Object Storage Service

Parallel File System Feature Guide

Issue 07

Date 2024-03-19

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei 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.

Contents

1 Introduction

1.1 About Parallel File System

Parallel File System (PFS) is a high-performance semantic file system provided by OBS. It features access latency in milliseconds, TB/s-level bandwidth, and millions of IOPS.

You can access data in a PFS via OBS APIs.

1.2 Application Scenarios

PFS is highly compatible, scalable, and reliable, and delivers amazing performance.

It is mainly used in the following scenarios:

Big data: log analysis, content recommendation, operation reports, user profiling, and interactive analysis

1.3 Constraints

Operations

- An existing OBS bucket cannot be changed to a parallel file system. For details about how to create a parallel file system, see **Creating a Parallel File System**.
- Custom domain names of parallel file systems cannot be configured on OBS Console. You can configure them using APIs by referring to **Configuring a Custom Domain Name**.

Functions

- **Image processing** currently cannot be used to process (such as downsize, resize, or watermark) images stored in parallel file systems.
- Server-side encryption is not supported.
- Cross-region replication is not supported.

- Versioning is not supported.
- Bucket inventory is not supported.
- Static website hosting is not supported.
- Configuration of default storage class for a parallel file system is not supported.
- Deep Archive storage is not supported.
- A parallel file system can be mounted to multiple Linux servers to implement concurrent reads, but this is not recommended for concurrent writes.

Performance

- A parallel file system provides a maximum bandwidth of 10 MB/s per TB by default.

Naming

- In a parallel file system, a file name cannot contain two consecutive slashes (//). For example, if you name a file as **test//123.txt**, an error will be reported.

1.4 Using PFS

You can use the console and REST APIs to process files stored in PFS anytime, anywhere, and quickly obtain the processed files. PFS supports both POSIX and OBS APIs, so you can process files the same way you process objects. There is flexible conversion supported between objects and files.

You can use PFS in the following ways:

NOTE

Access permissions for OBS buckets also apply to parallel file systems. Before using a parallel file system, ensure that you have the required permissions to access OBS buckets.

Table 1-1 How to use PFS

Method	Function	Reference
Console	On the console, you can create parallel file systems and manage them.	**Creating a Parallel File System**
OBS API	Use parallel file systems by calling OBS APIs.	**Supported APIs**
Other OBS tools	PFS allows for conversion between objects and files, so you can also use other OBS tools (such as obsutil and OBS Browser+) to access PFS.	**OBS Tools**

1.5 Pricing

Parallel file systems support both pay-per-use and yearly/monthly (resource packages) billing modes. For details about resource packages, see **Resource Package Overview**.

For details about PFS pricing, see **Product Pricing Details**.

1.6 Features

1.6.1 Lifecycle Management

The application scenarios and main functions of lifecycle management for PFS is consistent with those of OBS. For more information, see **Lifecycle Management**.

For details about the SDK reference for lifecycle management, see **SDK Overview**.

Differences from the Object Lifecycle Management

- You can use the lifecycle management function to manage files. Lifecycle management cannot transition a folder to the Archive storage class, but it can delete an empty folder upon expiration.
- You can use APIs to configure lifecycle rules to expire and then delete the specified files or to transition the specified files to the Archive or Infrequent Access storage class. Currently, OBS Console allows only deletion upon expiration and transition to the Archive storage class. PFS does not support versioning, so lifecycle rules (including deletion upon expiration, and transition to the Archive or Infrequent Access storage class) that are related to versioning cannot be applied.
- If direct reading is enabled for a parallel file system, you can read files stored in the Archive storage class without restoring them first.
- A maximum of 20 lifecycle rules can be configured for a parallel file system.
- A lifecycle rule adjusts its baseline time to keep in line with when relevant files have been most recently updated.
- Lifecycle rules cannot transition files to the Deep Archive storage.
- After a lifecycle rule is configured for a parallel file system, there are limits on how many directories that the rule can be applied to. If the configuration exceeds the limit, the lifecycle rule execution will be prolonged.

  a. There can be no more than 100,000 level-1 subdirectories in each directory.

  b. There can be no more than 10 million subdirectories (folders) matching the prefix defined in the rule in total.

  c. The total number of files matching the prefix defined in the rule cannot exceed 30 million.

Other Descriptions

- If you rename a file or a folder, and the renamed file or files in the renamed folder meet the requirements of a lifecycle rule, the baseline time of the lifecycle rule will be the time when the content data of the files was most recently updated, not when the file was renamed. In addition, the effective time of the lifecycle rule may be delayed for up to 7 days.
- For a file copy on a client, its expiration time or time when it is transitioned to the Archive storage class is based on when the file was most recently copied.
  - For example, if a file, **src.txt**, was created on January 1, 2019, and was then copied to the **des.txt** file by running the **cp -a src.txt des.txt** command on September 1, 2019. Then the lifecycle rule is based on September 1, 2019.
- In the lifecycle rule of a parallel file system, directories in the file system are periodically scanned, and then deleted if they meet the expiration conditions. Scan intervals (usually seven days) vary depending on cluster configurations. A scan starts from the deepest directory, and the scanned empty directories that meet the expiration conditions will be deleted, but those non-empty directories will not be processed. For this mechanism, a single-level directory that meets the expiration conditions will be deleted 0 to 7 days after it has been emptied. Accordingly, a two-level directory will be deleted 0 to 14 days after it has been emptied. Each time a directory level is added, the waiting time for deletion increases by seven days.

1.6.2 Permission Configurations

Application scenarios and main functions of permission configurations for parallel file systems are the same as those for object storage. For more information, see **OBS Permission Control**.

Main Permission Configuration Differences

To exactly match a specific directory, the resource path in the policy must end with a slash (/). When checking permissions, parallel file systems consider objects as directories. If the object identifier does not end with a slash (/), the system will add a slash (/) to the end of the object identifier and then performs a policy matching.

IAM Permission Examples

Example 1: Grant a user the permissions required to download **dir_1**, excluding its subdirectories.

The policy is configured as follows. The resource path ends with a slash (/). In such case, a success response can be returned when **dir_1** or **dir_1/** is contained in the URL of a head request.

Note: This policy is not applied to subdirectories or files in **dir_1**. Therefore, a failure response will be returned if a head request is sent to **dir_1/file1**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject",
            ],
            "Resource": [
                "obs:*:*:object:examplebucket/dir_1/",
            ]
        }
    ]
}
```

Example 2: Grant a user the permissions required to download **dir_1** and its subdirectories.

The policy is configured as follows. The resource path uses prefix match and ends with a wildcard (*). In such case, a success response can be returned when a head request is sent to **dir_1/file1**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:object:GetObject",
            ],
            "Resource": [
                "obs:*:*:object:examplebucket/dir_1/*",
            ]
        }
    ]
}
```

Bucket Policy Examples

Example 1: Grant a user the permissions required to download **dir_1**, excluding its subdirectories.

The policy is configured as follows. The resource path ends with a slash (/). In such case, a success response can be returned when **dir_1** or **dir_1/** is contained in the URL of a head request.

```
{
    "Statement":[
    {
      "Sid":"test",
      "Effect":"Allow",
      "Principal": {"ID": ["domain/b4bf1b36d9ca43d984fbcb9491b6fce9:user/
71f3901173514e6988115ea2c26d1999"]},
      "Action":["*"],
      "Resource":[
        "examplebucket/dir_1/",
      ]
    }
  ]
}
```

2 Managing Parallel File Systems on OBS Console

2.1 Creating a Parallel File System

You can create a parallel file system on OBS Console.

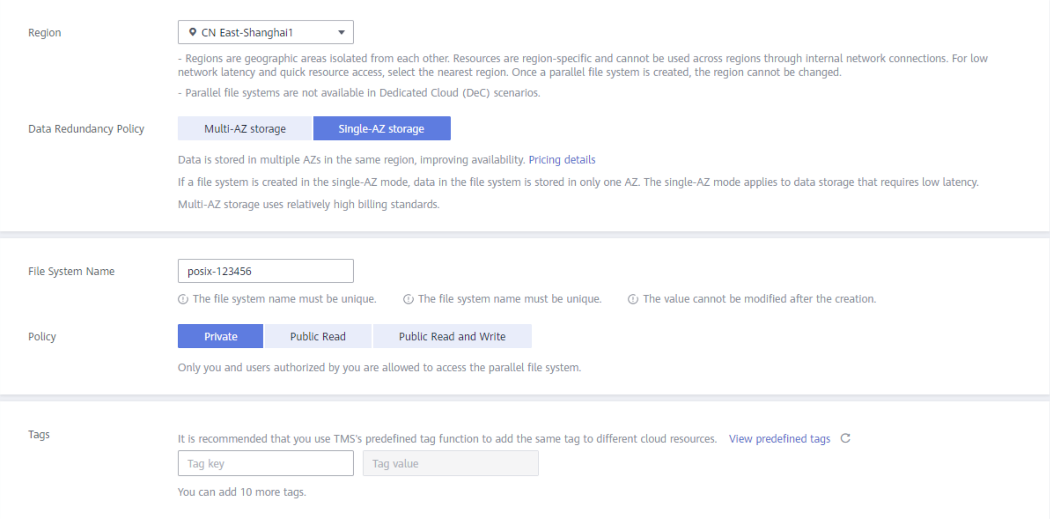
Procedure

Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the navigation pane, select **Parallel File System**.

Step 3 In the upper right corner of the page, click **Create Parallel File System**.

Figure 2-1 Creating a parallel file system



Region: CN East-Shanghai1

- Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region. Once a parallel file system is created, the region cannot be changed.

- Parallel file systems are not available in Dedicated Cloud (DeC) scenarios.

Data Redundancy Policy: Multi-AZ storage [✓ Single-AZ storage]

Data is stored in multiple AZs in the same region, improving availability. Pricing details

If a file system is created in the single-AZ mode, data in the file system is stored in only one AZ. The single-AZ mode applies to data storage that requires low latency.

Multi-AZ storage uses relatively high billing standards.

File System Name: posix-123456

The file system name must be unique. The file system name must be unique. The value cannot be modified after the creation.

Policy: [✓ Private] Public Read Public Read and Write

Only you and users authorized by you are allowed to access the parallel file system.

Tags

It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

You can add 10 more tags.

Step 4 Select a region and enter a name for the parallel file system.

NOTE

- Once a parallel file system is created, its name cannot be changed.
- URLs do not support uppercase letters and cannot distinguish between names containing uppercase or lowercase letters. For example, if you attempt to access the parallel file system **MyFileSystem** using a URL, the file system name will be resolved to **myfilesystem**, causing an access error. For this reason, a parallel file system name can contain only lowercase letters, digits, periods (.), and hyphens (-).

Step 5 Configure a data redundancy policy. If you select multi-AZ storage, data will be stored in multiple AZs for a higher reliability.

NOTE

Once a parallel file system is created, you cannot change the data redundancy policy, so take care when selecting at data redundancy policy.

Step 6 Configure a policy. You can select **Private**, **Public Read**, or **Public Read and Write** for the parallel file system.

Step 7 Configure direct reading. With direct reading enabled, you can directly download objects in the Archive storage class without restoring them first. Direct reading is a billable function. For details, see **Product Pricing Details**.

Step 8 (Optional) Add tags. Tags are used to identify parallel file systems in OBS, for the purpose of classification. Each tag is represented by one key-value pair. For details about how to add a tag, see **Tags**.

Step 9 (Optional) Buy storage packages. By default, you are billed on a pay-per-use basis for using parallel file systems. You can also purchase **storage packages** to save more. After selecting a required package, go to the **Confirm** page to complete the purchase.

Storage packages can also be purchased after the parallel file system is created.

Step 10 Confirm the settings at the bottom of the page and click **Create Now**.

Step 11 View the file system you created just now in the parallel file system list.

Then, you can use the parallel file system the same way you use a bucket. For details about how to use PFS, see **Using PFS**.

----End

3 Managing Parallel File Systems by APIs

3.1 Supported APIs

This section focuses on the compatibility of parallel file systems and existing OBS APIs, and provides a description of APIs that are partly compatible.

For details about the OBS APIs, see **Object Storage Service API Reference**.

API Operations on Buckets

Table 3-1 API operations on buckets

API	PFS Compatible	Differences
Listing buckets	Yes	The **x-obs-bucket-type:POSIX** header field is required for obtaining the list of parallel file systems.
Creating a bucket	Yes	The **x-obs-fs-file-interface:Enabled** header field is required for creating a parallel file system.
Listing objects in a bucket	Yes	-
Obtaining bucket metadata	Yes	-
Obtaining bucket region locations	Yes	-
Deleting a bucket	Yes	-

API Operations on Advanced Bucket Settings

Table 3-2 API operations on advanced bucket settings

API	PFS Compatible	Differences
Configuring a bucket policy	Yes	-
Obtaining bucket policy information	Yes	-
Deleting a bucket policy	Yes	-
Configuring a bucket ACL	Yes	-
Obtaining bucket ACL information	Yes	-
Configuring logging for a bucket	Yes	-
Obtaining a bucket logging configuration	Yes	-
Configuring bucket lifecycle rules	Yes	-
Obtaining bucket lifecycle configuration	Yes	-
Deleting bucket lifecycle rules	Yes	-
Configuring versioning for a bucket	No	-
Obtaining bucket versioning status	No	-
Configuring event notification for a bucket	Yes	-

API	PFS Compatible	Differences
Obtaining the event notification configuration of a bucket	Yes	-
Configuring storage class for a bucket	No	-
Obtaining bucket storage class information	No	-
Configuring cross-region replication for a bucket	No	-
Obtaining the cross-region replication configuration of a bucket	No	-
Deleting the cross-region replication configuration of a bucket	No	-
Configuring tags for a bucket	Yes	-
Obtaining bucket tags	Yes	-
Deleting bucket tags	Yes	-
Configuring bucket storage quota	Yes	-
Querying bucket storage quota	Yes	-
Querying information about used space in a bucket	Yes	-

API	PFS Compatible	Differences
Configuring bucket inventories	No	-
Obtaining bucket inventories	No	-
Listing bucket inventories	No	-
Deleting bucket inventories	No	-
Configuring a custom domain name for a bucket	Yes	-
Obtaining the custom domain name of a bucket	Yes	-
Deleting a custom domain name of a bucket	Yes	-
Configuring bucket encryption	No	-
Obtaining bucket encryption configuration	No	-
Deleting the encryption configuration of a bucket	No	-
Configuring the direct reading policy for Archive objects in a bucket	Yes	-
Obtaining the direct reading policy for Archive objects in a bucket	Yes	-

API	PFS Compatible	Differences
Deleting the direct reading policy for Archive objects in a bucket	Yes	-

API Operations for Static Website Hosting

Table 3-3 API operations for static website hosting

API	PFS Compatible	Differences
Configuring static website hosting for a bucket	No	-
Obtaining the static website hosting configuration of a bucket	No	-
Deleting the static website hosting configuration of a bucket	No	-
Configuring bucket CORS	Yes	-
Obtaining the CORS configuration of a bucket	Yes	-
Deleting the CORS configuration of a bucket	Yes	-
OPTIONS buckets	No	-
OPTIONS objects	No	-

API Operations on Objects

Table 3-4 API operations on objects

API	PFS Compatible	Differences
PUT objects	Yes	- Headers not supported: **x-obs-storage-class**, **x-obs-website-redirect-location**, **x-obs-server-side-encryption**, **x-obs-server-side-encryption-kms-key-id**, **x-obs-server-side-encryption-customer-algorithm**, **x-obs-server-side-encryption-customer-key**, **x-obs-server-side-encryption-customer-key-MD5**, **success-action-redirect**, and **x-obs-expires** - Objects uploaded using this API cannot be directly stored in the Infrequent Access or Archive storage class and are stored in the Standard storage class by default. You can later change the storage class by using a lifecycle rule or modifying the metadata.
POST objects	Yes	Headers not supported: **x-obs-storage-class**, **x-obs-website-redirect-location**, **x-obs-server-side-encryption**, **x-obs-server-side-encryption-kms-key-id**, **x-obs-server-side-encryption-customer-algorithm**, **x-obs-server-side-encryption-customer-key**, **x-obs-server-side-encryption-customer-key-MD5**, **success-action-redirect**, and **x-obs-expires**
Copying objects	Yes	Data can only be replicated between a parallel file system and a bucket if the parallel file system and the bucket are in the same cluster.
Obtaining object content	Yes	-
Obtaining object metadata	Yes	-
Deleting objects	Yes	-
Batch deleting objects	Yes	-
Restoring Archive objects	Yes	-

API	PFS Compatible	Differences
Appending objects	No	-
Configuring object ACL	Yes	-
Obtaining object ACL information	Yes	-
Modifying object metadata	Yes	In a parallel file system, the storage class of a directory cannot be changed. To change the storage class of a file in the directory, modify the metadata of the file or use a lifecycle rule to change the storage class of files in batches.
Modifying an object	Yes	This is a PFS only API, and is not supported by OBS buckets.
Truncating an object	Yes	This is a PFS only API, and is not supported by OBS buckets.
Renaming an object	Yes	This is a PFS only API, and is not supported by OBS buckets.

API Operations for Multipart Tasks

Table 3-5 API operations for multipart tasks

API	PFS Compatible	Differences
Listing initialized multipart tasks in a bucket	Yes	-
Initiating multipart upload tasks	Yes	-
Uploading parts	Yes	-
Copying parts	Yes	Copying parts is not supported for an appended file.
Listing uploaded parts	Yes	-
Merging parts	Yes	-

API	PFS Compatible	Differences
Canceling multipart tasks	Yes	-

4 FAQs

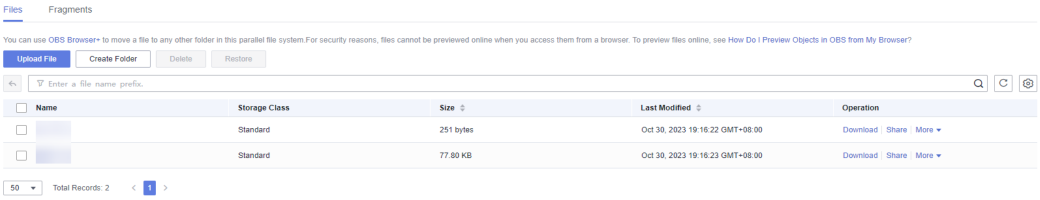
4.1 Why Is 256 TB Displayed After I Mounted a Parallel File System?

There is no limit on the capacity of a parallel file system. When you run the **df -h** command on a client, the system returns 256 TB just for display purposes. 256 TB itself does not have any meaning.

4.2 Can I Sort Files in the File List?

In the file list, you can click ↕ next to the size or last modification time to sort files. File search by last modification time can only display the first 1,000 records. If there are more than 5,000 files in a parallel file system, the files are sorted in alphabetical order and can be searched only by file name prefix.

Figure 4-1 File list



You can use OBS Browser+ to move a file to any other folder in this parallel file system.For security reasons, files cannot be previewed online when you access them from a browser. To preview files online, see How Do I Preview Objects in OBS from My Browser?

□	Name	Storage Class	Size	Last Modified	Operation
□	[redacted]	Standard	251 bytes	Oct 30, 2023 19:16:22 GMT+08:00	Download | Share | More
□	[redacted]	Standard	77.80 KB	Oct 30, 2023 19:16:23 GMT+08:00	Download | Share | More

A Change History

Release Date	What's New
2024-03-19	This is the seventh official release. This issue incorporates the following change: ● Added naming restrictions in **Constraints**.
2024-02-28	This is the sixth official release. This issue incorporates the following change: ● Added the content related to Deep Archive storage (under limited beta testing).
2023-10-30	This is the fifth official release. This issue incorporates the following change: ● Added **Can I Sort Files in the File List?**
2023-07-10	This is the fourth official release. This issue incorporates the following change: ● Updated the content under **Other Descriptions** in section "Lifecycle Management."
2023-06-16	This is the third official release. This issue incorporates the following change: ● Added **Why Is 256 TB Displayed After I Mounted a Parallel File System?**
2020-07-30	This is the second official release. This issue incorporates the following change: ● Added the lifecycle management feature.
2019-11-22	This is the first official release.