.012x, 28=17.92x, 59=20.06x lat (msec) : 2=0.28x, 4=33.15x, 10=21.012x, 28=17.92x, 59=20.06x lat (msec) : 100=1.62x, 250=0.02x, 10=21.012x, 230=0.02x, 50=20.00x lat (msec) : 100=1.62x, 250=0.02x, 10=20.02x, 59=2.006x lat (msec) : 100=1.62x, 250=0.02x, 10=20.02x, 59=2.006x lat (msec) : 100=1.62x, 250=0.02x, 10=20.02x, 59=2.006x lat (msec) : 100=1.62x, 250=0.02x, 10=0.02x, 23=0.12x, 50=130.08x submit : 0=0=0.02x, 4=00=0.02x, 10=0.02x, 50=20.02x gu : use=1.35x, sys=6.43x, ctx=1072310, majf=0, minf=30 l0 depths : 1=0.12x, 2=0.12x, 4=0.12x, 3==0.12x, 32=0.12x, 50=130.08x submit : 0=0=0.8x, 4</pre>	clat (usec): min=245, max=13325, avg=11162.10, stdev=12136.31
<pre>clat percentiles (usec): i 1.08th=[2548], 10.90th=[2671], 20.90th=[7832], i 38.80th=[3138], 40.80th=[3458], 58.80th=[4293], 60.80th=[7832], i 78.80th=[13173], 80.80th=[1972], 90.80th=[2043], 99.95th=[36439], i 99.95th=[36042] bw (KiBxs): min=16668, max=45568, per=180.80tx, avg=29696.88, stdev=5544.46, samples=494 iops : min= 4158, max=14390, avg=7424.81, stdev=1386.11, samples=494 iops : min=1488, max=382738, avg=7424.81, stdev=3359.68 clat (usec): min=1487, max=148666, avg=14166.85, stdev=3373.72 lat (usec): min=1487, max=148666, avg=14166.85, stdev=13373.72 lat (usec): min=1477, max=148666, avg=14178.73, stdev=3373.74 clat percentiles (msec): i 1.88th=[51, 48.80th=[41, 19.80th=[41, 28.80th=[51, 138.80th=[51, 138.80th=[51, 48.80th=[61, 59.80th=[41], 28.80th=[141, 178.80th=[51, 19.90th=[671, 99.90th=[671, 99.95th=[941, 19.90th=[421, 19.90th=[671, 99.95th=[941, 19.90th=[421, 19.90th=[671, 99.95th=[941, 19.90th=[141, 19.90th=[61], 99.90th=[671, 99.95th=[941, 19.90th=[141, 19.90th=[671, 99.95th=[51, 19.90th=[671, 99.95th=[941, 19.90th=[671, 99.95th=[671, 99.95th=[941, 19.90th=[122] bw (KiBx's): min= 7144, max=19688, per=180.80tx, avg=12738.98, stdev=2395.77, samples=494 iops : min= 1786, max=4968, avg=3182.78, stdev=598.96, samples=494 iops : min= 1786, max=4968, avg=3182.78, stdev=598.96, samples=494 iops : min= 1786, max=4968, avg=3182.78, stdev=598.96, samples=494 iops : sim= 1786, max=4968, avg=3182.78, stdev=588.96, samples=494 iops : sim= 1786, max=4968, avg=3182.78, stdev=588.96, samples=494 iops : sim= 1686.80tx, ctx=1872918, ma_f=8, id=80, kdex cpu : usr=1.35tx, zys=6.43tx, ctx=1872918, ma_f=8, id=80, kdex complete : 8=8.</pre>	lat (usec): min=252, max=133338, avg=11166.32, stdev=12136.34
 i 1.08th=[2245], 5.08th=[2540], 10.08th=[2671], 20.08th=[2900], i 80.08th=[13173], 00.08th=[15722], 90.08th=[2013], 95.08th=[26433], i 99.08th=[13173], 00.08th=[15722], 90.08th=[20143], 95.08th=[36433], i 99.08th=[13173], 00.08th=[15722], 90.08th=[20143], 95.08th=[36433], i 99.99th=[30042] bu (KiB/s): min=1650, max=45560, per=100.082, avg=29696.08, stdev=5544.46, samples=494 iops : min=1450, max=382730, avg=4613.59, stdev=3356.11, samples=494 iops : min=1468, max=382730, avg=4613.59, stdev=3359.60 clat (usec): min=1477, max=140666, avg=14166.65, stdev=13373.72 lat (usec): min=1477, max=140667, avg=14178.73, stdev=13373.74 clat percentiles (msec): i .08th=[41, 18.08th=[41, 18.08th=[41, 28.08th=[51, 13.08th=[51, 13.08th=] 51, 13.08th=[51, 13.08th=[51, 13.08th=] 51, 13.08th=] 51, 13.08th=[51, 13.08th=] 51, 13.08th=] 51, 13.08th=[51, 13.08th=] 52, 13.08th=] 52, 13.08th=] 52, 13.08th=[51, 13.08th=] 52, 13.08th=] 52, 13.08th=] 52, 13.08th=[51, 13.08th=] 52, 13.08th=] 52, 13.08th=] 52, 13.08th=] 5	clat percentiles (usec):
<pre>i 38.88th=[3138], 48.08th=[3458], 58.88th=[4293], 68.08th=[7832],</pre>	; 1.80th=[2245], 5.80th=[2540], 10.80th=[2671], 20.80th=[2980],
<pre>i 78.88th=[13173], 88.88th=[15792], 98.88th=[2843], 95.88th=[36439], i 99.98th=[53216], 99.58th=[68031], 99.98th=[79168], 99.95th=[85459], i 99.99th=[98042] bw (KiBrs): min=16688, max=45568, per=188.88z, avg=29696.88, stdev=5544.46, samples=494 iops : min=4158, max=11398, avg=7424.81, stdev=1386.11, samples=494 iorite: IDFS=3182, B&=12.4HiBrs (13.8HBrs)]3973HiBr247168mesec) slat (usec): min=1447, max=1486671, avg=1416.85, stdev=13373.72 lat (usec): min=1477, max=148671, avg=14178.73, stdev=13373.74 clat percentiles (msec): i 1.88th=[41, 5.88th=[41, 18.88th=[41, 28.88th=[51, i 38.88th=[5], 48.88th=[61, 58.88th=[81, 68.08th=[421, i 99.88th=[53], 99.58th=[671, 99.99th=[87], 99.95th=[941], i 99.88th=[53], 99.58th=[67], 99.99th=[67], 99.95th=[941], i 99.98th=[53], 99.58th=[67], 99.99th=[87], 99.95th=[941], i 99.88th=[53], 99.58th=[67], 99.98th=[67], 99.95th=[94], i 99.88th=[53], 99.58th=[67], 99.98th=[67], 99.95th=[94], i 99.88th=[53], 99.58th=[67], 99.98th=[67], 99.95th=[94], i 99.98th=[53], 99.58th=[67], 99.98th=[67], 99.95th=[94], i 1221 bw (KiBrs): min= 7144, max=19688, per=188.88z, avg=12738.98, stdev=2395.77, samples=494 iops : min= 1786, max= 4988, avg=3182.78, stdev=598.96, samples=494 iat (msec) : 258=8.81x, 508=8.81x, 708=8.81x, 1088=8.01x lat (msec) : 180=1.62x, 258=8.82x, cpu : usr=1.35x, sys=6.43x, ctx=1872918, majf=8, minf=38 10 depths : 1=0.1z, 2=0.1z, 4=0.1z, 10=1z, 2=0.1z, 3=0.1z, 3=0.418, 3=0.48x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16=8.8x, 3=4.8x, 64=8.8x, 54=4.8x, 54=4.8x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16=8.8x, 3=4.8x, 64=8.8x, 54=4.8x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16=8.8x, 3=2-8.8x, 64=8.8x, 54=4.8x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16=8.8x, 3=2-8.8x, 64=8.8x, 3=0=4.8x, latency : target=8, window=8, percentile=160.88x, dept=428. Bun status group 8 (a11 jobs): READ: bw=28.0811Brs (38.4HErs), 28.041Brs (38.4HErs=38.4HBrs), io=7167HiB (7515HB), run=247168-247168mesec</pre>	: 30.00th=[3130], 40.00th=[3458], 50.00th=[4293], 60.00th=[7832],
<pre>i 99.88th=[53216], 99.58th=[68031], 99.98th=[79168], 99.95th=[85459],</pre>	; 78.80th=[13173], 80.00th=[19792], 90.00th=[28443], 95.00th=[36439],
<pre>i 99,99th=[30042] bw (K18×3): min=116600, max=45560, per=100.002, avg=29696.00, stdev=5544.46, samples=494 iops : min= 4150, max=41390, avg=7424.01, stdev=1306.11, samples=494 write: 1078=3102, BW+12.4M18/x (13.0M1/x) 3073M1B>247160msec) slat (usec): min=1480, max=302730, avg=7413.65, stdev=13373.72 lat (usec): min=1487, max=140666, avg=14170.73, stdev=13373.72 lat (usec): min=1477, max=140666, avg=14170.73, stdev=13373.74 clat percentiles (msec): i 1.000000000000000000000000000000000000</pre>	: 99.80th=[53216], 99.50th=[60031], 99.90th=[79168], 99.95th=[85459],
 bu (KiB/s): min=16600, max=45560, per=100.002, avg=29656.00, stdev=5544.46, samples=494 iops : min=4150, max=1330, avg=4613.59, stdev=3356.11, samples=494 iorte: IOPS=3182, BW=12.4MiB/s (13.0MB/s) 3073MiB/247160msec) slat (nsec): min=1407, max=3082730, avg=4613.59, stdev=3373.72 lat (usec): min=1477, max=140666, avg=14166.65, stdev=13373.74 clat (usec): min=1477, max=140671, avg=14170.73, stdev=13373.74 clat (usec): min=1474, max=140671, avg=14170.73, stdev=13373.74 clat (usec): 101, 00.0001e1 (241, 90.0001e1 (331, 95.001e1 (421, 141, 141, 141, 141, 141, 141, 141,	1 99.99th=[98042]
<pre>iops : min= 4158, max=11398, avg=7424.81, stdev=1386.11, samples=494 inite: 10PS=3182, Bu=12.44HExs (13.80%=x)392734HEx247166Mesec) slat (nsec): min=1488, max=342738, avg=4613.59, stdev=3359.68 clat (usec): min=1477, max=148666, avg=14166.65, stdev=13373.72 lat (usec): min=1477, max=148671, avg=14178.73, stdev=13373.74 clat percentiles (msec): i 1.88th=[41, 5.88th=[41, 18.88th=[41, 28.88th=[51, i 38.88th=[51, 48.88th=[61, 58.88th=[81, 68.88th=[141, i 78.88th=[53, 99.89th=[79.99.88th=[81, 68.88th=[42], i 99.98th=[59], 99.58th=[67], 99.98th=[67], 99.99th=[42], i 99.98th=[181, 88.88th=[141, 99.88th=[42], 99.99th=[42], i 99.98th=[122] bu (KHs:): min= 7144, max=19608, per=180.88z, avg=12738.98, stdev=2395.77, samples=494 iops :: min= 7166, max= 4988, avg=3182.78, stdev=538.96, samples=494 lat (usec) : 258-8.81x, 598-8.81x, 758-8.8x, 618-8.8x, 168-8.8x, cpu : use=1.35x, sys=5.43x, ctx=1872918, majf=8, minf=38 I0 depths : 1=8.1z, z=8.1z, 4=8.1z, 16=8.1z, 32=8.1z, x>564=188.8z, submit : 9=8.8z, 4=188.8z, 8=.8z, 16=8.8z, 32=8.8z, 64=8.8z, >564=8.8z, complete: 8=8.8z, 4=188.8z, 8=8.8z, 16=8.8z, 32=8.8z, 64=8.8z, >564=8.8z, co</pre>	bw (KiB/s); min=16600, max=45560, per=100.00%, avg=29696.00, stdev=5544.46, samples=494
<pre>burite: 10F3=3182, BW=12.4HiB/s (13.0HB/s)(39734)B/2471060mec) slat (usec): min=1488, max=382738, aug=4613.59, stdev=3359.68 clat (usec): min=1447, max=148666, aug=14166.05, stdev=13373.72 lat (usec): min=1447, max=148666, aug=14178.73, stdev=13373.74 clat percentiles (msec): i 1.08th=[41, 5.08th=[41, 18.08th=[41, 28.08th=[51, i 38.08th=[51, 48.08th=[61, 58.08th=[81, 60.08th=[141, i 79.08th=[53], 99.58th=[67], 99.99th=[87], 99.95th=[42], i 99.08th=[53], 99.58th=[67], 99.99th=[67], 99.95th=[94], i 99.99th=[122] bw (KiB/s): min= 7144, max=19608, per=100.082%, aug=12738.98, stdev=2395.77, samples=494 iops : min= 1786, max= 4908, aug=3182.78, stdev=598.96, samples=494 lat (usec) : 258=8.81%, 508=8.01%, 788=8.01%, 1080=6.01% lat (msec) : 2.8=8.81%, 508=6.81%, 758=8.01%, 1080=6.01% lat (msec) : 108=1.62%, 258=8.02% cpu : usr=1.35%, sys=6.43%, ctx=1072910, majf=0, minf=38 l0 depths : 1=0.1%, 2=0.1%, 4=0.1%, 16=8.1%, 32=8.0%, 54=8.0%, 54=8.0% complete : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.0%, 32=8.0%, 54=8.0%, 54=8.0% complete : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.0%, 32=8.0%, 54=8.0%, 8= complete : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.0%, 32=8.0%, 54=8.0%, 8= complete : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.0%, 32=8.0%, 54=8.0%, 8= latency : target=0, unndow=0, percentile=160.08%, dep=128 Run status group 0 (al1 jobs):</pre>	iops : min= 4150, max=11390, avg=?424.01, stdev=1386.11, samples=494
<pre>slat (nsec): min=1488, max=382738, avg=4613.59, stdev=3359.68 clat (usec): min=1477, max=148667, avg=1416.65, stdev=13373.72 lat (usec): min=1477, max=148671, avg=14178.73, stdev=13373.74 clat percentiles (msec): i 1.88th=t 41, 5.88th=t 41, 18.88th=t 41, 28.88th=t 51, i 38.88th=t 51, 40.88th=t 61, 59.88th=t 331, 95.88th=t 421, i 99.88th=t 531, 99.58th=t 671, 99.98th=t 331, 95.88th=t 421, i 99.88th=t 1531, 90.88th=t 241, 90.98th=t 871, 99.95th=t 941, i 99.98th=t 1221 bw (KiBx): min= 7144, max=19608, per=190.88%, avg=12738.98, stdev=2395.77, samples=494 iops : min= 7186, max= 9988, avg=3182.78, stdev=538, 96, samples=494 lat (usec) : 258-8.81%, 508-8.81%, 758-8.81%, 1888-8.81% lat (msec) : 2 = 2.82%, 4=33.15%, 18=21.01%, 28=17.92%, 58=28.06% lat (msec) : 108=1.62%, 250=8.82%, 250=8.82%, cpu : use=1.35%, sys=5.43%, ctx=1872918, majf=8, minf=38 10 depths : 1=8.1%, 2=8.1%, 4=0.1%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 8=8.8%, 4=188.8%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 8=8.8%, 4=188.4%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.4%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.3%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.3%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.4%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.4%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.3%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%, 564=8.8% complete : 4=8.8%, 4=188.4%, 8=8.8%, 16=8.8%, 32=8.8%, 564=8.8%</pre>	urite: IOPS=3182, BW=12.4MiB/s (13.8MB/s)(3073MiB/247160msec)
<pre>clat (usec): min=1447, max=148666, aug=14166.95, stdev=13373.72 lat (usec): min=1457, max=148671, aug=14170.73, stdev=13373.72 clat percentiles (msec): i 1.08th=[41, 5.80th=[41, 18.00th=[41, 28.00th=[51, i 38.00th=[51, 40.00th=[61, 58.00th=[81, 60.00th=[141, i 78.00th=[51, 40.00th=[61, 58.00th=[331, 95.00th=[421, i 99.00th=[531, 99.50th=[671, 99.90th=[671, 99.95th=[941, i 99.00th=[531, 99.50th=[671, 99.90th=[671, 99.95th=[941, i 99.90th=[122] bw (KiBxs): min= 7144, max=19608, per=100.082x, aug=12738.98, stdev=2395.77, samples=494 iops : min= 7166, max= 4908, aug=3182.78, stdev=598.96, samples=494 lat (usec) : 2.261-8.01x, 500-8.01x, 1800-8.01x, 1800-8.01x lat (msec) : 2.08.28x, 4=39.15x, 10=21.01x, 28=17.92x, 58=20.06x lat (msec) : 100=1.62x, 250-8.02x, cpu : usr=1.35x, sys=6.43x, ctx=1072918, majf=8, minf=38 l0 depths : 1=0.1x, 2=0.1x, 4=0.1x, 10=0.1x, 32=0.1x, >=04-180.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.8x, 54=8.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.0x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.0x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.0x, 32=0.8x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.08x, 64=8.8x, >=64=8.8x, complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.08x, 4epth=128 Run status group 0 (a11 jobs): READ: bw=28.0M1Bxs (38.4MBxs), 28.0M1Bxs (38.4MBxs), io=7167M1B (7515MB), run=247160-247160msecd READ: bw=28.0M1Bxs (38.4MBxs), 28.0M1Bxs (38.4MBxs), io=7167M1B (7515MB), run=247160-247160msecd ReAD:</pre>	slat (nsec): min=1488, max=302738, avg=4613.59, stdev=3359.60
<pre>lat (usec): min=1457, max=148671, aug=14178.73, stdev=13373.74 clat percentiles (msec): i 1.88th=[41, 5.88th=[41, 18.88th=[41, 28.88th=[51, i 38.88th=[51, 40.88th=[61, 50.88th=[331, 95.88th=[141, i 78.88th=[51, 40.88th=[671, 99.98th=[331, 95.88th=[421, i 99.88th=[531, 99.59th=[671, 99.99th=[871, 99.95th=[941, i 99.99th=[122] bu (K1B×3: min= 7144, max=19608, per=100.082, aug=12738.90, stdev=2395.77, samples=494 iops : min= 7056, max= 4988, aug=3182.78, stdev=538, 96, samples=494 iops : min= 1766, max= 4988, aug=3182.78, stdev=538, 96, samples=494 iops : sin= 1766, max= 4988, aug=3182.78, stdev=538, 96, samples=494 id (usec) : 258-8.81x, 508-8.81x, 758-8.81x, 1882-8.81x lat (msec) : 2.8-8.81x, 508-8.81x, 758-8.81x, 1882-8.81x lat (usec) : 108-1.62x, 258-8.82x, 508-8.81x, 1882-1.81x, 28=17.92x, 58=28.86x, lat (usec) : 108-1.62x, 258-8.82x, ctx=1072910, majf=0, minf=38 l0 depths : 1=0.1x, 2=0.1x, 4=0.1x, 18-8.1x, 16-8.1x, 32=0.1x, >=64-180.8x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16-8.8x, 54=8.8x, >=64-8.8x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16=8.8x, 54=8.8x, 54=8.8x, >=64-8.8x, complete : 8=8.8x, 4=108.8x, 8=8.8x, 16=8.8x, 64=8.8x, >=64-8.1x, issued ruts: total=1834036.786604.8, 8 short=0.8, 8, 64=8.8x, >=64-8.1x, issued ruts: total=1834036.786604.8, 8 short=0.8, 8, dep=4.8, 8.8 latency : target=0, window=0, percentile=160.68x, dept=4.8, 8.8 latency : target=0, window=0, percentile=160.8x, dept=3.8, 448x, sole 3.8, 448</pre>	clat (usec): min=1447, max=148666, avg=14166.05, stdev=13373.72
<pre>clat percentiles (msec):</pre>	lat (usec): min=1457, max=148671, avg=14178.73, stdev=13373.74
<pre>i 1.08th=[41, 5.08th=[41, 10.08th=[41, 20.08th=[5],</pre>	clat percentiles (msec):
<pre>i 38.08th=[5], 40.08th=[6], 59.08th=[8], 60.08th=[14],</pre>	i 1.00th=[4], 5.00th=[4], 10.00th=[4], 20.00th=[5],
<pre>1 78.08th=[18],80.08th=[24],99.08th=[33],95.08th=[42], 1 99.08th=[55],99.58th=[67],99.98th=[67],99.99th=[94], 1 99.99th=[122] bw (KiBxs): min= 7144, max=19608, avg=3182.78, avg=12738.98, stdev=2395.77, samples=494 iops : min= 7186, max=4908, avg=3182.78, stdev=589.66, samples=494 lat (usec) : 250=0.81x, 580=0.81x, 1000=0.81x, 200=0.81x lat (usec) : 260=0.81x, 580=0.81x, 1000=0.81x, 200=0.81x lat (usec) : 100=1.62x, 250=0.82x cpu : usr=1.35x, sys=6.43x, ctx=1072910, majf=0, minf=38 l0 depths : 1=0.1x, 2=0.1x, 4=0.1x, 10=0.1x, 32=0.8x, 64=0.8x, >=64=0.8x complete : 0=0.8x, 4=100.8x, 0=0.8x, 32=0.8x, 64=0.8x, >=64=0.8x complete : 0=0.8x, 4=108.8x, 0=0.8x, 16=0.8x, 32=0.8x, 64=0.8x, >=64=0.1x issued rwts: total=1834036,706604,0.0 short=0,0,0.8 dopped=3.0,0.0 latency : target=0, unidow=0, percentile=100.00x, depth=128 Run status group 0 (all jobs): READ: bw=28.0MIB/s (38.4MB/s), 28.0MIB/s-20.0MIB/s (38.4MB/s-38.4MB/s), io=7167MIB (7515MB), run=247160-24716000000 </pre>	30.00th=[5], 40.00th=[6], 50.00th=[8], 60.00th=[14],
<pre>i 99.88th=[59], 99.58th=[67], 99.98th=[87], 99.95th=[94], i 99.99th=[122] bw (XiB/s): min= 7144, max=19608, per=180.08%, avg=12738.98, stdev=2395.77, samples=494 iops : min= 1786, max= 49808, avg=3182.78, stdev=598.96, samples=494 lat (usec) : 258-88.01, 580-8.01%, 759-8.01%, 1080-8.01% lat (usec) : 2.8-8.01%, 759-8.01%, 1080-8.01% lat (usec) : 108-1.62%, 258-8.02% cpu : usr=1.35%, sys=5.43%, ctx=1072910, majf=8, minf=38 10 depths : 1=0.1%, 2=0.1%, 4=0.1%, 0=0.1%, 02=0.1%, 0=0.48%,</pre>	1 78.68th=[18], 80.68th=[24], 90.68th=[33], 95.68th=[42],
<pre>i 99.99th=[122] bw (KiB/s): min= 7144, max=19600, per=100.000%, avg=12730.90, stdev=2395.77, samples=494 iops : min= 1786, max= 4900, avg=3182.70, stdev=590.96, samples=494 lat (usec) : 250=0.01%, 500=0.01%, 100=0.01% lat (usec) : 260=0.01%, 500=0.01%, 100=0.01% lat (usec) : 100=1.62%, 250=0.02% cpu : use=1.35%, sys=6.43%, ctx=1072910, majf=0, minf=30 10 depths : 1=0.1%, 2=0.1%, 4=0.1%, 16=0.1%, 32=0.0%, 54=0.0% submit : 0=0.0%, 4=100.0%, 0=0.0%, 16=0.0%, 32=0.0%, 54=0.0% complete : 0=0.0%, 4=100.0%, 0=0.0%, 16=0.0%, 32=0.0%, 54=0.0%, 54=0.1% issued rwts: total=1834036,706604,0,0 short=0.0,0,0 depth=120 Rum status group 0 (all jobs): READ: bw=28.0MiB/s (30.4MB/s), 20.0MiB/s=20.0MiB/s (30.4MB/s=30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec</pre>	i 99.80th=[59], 99.50th=[67], 99.90th=[87], 99.95th=[94],
 bu (KiB/s): min= 7144, max=19608, per=100.0082, avg=12730.90, stdev=2395.77, samples=494 iops : min= 1786, max=4908, avg=3182.78, stdev=598.96, samples=494 lat (usec) : 258=0.01x, 580=0.01x, 1800=0.01x lat (usec) : 2=0.20x, 4=39.15x, 10=21.01x, 20=17.92x, 58=20.06x lat (usec) : 100=1.62x, 2550=0.02x cpu : usr=1.35x, sys=6.43x, ctx=1072910, majf=0, minf=30 l0 depths : 1=0.1x, 2=0.1x, 4=0.1x, 10=0.1x, 32=0.1x, 3=64.180.8x complete : 0=0.8x, 4=100.8x, 50=0.8x, 50=0.8x, 54=0.8x, 54=0.1x complete : 0=0.8x, 4=100.8x, 8=0.8x, 16=0.8x, 32=0.8x, 54=0.1x issued rwts: total=1834036,786604,00 short=0,0,0,0 dropped=8,0,0.0 latency : target=0, window=0, percentile=100.00x, depth=128 Run status group 0 (all jobs): READ: bw=28.0MIB/s (38.4MB/s), 28.0MIB/s-20.0MIB/s (38.4MB/s), io=7167MIB (7515MB), run=247160-247160msec 	i 99.99th=[122]
<pre>iops : min= 1786, max= 4988, avg=3182.78, stdev=538,96, samples=494 lat (usec) : 258-8.81x, S08-8.81x, S08-8.81x,</pre>	bw (KiB/s); min= 7144, max=19608, per=100.08%, avg=12738.98, stdev=2395.77, samples=494
<pre>lat (usec) : 258-8.81x, 580-8.81x, 758-8.81x, 1808-8.81x lat (usec) : 280-8.81x, 580-8.81x, 780-8.81x, 188-21.01x, 28=17.92x, 58=20.06x lat (usec) : 108-1.62x, 250-8.62x cpu : use-1.35x, sys=6.43x, ctx=1872210, majf=0, minf=30 l0 depths : 1=8.1x, 2=8.1x, 4=8.1x, 16=8.1x, 32=8.1x, >=64=188.8x complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.8x, 32=8.8x, 64=8.8x, >=64=8.8x complete : 0=8.8x, 4=108.8x, 8=0.8x, 16=8.8x, 32=8.8x, 64=8.8x, 5=64=8.1x issued rwts: total=1834836,786604,8,0 short=0.8,0,0 dropped=0.68,0 latency : target=0, window=0, percentile=180.08x, depth=128 Run status group 0 (all jobs): READ: bw=28.8MiBrs (38.4MBrs), 28.0MiBrs-28.0MiBrs (38.4MBrs-38.4MBrs), io=7167MiB (7515MB), run=247160-247160msec </pre>	iops : min= 1786, max= 4900, avg=3182.70, stdev=598.96, samples=494
<pre>lat (msec) : 2=0.28x, 4=39.15x, 10=21.01x, 20=17.92x, 50=20.06x lat (msec) : 100=1.62x, 250=0.02x cpu : usr=1.35x, sys=6.43x, ctx=1072910, majf=0, minf=30 10 depths : 1=0.1x, 2=0.1x, 4=0.1x, 10=0.1x, 32=0.1x, >=64=100.8x submit : 0=0.8x, 4=100.8x, 16=0.8x, 32=0.8x, 64=0.8x, >=64=0.8x complete : 0=0.8x, 4=100.8x, 0=0.8x, 16=0.8x, 32=0.8x, 64=0.8x, >=64=0.1x issued rwts: total=1034036.786004.00 short=0.0,00 depth=128 Run status group 0 (all jobs): READ: bw=28.0M18xs (30.4M8xs), 20.0M18xs-20.0M18xs (30.4M8xs-30.4M8xs), io=7167M18 (7515M8), run=247160-247160msecd </pre>	lat (usec) : 250=0.01%, 500=0.01%, 750=0.01%, 1000=0.01%
 lat (msc) : 100=1.62x, 250=0.02x cpu : usr=1.35x, sys=6.43x, ctx=1072910, majf=0, minf=30 l0 depths : 1=0.1x, 2=0.1x, 4=0.1x, 16=0.1x, 32=0.1x, >=64=100.0x submit : 8=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=100.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 8=0.0x, 16=0.0x, 32=0.0x, 64=0.0x, >=64=0.0x complete : 9=8.0x, 4=100.0x, 7866004, 90, 90, hont=9.0x, 90, 90, 00, 00, 00, 00, 00, 00, 00, 00	lat (msec) : 2=8.28%, 4=39.15%, 18=21.81%, 28=17.92%, 58=28.86%
<pre>cpu : usr=1.35%, sus=5.43%, ctx=1872918, majf=8, minf=30 ID depths : 1=8.1%, 2=8.1%, 4=8.1%, 4=8.1%, 16=8.1%, 32=8.1%, >=64=180.8% submit : 0=8.0%, 4=100.0%, 0=8.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=8.0% complete : 0=8.0%, 4=100.0%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=8.0% issued rwts: total=1834036,706604,0.0 short=0.0,0.0 dopped=0.0,0.0 latency : target=0, window=0, percentile=100.000%, depth=120 Run status group 0 (all jobs): READ: bw=28.0MIB/s (30.4MB/s), 28.0MIB/s-20.0MIB/s (30.4MB/s-30.4MB/s), io=7167MIB (7515MB), run=247160-247160msec</pre>	lat (msec) : 198=1,62×, 258=8.82×
<pre>ID depths : 1=0.1x, 2=0.1x, 4=0.1x, 10=0.1x, 12=0.1x, 32=0.1x, 3=0-1x, 3</pre>	cpu : usr=1.35%, sys=6.43%, ctx=1072910, majf=0, minf=30
submit : 8=8.8%, 4=108.8%, 8=8.8%, 16=0.8%, 32=8.8%, 54=8.8%, 5=64=8.8% complete : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.8%, 32=8.8%, 64=8.8%, >=64=8.1% issued rwts: total=1834836,786604,8,8 short=8,8,8,8 dropped=8,8,8,8 latency : target=8, window=8, percentile=108.08%, depth=128 Run status group 0 (all jobs): READ: bw=28.8MiB/s (38.4MB/s), 28.0MiB/s-28.0MiB/s (38.4MB/s-38.4MB/s), io=7167MiB (7515MB), run=247160-247168msec	IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
complete : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.8%, 32=0.8%, 54=8.8%, >=54=8.1% issued rwts: total=1834836,786604,8.0 short=0.0,0.0 dropped=8,0.0.0 latency : target=0, window=0, percentile=100.00%, depth=128 Run status group 0 (all jobs): READ: bw=28.0MIB/s (38.4MB/s), 28.0MIB/s-28.0MIB/s (30.4MB/s-30.4MB/s), io=7167MIB (7515MB), run=247160-247160msec	submit : 8=8.8%, 4=108.0%, 8=8.8%, 16=8.8%, 32=8.8%, 64=8.8%, >=64=8.8%
issued rwts: total=1834836,786604,8,0 short=8,8,8,8 dropped=0,0,0,0 latency : target=8, window=8, percentile=100.00%, depth=128 Run status group 0 (all jobs): READ: bw=28.0MiB/s (30.4MB/s), 28.0MiB/s-28.0MiB/s (30.4MB/s-30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec	complete : 8=8.8%, 4=100.8%, 8=0.8%, 16=8.8%, 32=0.8%, 64=8.8%, >=64=0.1%
latency : target=0, window=0, percentile=100.00%, depth=120 Run status group 0 (all jobs): READ: bw=28.0MiB/s (30.4MB/s), 28.0MiB/s-28.0MiB/s (30.4MB/s-30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec	issued rwts: total=1834836,786684,8,8 short=8,8,8,8 dropped=8,8,8,8
Run status group θ (all jobs): READ: bw=28.0MIB/s (38.4MB/s), 28.0MIB/s-28.0MIB/s (38.4MB/s-38.4MB/s), io=7167MiB (7515MB), run=247160-247160msec	latency : target=0, window=0, percentile=100.00×, depth=128
Run status group 0 (all jobs): READ: bw=28.0MiB/s (30.4MB/s), 28.0MiB/s-28.0MiB/s (30.4MB/s-30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec	
READ: bw=28.0MiB/s (30.4MB/s), 28.0MiB/s-28.0MiB/s (30.4MB/s-30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec	Run status group Ø (all jobs):
	READ: bw=28.0M1B/s (30.4MB/s), 28.0M1B/s-28.0M1B/s (30.4MB/s), io=7167M1B (7515MB), run=247160-247160msec

• fio command:

```
fio --randrepeat=1 --ioengine=libaio --name=test -output=output.log --
direct=1 --filename=/mnt/nfs/test_fio --bs=1M --iodepth=128 --
size=10240M --readwrite=rw --rwmixwrite=70 --fallocate=none
```

NOTE

Variable */mnt/nfs/test_fio* is the location of the file to be tested. The location must be specific to the file name, which is the **test_fio** file in the */mnt/nfs* directory in this example. Set it based on the site requirements.

• fio result:

test; (grounid=0, jobs=1); err= 0; nid=20358; Mon Jun 8 11:57:14 2020
read: 10PS=5865, BH=19.8MiB/s (28.7MB/s)(3873MiB/155288msec)
slat (nsec); min=1271, max=269588, aug=4873.51, stdeu=3848.12
clat (usec): min=226, max=88185, avg=5711.35, stdev=7829.46
lat (usec): min=232, max=88187, avg=5715,49, stdev=7879,48
clat necentiles (usec):
1 1.88th=[1221], 5.88th=[1958], 18.88th=[2188], 28.88th=[2442],
38_88th=f_26861, 48_88th=f_28821, 58_88th=f_29931, 68_88th=f_32281.
28.88th=[3687], 88.98th=[5684], 98.88th=[14222], 95.88th=[21898],
99.88th=[35914], 99.58th=[48633], 99.98th=[51643], 99.95th=[55837].
1 99.99th=[66847]
bw (KiB/s): min=13360, max=28848, ner=99.99%, avg=20257.97, stdev=2913.05, samples=310
1005 : min= 3340, max= 7212, avg=5064,48, stdev=728.27, samples=310
write: IOPS=11.8k, BW=46.2MiB/s (48.4MB/s) 7167MiB/155288msec)
slat (nsec): min=1396, max=398684, avg=4485.68, stdev=3091.75
clat (usec): min=857, max=148259, avg=8377.47, stdev=8488.15
lat (usec): min=867, max=148264, avg=8382.82, stdev=8488.16
clat percentiles (mscc):
1,68th=[3], 5,68th=[4], 10,68th=[4], 20,68th=[4],
i 30.00th=[5], 40.00th=[5], 50.00th=[5], 60.00th=[6],
1 78.88th=[7], 88.88th=[13], 98.88th=[21], 95.88th=[28],
1 99.88th=[42], 99.58th=[47], 99.98th=[68], 99.95th=[68],
i 99.99th=[128]
bw (KiB/s): min=32224, max=67456, per=99.98%, avg=47254.23, stdev=6792.41, samples=318
iops : min= 8956, max=16864, avg=11813.55, stdev=1698.11, samples=310
lat (usec) : 258=8.01%, 508=8.04%, 758=8.07%, 1008=8.09%
lat (msec) : 2=1.53×, 4=36.85×, 18=41.27×, 28=11.38×, 58=8.61×
lat (msec) : 188=8.23%, 258=8.01%
cpu : usr=2.13z, sys=9.90z, ctx=925770, majf=0, minf=31
IO depths : 1=8.1%, 2=8.1%, 4=8.1%, 8=8.1%, 16=8.1%, 32=8.1%, >=64=180.8%
submit : 8=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 8=8.0%, 4=100.0%, 8=8.0%, 16=8.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=786597,1834843,0,0 short=0,0,0,0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00×, depth=120
Run status group 0 (all jobs):
READ: bw=19.8MiB/s (20.7MB/s), 19.8MiB/s-19.8MiB/s (20.7MB/s-20.7MB/s), io=3073MiB (3222MB), run=155288-155288ms
WRITE: bw=46.2MiB/s (48.4MB/s), 46.2MiB/s-46.2MiB/s (48.4MB/s-48.4MB/s), io=7167MiB (7516MB), run=155288-155288ms

Sequential read IOPS

• fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/--rw=read
--bs=4k --size=1G --iodepth=128 --runtime=120 --numjobs=10
```

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

• fio result:

read: IOPS=9654, BW=37.7MiB/s (39.5MB/s)(10.0GiB/271519msec)
slat (nsec): min=1233, max=662160, avg=4118.17, stdev=4773.23
clat (usec): min=365, max=131116, avg=13253.10, stdev=13958.09
lat (usec): min=371, max=131118, avg=13257.29, stdev=13958.89
clat percentiles (usec):
1.00th=[1762], 5.00th=[1991], 10.00th=[2147], 20.00th=[2376],
38.00th=[2704], 40.00th=[3621], 50.00th=[7767], 60.00th=[11994].
; 70,00th=[16909], 80,00th=[23462], 90,00th=[33162], 95,00th=[41681],
1 99.88th=[59582], 99.58th=[66842], 99.98th=[83362], 99.95th=[98782],
1 99.99th=[183285]
bw (KiB/s): min=18656, max=61576, per=99.99%, avg=38615.41, stdev=7783.32, samples=543 iops : min= 4664, max=15394, avg=9653.82, stdev=1925.83, samples=543 lat (usec) : 588=8.81%, 758=8.81%, 1888=8.82%
lat (msec) : 2=5.252, 4=36.352, 18=12.762, 28=28.562, 58=22.622
lat (msec) : 198=2,42%, 258=8,82%
cpu ; usr=1.84%, sus=5.35%, ctx=913138, maif=8, minf=159
ID depths : $1=0,12, 2=0,12, 4=0,12, 8=0,12, 16=0,12, 32=0,12, >=54=180,92$
submit : A=A, Az, 4=1AA, Az, 8=A, Az, 16=A, Az, 32=B, Az, 64=A, Az, >=64=A, Az
commulete : 8=8,82, 4=198,82, 8=8,82, 15=9,82, 32=8,82, 54=8,82, >=54=8,12
issued ruts: total=2621448.9.9.9 short=8.9.9.8 dromed=8.8.9.8
latencu + tawat-a uindau-a necentile=160 92 dente=128
Attendy , target b, window b, percentite 100.00%, appli-120
Run status group 0 (all jobs):

Random read IOPS

• fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/--
```

rw=randread --bs=4k --size=1G --iodepth=128 --runtime=120 -numjobs=10

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

• fio result:

test: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
Starting 1 process
John 1 (f=1), [r] [100 0% donal [1782/KB/0KB/0KB /s] [4/156/0/0 jone] [ata 00m/00s]
505.1(1-1), [1] [100.00 doine] [1002400/000/00 /3] [4400/00 1003] [100000] test (requiride interval
rad - in-102/00MB hu-18597KB/s insc-16/9 runt-563837msr
c_{1} (user) which may 375 ave - 56 stdev - 252
c_{1} (use c). min-1; max-3/3, avg 2.04, state 2.02
$t_{1} = t_{1} = t_{1$
clat (usec). mini-710, max-75505, avg=27550.05, stdev=100255.55
1 = 0 (mode) $3 = 5 = 0$ (mode) $5 = 1 = 0 = 0$ (mode) $6 = 20 = 0$ (mode) $6 = 1$
[50.00 th - [1]], 40.00 th - [15] , 90.00 th - [21] 95.00 th - [23]
[90,00h-[573]] 00,500h-[633] 00,00h-[751] 00,00h-[753]] 00,00h-[731]] 00,00h-[676] 00,500h-[676] 00,00h-[676] 00,0
93.00th-[750], 53.00th-[755], 53.00th-[725], 53.00th-[754],
55.5561-[/50] bw (KR /s) min= 1896 may=35752 par=100 00% avg=18605 56 stdev=1980 86
lat (user) · 750-0.01% 1000-0.01%
lat (user) - 2-0 32% 4=3 28% 10-63 65% 20=22 42% 50=7 50%
lat (msec) : 100-0 07% 250-0 01% 500-0 03% 750-2 77% 1000-0 01%
cou (mod) : usr=0.82% sv=2.41% ctx=1231561 maif=0 minf=155
ID denths 1=0 1% 2=0 1% 4=0 1% 8=0 1% 16=0 1% 32=0 1% >=64=100 0%
submit 0=0.0% 4=100.0% 8=0.0% 13=0.0% 32=0.0% 64=0.0% ==64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0% >=64=0.1%
is such to tal = $r=2621440$ /w=0 /d=0. short= $r=0$ /w=0 /d=0
latency : target=0 window=0 percentile=100.00% denth=128
Run status group 0 (all jobs):
READ: io=10240MB, aggrb=18597KB/s, minb=18597KB/s, maxb=18597KB/s, mint=563832msec, maxt=563832msec

Sequential write IOPS

• fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/--
rw=write --bs=4k --size=1G --iodepth=128 --runtime=120 --numjobs=10
```

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

• fio result:

test: (groupid=8, jobs=1): err= 8: pid=20874: Mon Jun 8 14:23:89 2020	
slat (neer); min=1483, may=36026, alw=361010/20115000567	
c_{1at} (user): min-1053 may=106548 are 11588 61 stder=5000.01	
Lat (usec) = min-1553, max-160510, avg-11502.01, state-5070.01	
at vaces tile (use):	
$\begin{bmatrix} 1 & 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0$	
1 1.001A-1 70131, 3.001A-1 33321, 10.001A-1 00321, 20.001A-1 (7331,	
1 38.801A=L 80273, 48.801A=L 60403, 58.801A=L 50343, 58.801A=L180143,	
7 /0.00th=1125181, 60.00th=1155331, 50.00th=1192681, 55.00th=1226761,	
i 99.00th=[32637], 99.50th=[37487], 99.90th=[49021], 99.95th=[53740],	
{ 99.99th=[69731]	
bw (KiB/s): min=31712, max=52431, per=99.99%, avg=44158.04, stdev=3987.31, samples=474	
iops : min= 7928, max=13107, avg=11039.50, stdev=996.83, samples=474	
lat (msec) : 2=0.01%, 4=1.00%, 10=51.94%, 20=38.58%, 50=8.39%	
lat (msec) : 100=0.08%, 250=0.01%	
cpu : usr=1.33%, sys=5.47%, ctx=392117, majf=8, minf=27	
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%	
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%	
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%	
issued rwts: total=0,2621440,0,0 short=0,0,0,0 dropped=0,0,0,0	
latency : target=0, window=0, percentile=100.00%, depth=128	
Run status group 8 (all jobs):	
WRITE: bu=43.1MiB/s (45.2MB/s), 43.1MiB/s-43.1MiB/s (45.2MB/s-45.2MB/s), io=10.0GiB (10.7GB),	run

Random write IOPS

• fio command:

fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 -group_reporting=1 --name=iops_fio --directory=/*mnt/sfs-turbo*/-rw=randwrite --bs=4k --size=1G --iodepth=128 --runtime=120 -numjobs=10

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

fio result:

test: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
starting 1 process
test, (ground-0, jobs-1), org-0, sid-16633, Thu Jan 12 10,12,33 2033
$(s_1) = (s_1) = (s_1) = (s_1) = (s_1) = (s_2) = (s_1) = (s_2) = (s_1) = (s_1) = (s_2) = (s_1) = (s_1) = (s_2) = (s_1) = (s_1) = (s_2) = (s_1) = (s_1) = (s_2) = (s_1) = (s_1) = (s_1) = (s_2) = (s_1) = (s_1) = (s_2$
while: $10=10240$ mb, $00=10400$ MD/S, 100 S=4010, 1010 S0/347 MSec
Stat (usec): $min=1$, $max=336$, $avg=3.21$, $state=2.04$
ctat (usec): $\min=0.00$, $\max=0.1500$, $avg=27/27.54$, store= 101207.14
tat (usec): min=055, max=015504, avg=2//50.05, stdev=10120/.14
39.00th=[044], 99.30th=[060], 99.90th=[/01], 99.95th=[/09],
55.55 Lin[(54)]
DW (ND /S): mini= 1004, max=30303, per=100.00%, avg=10403.11, Stdev=3/03.04
Lat (USEC) : 1000=0.01%
Lat $(msec)$: 2=0.20%, 4=1.05%, 10=00.35%, 20=24.30%, 30=3.05%
Lat (MSec) : 100=0.09%, 250=0.01%, 500=0.00%, 750=2.00%, 1000=0.01%
Cpu : usr=0.30%, sys=2.30%, ctx=1352/44, majr=0, minr=2/
IU deptns : $1=0.1^{\circ}$, $2=0.1^{\circ}$, $4=0.1^{\circ}$, $3=0.1^{\circ}$, $15=0.1^{\circ}$, $15=0.1^{\circ}$, $32=0.1^{\circ}$, $5=04=100.0^{\circ}$
SUBMIT : $U = 0.0^{\circ}, 4 = 100.0^{\circ}, 5 = 0.0^{\circ}, 15 = 0.0^{\circ}, 32 = 0.0^{\circ}, 54 = 0.0^{\circ}, 5 = 0.4 = 0.0^{\circ}$
complete : 0=0.0%, 4=100.0%, 0=0.0%, 10=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
155ued : total=r=0/w=2621440/d=0; short=r=0/w=0/d=0
latency : target=0, window=0, percentile=100.00%, deptn=128
Pup status group 0 (all isks).
NUT Status group + (att jubs):
$\frac{1}{10} = 10240$ MB, aggrb=10402KB/S, minb=10402KB/S, maxb=10402KB/S, mint=50/94/msec, maxt=50/94/msec

Sequential read bandwidth

• fio command:

fio --randrepeat=1 --ioengine=libaio --name=test -output=output.log -direct=1 --filename=/*mnt/sfs-turbo/test_fio* --bs=1M --iodepth=128 -size=10240M --readwrite=read --fallocate=none

NOTE

Variable */mnt/sfs-turbo/test_fio* is the location of the file to be tested. The location must be specific to the file name, which is the **test_fio** file in the */mnt/sfs-turbo* directory in this example. Set it based on the site requirements.

fio result:

test: (groupid=8, j <u>obs=1): err=</u> 8: pid=28962: Mon Jun 8 14:37:48 2828
read: IOPS=398, BW=391MiB/s (489MB/s)(10.06iB/26221msec)
slat (usec): min=78, max=595, avg=99.58, stdev=39.09
clat (msec): min=35, max=544, avg=327.38, stdev=99.64
lat (msec): min=36, max=545, avg=327.48, stdev=99.63
clat percentiles (msec):
1.00th=[155], 5.00th=[161], 10.00th=[167], 20.00th=[180],
30.00th=[368], 40.00th=[372], 50.00th=[380], 60.00th=[384],
70.00th=[388], 80.00th=[393], 90.00th=[401], 95.00th=[414],
99.00th=[472], 99.50th=[506], 99.90th=[535], 99.95th=[542],
1 99.99th=[542]
bw (KiB/s): min=301056, max=768000, per=99.52%, avg=397987.65, stdev=81583.56, samples=52
iops : min= 294, max= 750, avg=388.65, stdev=79.67, samples=52
lat (msec) : 50=0.17%, 180=0.28%, 250=27.61%, 500=71.37%, 750=0.58%
cpu : usr=0.08z, sys=4.21z, ctx=10395, ma if=0, minf=97
10 depths : $1=0.12$, $2=0.12$, $4=0.12$, $8=0.12$, $16=0.22$, $32=0.32$, $>=64=99.42$
submit : 8=8.8%, 4=188.8%, 8=8.8%, 16=8.8%, 32=8.8%, 64=8.8%, >=64=8.8%
complete : 8=8.8%, 4=188.8%, 8=8.8%, 16=8.8%, 32=8.8%, 64=8.8%, >=64=8.1%
issued rwts: total=10240,0,0,0 short=0,0,0,0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00%, depth=128
Run status group 0 (all jobs): READ: bw=391MiB/s (409MB/s), 391MiB/s-391MiB/s (409MB/s-409MB/s), io=10.06iB (10.76B), run=26221-26221mse

Random read bandwidth

fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/--
rw=randread --bs=1M --size=10G --iodepth=128 --runtime=120 --
numjobs=1
```

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

fio result:

Sequential write bandwidth

fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --
rw=write --bs=1M --size=10G --iodepth=128 --runtime=120 --numjobs=1
```

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

• fio result:

test: (groupid=8, jobs=1): err= 8: pid=21889: Mon Jun 8 14:53:44 2828
write: IOPS=243, BW=244MiB/s (255MB/s)(10.06iB/42048msec)
slat (usec): min=183, max=584, avg=198.38, stdev=29.47
clat (msec): min=18, max=1184, avg=525.23, stdev=253.35
lat (msec): min=18, max=1104, avg=525.42, stdev=253.35
clat percentiles (msec):
<pre>i 1.00th=[51], 5.00th=[108], 10.00th=[167], 20.00th=[292],</pre>
30.89th=[422], 40.88th=[468], 50.89th=[586], 60.88th=[550],
; 78.00th=[625], 80.00th=[768], 90.80th=[902], 95.00th=[978],
99.00th=[1036], 99.50th=[1045], 99.90th=[1070], 99.95th=[1099],
99.99th=[1099]
bw (KiB/s): min= 4096, max=468992, per=100.00%, avg=249508.99, stdev=147656.62, samples=83
iops : min= 4, max= 458, avg=243.63, stdev=144.22, samples=83
lat (msec) : 20=0.03×, 50=0.96×, 100=3.36×, 250=12.55×, 500=31.63×
lat (msec) : 750=38.07%, 1000=18.96%
cpu : usr=2.28×, sys=2.58×, ctx=3972, majf=8, minf=27
10 depths : 1=8.1%, 2=8.1%, 4=8.1%, 8=8.1%, 16=8.2%, 32=8.3%, >=64=99.4%
submit : 8=8.8%, 4=108.8%, 8=8.8%, 16=8.8%, 32=8.8%, 64=8.8%, >=64=8.8%
complete : 8=8.8%, 4=188.8%, 8=8.8%, 16=8.8%, 32=8.8%, 64=8.8%, >=64=8.1%
issued rwts: total=0,10240,0,0 short=0,0,0,0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00%, depth=120
Run status group 0 (all jobs):
LRITE: hu=244MiB/e (255MB/e) 244MiB/e 244MiB/e (255MB/e) in=19 BCiR (19 7CB) num=42948-42948mee

Random write bandwidth

• fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/--
rw=randwrite --bs=1M --size=10G --iodepth=128 --runtime=120 --
numjobs=1
```

NOTE

Variable */mnt/sfs-turbo/* is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

• fio result:

test: (g=0): rw=randwrite, bs=1M-1M/1M-1M/1M-1M, ioengine=libaio, iodepth=128 fio-2.1.10
Starting 1 process
test: (groupid=0, jobs=1): err= 0: pid=16370: Tue Dec 28 09:22:59 2021
write: io=10240MB, bw=156001KB/s, iops=152, runt= 67216msec
slat (usec): min=93, max=349, avg=156.14, stdev=22.29
clat (msec): min=17, max=1964, avg=839.92, stdev=345.94
lat (msec): min=17, max=1964, avg=840.08, stdev=345.94
clat percentiles (msec):
1.00th=[30], 5.00th=[37], 10.00th=[42], 20.00th=[971],
30.00th=[979], 40.00th=[988], 50.00th=[988], 60.00th=[996],
70.00th=[996], 80.00th=[1004], 90.00th=[1004], 95.00th=[1012],
99.00th=[1020], 99.50th=[1029], 99.90th=[1037], 99.95th=[1045],
99.99th=[1958]
bw (KB /s): min=150104, max=180654, per=98.76%, avg=154058.04, stdev=3404.48
lat (msec) : 20=0.04%, 50=13.44%, 100=1.04%, 250=0.73%, 500=1.05%
lat (msec) : 750=0.04%, 1000=60.69%, 2000=22.97%
cpu : usr=0.91%, sys=1.52%, ctx=2011, majf=0, minf=28
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued : total=r=0/w=10240/d=0, short=r=0/w=0/d=0
latency : target=0, window=0, percentile=100.00%, depth=128
Run status group 0 (all jobs):
WRITE: io=10240MB, aggrb=156000KB/s, minb=156000KB/s, maxb=156000KB/s, mint=67216msec, maxt=67216msec

3 Testing SFS Turbo Latency

This section uses the SFS Turbo Performance-Enhanced type to test the file system latency. The specifications of the ECSs to be used are as follows:

Specifications: General computing-plus | c6.4xlarge.4 | 16 vCPUs | 64 GB

Image: EulerOS 2.5

Single-queue random read

fio command:

fio -direct=1 -iodepth=1 -rw=randread -ioengine=libaio -bs=4k -size=10G numjobs=1 -runtime=60 -group_reporting -filename=/*mnt/sfsturbo/ fio_test_01 -name*=randread_test

NOTE

Variable */mnt/sfsturbo/fio_test_01 -name* indicates the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_01 -name** file in the **/mnt/sfsturbo** directory in this example. Set the location based on site requirements.

• fio result:

[root@100 ~]# fio -direct=1 -iodepth=1 -nw=randread -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/fio_test_01 -name=randread t
randread_test: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.5
Starting 1 process
randread test: Laying out IO file (1 file / 10240MiB)
Jobs: 1 (f=1): [r(1)][100.0%][r=5600KiB/s,w=0KiB/s][r=1400,w=0 IOPS][eta 00m:00s]
randread test: (groupid=0, jobs=1): err= 0: pid=29015: Mon Nov 7 11:44:12 2022
read: IOPS=1401, BW=5606KiB/s (5740kB/s)(328MiB/60001msec)
slat (nsec): min=2370, max=51192, avg=4228.52, stdev=1319.26
clat (usec): min=559, max=8403, avg=707.69, stdev=197.66
lat (usec): min=562, max=8407, avg=712.08, stdev=197.71
clat percentiles (usec):
1.00th=[603], 5.00th=[627], 10.00th=[635], 20.00th=[652],
] 30.00th=[660], 40.00th=[668], 50.00th=[676], 60.00th=[693],
70.00th=[701], 80.00th=[725], 90.00th=[758], 95.00th=[807],
99.00th=[1369], 99.50th=[2040], 99.90th=[3523], 99.95th=[3982],
99.99th=[5735]
bw (KiB/s): min= 5120, max= 5856, per=100.00%, avg=5605.12, stdev=134.52, samples=119
iops : min= 1280, max= 1464, avg=1401.28, stdev=33.63, samples=119
lat (usec) : 750=88.78%, 1000=9.31%
lat (msec) : 2=1.39%, 4=0.46%, 10=0.05%
cpu : usr=0.86%, sys=1.77%, ctx=84092, majf=0, minf=33
IO depths : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
issued rwts: total=84087,0,0,0 short=0,0,0,0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00%, depth=1
Run status group 0 (all jobs):
READ: bw=5666K1B/s (5740kB/s), 5666K1B/s 55666K1B/s (5740kB/s-5740kB/s), io=328M1B (344MB), run=60001-60001msec
[root@100 ~]#

Single-queue random write

• fio command:

fio -direct=1 -iodepth=1 -rw=randwrite -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=/*mnt/sfsturbo/ fio_test_02 -name*=randwrite_test

NOTE

Variable */mnt/sfsturbo/fio_test_02 -name* indicates the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_02 -name** file in the **/mnt/sfsturbo** directory in this example. Set the location based on site requirements.

fio result:

[root@100 ~j# fio -direct=1 -lodepth=1 -rw=randwrite -loengine=llbaio -bs=4k -size=106 -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/ststurbo/fio_test_02 -name=randwrite_te ≈t
fantamile test: (gmo): fw=fantawile, Ds=(K) 40900-40900, (W) 40900-40900, (T) 40900-40900, idengine=(iDaio, iddeptn=1 fin_2 5
Starting 1 process
John: 1 (f=1): [w(1)][]AA AN][r=AK;B/s w=67A8K;B/s][r=A w=1677 IODS][ata AAm:AAs]
sadwrife test: (arcupid=8) (bs=1): err= 8: pid=25/83: Tue Nov. 8: 21:37:13:2022
write: $TOPS=1661$ BW= 6645 iB/c (6804 bB/c)(380 MiB/ 6000 Beer)
slat (soc) = min=2566 msr=34191 avn=4562 stdow=1661 31
clat ($ sec\rangle$), $ =473$ $ =473$ $ =474$ $ $
lat (usec): min-477 max=2234 avr=660 38 stdp=160.55
clat hereintille (iser):
1 APTH=[515] 5 APTH=[529] 10 APTH=[537] 20 APTH=[553]
1 0. 00th [552] 40.00th [570] 50.00th [578] 60.00th [556]
70.00th=[594], 80.00th=[611], 90.00th=[635], 95.00th=[668].
99.00th=[938], 99.50th=[1303], 99.90th=[3130], 99.95th=[3949].
99.99th=[5669]
bw (KiB/s): min= 6288, max= 6968, per=100.00%, avg=6644.03, stdev=152.58, samples=119
iops : min= 1572, max= 1742, avg=1660.99, stdev=38.16, samples=119
lat (usec) : 500=0.12%, 750=97.53%, 1000=1.51%
lat (msec) : 2=0.59%, 4=0.20%, 10=0.05%
cpu : usr=1.13%, sys=2.03%, ctx=99679, maif=0, minf=34
IO depths : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
issued rwts: total=0,99676.0,0 short=0,0,0.0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00%, depth=1
Run status group θ (all jobs):
WRITE: bw=6645KiB/s (6804kB/s), 6645KiB/s -6645KiB/s (6804kB/s -6804kB/s), io=389MiB (408MB), run=60001-60001msec
[root@100 ~]#

Single-queue sequential read

• fio command:

```
fio -direct=1 -iodepth=1 -rw=read -ioengine=libaio -bs=4k -size=10G -
numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/
fio_test_03 -name=read_test
```

NOTE

Variable */mnt/sfsturbo/fio_test_03 -name* indicates the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_03 -name** file in the **/mnt/sfsturbo** directory in this example. Set the location based on site requirements.

• fio result:

[root@100 -]# fio -direct=1 -iodepth=1 -rw=read -ioengine=libaio -bs=4k -size=106 -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/fio_test_03 -name=read_test
read_test: (g=0): rw=read, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
10-3.5
Starting 1 process
JODS: 1 (T=1): [R(1)][100.0%][T=0412A1B/S,W=0K1B/S][T=1003,W=0 10/S][ETa 00m:00S]
read_test: (groupid=0, jobs=1): err= 0: pid=3004/: lue NoV 8 21:39:05 2022
read: 10/5=15/0, Bw=0300K1B/5 (045/KB/5)(13/041B/00001msec)
stat (nsec): min=/// max=30933, avg=#152.29, state=1412./1
Ctat (Usec): min+4/3, max-bo42, avg-b22.//, Study-133.33
tat (usec): mil+*//, max-8040, avg=055.10, Stuev=155.02
Ctat percentitues (usec):
1 1.00th=[543], 5.00th=[502], 10.00th=[576], 20.00th=[576],
70.00th [535] % 0.00th [535] % 0.00th [517] % 0.00th [517]
0.04th=[1020] 0.05th=[1360] 0.00th=[000] 0.00th=[3818]
990th=[5473]
$h_{\rm W}$ (K1R/s); min= 5808 max= 6552 por=99 96% avg=6303 65 stdov=141 48 samples=119
1005 : min= 1452, max= 1638, avo=1575.87. stdev=35.37. samples=119
lat (usec) : 500=0.01%, 750=96.40%, 1000=2.54%
lat (msoc) : 2=0.81%, 4=0.20%, 10=0.04%
cpu : usr=0.99%, sys=1.84%, ctx=94596, maif=0, minf=34
IO depths : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
issued rwts: total=94593,0,0,0 short=0,0,0,0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00%, depth=1
Run status oroun A (all jobs):
READ: bue-5366K192(c.fd575R2(c)_5396K18/c-5396K18/c-6457k8/c-6457k8/c)_jo=370M18_(387M8)_run=60001-60001meor
[root@100 -]#

Single-queue sequential write

• fio command:

fio -direct=1 -iodepth=1 -rw=write -ioengine=libaio -bs=4k -size=10G numjobs=1 -runtime=60 -group_reporting -filename=/*mnt/sfsturbo/ fio_test_04 -name*=write_test

NOTE

Variable */mnt/sfsturbo/fio_test_04 -name* indicates the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_04 -name** file in the **/mnt/sfsturbo** directory in this example. Set the location based on site requirements.

• fio result:

