

弹性文件服务

最佳实践

文档版本 01
发布日期 2025-02-06



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 弹性文件服务数据迁移	1
1.1 方案概述	1
1.2 通过云专线迁移（rclone 工具）	1
1.3 跨服务器迁移（rclone 工具）	3
1.4 通过云专线迁移（rsync 工具）	7
1.5 文件系统之间迁移数据	9
1.6 SFS 容量型文件系统迁移至其他文件系统	11
2 SFS Turbo 性能测试	13
3 SFS Turbo 时延测试	21
4 为文件系统创建普通用户可读写的子目录	24
4.1 方案概述	24
4.2 资源和成本规划	25
4.3 实施步骤	26
4.3.1 创建文件系统本地目录	26
4.3.2 为每个用户创建有读写权限的子目录	28
5 IDC 专线通过域名挂载 SFS Turbo HPC 型	31
6 面向 AI 场景使用 OBS+SFS Turbo 的存储加速实践	34
6.1 方案概述	34
6.2 资源和成本规划	36
6.3 操作流程	36
6.4 实施步骤	38
6.4.1 创建资源	38
6.4.2 基本配置	38
6.4.2.1 配置 ModelArts 和 SFS Turbo 间网络直通	38
6.4.2.2 配置 SFS Turbo 和 OBS 联动	41
6.4.2.3 配置 SFS Turbo 数据自动导出到 OBS 桶	42
6.4.2.4 配置 SFS Turbo 数据淘汰策略	42
6.4.3 训练	43
6.4.3.1 上传数据至 OBS 并预热到 SFS Turbo 中	43
6.4.3.2 创建训练任务	43
6.4.4 例行维护	44

6.5 常见问题.....	45
7 提升 Linux 操作系统单客户端访问 SFS Turbo 的性能.....	46

1 弹性文件服务数据迁移

1.1 方案概述

SFS Turbo默认只能被云上同一个VPC内的ECS/CCE访问，用户可通过云专线/VPN/对等连接等方式打通网络，实现多种访问方式。

- 云下或其他云访问：云专线/VPN
- 云上同区域同一账号不同VPC：VPC对等连接
- 云上同区域跨账号访问：VPC对等连接
- 不同区域访问：云连接

迁移数据分为两种网络条件，通过可访问公网的ECS直接挂载迁移。

- 通过mount方式挂载访问SFS Turbo，将本地NAS存储中的数据迁移至SFS Turbo。
 - [通过云专线迁移（rclone工具）](#)
 - [通过云专线迁移（rsync工具）](#)
- 无法打通网络直接挂载，可以通过公网访问华为云ECS绑定的EIP，也可进行传输迁移。
 - [跨服务器迁移（rclone工具）](#)

1.2 通过云专线迁移（rclone 工具）

方案概述

用户可以将本地NAS存储中的数据，通过云专线和rclone工具迁移至云上SFS Turbo文件系统中，进行云上业务拓展。

此方案通过在云上创建一台Linux操作系统的云服务器，来连接本地NAS存储以及云上SFS Turbo文件系统的通信，并通过这台服务器将本地NAS存储中的数据迁移至云上。

将云上NAS存储数据迁移至SFS Turbo文件系统中也可以参考本方案，具体说明请参考[云上NAS数据迁移至弹性文件服务](#)。

约束与限制

- 仅支持使用Linux系统的云服务器进行数据迁移。
- 文件UID和GID在同步操作后将不再保持一致。
- 文件访问模式同步操作后将不再保持一致。
- 支持实现增量迁移，即只迁移发生变化的数据。
- 通过 rclone 工具进行同步时，在执行命令之前写入的数据，可以完全同步；在执行命令之后写入的数据，不能保证完全同步。

前提条件

- 已经完成云专线的购买和配置，详情请参考《云专线服务用户指南》。
- 已创建一台操作系统为Linux的云服务器。
- 已创建SFS Turbo文件系统，并获取到文件系统的挂载地址。
- 已经获取到本地NAS存储的挂载地址。

资源规划

本章节介绍案例中使用的资源信息。具体如表1-1所示。

表 1-1 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs 16GB c7.2xlarge.2 操作系统：Linux 区域：华北-北京四 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

操作步骤

步骤1 登录弹性云服务器管理控制台。

步骤2 登录已创建好的Linux系统云服务器，用于同时访问本地NAS存储和云上SFS Turbo文件系统。

步骤3 输入以下挂载命令，用于访问本地NAS存储。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 本地NAS挂载地址 /mnt/src
```

步骤4 输入以下挂载命令，用于访问云上文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统挂载地址 /mnt/dst
```

步骤5 在Linux云服务器中执行以下命令安装rclone工具。

```
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-*
```

步骤6 执行以下命令，进行数据同步。

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

📖 说明

参数说明如下，transfers和checkers数目可以根据系统规格自行配置：

- --transfers：传输文件的并发数目。
- --checkers：扫描本地文件的并发数目。
- -P：数据拷贝进度。
- --links：复制源端的软链接，目的端保持为软链接的形式。
--copy-links：复制源端软链接指向的文件内容，目的端变成文件的形式，不再是软链接。
- --create-empty-src-dirs：复制源端的空目录到目的端。

等待数据完成同步后，可前往目标文件系统查看是否已成功迁移。

----结束

验证

步骤1 登录已创建好的Linux系统云服务器。

步骤2 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst  
ls | wc -l
```

步骤3 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

云上 NAS 数据迁移至弹性文件服务

云上NAS和SFS Turbo文件系统只需配置在同一个VPC下，或使用云连接（CC）配置网络后，再参考以上操作步骤，即可进行云上NAS数据迁移至弹性文件服务中。

云连接配置操作请参考[云连接用户指南](#)。

1.3 跨服务器迁移（rclone 工具）

方案概述

用户可以将本地NAS存储中的数据，使用rclone工具通过公网或内网迁移至云上SFS Turbo中，进行云上业务拓展。

此方案通过在云上和本地分别创建一台Linux操作系统的服务器，将本地NAS存储中的数据迁移至云上。两台服务器端口号22的出入方向需允许访问。本地服务器用于访问本地NAS存储，云上服务器用于访问云上SFS Turbo。

通过公网或内网将云上NAS存储数据迁移至SFS Turbo中也可以参考本方案完成。

约束与限制

- 暂不支持使用公网迁移本地NAS存储至SFS容量型文件系统。
- 仅支持使用Linux系统的云服务器进行数据迁移。
- 文件UID和GID在同步操作后将不再保持一致。

- 文件访问模式同步操作后不再保持一致。
- 端口号22的出入方向需允许访问。
- 支持实现增量迁移，即只迁移发生变化的数据。
- 通过 rclone 工具进行同步时，在执行命令之前写入的数据，可以完全同步；在执行命令之后写入的数据，不能保证完全同步。

前提条件

- 已在云上和本地分别创建一台Linux操作系统的服务器。
- 服务器已完成弹性IP的配置，保证两台服务器间可以相互访问。
- 已创建SFS Turbo文件系统，并获取到文件系统的挂载地址。
- 已经获取到本地NAS存储的挂载地址。

资源规划

本章节介绍案例中使用的资源信息。具体如表1-2所示。

表 1-2 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs 16GB c7.2xlarge.2 操作系统：Linux 区域：华北-北京四 VPC名称：VPC1 端口开放：22 弹性IP：xxx.xxx.xxx.xxx	已创建/mnt/dst的目录

操作步骤

步骤1 登录弹性云服务器管理控制台。

步骤2 登录已创建的本地服务器client1，执行以下命令，用于访问本地NAS存储。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 本地NAS挂载地址 /mnt/src
```

步骤3 登录已创建的Linux云服务器client2，执行以下命令，用于访问云上SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock SFS Turbo挂载地址 /mnt/dst
```

步骤4 在本地服务器client1中执行以下命令安装rclone工具。

```
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-*
```

步骤5 在本地服务器client1中执行以下命令，进行环境配置。

```
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 (新建名字)
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 (选择ssh/sftp号码)
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 (client2机器ip地址)
SSH username, leave blank for current username, root
Enter a string value. Press Enter for the default ("").
user> user name(client2机器的用户名)
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:(client2机器的登录密码)
Confirm the password:
password:(确认client2机器的登录密码)
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> (默认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").
key_use_agent> (默认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> (默认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          Type
=====
remote_name   sftp

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
```

说明

client2机器ip地址为公网地址。

步骤6 执行以下命令查看rclone.conf文件，路径为/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 = ***
```

步骤7 在本地服务器client1执行以下命令，同步数据。

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

说明

- 命令中的`remote_name`需替换为实际的远端名称。
- 参数说明如下，`transfers`和`checkers`数目可以根据系统规格自行配置：
 - `transfers`：传输文件的并发数目。
 - `checkers`：扫描本地文件的并发数目。
 - `P`：数据拷贝进度。

等待数据完成同步后，可前往目标SFS Turbo文件系统查看是否已成功迁移。

----结束

验证

步骤1 登录已创建好的Linux系统云服务器。

步骤2 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst
ls | wc -l
```

步骤3 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

1.4 通过云专线迁移（rsync 工具）

方案概述

用户可以将本地NAS存储中的数据，通过云专线和rsync工具迁移至云上SFS Turbo文件系统中，进行云上业务拓展。

此方案通过在云上创建一台Linux操作系统的云服务器，来连接本地NAS存储以及云上SFS Turbo文件系统的通信，并通过这台服务器将本地NAS存储中的数据迁移至云上。

将云上NAS存储数据迁移至SFS Turbo文件系统中也可以参考本方案，云上NAS和SFS Turbo文件系统需要配置在同一个VPC下。

约束与限制

- 支持拷贝特殊文件，如连接文件、设备等。
- 支持单文件断点续传。
- 支持保持原文件或目录的权限、时间、软硬链接、属主、组等所有属性。
- 支持使用rcp、rsh、ssh等方式来配合传输文件。
- 支持实现增量迁移，即只迁移发生变化的数据。
- 如果存在迁移多层级目录且小文件数量较多的情况时，rsync迁移效率将会非常低，建议使用多进程脚本解决或rclone工具迁移。

前提条件

- 已经完成云专线的购买和配置，详情请参考《云专线服务用户指南》。
- 已创建一台操作系统为Linux的云服务器。
- 已创建SFS Turbo文件系统，并获取到文件系统的挂载地址。
- 已经获取到本地NAS存储的挂载地址。
- 本地NAS和云服务器已完成互信关系建立。

资源规划

本章节介绍案例中使用的资源信息。具体如表1-3所示。

表 1-3 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs 16GB c7.2xlarge.2 操作系统：Linux 区域：华北-北京四 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

操作步骤

步骤1 登录已创建好的Linux系统云服务器，用于同时访问本地NAS存储和SFS Turbo文件系统。

步骤2 输入以下挂载命令，用于访问本地NAS存储。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 本地NAS挂载地址 /mnt/src
```

步骤3 输入以下挂载命令，用于访问SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统挂载地址 /mnt/dst
```

步骤4 在Linux云服务器中执行以下命令安装rsync工具。

```
yum install rsync
```

图 1-1 安装 rsync 工具

```
[root@ ~]# yum install rsync
Loaded plugins: fastestmirror
Determining fastest mirrors
epel/x86_64/metalink
 * base: mirrors.huaweicloud.com
 * epel: mirrors.bfsu.edu.cn
 * extras: mirrors.ustc.edu.cn
 * updates: mirrors.huaweicloud.com
```

说明

源服务器和目的端服务器均需要安装rsync工具，否则将会报错。

步骤5 安装完成后，可以输入以下命令查询rsync工具的安装和版本情况。

```
rsync -version
```

图 1-2 查看工具安装情况

```
[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 [OPTION]... SRC [SRC]... [USER@]HOST::DEST
      or rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/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.
```

步骤6 执行以下命令，将源端服务器/mnt/src目录中的数据全量同步迁移至目的端服务器/mnt/dst目录（文件系统）中。

```
rsync -avP /mnt/src /mnt/dst
```

图 1-3 全量同步数据

```
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 (xfr#4083, to-chk=295917/300001)
103672.dat
      1,024 100% 35.71kB/s  0:00:00 (xfr#4084, to-chk=295916/300001)
103673.dat
      1,024 100% 34.48kB/s  0:00:00 (xfr#4085, to-chk=295915/300001)
103674.dat
      1,024 100% 34.48kB/s  0:00:00 (xfr#4086, to-chk=295914/300001)
103675.dat
      1,024 100% 33.33kB/s  0:00:00 (xfr#4087, to-chk=295913/300001)
```

----结束

验证结果

步骤1 登录已创建好的Linux系统云服务器。

步骤2 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst
ls | wc -l
```

步骤3 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

1.5 文件系统之间迁移数据

方案概述

用户可以将SFS容量型文件系统中的数据迁移至SFS Turbo文件系统中，也可以将SFS Turbo文件系统中的数据迁移至SFS容量型文件系统中，进行云上业务拓展。

此方案通过创建一台Linux操作系统的云服务器，来连接SFS容量型文件系统和SFS Turbo文件系统的通信。

约束与限制

- 仅支持使用Linux系统的云服务器进行数据迁移。
- Linux系统云服务器、SFS容量型文件系统和SFS Turbo文件系统需在同一VPC下。
- 支持实现增量迁移，即只迁移发生变化的数据。

前提条件

- 已创建一台操作系统为Linux的云服务器。

- 已创建SFS容量型或SFS Turbo文件系统，并获取到文件系统的挂载地址。

资源规划

本章节介绍案例中使用的资源信息。具体如表1-4所示。

表 1-4 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs 16GB c7.2xlarge.2 操作系统：Linux 区域：华北-北京四 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

操作步骤

步骤1 登录弹性云服务器管理控制台。

步骤2 登录已创建好的Linux系统云服务器，用于同时访问SFS容量型文件和SFS Turbo文件系统。

步骤3 输入以下挂载命令，用于访问文件系统1。文件系统1可以是SFS容量型文件系统或SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统1挂载地址 /mnt/src
```

步骤4 输入以下挂载命令，用于访问文件系统2。文件系统2可以是SFS容量型文件或SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统2挂载地址 /mnt/dst
```

步骤5 下载并安装rclone工具。下载地址请参见<https://rclone.org/downloads/>。

步骤6 执行以下命令，进行数据同步。

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

📖 说明

参数说明如下，transfers和checkers数目可以根据系统规格自行配置：

- /mnt/src：源路径
- /mnt/dst：目标路径
- --transfers：传输文件的并发数目。
- --checkers：扫描本地文件的并发数目。
- -P：数据拷贝进度。
- --links：复制源端的软链接，目的端保持为软链接的形式。
- --copy-links：复制源端软链接指向的文件内容，目的端变成文件的形式，不再是软链接。
- --create-empty-src-dirs：复制源端的空目录到目的端。

等待数据完成同步后，可前往目标文件系统查看是否已成功迁移。

----结束

验证

步骤1 登录已创建好的Linux系统云服务器。

步骤2 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst  
ls | wc -l
```

步骤3 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

1.6 SFS 容量型文件系统迁移至其他文件系统

方案概述

用户可以将SFS容量型文件系统中的数据迁移至通用文件系统或SFS Turbo文件系统中。

此方案通过创建一台Linux操作系统的云服务器，来连接SFS容量型文件系统和其他文件系统的通信。

约束与限制

- 仅支持使用Linux系统的云服务器进行数据迁移。
- Linux系统云服务器、SFS容量型文件系统和其他文件系统需在同一VPC下。如果目标端是通用文件系统，需配置VPC终端节点，具体操作，请参见[配置VPC终端节点](#)。
- 支持实现增量迁移，即只迁移发生变化的数据。

前提条件

- 已创建一台操作系统为Linux的云服务器。
- 已创建SFS容量型和其他文件系统，并获取到文件系统的挂载地址。

资源规划

本章节介绍案例中使用的资源信息。具体如[表1-5](#)所示。

表 1-5 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs 16GB c7.2xlarge.2 操作系统：Linux 区域：华北-北京四 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

操作步骤

步骤1 登录弹性云服务器管理控制台。

步骤2 登录已创建好的Linux系统云服务器，用于同时访问SFS容量型文件系统和其他文件系统。

步骤3 输入以下挂载命令，用于访问文件系统1。文件系统1是SFS容量型文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统1挂载地址 /mnt/src
```

步骤4 输入以下挂载命令，用于访问文件系统2。文件系统2可以是通用文件系统或SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统2挂载地址 /mnt/dst
```

步骤5 在Linux云服务器中执行以下命令安装rclone工具。

```
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-*
```

📖 说明

rclone工具不保留源端的文件权限或属组信息，有相关需求请使用rsync工具。

步骤6 执行以下命令，进行数据同步。

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

📖 说明

参数说明如下，transfers和checkers数目可以根据系统规格自行配置：

- /mnt/src：源路径
- /mnt/dst：目标路径
- --transfers：传输文件的并发数目。
- --checkers：扫描本地文件的并发数目。
- -P：数据拷贝进度。
- --links：复制源端的软链接，目的端保持为软链接的形式。
- --copy-links：复制源端软链接指向的文件内容，目的端变成文件的形式，不再是软链接。
- --create-empty-src-dirs：复制源端的空目录到目的端。

等待数据完成同步后，可前往目标文件系统查看是否已成功迁移。

----结束

验证

步骤1 登录已创建好的Linux系统云服务器。

步骤2 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst
ls | wc -l
```

数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

2 SFS Turbo 性能测试

fio是一个开源的I/O压力测试工具，可以使用fio工具对SFS进行吞吐量和IOPS的性能测试。

前提条件

已在云服务器上安装fio工具。fio可从[官网](#)或[GitHub](#)下载。

注意和说明

测试性能依赖client和server之间的网络带宽及文件系统的容量大小。

安装 fio

以Linux CentOS系统为例说明：

1. 在官网下载fio。
yum install fio
2. 安装libaio引擎。
yum install libaio-devel
3. 查看fio版本。
fio --version

文件系统性能数据

SFS Turbo文件系统的性能主要有IOPS和吞吐量等指标，具体各指标数据参见[表2-1](#)。

表 2-1 性能数据表

参数	HPC型					
	20MB/s/ TiB	40MB/s/ TiB	125MB/s/ TiB	250MB/s/ TiB	500MB/s/ TiB	1000MB/s /TiB
最大容量	1PB	1PB	1PB	1PB	1PB	1PB

参数	HPC型					
	20MB/s/ TiB	40MB/s/ TiB	125MB/s/ TiB	250MB/s/ TiB	500MB/s/ TiB	1000MB/s /TiB
最大 IOPS	25万	25万	100万	100万	400万	400万
最大吞吐量	8GB/s	8GB/s	20GB/s	20GB/s	80GB/s	80GB/s
IOPS 性能计 算公式	IOPS=mi n(25000 0,600× 容量) 其中，容 量单位为 TB	IOPS=mi n(25000 0,1200× 容量) 其中，容 量单位为 TB	IOPS=mi n(100000 0,6000×容 量) 其中，容 量单位为 TB	IOPS=mi n(100000 0,12500× 容量) 其中，容 量单位为 TB	IOPS=mi n(400000 0,25000× 容量) 其中，容 量单位为 TB	IOPS=mi n(400000 0,50000× 容量) 其中，容 量单位为 TB

通用测试配置样例

📖 说明

以下提供的预估值为单台弹性云服务器ECS测试的结果。建议使用多台ECS测试，以达到[弹性文件服务](#)的性能指标。

本文以SFS Turbo性能型，云服务器规格如下为例说明。

规格：通用计算增强型 | c3.xlarge.4 | 4vCPUs | 16GB

镜像：CentOS 7.564bit

- fio命令：

```
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
```

📖 说明

其中，“/mnt/nfs/test_fio”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/nfs”目录下的“test_fio”文件，请根据实际填写。

- fio结果：

```
test: (groupid=0, jobs=1): err= 0: pid=10110: Mon Jun 0 11:40:57 2020
read: IOPS=7423, BW=28.0MiB/s (30.4MB/s)(7167MiB/247160msec)
slat (nsec): min=1234, max=397477, avg=4145.45, stdev=3344.40
clat (usec): min=245, max=133325, avg=11162.10, stdev=12136.31
lat (usec): min=252, max=133330, avg=11166.32, stdev=12136.34
clat percentiles (usec):
| 1.00th=[ 2245], 5.00th=[ 2540], 10.00th=[ 2671], 20.00th=[ 2900],
| 30.00th=[ 3130], 40.00th=[ 3450], 50.00th=[ 4293], 60.00th=[ 7032],
| 70.00th=[13173], 80.00th=[19792], 90.00th=[20443], 95.00th=[36439],
| 99.00th=[53216], 99.50th=[60031], 99.90th=[79160], 99.95th=[85459],
| 99.99th=[98042]
bw ( KIB/s): min=16600, max=45560, per=100.00%, avg=29696.00, stdev=5544.46, samples=494
iops      : min= 4150, max=11390, avg=7424.01, stdev=1386.11, samples=494
write: IOPS=3182, BW=12.4MiB/s (13.0MB/s)(3073MiB/247160msec)
slat (nsec): min=1480, max=302730, avg=4613.59, stdev=3359.60
clat (usec): min=1447, max=140666, avg=14166.05, stdev=13373.72
lat (usec): min=1457, max=140671, avg=14170.73, stdev=13373.74
clat percentiles (msec):
| 1.00th=[  41], 5.00th=[  41], 10.00th=[  41], 20.00th=[  51],
| 30.00th=[  51], 40.00th=[  61], 50.00th=[  81], 60.00th=[ 141],
| 70.00th=[ 101], 80.00th=[ 241], 90.00th=[ 331], 95.00th=[ 421],
| 99.00th=[ 591], 99.50th=[ 671], 99.90th=[ 871], 99.95th=[ 941],
| 99.99th=[ 1221]
bw ( KIB/s): min= 7144, max=19600, per=100.00%, avg=12730.90, stdev=2395.77, samples=494
iops      : min= 1706, max= 4900, avg=3182.70, stdev=590.96, samples=494
lat (usec) : 250=0.01%, 500=0.01%, 750=0.01%, 1000=0.01%
lat (msec)  : 2=0.20%, 4=39.15%, 10=21.01%, 20=17.92%, 50=20.06%
cpu (msec)  : 100=1.62%, 250=0.02%
cpu         : usr=1.35%, sys=6.43%, ctx=1072910, majf=0, minf=30
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%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete   : 0=0.0%, 4=100.0%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=1034036,706604,0,0 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=28.0MiB/s (30.4MB/s), 28.0MiB/s-28.0MiB/s (30.4MB/s-30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec
WRITE: bw=12.4MiB/s (13.0MB/s), 12.4MiB/s-12.4MiB/s (13.0MB/s-13.0MB/s), io=3073MiB (3222MB), run=247160-247160msec
```

- fio命令:

```
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
```

说明

其中，“/mnt/nfs/test_fio”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/nfs”目录下的“test_fio”文件，请根据实际情况填写。

- fio结果:

```
test: (groupid=0, jobs=1): err= 0: pid=20358: Mon Jun 0 11:57:14 2020
read: IOPS=5065, BW=19.8MiB/s (20.7MB/s)(3073MiB/155200msec)
slat (nsec): min=1271, max=265900, avg=4073.51, stdev=3040.12
clat (usec): min=226, max=80105, avg=5711.35, stdev=7079.46
lat (usec): min=232, max=80107, avg=5715.49, stdev=7079.48
clat percentiles (usec):
| 1.00th=[ 1221], 5.00th=[ 1950], 10.00th=[ 2100], 20.00th=[ 2442],
| 30.00th=[ 2606], 40.00th=[ 2802], 50.00th=[ 2991], 60.00th=[ 3220],
| 70.00th=[ 3687], 80.00th=[ 5604], 90.00th=[14222], 95.00th=[21890],
| 99.00th=[35914], 99.50th=[40633], 99.90th=[51643], 99.95th=[55837],
| 99.99th=[66047]
bw ( KIB/s): min=13360, max=28040, per=99.99%, avg=20257.97, stdev=2913.05, samples=310
iops      : min= 3340, max= 7212, avg=5064.40, stdev=720.27, samples=310
write: IOPS=11.0k, BW=46.2MiB/s (48.4MB/s)(7167MiB/155200msec)
slat (nsec): min=1396, max=390604, avg=4405.60, stdev=3091.75
clat (usec): min=057, max=140259, avg=0377.47, stdev=0400.15
lat (usec): min=067, max=140264, avg=0302.02, stdev=0400.16
clat percentiles (msec):
| 1.00th=[  31], 5.00th=[  41], 10.00th=[  41], 20.00th=[  41],
| 30.00th=[  51], 40.00th=[  51], 50.00th=[  51], 60.00th=[  61],
| 70.00th=[  71], 80.00th=[ 131], 90.00th=[ 211], 95.00th=[ 201],
| 99.00th=[ 421], 99.50th=[ 471], 99.90th=[ 601], 99.95th=[ 601],
| 99.99th=[ 1201]
bw ( KIB/s): min=32224, max=67456, per=99.90%, avg=47254.23, stdev=6792.41, samples=310
iops      : min= 8056, max=16064, avg=11013.55, stdev=1690.11, samples=310
lat (usec) : 250=0.01%, 500=0.04%, 750=0.07%, 1000=0.09%
lat (msec)  : 2=1.53%, 4=36.85%, 10=41.27%, 20=11.30%, 50=0.61%
cpu (msec)  : 100=0.23%, 250=0.01%
cpu         : usr=2.13%, sys=9.90%, ctx=925770, majf=0, minf=31
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%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete   : 0=0.0%, 4=100.0%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=706597,1034043,0,0 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=155200-155200msec
WRITE: bw=46.2MiB/s (48.4MB/s), 46.2MiB/s-46.2MiB/s (48.4MB/s-48.4MB/s), io=7167MiB (7516MB), run=155200-155200msec
```

顺序读IOPS

- fio命令:

```
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
```

📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
test: (groupid=0, jobs=1): err= 0: pid=20459: Mon Jun  8 12:28:18 2020  
read: IOPS=9654, BW=37.7MiB/s (39.5MB/s)(10.0GiB/271519msec)  
slat (usec): 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=13958.09  
clat percentiles (usec):  
| 1.00th=[ 17621], 5.00th=[ 19911], 10.00th=[ 21471], 20.00th=[ 23761],  
| 30.00th=[ 27041], 40.00th=[ 36211], 50.00th=[ 77671], 60.00th=[ 119941],  
| 70.00th=[ 169091], 80.00th=[ 234621], 90.00th=[ 331621], 95.00th=[ 416811],  
| 99.00th=[ 595071], 99.50th=[ 668471], 99.90th=[ 833621], 99.95th=[ 907021],  
| 99.99th=[1032851]  
bw ( Kib/s): min=10656, max=61576, per=99.99%, avg=30615.41, stdev=7703.32, samples=543  
iops      : min= 4664, max=15394, avg=9653.02, stdev=1925.03, samples=543  
lat (usec) : 500=0.01%, 750=0.01%, 1000=0.02%  
lat (msec)  : 2=5.25%, 4=36.35%, 10=12.76%, 20=20.56%, 50=22.62%  
lat (msec)  : 100=2.42%, 250=0.02%  
cpu         : usr=1.04%, sys=5.35%, ctx=913138, majf=0, minf=159  
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 ruts: total=2621440,0,0,0 short=0,0,0 dropped=0,0,0  
latency    : target=0, window=0, percentile=100.00%, depth=128  
  
Run status group 0 (all jobs):  
READ: bw=37.7MiB/s (39.5MB/s), 37.7MiB/s-37.7MiB/s (39.5MB/s-39.5MB/s), io=10.0GiB (10.7GB), run=2
```

随机读IOPS

- fio命令：

```
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
```

📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
test: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
fio-2.1.10
Starting 1 process
Jobs: 1 (f=1): [r] [100.0% done] [17824KB/0KB/0KB /s] [4456/0/0 iops] [eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=20755: Tue Dec 28 09:41:43 2021
  read : io=10240MB, bw=18597KB/s, iops=4649, runt=563832msec
    slat (usec): min=1, max=375, avg= 2.64, stdev= 2.52
    clat (usec): min=715, max=755902, avg=27527.31, stdev=106233.39
      lat (usec): min=718, max=755903, avg=27530.03, stdev=106233.39
    clat percentiles (msec):
      | 1.00th=[  3],  5.00th=[  5], 10.00th=[  6], 20.00th=[  6],
      | 30.00th=[  7], 40.00th=[  7], 50.00th=[  8], 60.00th=[  9],
      | 70.00th=[ 11], 80.00th=[ 15], 90.00th=[ 21], 95.00th=[ 28],
      | 99.00th=[ 676], 99.50th=[ 693], 99.90th=[ 725], 99.95th=[ 734],
      | 99.99th=[ 750]
    bw (KB /s): min= 1896, max=35752, per=100.00%, avg=18605.56, stdev=1980.86
    lat (usec) : 750=0.01%, 1000=0.01%
    lat (msec) : 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=2.72%, 1000=0.01%
  cpu      : usr=0.82%, sys=2.41%, ctx=1231561, majf=0, minf=155
  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   : total=r=2621440/w=0/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):
  READ: io=10240MB, aggrb=18597KB/s, minb=18597KB/s, maxb=18597KB/s, mint=563832msec, maxt=563832msec
```

顺序写IOPS

- fio命令:

```
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
```

📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果:

```
test: (groupid=0, jobs=1): err= 0: pid=20874: Mon Jun  8 14:23:09 2020
write: IOPS=11.0k, BW=43.1MiB/s (45.2MB/s)(10.0GiB/237436msec)
  slat (nsec): min=1483, max=368726, avg=4388.87, stdev=3688.87
  clat (usec): min=1953, max=186548, avg=11588.61, stdev=5876.84
    lat (usec): min=1959, max=186552, avg=11593.86, stdev=5876.86
  clat percentiles (usec):
    | 1.00th=[ 4815],  5.00th=[ 5932], 10.00th=[ 6652], 20.00th=[ 7439],
    | 30.00th=[ 8829], 40.00th=[ 8848], 50.00th=[ 9634], 60.00th=[10814],
    | 70.00th=[12518], 80.00th=[15533], 90.00th=[19268], 95.00th=[22676],
    | 99.00th=[32637], 99.50th=[37487], 99.90th=[49821], 99.95th=[53748],
    | 99.99th=[69731]
  bw ( KiB/s): min=31712, max=52431, per=99.99%, avg=44158.84, stdev=3987.31, samples=474
  iops       : min= 7928, max=13187, avg=11839.58, stdev=996.83, samples=474
  lat (msec) : 2=0.81%, 4=1.00%, 10=51.94%, 20=38.58%, 50=8.39%
  lat (msec) : 100=0.88%, 250=0.81%
  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 rwt: 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 0 (all jobs):
  WRITE: bw=43.1MiB/s (45.2MB/s), 43.1MiB/s-43.1MiB/s (45.2MB/s-45.2MB/s), io=10.0GiB (10.7GB), runt=
```

随机写IOPS

- fio命令:

```
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
```


说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
test: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
fio-2.1.10
Starting 1 process

test: (groupid=0, jobs=1): err= 0: pid=16622: Thu Jan 13 10:13:22 2022
write: io=10240MB, bw=18463KB/s, iops=4615, runt=567947msec
slat (usec): min=1, max=356, avg= 3.21, stdev= 2.04
clat (usec): min=890, max=815560, avg=27727.54, stdev=101207.14
lat (usec): min=893, max=815564, avg=27730.83, stdev=101207.14
clat percentiles (msec):
| 1.00th=[ 4], 5.00th=[ 6], 10.00th=[ 6], 20.00th=[ 7],
| 30.00th=[ 7], 40.00th=[ 8], 50.00th=[ 8], 60.00th=[ 10],
| 70.00th=[ 13], 80.00th=[ 16], 90.00th=[ 23], 95.00th=[ 30],
| 99.00th=[ 644], 99.50th=[ 668], 99.90th=[ 701], 99.95th=[ 709],
| 99.99th=[ 734]
bw (KB /s): min= 1064, max=36589, per=100.00%, avg=18469.11, stdev=3769.64
lat (usec) : 1000=0.01%
lat (msec) : 2=0.20%, 4=1.85%, 10=60.93%, 20=24.30%, 50=9.85%
lat (msec) : 100=0.09%, 250=0.01%, 500=0.08%, 750=2.68%, 1000=0.01%
cpu : usr=0.98%, sys=2.90%, ctx=1552744, majf=0, 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 : total=r=0/w=2621440/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=18462KB/s, minb=18462KB/s, maxb=18462KB/s, mint=567947msec, maxt=567947msec
```

顺序读带宽

- fio命令：

```
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
```

说明

其中，“/mnt/sfs-turbo/test_fio”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfs-turbo”目录下的“test_fio”文件，请根据实际情况填写。

- fio结果：

```
test: (groupid=0, jobs=1): err= 0: pid=28962: Mon Jun 8 14:37:48 2020
read: IOPS=398, BW=391MiB/s (409MB/s)(10.0GiB/26221msec)
slat (usec): min=78, max=595, avg=99.58, stdev=39.89
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],
| 99.99th=[ 542]
bw ( KiB/s): min=301856, 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%, 100=0.28%, 250=27.61%, 500=71.37%, 750=0.58%
cpu : usr=0.08%, sys=4.21%, ctx=18395, majf=0, minf=97
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 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=10.0GiB (10.7GB), run=26221-26221msec
```

随机读带宽

- fio命令：

```
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
```

📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
test: (g=0): rw=randread, bs=1M-1M/1M-1M-1M, ioengine=libaio, iodepth=128  
fio-2.1.10  
Starting 1 process  
  
test: (groupid=0, jobs=1): err= 0: pid=14261: Tue Dec 28 09:18:04 2021  
read : io=10240MB, bw=154130KB/s, iops=150, runt= 68032msec  
slat (usec): min=61, max=8550, avg=142.99, stdev=187.96  
clat (msec): min=12, max=2002, avg=849.91, stdev=347.27  
lat (msec): min=12, max=2003, avg=850.05, stdev=347.26  
clat percentiles (msec):  
| 1.00th=[ 47], 5.00th=[ 84], 10.00th=[ 105], 20.00th=[ 914],  
| 30.00th=[ 947], 40.00th=[ 963], 50.00th=[ 971], 60.00th=[ 988],  
| 70.00th=[ 996], 80.00th=[ 1012], 90.00th=[ 1037], 95.00th=[ 1057],  
| 99.00th=[ 1876], 99.50th=[ 1926], 99.90th=[ 1975], 99.95th=[ 1975],  
| 99.99th=[ 2008]  
bw (KB /s): min=69974, max=167768, per=98.85%, avg=152360.15, stdev=10783.47  
lat (msec) : 20=0.33%, 50=0.80%, 100=7.02%, 250=7.95%, 1000=55.30%  
lat (msec) : 2000=28.57%, >=2000=0.02%  
cpu      : usr=0.02%, sys=1.93%, ctx=4399, majf=0, minf=602  
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=10240/w=0/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):  
  READ: io=10240MB, aggrb=154129KB/s, minb=154129KB/s, maxb=154129KB/s, mint=68032msec, max  
t=68032msec
```

顺序写带宽

- fio命令：

```
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
```

📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
test: (groupid=0, jobs=1): err= 0: pid=21889: Mon Jun 8 14:53:44 2020  
write: IOPS=243, BW=244MiB/s (255MB/s)(18.8GiB/42848msec)  
slat (usec): min=183, max=504, avg=198.38, stdev=29.47  
clat (msec): min=18, max=1184, avg=525.23, stdev=253.35  
lat (msec): min=18, max=1184, avg=525.42, stdev=253.35  
clat percentiles (msec):  
| 1.00th=[ 51], 5.00th=[ 188], 10.00th=[ 167], 20.00th=[ 292],  
| 30.00th=[ 422], 40.00th=[ 468], 50.00th=[ 586], 60.00th=[ 558],  
| 70.00th=[ 625], 80.00th=[ 768], 90.00th=[ 982], 95.00th=[ 978],  
| 99.00th=[ 1836], 99.50th=[ 1845], 99.90th=[ 1878], 99.95th=[ 1899],  
| 99.99th=[ 1899]  
bw (KiB/s): min= 4896, max=468992, per=100.00%, avg=249588.99, stdev=147656.62, samples=83  
iops      : min= 4, max= 458, avg=243.63, stdev=144.22, samples=83  
lat (msec) : 28=0.83%, 58=0.96%, 188=3.36%, 258=12.55%, 588=31.63%  
lat (msec) : 758=38.87%, 1888=18.96%  
cpu      : usr=2.28%, sys=2.58%, ctx=3972, majf=0, minf=27  
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=18240,0 short=r=0,0,0 dropped=0,0,0  
latency  : target=0, window=0, percentile=100.00%, depth=128  
  
Run status group 0 (all jobs):  
  WRITE: bw=244MiB/s (255MB/s), 244MiB/s-244MiB/s (255MB/s-255MB/s), io=18.8GiB (18.7GB), run=42848-42848msec
```

随机写带宽

- **fio命令:**
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

说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际填写。

- **fio结果:**

```
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=[ 97],
  | 30.00th=[ 97], 40.00th=[ 98], 50.00th=[ 98], 60.00th=[ 99],
  | 70.00th=[ 99], 80.00th=[ 100], 90.00th=[ 100], 95.00th=[ 101],
  | 99.00th=[ 102], 99.50th=[ 102], 99.90th=[ 103], 99.95th=[ 104],
  | 99.99th=[ 195]
  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 SFS Turbo 时延测试

本文以SFS Turbo性能增强型，云服务器规格如下为例说明。

规格：通用计算增强型 | c6.4xlarge.4 | 16vCPUs | 64GB

镜像：EulerOS 2.5

单队列随机读

- fio命令：

```
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
```

📖 说明

其中，“/mnt/sfsturbo/fio_test_01 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio_test_01 -name”文件，请根据实际情况填写。

- fio结果：

```
[root@100 ~]# 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
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=29815: Mon Nov 7 11:44:12 2022
reads: IOPS=1401, 0m=5600KiB/s (5740KB/s) (320MiB/60001msec)
slat (nsec): min=2379, max=51192, avg=4228.52, stdev=1319.26
clat (usec): min=562, max=8407, 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=[ 1350], 99.50th=[ 2040], 99.90th=[ 3523], 99.95th=[ 3982],
| 99.99th=[ 5755]
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=80.70%, 1000=9.21%
lat (msec) : 2=1.39%, 4=0.40%, 10=0.05%
cpu : usr=0.86%, sys=1.77%, ctx=84892, 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=84807,0,0,0 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=5600KiB/s (5740KB/s), 5600KiB/s-5600KiB/s (5740KB/s-5740KB/s), io=320MiB (344MB), run=60001-60001msec
[root@100 ~]#
```

单队列随机写

- fio命令：

```
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
```

说明

其中，“/mnt/sfsturbo/fio_test_02 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio_test_02 -name”文件，请根据实际填写。

- fio结果：

```
[root@100 ~]# 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
Starting 1 process
Jobs: 1 (f=1): [w(1)][100.0%][r=0kIB/s,w=6708KIB/s][r=0,w=1677 IOPS][eta 00m:00s]
randwrite test: (groupid=0, jobs=1): err=0: pid=25783: Tue Nov 8 21:37:13 2022
write: IOPS=1661, BW=6645KIB/s (6884kB/s) (389MB/6000msec)
slat (msec): min=2566, max=34191, avg=4528.29, stdev=1661.31
clat (usec): min=473, max=8234, avg=595.67, stdev=169.48
lat (usec): min=477, max=8234, avg=600.38, stdev=169.55
clat percentiles (usec):
| 1.00th=[ 515], 5.00th=[ 529], 10.00th=[ 537], 20.00th=[ 553],
| 30.00th=[ 562], 40.00th=[ 578], 50.00th=[ 578], 60.00th=[ 586],
| 70.00th=[ 594], 80.00th=[ 611], 90.00th=[ 635], 95.00th=[ 668],
| 99.00th=[ 938], 99.50th=[ 1383], 99.90th=[ 3130], 99.95th=[ 3949],
| 99.99th=[ 5669]
bw ( KIB/s): min= 6288, max= 6968, per=100.00%, avg=6044.83, stdev=152.58, samples=119
iops   : min=1572, max=1742, avg=1668.99, stdev=30.16, 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=1.12%, sys=2.03%, ctx=98979, majf=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 rts: total=0,99976,0,0 short=0,0,0,0,0 dropped=0,0,0,0
latency    : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
WRITE: bw=6645KIB/s (6884kB/s), 6645KIB/s-6645KIB/s (6884kB/s-6884kB/s), io=389MB (489MB), run=60001.60001msec
[root@100 ~]#
```

单队列顺序读

- fio命令：

```
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
```

说明

其中，“/mnt/sfsturbo/fio_test_03 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio_test_03 -name”文件，请根据实际填写。

- fio结果：

```
[root@100 ~]# 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
Starting 1 process
Jobs: 1 (f=1): [R(1)][100.0%][r=6412KIB/s,w=0KIB/s][r=1602,w=0 IOPS][eta 00m:00s]
read test: (groupid=0, jobs=1): err=0: pid=38647: Tue Nov 8 21:39:05 2022
read: IOPS=1576, BW=6306KIB/s (6457kB/s) (370MB/6000msec)
slat (msec): min=2349, max=30953, avg=4152.29, stdev=1412.71
clat (usec): min=472, max=8642, avg=629.77, stdev=155.53
lat (usec): min=477, max=8646, avg=633.10, stdev=155.62
clat percentiles (usec):
| 1.00th=[ 545], 5.00th=[ 562], 10.00th=[ 570], 20.00th=[ 578],
| 30.00th=[ 594], 40.00th=[ 603], 50.00th=[ 611], 60.00th=[ 619],
| 70.00th=[ 635], 80.00th=[ 644], 90.00th=[ 676], 95.00th=[ 717],
| 99.00th=[ 1020], 99.50th=[ 1369], 99.90th=[ 2900], 99.95th=[ 3818],
| 99.99th=[ 5473]
bw ( KIB/s): min= 5888, max= 6552, per=99.96%, avg=6303.65, stdev=141.48, samples=119
iops   : min=1452, max=1638, avg=1575.87, stdev=35.37, samples=119
lat (usec)  : 500=0.01%, 750=96.40%, 1000=2.54%
lat (msec)  : 2=0.01%, 4=0.20%, 10=0.04%
cpu         : usr=0.99%, sys=1.84%, ctx=94596, majf=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 rts: total=0,99999,0,0 short=0,0,0,0,0 dropped=0,0,0,0
latency    : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
READ: bw=6306KIB/s (6457kB/s), 6306KIB/s-6306KIB/s (6457kB/s-6457kB/s), io=370MB (387MB), run=60001.60001msec
[root@100 ~]#
```

单队列顺序写

- fio命令：

```
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
```

说明

其中，“/mnt/sfsturbo/fio_test_04 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio_test_04 -name”文件，请根据实际填写。

- fio结果:

```
[root@100 ~]# fio --direct=1 --ioengine=libaio --bs=4k --size=100 --numjobs=1 --runtime=60 --group_reporting --filename=/mnt/sfsturbo/fio_test_04 --name=write_test
write_test: (g=0): rw=write, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.5
Starting 1 process
Jobs: 1 (f=1): [(w1)] [100.0%][r=0kIB/s,w=6886kIB/s][r=0,w=1721 IOPS][eta 00m:00s]
write_test: [groupid=0], jobs=1: err=0: pid=389: Tue Nov 9 21:40:23 2022
write: IOPS=1799, BW=7194KIB/s (7367kB/s)(422MB/6000Imsec)
slat (msec): min=2266, max=32022, avg=4256.55, stdev=1452.74
clat (usec): min=443, max=10894, avg=550.41, stdev=168.67
lat (usec): min=447, max=10900, avg=554.82, stdev=168.73
lat percentiles (usec):
| 1.00th=[ 469], 5.00th=[ 486], 10.00th=[ 494], 20.00th=[ 502],
| 30.00th=[ 515], 40.00th=[ 523], 50.00th=[ 529], 60.00th=[ 537],
| 70.00th=[ 553], 80.00th=[ 562], 90.00th=[ 594], 95.00th=[ 627],
| 99.00th=[ 963], 99.50th=[ 1483], 99.90th=[ 3064], 99.95th=[ 3654],
| 99.99th=[ 5211]
bw ( KIB/s): min= 6528, max= 7688, per=99.99%, avg=7193.45, stdev=213.10, samples=120
iops      : min= 1692, max= 1922, avg=1799.35, stdev=53.20, samples=120
lat (usec) : 500=16.99%, 750=80.94%, 1000=1.14%
lat (msec) : 2=0.63%, 4=0.26%, 10=0.03%, 20=0.01%
cpu       : usr=0.97%, sys=2.16%, ctx=107916, 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=0,107912,0,0 short=0,0,0 dropped=0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
WRITE: bw=7194KIB/s (7367kB/s), 7194KIB/s-7194KIB/s (7367kB/s-7367kB/s), io=422MB (442MB), run=60001-60001msec
[root@100 ~]#
```

4 为文件系统创建普通用户可读写的子目录

4.1 方案概述

应用场景

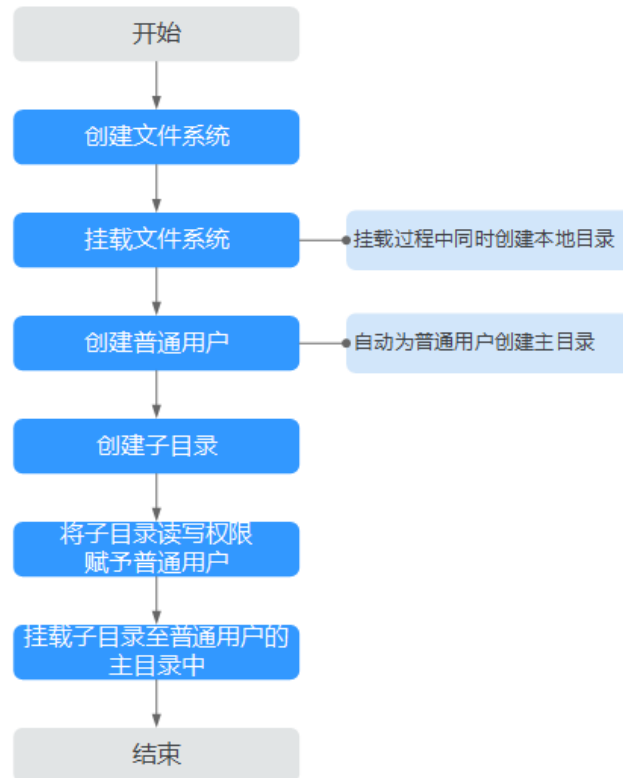
文件系统创建成功后，文件系统默认情况下只有root用户具有读写权限。

root用户可以根据需要，可以授权多个普通用户访问权限，并在文件主目录下为每个普通用户创建多个子目录并挂载至主目录上，同时使普通用户拥有读写权限。

操作流程

为文件系统创建普通用户可读写的子目录的流程如[图4-1](#)所示。

图 4-1 操作流程



4.2 资源和成本规划

本节介绍最佳实践中资源规划情况，包含以下内容：

表 4-1 资源和成本规划

资源	资源说明
弹性云服务器ECS	云服务器和文件系统归属于同一项目下，用于挂载共享路径实现数据共享。
虚拟私有云VPC	VPC为弹性云服务器构建隔离的、用户自主配置和管理的虚拟网络环境，提升用户云中资源的安全性，简化用户的网络部署。 云服务器无法访问不在同一VPC下的文件系统，使用弹性文件服务时需将文件系统和云服务器归属于同一VPC下。
文件系统	文件系统通过标准的NFS协议和CIFS协议为客户提供文件存储服务，用于网络文件远程访问，用户通过管理控制台创建挂载地址后，即可在多个云服务器上进行挂载，并通过标准的POSIX接口对文件系统进行访问。

4.3 实施步骤

4.3.1 创建文件系统本地目录

当创建文件系统后，您需要将文件系统挂载至云服务器上，再为root用户创建一个本地目录。

如已成功挂载文件系统，可跳过此章节。请记录[步骤4](#)的本地目录后，按照[为每个用户创建有读写权限的子目录](#)执行操作。

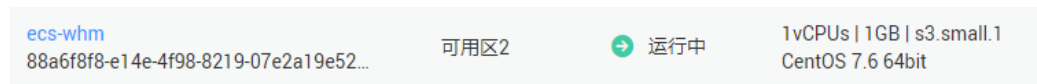
前提条件

- 确定云服务器操作系统类型，不同操作系统安装NFS客户端的命令不同。
- 已完成创建文件系统，并获取到文件系统的挂载地址。
- 选择与文件系统所属VPC相同的云服务器进行挂载。
- 云服务器上已配置了用于内网解析文件系统域名的DNS服务器的IP地址。

操作步骤

步骤1 已经成功在北京一可用区2创建一台CentOS系统的云服务器如ecs-whm，如[图4-2](#)所示。

图 4-2 成功创建云服务器



步骤2 以root用户登录弹性云服务器。安装NFS客户端。

1. 查看系统是否安装NFS软件包。

- CentOS、Red Hat、Oracle Enterprise Linux、SUSE、Euler OS、Fedora或OpenSUSE系统下，执行如下命令：

```
rpm -qa|grep nfs
```

- Debian或Ubuntu系统下，执行如下命令：

```
dpkg -l nfs-common
```

不同操作系统回显会有所不同，如果回显如下类似信息，说明已经成功安装NFS软件包，执行[步骤3](#)。如未显示，执行[步骤2.2](#)。

- CentOS、Red Hat、Euler OS、Fedora或Oracle Enterprise Linux系统下，回显如下类似信息：

```
libnfsidmap  
nfs-utils
```

- SUSE或OpenSUSE系统下，回显如下类似信息：

```
nfsidmap  
nfs-client
```

- Debian或Ubuntu系统下，回显如下类似信息：

```
nfs-common
```

2. 如果查看到未安装，根据不同的操作系统，执行不同命令。

📖 说明

执行以下命令前要求云服务器已连接到互联网，否则安装NFS客户端失败。

- CentOS、Red Hat、Euler OS、Fedora或Oracle Enterprise Linux系统下，执行如下命令：

```
sudo yum -y install nfs-utils
```

- Debian或Ubuntu系统下，执行如下命令：

```
sudo apt-get install nfs-common
```

- SUSE或OpenSUSE系统下，执行如下命令：

```
zypper install nfs-client
```

步骤3 执行如下命令，查看是否能解析文件系统挂载地址中的域名。如图4-3所示。

nslookup 文件系统域名

```
nslookup sfs-nas1.xx-xxxx-xx.xxxxxxxxxxxx.com
```

📖 说明

- 文件系统域名仅为域名，如：sfs-nas1.xxx.com。文件系统域名请从文件系统的挂载地址中获取，不需要输入整个挂载地址。
- 无法使用nslookup命令时，需要先安装bind-utils软件包。（可通过执行yum install bind-utils命令安装）
- 解析成功，执行**步骤4**。
- 解析失败，请先完成DNS服务器IP地址的配置再执行挂载文件系统的操作，具体配置操作请参见**配置DNS**。

图 4-3 解析域名

```
[root@ecs-whm ~]# rpm -qa|grep nfs
libnfsidmap-0.25-19.el7.x86_64
nfs-utils-1.3.0-0.61.el7.x86_64
[root@ecs-whm ~]# nslookup sfs-nas1.
-bash: nslookup: command not found
[root@ecs-whm ~]# yum install bind-utils
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
```

步骤4 执行如下命令，创建用于挂载文件系统的本地目录。此时记录本地目录名字，此处以root001为例。

mkdir 本地目录

```
mkdir root001
```

步骤5 执行如下命令，将文件系统挂载到云服务器上。文件系统目前仅支持NFSv3协议挂载到Linux云服务器，其中变量说明见**表4-2**。

```
mount -t nfs -o vers=3,timeo=600,noLOCK 挂载地址 本地目录
```

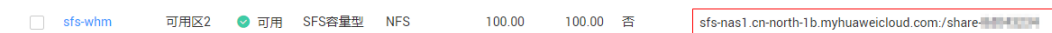
须知

已挂载文件系统的云服务器重启后，该云服务器上的挂载信息将会丢失，您可以通过在fstab文件中配置自动挂载来保证云服务器重启时自动挂载文件系统，具体操作请参见**自动挂载文件系统**。

表 4-2 参数说明

参数	说明
vers	文件系统版本，目前只支持NFSv3。取值：3。
timeo	NFS客户端重传请求前的等待时间(单位为0.1秒)。建议值：600。
lock/nolock	选择是否使用NLM协议在服务器上锁文件。当选择nolock选项时，锁对于同一主机的应用有效，对不同主机不受锁的影响。建议值：nolock。
挂载地址	SFS文件系统的格式为：文件系统域名:/路径，例如：example.com:/share-xxx。SFS Turbo文件系统的格式为：文件系统IP:/，例如192.168.0.0:/。 说明 x是数字或字母。 由于挂载地址名称较长，需要拉宽该栏以便完整显示。
本地目录	云服务上用于挂载文件系统的本地目录，例如“/local_path”。

图 4-4 挂载地址



步骤6 挂载完成后，执行如下命令，查看已挂载的文件系统。

```
mount -l
```

如果回显包含如下类似信息，说明挂载成功。

```
example.com:/share-xxx on /local_path type nfs (rw,vers=3,timeo=600,nolock,addr=)
```

步骤7 挂载成功后，参考下一个章节创建普通用户和子目录。

如果挂载失败或超时，请参考[故障排除](#)处理。

📖 说明

支持写入的单个文件最大容量为240TB。

----结束

4.3.2 为每个用户创建有读写权限的子目录

前提条件

- 已创建文件系统，并能通过root账号成功挂载到云服务器上。操作步骤请参考[创建文件系统本地目录](#)。
- 已获取到文件系统的挂载地址。

操作步骤

步骤1 以root账号登录弹性云服务器。

步骤2 使用以下命令在root账号下使用以下命令添加普通用户账号，下面以添加普通用户Tom为例。

```
adduser Tom  
passwd Tom
```

根据回显提示修改普通用户Tom的密码，创建成功后会自动创建用户Tom的主目录/home/Tom。

步骤3 添加普通用户Tom成功后，再在root的本地目录下使用以下命令为普通用户Tom创建子目录。

由[创建文件系统本地目录](#)章节的**步骤4**可知root的本地目录为root001，使用以下命令为普通用户Tom创建子目录Tom。其中root001需替换为实际的本地目录。

```
mkdir /root/root001/Tom
```

步骤4 使用以下命令将子目录Tom的读写权限赋予普通用户Tom。其中root001需替换为实际的本地目录。

```
chown Tom:Tom /root/root001/Tom
```

创建完后可以通过以下命令验证普通用户Tom是否已有子目录Tom的读写权限。如[图4-5](#)所示。

```
cd /home  
cd /root/root001  
ll
```

图 4-5 查询权限

```
[root@ecs-whm home]# cd /root/root001  
[root@ecs-whm root001]# ll  
total 4  
drwxr-xr-x 2 Tom Tom 4096 May 13 14:31 Tom
```

可以看到用户Tom已经获取了子目录Tom的读写权限。

步骤5 将root001/Tom子目录使用挂载命令mount挂载到Tom的主目录/home/Tom中。其中xx-xxxx-xx需要替换为文件系统所在区域，share-xxxx需要修改为实际的文件系统，第一个Tom需要修改为实际的子目录名称。如[图4-6](#)所示。

```
mount -t nfs sfs-nas1.xx-xxxx-xx.xxxxxxxxxx.com:/share-xxxx/Tom /home/Tom
```

图 4-6 挂载目录

```
[root@ecs-whm root001]# mount -t nfs sfs-nas1.xxxxxxxxxx.com:/share-xxxx/Tom /home/Tom  
[1282638.340788] Key type dns_resolver registered  
[1282638.381991] NFS: Registering the id_resolver key type  
[1282638.382851] Key type id_resolver registered  
[1282638.383689] Key type id_legacy registered
```

执行以下命令可以查看到该子目录已经成功挂载至Tom的主目录中。如[图4-7](#)所示。

```
df -h
```

图 4-7 查询挂载结果

```
[root@ecs-whm root001]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/vda1                  40G        1.9G   36G   5% /
devtmpfs                   486M         0   486M   0% /dev
tmpfs                      496M         0   496M   0% /dev/shm
tmpfs                      496M        26M   471M   6% /run
tmpfs                      496M         0   496M   0% /sys/fs/cgroup
tmpfs                      100M         0    100M   0% /run/user/0
sfs-nas1.                  100G         0    100G   0% /root/root001
sfs-nas1.                  100G         0    100G   0% /home/Tom
```

----结束

5 IDC 专线通过域名挂载 SFS Turbo HPC 型

操作场景


虚拟专用网络、云专线线下节点通过终端节点高速访问SFS Turbo HPC型文件系统。

SFS Turbo HPC型文件系统的挂载点使用的是域名，IDC环境如果不适用云上DNS则无法解析SFS Turbo的域名，导致此场景无法使用。

前提条件

- 本地数据中心已通过虚拟专用网络或者云专线与VPC连通。
- 虚拟专用网络VPN网关允许访问的VPC子网网段，需要包含SFS Turbo的网段。

操作步骤

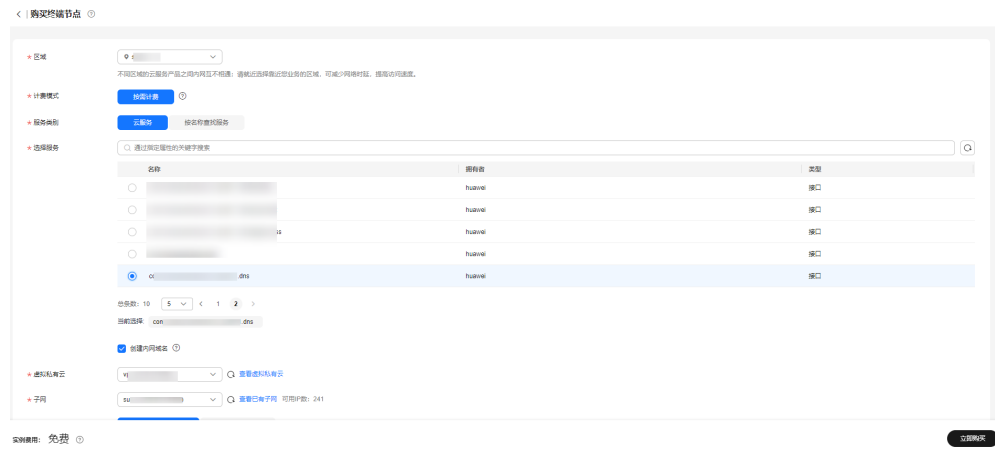
1. 在管理控制台左上角单击  图标，选择区域和项目。
2. 在“服务列表”中，选择“网络 > VPC终端节点”，进入终端节点页面。
3. 在左侧导航栏，选择“VPC终端节点 > 终端节点”。
4. 在终端节点界面，单击“购买终端节点”，创建连接DNS服务的终端节点。
5. 选择“云服务 > com.myhuaweicloud.region.dns”。

说明

region为购买的终端节点所在的区域。

6. 根据界面提示配置参数。

图 5-1 购买终端节点（云服务-接口型）



7. 选择“立即购买 > 提交”，完成创建。
8. 查看连接DNS服务的终端节点创建完成后返回的节点IP。
9. 在用户本地数据中心的DNS服务器配置相应的DNS转发规则，将解析SFS Turbo域名的请求转发到连接DNS服务的终端节点。

不同操作系统中配置DNS转发规则的方法不同，具体操作请参考对应DNS软件的操作指导。

本步骤以Unix操作系统，常见的DNS软件Bind为例介绍：

方式1：在/etc/named.conf文件中增加DNS转发器的配置，“forwarders”为连接DNS服务的终端节点的IP地址。

```
options {
forward only;
forwarders{ xx.xx.xx.xx;};
};
```

方式2：在/etc/named.rfc1912.zones文件中增加如下内容，“forwarders”为连接DNS服务的终端节点的IP地址。

```
zone "xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx.sfsturbo.internal" {
type forward;
forward only;
forwarders{ xx.xx.xx.xx;};
};
```

📖 说明

- 用户本地数据中心如果没有DNS服务器，需要将连接DNS服务的终端节点的节点IP增加到用户本地数据中心节点的/etc/resolv.conf文件中。
 - “xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx.sfsturbo.internal”表示需要访问的云上SFS Turbo的域名。
10. 配置线下节点到专线网关或VPN网关的DNS路由。

DNS终端节点IP的IP地址为xx.xx.xx.xx，所以需要将节点访问DNS的流量指向线下专线网关或VPN网关，然后走专线或VPN访问SFS Turbo。在线下节点配置永久路由，指定访问SFS Turbo的流量下一跳为线下专线网关或VPN网关的IP地址。

```
route -p add xx.xx.xx.xx mask 255.255.255.255 xxx.xxx.xxx.xxx
```

说明

- xx.xx.xx.xx为步骤8中的节点IP。
- xxx.xxx.xxx.xxx为线下专线网关或VPN网关的IP地址。
- 不同操作系统的Route命令格式存在差异，请以用户实际操作系统对应的Route命令格式为准。

测试 VPCEP 是否可以正常访问域名

```
dig 域名@xx.xx.xx.xx
```

说明

xx.xx.xx.xx为dns对应终端节点的IP。

如果返回可以拿到sfs Turbo的挂载点ip，则表明通过此终端节点可以解析SFS Turbo的域名。

6 面向 AI 场景使用 OBS+SFS Turbo 的存储加速实践

6.1 方案概述

应用场景

近年来，AI快速发展并应用到很多领域中，AI新产品掀起一波又一波热潮，AI应用场景越来越多，有自动驾驶、大模型、AIGC、科学AI等不同行业。AI人工智能的实现需要大量的基础设施资源，包括高性能算力，高速存储和网络带宽等基础设施，即“大算力、大存力、大运力”的AI基础大设施底座，让算力发展不要偏斜。

从过去的经典AI，到今天人人谈论的大模型，自动驾驶，我们看到AI模型的参数及AI算力规模呈现出指数级的爆发增长，对存储基础设施也带来全新的挑战。

- 高吞吐的数据访问挑战：**随着企业使用 GPU/NPU 越来越多，底层存储的 IO 已经跟不上计算能力，企业希望存储系统能提供高吞吐的数据访问能力，充分发挥 GPU/NPU 的计算性能，包括训练数据的读取，以及为了容错做的检查点（以下简称Checkpoint）保存和加载。训练数据的读取要尽量读得快，减少计算对 I/O 的等待，而 Checkpoint主要要求高吞吐、减少训练中断的时间。
- 文件接口方式的数据共享访问：**由于 AI 架构需要使用到大规模的计算集群（GPU/NPU服务器），集群中的服务器访问的数据来自一个统一的数据源，即一个共享的存储空间。这种共享访问的数据有诸多好处，它可以保证不同服务器上访问数据的一致性，减少不同服务器上分别保留数据带来的数据冗余等。另外以 AI 生态中非常流行的开源深度学习框架PyTorch为例，PyTorch默认会通过文件接口访问数据，AI算法开发人员也习惯使用文件接口，因此文件接口是最友好的共享存储访问方式。

📖 说明

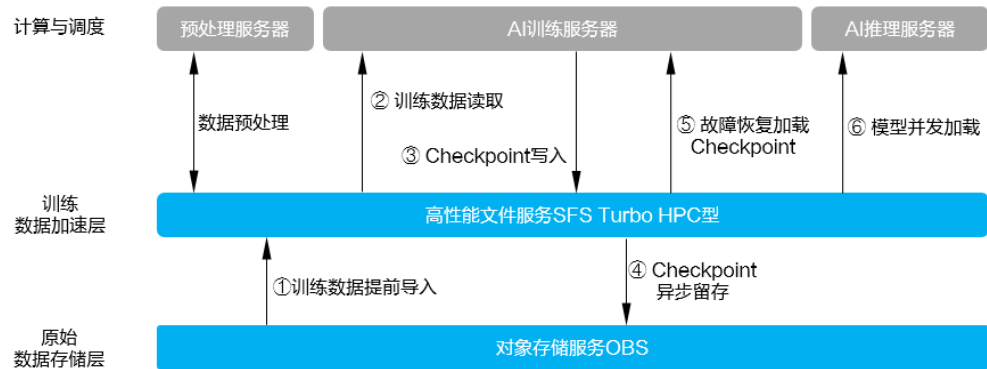
如果您想了解更多本方案相关信息，或在方案使用过程中存在疑问，可通过[方案咨询](#)渠道，寻求专业人员支持。

方案架构

针对AI训练场景中面临的问题，华为云提供了基于对象存储服务OBS+高性能文件服务SFS Turbo的AI云存储解决方案，如[图](#)所示，华为云高性能文件服务SFS Turbo HPC型

支持和OBS数据联动，您可以通过SFS Turbo HPC型文件系统来加速对OBS对象存储中的数据访问，并将生成的结果数据异步持久化到OBS对象存储中长期低成本保存。

图 6-1 基于 OBS+SFS Turbo 的华为云 AI 云存储解决方案



方案优势

华为云AI云存储解决方案的主要优势如下表所示。

表 6-1 华为云 AI 云存储解决方案的主要优势

序号	主要优势	详细描述
1	存算分离，资源利用率高	GPU/NPU算力和SFS Turbo存储解耦，各自按需扩容，资源利用率提升。
2	SFS Turbo高性能，加速训练过程	<ul style="list-style-type: none"> 训练数据集高速读取，避免GPU/NPU因存储I/O等待产生空闲，提升GPU/NPU利用率。 大模型TB级Checkpoint文件秒级保存和加载，减少训练任务中断时间。
3	数据导入导出异步化，不占用训练任务时长，无需部署外部迁移工具	<ul style="list-style-type: none"> 训练任务开始前将数据从OBS导入到SFS Turbo，训练过程中写入到SFS Turbo的Checkpoint数据异步导出到OBS，均不占用训练任务时长。 SFS Turbo和OBS存储服务之间数据直接导入导出，无需部署外部数据拷贝机器及工具。
4	冷热数据自动流动，降低存储成本	<ul style="list-style-type: none"> SFS Turbo支持自定义数据淘汰策略，冷数据自动分级到OBS，释放高性能存储空间用于接收新的热数据。 访问冷数据时SFS Turbo从OBS自动加载数据提升访问性能。
5	多AI开发平台、生态兼容	pytorch、mindspore等主流AI应用框架，kubernetes容器引擎、算法开发场景通过文件语义访问共享数据，无需适配开发。

📖 说明

如果您想了解更多本方案相关信息，或在方案使用过程中存在疑问，可通过[方案咨询](#)渠道，寻求专业人员支持。

6.2 资源和成本规划

本节介绍最佳实践中资源规划情况，包含以下内容：

表 6-2 资源和成本规划内容说明

维度	说明
资源规划	<ul style="list-style-type: none">● OBS：存放训练数据集、预训练模型等数据资源的桶，桶存储类别为“标准存储”，桶策略为“私有”。● SFS Turbo：文件系统类型为“HPC型”，存储类型请根据存储容量和性能需求选择，AI场景建议选择250MB/s/TiB及以上的存储类型。● ModelArts：AI开发平台，采用多机多卡分布式训练。● VPC：虚拟私有云和子网。● 算法及数据：准备AI训练需要的算法及数据集，如Swin-Transformer算法，及ImageNet21K数据集。 <p>说明 为了提供最佳加速性能，建议SFS Turbo HPC文件系统和ModelArts资源池就近选择在同一Region的同一个可用区（AZ）。</p>
成本规划	<ul style="list-style-type: none">● OBS费用：详见OBS计费说明。● SFS Turbo费用：详见SFS计费说明。● ModelArts费用：详见ModelArts计费说明。 <p>须知 本文提供的成本预估费用仅供参考，资源的实际费用以华为云管理控制台或价格计算器显示为准。</p>

📖 说明

如果您想了解更多本方案相关信息，或在方案使用过程中存在疑问，可通过[方案咨询](#)渠道，寻求专业人员支持。

6.3 操作流程

本档介绍面向AI场景如何使用OBS+SFS Turbo的存储加速，流程如[图6-2](#)所示。

图 6-2 面向 AI 场景使用 OBS+SFS Turbo 的存储加速方案步骤

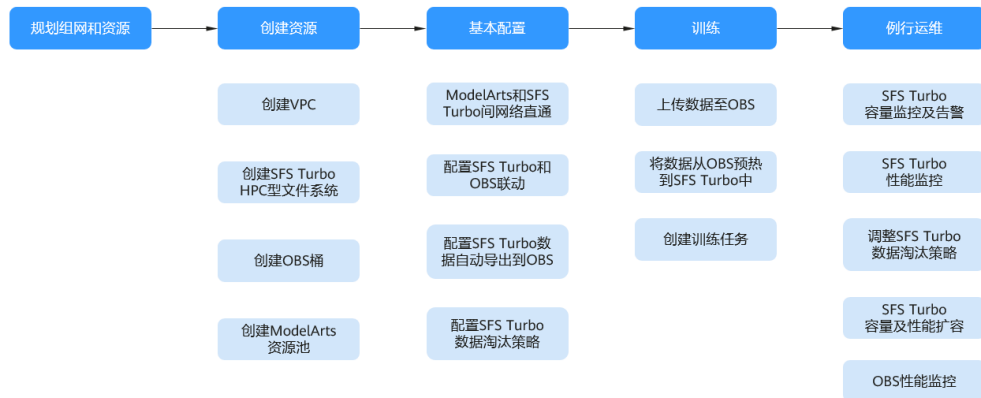


表 6-3 面向 AI 场景使用 OBS+SFS Turbo 的存储加速流程说明

序号	步骤	说明
1	规划组网和资源	此步骤请 提交工单 联系技术支持人员进行支撑配置。
2	创建资源	<ol style="list-style-type: none"> 创建VPC：创建1个虚拟私有云和子网。 创建SFS Turbo HPC型文件系统：创建1个SFS Turbo文件系统，文件系统类型选择“HPC型”，存储类型请根据存储容量和性能需求选择，AI场景建议选择250MB/s/TiB及以上的存储类型。 创建OBS桶：创建1个OBS桶，存储类别为“标准存储”，桶策略为“私有”。 创建ModelArts资源池：创建1个专属资源池。
3	基本配置	<ol style="list-style-type: none"> 配置ModelArts和SFS Turbo间网络直通。 <ol style="list-style-type: none"> 创建委托授权ModelArts云服务使用SFS Turbo。 配置ModelArts网络关联SFS Turbo。 配置SFS Turbo和OBS联动。 配置SFS Turbo数据自动导出到OBS桶。 配置SFS Turbo数据淘汰策略。
4	训练	<ol style="list-style-type: none"> 上传数据至OBS并预热到SFS Turbo中。 创建训练任务。
5	例行运维	使用OBS+SFS Turbo的存储加速方案的过程中，您可以进行采取以下运维措施，保证系统正常高效运行： <ul style="list-style-type: none"> SFS Turbo容量监控及告警。 SFS Turbo性能监控。 调整SFS Turbo数据淘汰策略。 SFS Turbo容量及性能扩容。 OBS性能监控。

📖 说明

如果您想了解更多本方案相关信息，或在方案使用过程中存在疑问，可通过[方案咨询](#)渠道，寻求专业人员支持。

6.4 实施步骤

6.4.1 创建资源

本最佳实践方案需要使用到VPC、SFS Turbo HPC型文件系统、OBS桶、ModelArts资源池资源。

📖 说明

为了提供最佳加速性能，建议SFS Turbo HPC文件系统和ModelArts资源池就近选择在同一Region的同一个可用区（AZ）。

创建 VPC

虚拟私有云可以为您构建隔离的、用户自主配置和管理的虚拟网络环境，操作指导请参考[创建虚拟私有云和子网](#)。

创建 SFS Turbo HPC 型文件系统

创建SFS Turbo文件系统，文件系统类型选择“HPC型”，操作指导请参考[创建SFS Turbo文件系统](#)。

创建 OBS 桶

创建OBS桶，存储类别为“标准存储”，桶策略为“私有”，操作指导请参考[创建OBS桶](#)。

创建 ModelArts 资源池

以常见的专属资源池为例，专属资源池提供独享的计算资源，可用于Notebook、训练作业、部署模型。专属资源池不与其他用户共享，更加高效。在使用专属资源池之前，您需要先创建一个专属资源池，操作指导请参考[创建专属资源池](#)。

6.4.2 基本配置

6.4.2.1 配置 ModelArts 和 SFS Turbo 间网络直通

创建委托授权 ModelArts 云服务使用 SFS Turbo

步骤1 使用IAM管理员账号登录[IAM控制台](#)。

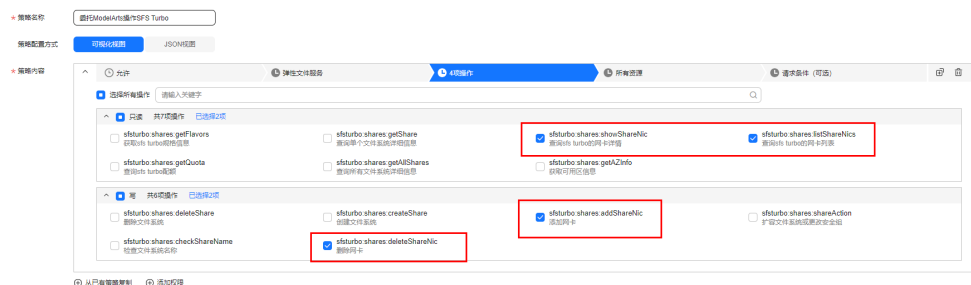
步骤2 在[IAM控制台](#)的左侧导航窗格中选择“权限管理 > 权限”。

步骤3 配置调用SFS Turbo接口的自定义策略：

1. 单击右上角的“创建自定义策略”，进入自定义策略配置页面。

2. 输入“策略名称”，用户可根据需要自定义，例如“委托modelarts操作SFS Turbo-1”
3. “策略配置方式”选择选择“可视化视图”或者“JSON视图”均可。如果选择“可视化视图”请跳转至[该步骤](#)，如果选择“JSON视图”请跳转至[该步骤](#)。
4. 在“策略内容”下配置策略。
 - a. 选择“允许”。
 - b. 选择云服务，勾选“弹性文件服务(SFSTurbo)”。
 - c. 选择“操作”，勾选只读操作“sfsturbo:shares:showShareNic”、“sfsturbo:shares:listShareNics”，勾选写操作“sfsturbo:shares:addShareNic”、“sfsturbo:shares:deleteShareNic”。
 - d. 选择“所有资源”

图 6-3 创建委托授权 ModelArts 云服务使用 SFS Turbo



5. 在“策略内容”区域，填写以下授权语句。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sfsturbo:shares:addShareNic",
        "sfsturbo:shares:listShareNics",
        "sfsturbo:shares:deleteShareNic",
        "sfsturbo:shares:showShareNic"
      ]
    }
  ]
}
```

6. 单击“确定”，完成自定义策略创建。

步骤4 配置调用VPC接口的自定义策略：

1. 单击右上角的“创建自定义策略”，进入自定义策略配置页面。
2. 输入“策略名称”，用户可根据需要自定义，例如“委托modelarts操作SFS Turbo-2”
3. “策略配置方式”选择选择“可视化视图”或者“JSON视图”均可。如果选择“可视化视图”请跳转至[该步骤](#)，如果选择“JSON视图”请跳转至[该步骤](#)。
4. 在“策略内容”下配置策略。
 - a. 选择“允许”。
 - b. 选择云服务，勾选“虚拟私有云 (VPC)”。
 - c. 选择“操作”，勾选写操作“vpc:ports:update”，勾选列表操作“vpc:ports:get”。

d. 选择“所有资源”

图 6-4 创建委托授权 ModelArts 云服务使用 VPC



5. 在“策略内容”区域，填写以下授权语句。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "vpc:ports:update",
        "vpc:ports:get"
      ]
    }
  ]
}
```

6. 单击“确定”，完成自定义策略创建。

步骤5 在IAM控制台页面的左侧导航窗格中选择“委托”，单击右上方的“创建委托”。

步骤6 在创建委托页面，设置“委托名称”，例如设置为“modelarts_agency”。

步骤7 “委托类型”选择“云服务”，在“云服务”中选择“ModelArts”，持续时间根据用户需要选取，单击“下一步”，进入给委托授权页面。

步骤8 勾选步骤1到4创建的自定义策略，给委托授权，单击“下一步”。

步骤9 选择授权范围方案，选择“所有资源”。

步骤10 单击“确定”，委托创建完成。

----结束

配置 ModelArts 网络关联 SFS Turbo

ModelArts网络关联SFS Turbo后，可直接在ModelArts的Notebook开发及训练环境中挂载SFS Turbo共享文件系统，并访问其中的数据。

步骤1 登录ModelArts管理控制台，创建网络并打通创建资源中创建的创建虚拟私有云和子网，详细步骤参见ModelArts网络。

步骤2 单击1中创建生成的资源池“网络”所在行的“更多”，选择“关联sfsturbo”。

步骤3 在“关联sfsturbo”弹窗中，选择创建资源中创建的SFS Turbo HPC型文件系统。

步骤4 选择完成后，单击“确定”创建关联。

----结束

说明

1. 使用过程中请不要解除关联，解除关联会导致ModelArts资源池无法访问SFS Turbo文件系统中的数据。
2. 一个SFS Turbo文件系统最多可关联1个网络。

6.4.2.2 配置 SFS Turbo 和 OBS 联动

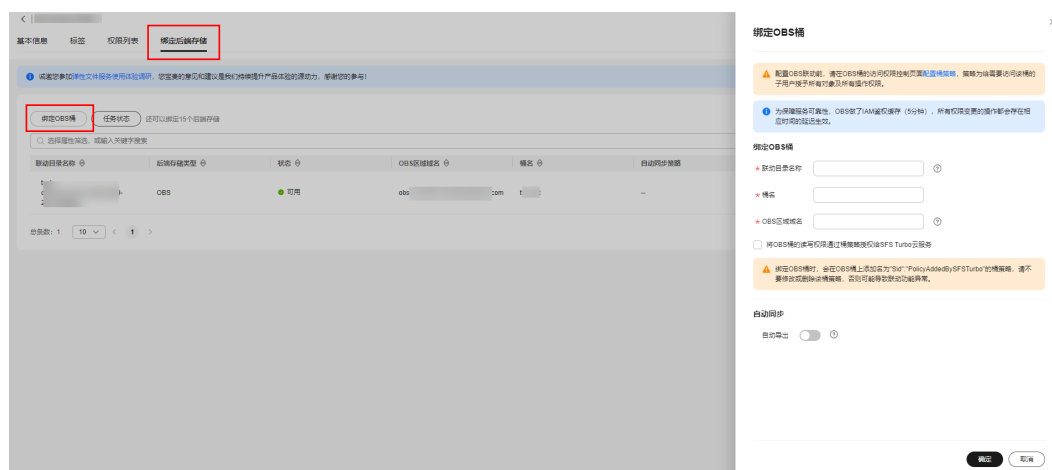
SFS Turbo HPC型文件系统支持无缝访问存储在对象存储OBS存储桶中的对象，您可以指定SFS Turbo内的文件目录与OBS对象存储桶进行关联。

步骤1 登录[SFS管理控制台](#)，在左侧导航窗格中选择“SFS Turbo”。

步骤2 在文件系统列表中，单击[创建资源](#)中创建的HPC型文件系统，进入文件系统详情页面。

步骤3 进入页签“绑定后端存储”，单击“绑定OBS桶”。

图 6-5 绑定 OBS 桶



步骤4 在右侧弹窗“绑定OBS桶”中，填写如下表所示参数。

表 6-4 绑定 OBS 桶配置参数

参数	含义	限制	配置后可编辑
联动目录名称	SFS Turbo文件系统根目录下会以该名称创建一个子目录，该目录将绑定对应的OBS桶，且该目录名称不能和已有目录重名。	<ul style="list-style-type: none"> 子目录名称不能重复，子目录名称长度不能超过63个字符。 子目录名称必须是文件系统根目录下不存在的目录名。 子目录名称不能是“.”或“..”。 	不支持

参数	含义	限制	配置后可编辑
桶名	OBS存储桶桶名。	<ul style="list-style-type: none">无法绑定不存在的存储桶。目前仅支持OBS存储桶，不支持OBS并行文件系统。	不支持
OBS区域域名	OBS区域域名，即OBS的终端节点。	OBS存储桶必须和SFS Turbo文件系统在同一个Region。	不支持

步骤5 勾选“将OBS桶的读写权限通过桶策略授权给SFS Turbo云服务”。

步骤6 单击“确定”，完成绑定。

步骤7 指定导入目录和文件的默认权限，请参考《高性能弹性文件服务API参考》的“[绑定后端存储](#)”和“[更新后端存储属性](#)”章节执行操作。

----结束

6.4.2.3 配置 SFS Turbo 数据自动导出到 OBS 桶

配置自动导出后，训练过程中周期性写入SFS Turbo文件系统的Checkpoint模型文件会自动以异步方式导出到关联的OBS桶中进行长期保存，无需手工导出，异步导出方式不会占用上层训练任务时间。

📖 说明

- SFS Turbo的自动导出功能当前处于**受限使用**状态，请[提交工单](#)联系技术支持人员进行配置。
- 文件导出速度受OBS服务的写入带宽上限影响，默认是16Gbit/s，如果大模型训练生成的Checkpoint文件过大、导出速度过慢，可[提交工单](#)申请调大OBS服务的写入带宽。

6.4.2.4 配置 SFS Turbo 数据淘汰策略

SFS Turbo HPC型文件系统绑定OBS后端之后，建议配置缓存数据淘汰功能。SFS Turbo会自动释放设定时间内没有访问过的文件数据内容，仅保留文件元数据，数据内容释放后不占用SFS Turbo文件系统上的存储空间，再次访问该文件时，将重新从OBS中加载文件数据内容。

步骤1 登录[SFS管理控制台](#)。

步骤2 在文件系统列表中，单击创建的HPC型文件系统名称，进入文件系统详情页面。

步骤3 在“基本信息”页签，设置冷数据淘汰时间。

图 6-6 设置冷数据淘汰时间

冷数据淘汰时间 (小时) ⓘ - ↻

----结束

📖 说明

只有已经导出到OBS且满足淘汰时间的数据才会被淘汰。

6.4.3 训练

6.4.3.1 上传数据至 OBS 并预热到 SFS Turbo 中

上传数据至 OBS

已经在OBS上创建好普通OBS桶，请参见[创建普通OBS桶](#)。

已经安装obsutil，请参考[下载和安装obsutil](#)。

步骤1 登录Imagenet数据集下载官网地址，下载Imagenet21k数据集：<http://image-net.org/>。

步骤2 下载格式转换后的annotation文件：[ILSVRC2021winner21k_whole_map_train.txt](#)和[ILSVRC2021winner21k_whole_map_val.txt](#)。

步骤3 下载完成后将上述3个文件数据上传至OBS桶中的imagenet21k_whole文件夹中。上传方法请参考[obsutil命令行工具使用指导](#)。

----结束

📖 说明

OBS针对不同场景提供了多种数据上云方案，您可根据数据量、耗时、费用等需求选择适合的方案上传数据至OBS，更多内容请参考[数据上云方案](#)。

将数据从 OBS 预热到 SFS Turbo 中

SFS Turbo HPC型文件系统绑定OBS桶后，可以使用数据预热功能，以减少后续训练首次访问数据耗时。

训练任务开始前可通过数据预热功能将文件元数据和数据内容全部从OBS导入到SFS Turbo高性能文件存储中，数据预热功能的具体操作请参考[创建SFS Turbo 和 OBS 之间的联动任务](#)。

📖 说明

1. 您可通过[查询联动任务详情接口](#)查看导入任务的完成状态。
2. 如果您觉得数据集规模较小或数据集变化不太频繁，不需要通过数据联动来做数据导入导出，您可借助外部工具将数据从OBS迁移到SFS Turbo中，操作指导请参考[OBS和SFS之间的数据迁移](#)，推荐使用obsutil工具。

6.4.3.2 创建训练任务

基于SFS Turbo共享文件存储创建ModelArts训练任务。

步骤1 登录[ModelArts管理控制台](#)。

步骤2 在左侧导航栏中选择“训练管理 > 训练作业”，进入“训练作业”列表。

步骤3 单击右上角的“创建训练作业”，进入“创建训练作业”页面，在该页面填写训练作业相关参数信息。

步骤4 填写训练作业相关参数信息，以下配置项请按要求填写，其余参数配置请参考[创建训练作业](#)根据您的自身情况选择。

- 资源池：专属资源池，选择[创建资源](#)中创建的ModelArts资源池。
- SFS Turbo：增加挂载配置，选择[创建资源](#)中创建的SFS Turbo HPC型文件系统。

步骤5 单击“提交”，完成训练作业的创建。

----结束

训练作业创建完成后，后台将自动完成容器镜像下载、代码目录下载、执行启动命令等动作。训练作业一般需要运行一段时间，根据您的训练业务逻辑和选择的资源不同，训练时长将持续几十分钟到几小时不等。要查看训练作业实时情况，您可以前往训练作业列表，查看训练作业的基本情况。

6.4.4 例行维护

SFS Turbo 容量监控及告警

如果SFS Turbo HPC型文件系统存储空间被写满，会影响业务运行，您可以在CES云监控服务上监控SFS Turbo文件系统的容量使用情况，并创建告警规则，当容量使用率超过一定阈值，可以发送邮件、短信等告警到运维人员。当收到容量监控告警时，您需要及时清理SFS Turbo存储空间、或缩短冷数据淘汰时间加速冷数据淘汰、或对SFS Turbo进行空间扩容。详情可参见[SFS Turbo监控指标说明](#)和[创建告警规则](#)。

SFS Turbo 性能监控

您可以在CES云监控服务上监控和SFS Turbo文件系统的性能使用情况。当AI算力集群规模变大，大模型参数量变大，导致Checkpoint读加载时间变长时，或训练数据集加载由于存储读写带宽不足导致拖慢AI训练时，您可以对SFS Turbo进行性能扩容，以缩短数据加载时长。详情可参见[SFS Turbo监控指标说明](#)和[创建告警规则](#)。

调整 SFS Turbo 数据淘汰策略

操作指导请参考[配置SFS Turbo数据淘汰策略](#)。

SFS Turbo 容量及性能扩容

当SFS Turbo存储空间不足时，您可以对SFS Turbo存储空间进行容量扩容。扩容操作步骤请参考[容量调整](#)。

SFS Turbo HPC型是按每TB单位容量来提供一定的带宽吞吐，因此当SFS Turbo HPC性能不足时，需要通过容量扩容来提高性能吞吐。

OBS 性能监控

您可以在CES云监控服务上监控SFS Turbo关联的OBS桶的性能使用情况，SFS Turbo和OBS之间的数据导入导出速度会受OBS服务的读写带宽上限QoS影响，默认是16Gbit/s，如果导入导出速度受到OBS读写带宽上限影响，可[提交工单](#)联系技术支持人员申请调大OBS服务的读写带宽。

6.5 常见问题

- **可以只使用SFS Turbo HPC型文件系统支撑AI训练吗？**
当数据规模较小，不存在冷热数据分级降本诉求，又希望能方便快捷的构建AI训练系统时，可以选择只使用SFS Turbo高性能文件存储支撑AI训练。
- **可以基于OBS对象存储支撑AI自动驾驶、大模型训练吗？**
OBS为容量型存储，在时延、带宽等存储性能上无法满足高性能AI训练，建议使用SFS Turbo HPC型高性能文件系统加速AI训练任务，训练速度加快可以节省AI算力费用。
- **文件系统使用空间不足，可以扩容吗？**
SFS Turbo文件系统支持在线扩容，扩容过程中挂载文件系统可能失败，建议业务低峰期扩容。

7 提升 Linux 操作系统单客户端访问 SFS Turbo 的性能

操作场景

nconnect是Linux操作系统下NFS客户端的挂载参数，开启后支持NFS客户端与服务端之间建立多条TCP连接来提升吞吐性能。

前提条件

Linux内核为5.3及以上版本。

📖 说明

- 当客户端与服务端之间网络延迟较高（例如在线下IDC访问云上NFS文件系统和跨Region挂载NFS文件系统的场景下），且您需要达到较高的带宽或IOPS时，建议开启该功能；
- 开启nconnect后对于单并发、小IO数据块等场景会有一定的时延增加，当您的业务为小并发的延迟敏感型时，不建议开启该功能；
- nconnect功能开启后，可以提升NFS客户端与服务端之间的吞吐性能，最终性能阈值受限于文件系统服务端的能力及客户端网卡或CPU等资源能提供的能力。

操作步骤

在挂载NFS文件系统的命令中，增加nconnect参数，建议设置nconnect=4。示例如下：

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp,nconnect=4 挂载地址 本地路径
```