



# 天翼云对象存储

## 并行文件系统特性指南

天翼云科技有限公司

# 目 录

---

<b>1 简介</b> .....	<b>3</b>
1.1 什么是并行文件系统 .....	3
1.2 应用场景 .....	3
1.3 约束限制 .....	3
1.4 使用方式 .....	4
<b>2 控制台方式</b> .....	<b>5</b>
2.1 创建并行文件系统 .....	5
<b>3 API 方式</b> .....	<b>6</b>
3.1 支持的 API 列表 .....	6
<b>A 修订记录</b> .....	<b>10</b>

# 1 简介

## 1.1 什么是并行文件系统

并行文件系统（Parallel File System）是对象存储服务（Object Storage Service, OBS）提供的一种经过优化的高性能文件系统，能够快速处理高性能计算（HPC）工作负载。

作为对象存储服务的子产品， 并行文件系统支持用户按照标准的 OBS 接口读取数据。

## 1.2 应用场景

并行文件系统提供高兼容性、高性能、高可扩展性、高可靠性的能力，适用各种高性能计算以及媒资归档场景。

**主要的应用场景如下：**

视频监控：公安社会视频、商业监控、家庭监控

点播：OTT 分发、媒资库

HPC：聚焦基因测序、制造业 CAE 场景

大数据：日志分析、内容推荐、运营报表、用户画像、交互式分析

## 1.3 约束限制

### 性能限制

- 单个天翼云账号内网默认的读写（GET/PUT）带宽上限是 16Gbit/s，同账号下并行文件系统与 OBS 共享 16Gbit/s 带宽上限；
- 单个天翼云账号公网默认的读写（GET/PUT）带宽上限是 1Gbit/s，同账号下并行文件系统与 OBS 共享 1Gbit/s 带宽上限；

### 操作限制

- 不支持将已有的 OBS 桶修改为并行文件系统，创建并行文件系统方法请参见 2.1 创建并行文件系统。
- 并行文件系统暂不支持配额配置功能，默认无配额限制。

#### 功能限制

- 暂不支持跨区域复制
- 暂不支持多版本控制
- 暂不支持配置静态网站托管

## 1.4 使用方式

并行文件系统提供管理控制台、obsfs 客户端工具以及 RESTful API 接口的使用方法，其中 obsfs 客户端工具提供标准的 POSIX 接口，使得用户的的应用程序无需任何修改即可无缝上云。用户可以在任何时间和地点对存储在并行文件系统内的文件进行处理，并且可以快速获取到处理后的文件。并行文件系统除支持 POSIX 文件语义外，支持原来对象的接口，即用户仍然可以使用对象的处理方式来处理文件，实现了对象和文件的互通。

用户可以通过以下几种方式使用并行文件系统。

#### 说明

OBS 桶的权限管理规则及权限要求同样适用于并行文件系统，用户在使用并行文件系统前需要确保具备 OBS 资源的访问权限。

表1-1 使用方式

方式	主要功能	相关链接
并行文件系统控制台	您可以在控制台创建并行文件系统，并进行查看、管理等基本操作。	2.1 创建并行文件系统
obsfs 工具	obsfs 是针对并行文件系统开发的客户端工具，支持 POSIX 标准接口，通过挂载即可使用，推荐您使用此方式。	《对象存储（OBS）工具指南(obsfs)》
OBS API	通过 OBS API 来使用并行文件系统。	3.1 支持的 API 列表

# 2 控制台方式

## 2.1 创建并行文件系统

用户可以通过控制台创建并行文件系统。

### 操作步骤

步骤 1 在天翼云首页，选择“控制中心 > 所有服务 > 存储 > 对象存储”（OBS）。

步骤 2 选择左侧导航栏的“并行文件系统”，进入并行文件系统控制台。

步骤 3 单击界面右上角的“创建并行文件系统”，进入创建页面。

步骤 4 设置并行文件系统的区域和名称。

#### 说明

- 并行文件系统创建成功后，不能修改名称，请提前规划合适的名称。
- 由于通过 URL 访问并行文件系统时，名称会作为 URL 的一部分，根据 DNS 标准，URL 不支持大写字母，无法区分带大写字母的并行文件系统。因此，名称仅支持小写字母、数字、“-”、“.”。例如：若想通过 URL 访问名为“MyFileSystem”的文件系统，该 URL 将解析成名为“myfilesystem”的文件系统，导致访问出错。

步骤 5 策略：您可以为并行文件系统配置私有、公共读、或公共读写策略。

步骤 6 在界面最下方确认配置信息后，单击“立即创建”。

步骤 7 创建成功后，回到并行文件系统控制台，即可看到已创建的并行文件系统。

至此，用户就可以像操作 OBS 桶一样使用并行文件系统了。如何使用并行文件系统，请参见 1.4 使用方式。

----结束

# 3 API 方式

## 3.1 支持的 API 列表

本节主要介绍并行文件系统和已有的 OBS API 的兼容情况，并对部分兼容的 API 进行差异说明。

关于 OBS 各个 API 接口的详细介绍，请参见《对象存储（OBS）API 参考》。

### 桶基础操作接口

表3-1 桶基础操作接口

接口	并行文件系统是否兼容	差异说明
获取桶列表	√	获取并行文件系统列表需要携带头域“x-obs-bucket-type: POSIX”。
创建桶	√	创建并行文件系统需要携带头域“x-obs-fs-file-interface:Enabled”
列举桶内对象	√	-
获取桶元数据	√	-
获取桶区域位置	√	-
删除桶	√	-

## 桶高级配置接口

表3-2 桶高级配置接口

接口	并行文件系统是否兼容	差异说明
设置桶策略	√	-
获取桶策略	√	-
删除桶策略	√	-
设置桶 ACL	√	-
获取桶 ACL	√	-
设置桶日志管理配置	√	-
获取桶日志管理配置	√	-
设置桶的生命周期配置	×	-
获取桶的生命周期配置	×	-
删除桶的生命周期配置	×	-
设置桶的多版本状态	×	-
获取桶的多版本状态	×	-
设置桶的跨区域复制配置	×	-
获取桶的跨区域复制配置	×	-
删除桶的跨区域复制配置	×	-
设置桶配额	√	-
获取桶配额	√	-
获取桶存量信息	√	-

## 静态网站托管接口

表3-3 静态网站托管接口

接口	并行文件系统是否兼容	差异说明
设置桶的网站配置	×	-
获取桶的网站配置	×	-

接口	并行文件系统是否兼容	差异说明
删除桶的网站配置	×	-
设置桶的 CORS 配置	×	-
获取桶的 CORS 配置	×	-
删除桶的 CORS 配置	×	-
OPTIONS 桶	×	-
OPTIONS 对象	×	-

## 对象操作接口

表3-4 对象操作接口

接口	并行文件系统是否兼容	差异说明
PUT 上传	√	不支持的头域: 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、x-obs-expires
POST 上传	√	不支持的头域: 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、x-obs-expires
复制对象	√	只支持相同集群内并行文件系统或 OBS 桶之间进行复制。
获取对象内容	√	-
获取对象元数据	√	-
删除对象	√	-

接口	并行文件系统是否兼容	差异说明
批量删除对象	√	-
追加写对象	×	-
设置对象 ACL	√	-
获取对象 ACL	√	-
修改对象元数据	√	-
修改写对象	√	并行文件系统新增接口，OBS 桶不支持。
截断对象	√	并行文件系统新增接口，OBS 桶不支持。
重命名对象	√	并行文件系统新增接口，OBS 桶不支持。

## 多段操作接口

表3-5 多段操作接口

接口	并行文件系统是否兼容	差异说明
列举桶中已初始化多段任务	√	-
初始化上传段任务	√	-
上传段	√	-
拷贝段	√	不支持对追加写过的文件执行拷贝段操作。
列举已上传的段	√	-
合并段	√	-
取消多段上传任务	√	-

# A 修订记录

发布日期	修订记录
2022-08-05	<p>第二次正式发布。</p> <p>本次更新说明如下：</p> <ul style="list-style-type: none"><li>使用方式新增 obsfs 客户端工具。</li></ul>
2021-10-21	第一次正式发布。
2023-12-21	产品名称对象存储（经典版）II 型修改为对象存储