Object Storage Service

Parallel File System Feature Guide

Date **2024-01-29**

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**.
- Parallel file systems do not support quota configuration. By default, there is no quota limit.

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.
- Versioning is not supported.
- Static website hosting is not supported.

- Changing file storage class is not supported.
- Configuration of default storage class for a parallel file system is not supported.

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**

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

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 (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 the "Tags" section in the *Object Storage Service User Guide*.

Step 8 Confirm the settings at the bottom of the page and click **Create Now**.

Step 9 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 the *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	No	-
Obtaining bucket lifecycle configuration	No	-
Deleting bucket lifecycle rules	No	-
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 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	-
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	-

API	PFS Compatible	Differences
Obtaining bucket encryption configuration	No	-
Deleting the encryption configuration of a bucket	No	-

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	No	-
Obtaining the CORS configuration of a bucket	No	-
Deleting the CORS configuration of a bucket	No	-
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 Cold objects	No	-

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	-

A Change History

Release Date	What's New
2024-01-29	This is the second official release. This issue incorporates the following change: ● Adapted to the new OBS Console.
2021-02-07	This is the first official release.