

Object Storage Service

Parallel File System Feature Guide (Kuala Lumpur Region)

Issue 01
Date 2022-08-15



Copyright © Huawei Technologies Co., Ltd. 2022. All rights reserved.

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

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei 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
1.1 About Parallel File System.....	1
1.2 Application Scenarios.....	1
1.3 Constraints.....	1
1.4 Using PFS.....	2
2 Managing Parallel File Systems on OBS Console.....	3
2.1 Creating a Parallel File System.....	3
3 Managing Parallel File Systems by APIs.....	5
3.1 Supported APIs.....	5
4 Managing Parallel File Systems by obsfs.....	11
A Change History.....	12

1 Introduction

1.1 About Parallel File System

Parallel File System (PFS), a sub-product of OBS, is a high-performance file system, with only milliseconds of access latency. PFS can support terabytes of bandwidth and can handle millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

You can access data in a parallel file system using standard OBS APIs. It also supports data read and write through obsfs, a PFS client that supports POSIX. obsfs can be deployed on a Linux ECS, and then you can use obsfs to mount a parallel file system to that server. Once mounted, PFS functions like a local file system. You can manage the PFS online, including creating, deleting, renaming files and folders, or modifying files.

1.2 Application Scenarios

PFS is highly compatible, scalable, and reliable, and delivers amazing performance. PFS is well suited to HPC and media asset archiving scenarios.

The main application scenarios of PFS are as follows:

Video surveillance: public security, business environments, and households

Video on demand (VOD): over-the-top (OTT) delivery and media assets library

HPC: DNA sequencing and computer-aided engineering (CAE) scenarios in manufacturing

Big data: log analysis, content recommendation, operation report, user profile, and interactive analysis

1.3 Constraints

Operational Limitations

- 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.

Function Restrictions

- 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

PFS provides the console, client (obsfs), and REST APIs for managing and accessing data. obsfs supports POSIX, so that your applications can be seamlessly interconnected with PFS. No modifications to your applications are needed. You can 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 you can also view and manage your file systems and files.	Creating a Parallel File System
obsfs	obsfs is a client developed and designed for PFS. It supports POSIX and can mount parallel file systems to cloud servers. obsfs is recommended for accessing PFS.	Object Storage Service Tools Guide (obsfs)
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 the console.

Procedure

- Step 1** On the homepage of the console, click **Service List** on the top navigation menu, and choose **Storage > Object Storage Service**.
- Step 2** In the navigation pane, select **Parallel File System**.
- Step 3** Click **Create Parallel File System** in the upper right corner of the page. The page for creating a parallel file system is displayed.
- 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 policy. You can select **Private**, **Public Read**, or **Public Read and Write** for the parallel file system.
- Step 6** (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 *Object Storage Service User Guide*.
- Step 7** Confirm the settings at the bottom of the page and click **Create Now**.
- Step 8** 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	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	The following header fields are 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
POST objects	Yes	The following header fields are 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	-
Appending objects	No	-
Configuring object ACL	Yes	-
Obtaining object ACL information	Yes	-

API	PFS Compatible	Differences
Modifying object metadata	Yes	-
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	-
Canceling multipart tasks	Yes	-

4 Managing Parallel File Systems by obsfs

obsfs is a file system tool provided by Object Storage Service (OBS) for mounting OBS parallel file systems to Linux operating systems. It uses FUSE and is compatible with POSIX. With obsfs, you can easily access the practically infinite storage of OBS as easily as accessing a local file system.

obsfs is a great option if you are used to storing data locally but your data is now stored in OBS.

For more information about obsfs, see [Introduction to obsfs](#).

A Change History

Release Date	What's New
2022-08-15	This is the first official release.