



# 对象存储数据迁移工具

用户使用指南

天翼云科技有限公司

## 目 录

|  |    |
|--|----|
| 1 介绍 .....   | 5  |
| 2 工具使用流程 .....   | 6  |
| 2.1 工具安装 .....   | 6  |
| 2.1.1 依赖的软硬件环境 .....                                     | 6  |
| 2.1.2 推荐的配置方案 .....                                      | 6  |
| 2.2 工具使用 .....   | 7  |
| 2.3 migration.conf 合规校验 .....                            | 8  |
| 2.3.1 Linux 系统 .....                                     | 13 |
| 2.3.2 windows 系统 .....                                   | 13 |
| 2.4 启动迁移 .....   | 14 |
| 2.4.1 Linux 系统 .....                                     | 14 |
| 2.4.2 Windows 系统 .....                                   | 14 |
| 2.5 停止迁移 .....   | 15 |
| 2.5.1 Linux 系统 .....                                     | 15 |
| 2.5.2 Windows 系统 .....                                   | 15 |
| 2.6 查询迁移进度 .....   | 15 |
| 2.6.1 Linux 系统 .....                                     | 15 |
| 2.6.2 Windows 系统 .....                                   | 16 |
| 2.7 查看迁移结果 .....   | 16 |
| 2.8 查看迁移日志 .....   | 16 |
| 3 migration.conf 填写示例 .....                              | 18 |
| 3.1 srcType .....  | 18 |
| 3.2 srcMigrationType .....                               | 18 |
| 3.3 migrationAfterModified/migrationBeforeModified ..... | 20 |
| 3.4 objectStorageClass/objectAcl .....                   | 20 |
| 3.5 destPrefix .....                                     | 21 |
| 3.6 conflictMode .....                                   | 21 |
| 3.7 multipartSize(megabytes) .....                       | 22 |
| 3.8 ratelimit .....                                      | 22 |

|                                  |    |
|----------------------------------|----|
| 3.9 processNums/threadNums ..... | 23 |
| 3.10 failRetryMode .....         | 23 |
| 3.11 ignoreGetObjTag .....       | 24 |
| 3.12 s3AddressingStyle .....     | 24 |
| 4 迁移实践 .....                     | 24 |
| 4.1 Linux 系统 .....               | 24 |
| 4.2 Windows 系统 .....             | 25 |
| 4.3 多任务迁移 .....                  | 28 |
| 4.4 数据迁移网络模型 .....               | 29 |
| 5 约束与限制 .....                    | 29 |
| 6 注意事项 .....                     | 31 |
| 7 常见问题及处理 .....                  | 32 |

## 文档修订记录表:

| 日期          | 修订内容                               |
|-------------|------------------------------------|
| 2024 年 11 月 | 新增迁移实践、约束与限制、注意事项、常见问题与处理章节。       |
| 2025 年 3 月  | 新增流量控制填写示例                         |
| 2025 年 8 月  | 新增配置文件校验，多桶迁移实践                    |
| 2026 年 3 月  | 新增 ignoreGetObjTag 填写示例，数据迁移网络模型实践 |
| 2026 年 4 月  | 新增 s3AddressingStyle 填写示例          |

# 1 介绍

天翼云对象存储（CT-ZOS, Zettabyte Object Storage）数据迁移工具（Zsync）是一款用于将存储在其他对象存储中的数据迁移至天翼云 ZOS 的工具。

Zsync 工具需部署在本地服务器或云主机上使用，支持多种操作系统，具有即开即用，用户交互简洁，快速启动等优点。

**Zsync 工具支持的特性如下：**

- **适配多源端：**支持多个数据源，包括：亚马逊 S3、阿里云 OSS、华为云 OBS、腾讯云 COS、天翼云 ZOS、天翼云 OOS 和其他遵循标准 S3 协议的对象存储服务。
- **源端配置**
  - 源端支持指定迁移方式：整桶迁移、指定文件夹、指定文件、指定前缀进行迁移。
  - 源端支持起止时间过滤：迁移源桶指定时间区间内的文件。
- **目的端配置**
  - 目的端支持存储类型配置：支持配置迁移至目的端的对象的存储类型。
  - 目的端支持访问控制配置：支持配置迁移至目的端的对象的访问控制权限。
  - 目的端指定前缀：支持将文件迁移至目的端桶时，为对象增加指定前缀。
- **高级配置**
  - 同名文件处理策略：支持配置多种同名文件处理方式。
  - 失败对象记录及重试：支持记录迁移失败对象，支持对迁移失败的对象进行重试迁移。
  - 一致性校验：支持迁移过程中的一致性校验（按最后修改时间和 size）。
  - 自定义分片：支持自定义分片大小。
  - 流量控制：支持全时/分时迁移限流。
- **工具易用性**
  - 配置文件校验：支持校验配置文件填写语法。
  - 并行传输和断点续传：支持对象并行下载和上传，支持断点续传。
  - 日志记录：支持迁移日志记录。
  - 进度查询：支持实时查询迁移进度。
  - 多任务迁移：支持填写多个配置文件，支持排队迁移。
  - 迁移报告：支持输出迁移报告。

## 2 工具使用流程

### 2.1 工具安装

#### 2.1.1 依赖的软硬件环境

表 2.1 硬件要求

| 名称    | 配置   | 说明    |
|-------|------|-------|
| CPU   | 8C   | 可按需调整 |
| 内存    | 16GB | 可按需调整 |
| 系统盘大小 | 40GB | 可按需调整 |

表 2.2 软件要求

| 名称   | 版本                      | 说明 |
|------|-------------------------|----|
| 操作系统 | Centos7.8/Kylin/Windows | —  |

表 2.3 网络要求

| 名称    | 说明  |
|-------|---|
| 公网/内网 | 可公网部署亦可内网部署,迁移工具所在的服务器可访问源端云存储资源池和 ZOS 资源池即可。 |
| 带宽    | 按需 (和迁移速度正相关)                                 |

天翼云官网的对象存储帮助文档中,提供迁移工具压缩包 ZOS\_Migration\_Tool.zip 的下载地址,在上述运行环境确定后,下载并解压迁移工具压缩包,完成迁移工具的安装。

#### 2.1.2 推荐的配置方案

为方便您快捷进行工具环境的准备,我们建议您可按照如下配置进行部署。以下推荐方案在确定硬件配置与并发参数配置的情况下,已经帮您计算出所需的最低网络带宽。

表 2.4 推荐配置方案

| 硬件配置                   | 工具并发数配置                                      | 网络带宽要求    |
|------------------------|--|-----------|
| 8C16G<br>(系统盘 40GB 以上) | processNums 参数配置为 12;<br>threadNums 参数配置为 8; | 60Mbps 以上 |
| 4C8G                   | processNums 参数配置为 6;                         | 30Mbps 以上 |

|               |                     |  |
|---------------|---------------------|--|
| (系统盘 40GB 以上) | threadNums 参数配置为 8; |  |
|---------------|---------------------|--|

**注意:**

若您无法提供如上表的推荐配置。例如：迁移主机配置为 4c8g，部署设备与源端，或部署设备与目的端的带宽仅有 10Mbps 时，建议您配置 processNums 参数不高于 2，threadNums 参数不高于 8，避免导致频繁出现对象迁移失败的情况。

若您提供的配置，特别是迁移的网络带宽较高（例如带宽为 1Gbps 以上），您可以进行迁移前的测试，通过提供更多核心和内存的服务器设备，按并发数调优方式进行并发数配置，来实现更好的迁移效果。

**说明:**

- (1) **迁移工具并发数如何调优:** 实际与主机性能相关，通常配置文件 migration.conf 进程数 processNums 设置不超过服务器 CPU 的核心数的 1.5 倍。每个进程内的并发线程数 threadNums，建议不超过 10。
- (2) **网络稳定性的影响:** 您部署的设备与您的源端的网络连通性会影响迁移的稳定性、您部署的设备与目的端 ZOS 的网络连通性也会影响迁移的稳定性，建议您迁移前检查网络连通性。
- (3) **迁移前测试的重要性:** 工具的并发数配置与您提供的带宽都会影响迁移，造成对象迁移失败的情况（这种情况您可进行失败对象重试），建议您进行迁移前的测试。

## 2.2 工具使用

**步骤 1:**

填写配置文件 migration.conf，详见 2.3 小节；

**步骤 2:**

启动工具 zsync，详见 2.4 小节。

工具操作相关指令如下：

表 2.5 zsync 工具指令

| 命令/选项         | 描述                                | Linux 示例              | Windows 示例         |
|---------------|-----------------------------------|-----------------------|--------------------|
| -h            | 查看工具帮助                            | ./zsync -h            | zsync.exe -h       |
| check         | 检查配置文件的语法是否正确，默认检查 migration.conf | ./zsync check         | zsync.exe check    |
| check -c file | 检查配置文件的语法是                        | ./zsync check -c file | zsync.exe check -c |

|               |  |                       |                         |
|---------------|--|-----------------------|-------------------------|
|               | 否正确, 检查指定配置文件 file                         |                       | file                    |
| check-all     | 检查配置文件的语法是否正确, 依次检查工具所在目录下以.conf 为拓展名的配置文件 | ./zsync check-all     | zsync.exe check-all     |
| start         | 启动工具, 默认执行 migration.conf                  | ./zsync start         | zsync.exe start         |
| start -c file | 启动工具, 指定配置文件 file                          | ./zsync start -c file | zsync.exe start -c file |
| start-all     | 启动工具, 依次迁移工具所在目录下以.conf 为拓展名的配置文件          | ./zsync start-all     | zsync.exe start-all     |
| stop          | 停止工具                                       | ./zsync stop          | zsync.exe stop          |
| info          | 查询迁移进度                                     | ./zsync info          | zsync.exe info          |
| version       | 查看工具版本                                     | ./zsync version       | zsync.exe version       |

## 2.3 migration.conf 合规校验

迁移工具包 ZOS\_Migration\_Tool.zip 文件解压完成后, 首先查看文件夹内的迁移配置文件 migration.conf, 该配置文件包含字段如表 2.6。

表 2.6 migration.conf

| 参数名称         | 说明   | 类型     |
|--------------|--|--------|
| srcType      | 迁移源类型, 全部可选参数为 S3、OSS、COS、OBS、OOS。<br><br><b>【参数说明】</b> <ul style="list-style-type: none"> <li>● S3: 亚马逊、ZOS、XOS (其他遵循标准 S3 协议的对象存储服务)</li> <li>● OSS: 阿里云</li> <li>● COS: 腾讯云</li> <li>● OBS: 华为云</li> <li>● OOS: 天翼云经典版 I 型</li> </ul> | string |
| srcUrl       | 源资源池 URL 地址, 通常为 http://<endpoint> 或 https://<endpoint>。<br><br>注意:<br>腾讯云 COS 较为特殊: 仅需填写<region>, 例如 ap-beijing, ap-shanghai。   | string |
| srcAccessKey | 源资源池 AccessKey   | string |
| srcSecretKey | 源资源池 SecretKey   | string |

|                         |  |        |
|-------------------------|--|--------|
| srcBucket               | 源资源池 Bucket 名称   | string |
| srcMigrationType        | 源资源池迁移模式，全部可选参数为 Bucket、Folder、Files、Prefix。<br>【参数说明】<br>● Bucket: 整桶迁移<br>● Folder: 指定文件夹迁移<br>● Files: 指定文件迁移<br>● Prefix: 指定前缀迁移   | string |
| srcMigrateFolder        | 指定源资源池迁移的文件夹列表，仅当 srcMigrationType 为 Folder 时有效  | list   |
| srcMigrateFiles         | 指定源资源池迁移的文件名列表，仅当 srcMigrationType 为 Files 时有效   | list   |
| srcMigratePrefix        | 指定源资源池迁移文件的前缀列表，仅当 srcMigrationType 为 Prefix 时有效   | list   |
| destUrl                 | 目的资源池 (ZOS) 的 URL 地址为 http://<endpoint>或 https://<endpoint>。<br><br>说明：<br>您可在对象存储 ZOS 的 bucket“概览-域名信息”中找到 终端节点 (Endpoint) 信息。  | string |
| destAccessKey           | 目的资源池(ZOS)AccessKey  | string |
| destSecretKey           | 目的资源池(ZOS)SecretKey  | string |
| destBucket              | 目的资源池(ZOS)Bucket 名称  | string |
| migrationAfterModified  | 起始时间，仅迁移最后修改时间大于设置时间的对象，可以设置以下两种格式：<br>● "year-month-day"<br>● "year-month-day hour:minute:second"   | string |
| migrationBeforeModified | 终止时间，仅迁移最后修改时间小于设置时间的对象，可以设置以下两种格式：<br>● "year-month-day"<br>● "year-month-day hour:minute:second"   | string |
| objectStorageClass      | 对象存储类型，默认适配源端对象的存储类型，全部可选参数为 STANDARD、STANDARD_IA、GLACIER。<br>【参数说明】<br>● STANDARD: 标准存储<br>● STANDARD_IA: 低频存储<br>● GLACIER: 归档存储<br>注意：<br>源端对象若为归档数据，无法直接迁移，需解冻后再执行迁移，并请您确保解冻时长，以防迁移期间源端数据再次变成冻结状态。 | string |

|                        |  |        |
|------------------------|--|--------|
| objectAcl              | <p>对象访问权限，默认适配源端对象的访问权限，全部可选参数为 <code>private</code>、<code>public-read</code>。</p> <p><b>【参数说明】</b></p> <ul style="list-style-type: none"> <li>● <code>private</code>: 私有</li> <li>● <code>public-read</code>: 公共读</li> </ul>  | string |
| destPrefix             | <p>指定目的端前缀，配置后将文件迁移至目的端桶时，为对象增加指定前缀。若不配置，默认迁移后目的桶内对象的路径和源桶将保持一致。</p> <p>说明：</p> <ol style="list-style-type: none"> <li>1、<b>增加前缀目录</b>：该前缀若为目录，则为迁移的对象增加配置的前缀目录，实现将对象迁移至指定的目录下的效果。</li> <li>2、<b>增加前缀</b>：该前缀若不为目录，则为迁移的对象名中增加该部分前缀。</li> </ol>  | string |
| recordFailedObject     | <p>迁移失败记录开关，全部可选参数为 <code>True</code>、<code>False</code>。</p> <p><b>【参数说明】</b></p> <ul style="list-style-type: none"> <li>● <code>True</code>: 记录迁移失败的对象</li> <li>● <code>False</code>: 不会记录迁移失败对象</li> </ul>  | string |
| enableConsistencyCheck | <p>一致性校验开关，全部可选参数为 <code>True</code>、<code>False</code>。</p> <p><b>【参数说明】</b></p> <ul style="list-style-type: none"> <li>● <code>True</code>: 迁移过程开启一致性校验，校验不通过的对象会记录在 <code>Possible_Inconsistent_Objects_xxx.txt</code> 文件中</li> <li>● <code>False</code>: 迁移过程关闭一致性校验</li> </ul>  | string |
| conflictMode           | <p>同名文件处理选项，全部可选参数为 <code>OVERWRITE</code>、<code>IGNORE</code>、<code>COMPARE</code>。</p> <p><b>【参数说明】</b></p> <ul style="list-style-type: none"> <li>● <code>OVERWRITE</code>: 对同名文件，不进行任何判断，一律执行覆盖。</li> <li>● <code>IGNORE</code>: 对同名文件，不进行任何判断，一律执行跳过。</li> <li>● <code>COMPARE</code>: 对同名文件，优先判断二者的最后修改时间，仅当源端文件的最后修改时间相较目的端最后修改时间更新时，进行覆盖。若源端与目的端文件最后修改时间一致，则判断两者的文件大小，大小一致则跳过，大小不一致则执行覆盖。</li> </ul> | string |
| logLevel               | <ul style="list-style-type: none"> <li>● 日志级别，全部可选参数为 <code>ERROR</code>、<code>INFO</code>、<code>DEBUG</code>。</li> </ul> <p><b>【参数说明】</b></p> <ul style="list-style-type: none"> <li>● <code>ERROR</code></li> </ul>  | string |

|                          |  |         |
|--------------------------|--|---------|
|                          | <ul style="list-style-type: none"> <li>● INFO</li> <li>● DEBUG</li> </ul>  |         |
| processNums              | 迁移任务的进程并发数量<br>说明：<br>建议您按照手册的“工具安装与配置建议”根据您的主机配置该参数。  | int     |
| threadNums               | 迁移任务的线程并发数量<br>说明：<br>建议您按照手册的“工具安装与配置建议”根据您的主机配置该参数。  | int     |
| multipartSize(megabytes) | 自定义设置分片大小，取值范围为[5, 5120]，单位为 MB。   | int     |
| ratelimit                | 对单个任务的迁移速度进行流量控制，单位为 MB/s。<br><b>【参数说明】</b> <ul style="list-style-type: none"> <li>● 填写 0 则表示不限流。</li> </ul> 说明：<br>1、迁移工具通过限制源端数据下载的流速实现流量控制；<br>2、流量控制采用单位时间内平均限流的方式，从源端下载大文件时，若限流参数 < 5MB/s，会出现瞬时流量超过限流参数，达到 5MB/s 的情况； | int/map |
| failRetryMode            | 失败重试开关，全部可选参数为 True、False。<br><b>【参数说明】</b> <ul style="list-style-type: none"> <li>● False：关闭失败重试功能；</li> <li>● True：使用失败重启功能，对 retryFailFiles 配置的文件列表进行重试。</li> </ul>   | string  |
| retryFailFiles           | 迁移过程中自动生成的错误日志列表，只支持对 _ZOS_Migration_Failure_ 命名的日志文件进行重试。   | string  |
| ignoreGetObjTag          | 忽略迁移源端对象 tagging 开关，可选参数为 True、False。<br><b>【参数说明】</b> <ul style="list-style-type: none"> <li>● False：迁移源端对象 tagging；</li> <li>● True：不迁移源端对象 tagging。</li> </ul>  | string  |
| s3AddressingStyle        | 当 srcType 设置为 S3 时，此参数生效，用于控制向 S3 存储桶发起请求时的 URL 格式。<br>可选值为 path、virtual。<br><b>【参数说明】</b> <ul style="list-style-type: none"> <li>● path：路径样式格式；</li> <li>● virtual：虚拟样式格式。</li> </ul>                                     | string  |

配置文件 migration.conf 示例：

```
{
  "srcType": "S3",
  "srcUrl": "http://sichuan-2.zos.ctyun.cn",
  "srcAccessKey": "xxxxxxxxxxxxxxxx",
  "srcSecretKey": "xxxxxxxxxxxxxxxx",
  "srcBucket": "bucket-name",
  "srcMigrationType": "Bucket",
  "srcMigrateFolder": ["test1/", "test2/"],
  "srcMigrateFiles": ["objects_list.txt"],
  "srcMigratePrefix": ["test"],
  "destUrl": "http://jiangsu-10.zos.ctyun.cn",
  "destAccessKey": "xxxxxxxxxxxxxxxx",
  "destSecretKey": "xxxxxxxxxxxxxxxx",
  "destBucket": "bucket-name",
  "migrationAfterModified": "",
  "migrationBeforeModified": "",
  "objectStorageClass": "",
  "objectAcl": "",
  "destPrefix": "",
  "recordFailedObject": "True",
  "enableConsistencyCheck": "False",
  "conflictMode": "IGNORE",
  "logLevel": "DEBUG",
  "processNums": 12,
  "threadNums": 8,
  "multipartSize(megabytes)": 5,
  "ratelimit": 0,
  "failRetryMode": "False",
  "retryFailFiles": [],
  "ignoreGetObjTag": "False",
  "s3AddressingStyle": "path"
}
```

上述配置文件 migration.conf 填写完毕后,可以使用 check 命令对配置文件进行合规校验。

### 2.3.1 Linux 系统

在 Linux 环境使用迁移工具前，需要先赋予工具可执行权限，执行如下指令：

```
chmod +x zsync
```

上述操作建议在**管理员权限下执行**。

在 Linux 平台中，配置文件 migration.conf 填写完毕后，使用 cd 命令进入到 zsync 可执行文件所在的目录下，即包含 migration.conf 和 zsync 的目录。

执行指令：

```
./zsync check
```

默认校验 migration.conf 文件填写语法是否正确。

执行指令：

```
./zsync check -c file
```

校验指定的配置文件 file 填写语法是否正确。

执行指令：

```
./zsync check-all
```

依次校验工具所在目录下以.conf 为拓展名的配置文件是否填写正确。

### 2.3.2 windows 系统

在 Windows 平台，配置文件 migration.conf 填写完毕后，打开 cmd 终端窗口，**使用 cd 命令切换到迁移工具包目录下**，即包含 migration.conf 和 zsync.exe 的目录。

执行指令：

```
zsync.exe check
```

默认校验 migration.conf 文件填写语法是否正确。

执行指令：

```
zsync.exe check -c file
```

校验指定的配置文件 file 填写语法是否正确。

执行指令：

```
zsync.exe check-all
```

依次校验工具所在目录下以.conf为拓展名的配置文件是否填写正确。

## 2.4 启动迁移

### 2.4.1 Linux 系统

在 Linux 环境使用迁移工具前，需要先赋予工具可执行权限，执行如下指令：

```
chmod +x zsync
```

上述操作建议**在管理员权限下执行**。

在 Linux 平台中，配置文件 migration.conf 填写完毕后，使用 cd 命令进入到 zsync 可执行文件所在的目录下，即包含 migration.conf 和 zsync 的目录。

执行指令：

```
./zsync start
```

开始迁移，默认执行 migration.conf 配置的迁移任务。

为避免关闭终端导致 zsync 进程中断，可使用指令：

```
nohup ./zsync start &
```

将 zsync 进程放在后台执行，不受终端关闭影响。

### 2.4.2 Windows 系统

在 Windows 平台，配置文件 migration.conf 填写完毕后，打开 cmd 终端窗口，**使用 cd 命令切换到迁移工具包目录下**，即包含 migration.conf 和 zsync.exe 的目录。执行指令：

```
zsync.exe start
```

开始迁移，默认执行 migration.conf 配置的迁移任务。

## 2.5 停止迁移

### 2.5.1 Linux 系统

在 Linux 平台，在 zsync 可执行文件所在的目录下，执行指令：

```
./zsync stop
```

停止执行中的任务。

### 2.5.2 Windows 系统

在 Windows 平台，直接关闭 zsync.exe start 运行的终端窗口，即可停止迁移。

## 2.6 查询迁移进度

表 2.7 迁移进度参数

| 参数             | 说明        |
|----------------|-----------|
| Query time     | 查询用时，单位秒  |
| Migration time | 迁移用时，单位秒  |
| Scanned nums   | 源桶扫描的对象个数 |
| Ignored nums   | 迁移忽略的对象个数 |
| Failed nums    | 迁移失败的对象个数 |
| Finished nums  | 迁移完成的对象个数 |
| Finished size  | 迁移完成的对象大小 |
| Download rate  | 当前下载速率    |
| Upload rate    | 当前上传速率    |

### 2.6.1 Linux 系统

迁移过程中，在 zsync 可执行文件所在的目录下，执行如下指令：

```
./zsync info
```

查询实时的迁移进度。

## 2.6.2 Windows 系统

迁移过程中，打开 cmd 终端窗口，使用 **cd 命令切换到迁移工具包目录下**，即包含 migration.conf 和 zsync.exe 的目录。执行指令：

```
zsync.exe info
```

即可查询实时的迁移进度。

## 2.7 查看迁移结果

迁移完成后，与 zsync 可执行文件同级目录下会生成 migration\_results.txt，即迁移结果文件。该文件记录了 zsync 工具执行迁移的结果信息，包括：ZOS 桶名、迁移状态、迁移开始时间、迁移结束时间、源桶扫描对象个数、迁移忽略个数、迁移失败个数、迁移完成个数和总的迁移数据量等信息。

表 2.8 migration\_results.txt

| 参数                   | 说明   |
|----------------------|--|
| bucket               | 源桶名→目的桶名，迁移状态： <ul style="list-style-type: none"><li>● migration completed: 迁移正常完成</li><li>● migration interrupted: 迁移异常中断</li></ul> |
| migration start time | 迁移开始时间   |
| migration end time   | 迁移结束时间   |
| scanned nums         | 源桶扫描的对象个数  |
| ignored nums         | 迁移忽略的对象个数  |
| failed nums          | 迁移失败的对象个数  |
| finished nums        | 迁移完成的对象个数  |
| migration size       | 迁移完成的对象大小  |

## 2.8 查看迁移日志

Windows 平台迁移产生的日志存储在 zsync.exe 同级 log/目录下。



Linux 平台迁移产生的日志存储在 `/var/log/zsync/`。根据实际需要可在 `migration.conf` 中配置 `debug`、`info` 和 `error` 等不同的日志级别。

## 3 migration.conf 填写示例

### 3.1 srcType

Zsync 工具支持将数据从不同的云厂商迁移到天翼云 ZOS，通过配置文件 migration.conf 的 srcType 字段实现。

| 参数填写示例            | 描述                          |
|-------------------|-----------------------------|
| "srcType": "S3",  | 源端是天翼云 ZOS，或支持标准 S3 接口的对象存储 |
| "srcType": "OSS", | 源端是阿里云                      |
| "srcType": "OBS", | 源端是华为云                      |
| "srcType": "COS", | 源端是腾讯云                      |
| "srcType": "OOS", | 源端是天翼云 OOS                  |

### 3.2 srcMigrationType

Zsync 工具支持对源端进行整桶、指定文件夹、指定文件或指定前缀的迁移，通过配置文件 migration.conf 中的 srcMigrationType 字段实现。

| 参数填写示例   | 描述   |
|--|--|
| "srcMigrationType": "Bucket",  | 整桶迁移   |
| "srcMigrationType": "Folder",<br>"srcMigrateFolder": ["test1/", "test2/"], | 指定文件夹迁移，将源桶内文件夹名称填写在 srcMigrateFolder 字段，以逗号分割。                          |
| "srcMigrationType": "Prefix",<br>"srcMigratePrefix": ["test"],             | 指定前缀迁移，如果源桶内文件名称从左至右存在相同的字符，可以指定按相同前缀进行迁移，填写在 srcMigratePrefix 字段，以逗号分割。 |
| "srcMigrationType": "Files",<br>"srcMigrateFiles": ["objects_list.txt"],   | 指定文件迁移，如果明确待迁移的对象名称，用户自定义一个文件 objects_list.txt，在 txt 文件中一行填写一个对象。        |

### 1、指定文件夹迁移示例

若源桶内存在按年整理的文件夹，例如 2021/，2022/，2023/， 2024/。如果需要迁移 2023/和 2024/文件夹下的对象，可以将配置文件 migration.conf 中 srcMigrationType 和 srcMigrateFolder 字段按如下方式填写：

```
"srcMigrationType": "Folder",  
"srcMigrateFolder": ["2023/", "2024/"],
```

### 2、指定前缀迁移示例

若源桶内存在按年月整理的文件夹，例如：202401/，202402/，……， 202411/，202412/，这些文件夹拥有相同的前缀“2024”。如果需要迁移 2024 年所有数据，可以将配置文件 migration.conf 中 srcMigrationType 和 srcMigratePrefix 字段按如下方式填写：

```
"srcMigrationType": "Prefix",  
"srcMigratePrefix": ["2024"],
```

### 3、指定文件迁移示例

如果明确源桶内待迁移对象的名称，可以将配置文件 migration.conf 中 srcMigrationType 和 srcMigrateFiles 字段按如下方式填写：

```
"srcMigrationType": "Files",  
"srcMigrateFiles": ["objects_list.txt"],
```

其中，objects\_list.txt 文件中记录的是待迁移的文件名称，一行填写一个对象名。例如：  
a.jpg, b.pdf, c.html, d.mp4 等 4 个文件待迁移，填写示例如下图。

```
[root@localhost ~]# cat objects_list.txt  
a.jpg  
b.pdf  
c.html  
d.mp4  
[root@localhost ~]# █
```

### 3.3 migrationAfterModified/migrationBeforeModified

| 参数填写示例  | 描述   |
|---|--|
| "migrationAfterModified": "2024-11-14 12:00:00",  | 仅迁移最后修改时间晚于 migrationAfterModified 的对象。                                |
| "migrationBeforeModified": "2024-11-14 12:00:00",   | 仅迁移最后修改时间早于 migrationBeforeModified 的对象。                               |
| "migrationAfterModified": "2024-11-14 12:00:00",<br>"migrationBeforeModified": "2024-11-15 12:00:00", | 仅迁移最后修改时间位于 (migrationBeforeModified, migrationBeforeModified) 区间内的对象。 |

**注意：**

同时设定 migrationAfterModified 和 migrationBeforeModified，工具限制 migrationBeforeModified 需晚于 migrationAfterModified。

### 3.4 objectStorageClass/objectAcl

| 参数填写示例                               | 描述                                    |
|--------------------------------------|---------------------------------------|
| "objectStorageClass": "",            | <b>匹配源端：</b> 会自动匹配源端存储类型，与源端存储类型保持一致。 |
| "objectStorageClass": "STANDARD",    | <b>标准存储：</b> 将会存储为标准类型。               |
| "objectStorageClass": "STANDARD_IA", | <b>低频存储：</b> 将会存储为低频类型。               |
| "objectStorageClass": "GLACIER",     | <b>归档存储：</b> 将会存储为归档类型。               |

**注意：**

“匹配源端”但仅能自动匹配源端的“标准”和“低频”类型，匹配源端的“归档”或“深度归档”类型，请您务必提前对源端归档数据进行手动解冻，并确保迁移任务完成前数据保持解冻状态，否则该部分数据会迁移失败。

| 参数填写示例                  | 描述                                    |
|-------------------------|---------------------------------------|
| "objectAcl": "",        | <b>匹配源端：</b> 会自动匹配源端的 ACL 配置，与源端保持一致。 |
| "objectAcl": "private", | <b>私有：</b> 会将目的端 ACL 配置为私有。           |

|                             |                                |
|-----------------------------|--------------------------------|
| "objectAcl": "public-read", | <b>公共读</b> : 会将目的端 ACL 配置为公共读。 |
|-----------------------------|--------------------------------|

### 3.5 destPrefix

| 参数填写示例                | 描述                                    |
|-----------------------|---------------------------------------|
| "destPrefix": "",     | 目的桶内对象和源端桶内对象名称保持一致                   |
| "destPrefix": "ZOS",  | 源端对象迁移到目的端, 会在对象名前增加设置的 <b>前缀</b> 。   |
| "destPrefix": "ZOS/", | 源端对象迁移到目的端, 会在对象名前增加设置的 <b>前缀目录</b> 。 |

#### 1、目的端增加前缀示例

源端迁移对象文件为: a.txt, b.txt, 设置“保存到指定前缀” destPrefix 参数为: “ZOS”, 那么迁移后的对象文件变为: ZOSa.txt、ZOSb.txt。

#### 2、目的端增加前缀目录示例

源端迁移对象文件为: a.txt、b.txt, 设置“保存到指定前缀” destPrefix 参数为: “ZOS/”, 那么迁移后的对象文件保存到目的端桶的路径为: ZOS/a.txt、ZOS/b.txt。

### 3.6 conflictMode

Zsync 工具通过设置同名文件配置参数, 可以实现增量迁移的效果。

用户需要增量迁移任务时, 可通过配置文件 migration.conf 中的 conflictMode 字段来实现。

当 conflictMode 字段参数配置为 OVERWRITE, 则对同名文件进行覆盖;

当 conflictMode 字段参数配置为 IGNORE, 则不对同名文件进行迁移, 实现一般的增量迁移效果;

当 conflictMode 字段参数配置为 COMPARE, 同名文件按最后修改时间比较, 源端最后修改时间比目的端最后修改时间更新时则会进行二次迁移, 适合源端对象内容会动态变化的增量迁移场景。

| 参数填写示例                    | 描述   |
|---------------------------|--|
| "conflictMode": "IGNORE", | <b>不覆盖</b> : 对同名文件, 不进行任何判断, 一律执行跳过, 即可实现增量迁移。 |

|                              |   |
|------------------------------|---|
| "conflictMode": "OVERWRITE", | <b>全覆盖:</b> 对于同名文件, 不进行任何判断, 一律执行覆盖。  |
| "conflictMode": "COMPARE",   | <b>按最后修改时间:</b> 对于同名文件, 优先判断二者的最后修改时间, 仅当源端文件相较目的端最后修改时间更新时, 进行覆盖。若源端与目的端文件最后修改时间一致, 则判断两者的文件大小, 大小一致则执行跳过, 大小不一致则执行覆盖。 |

**注意:**

若您迁移的目的桶存有存量数据, **配置为 OVERWRITE 进行全量迁移可能会导致目的桶存量数据被覆盖, 造成数据丢失, 请谨慎操作。**

### 3.7 multipartSize(megabytes)

| 参数填写示例                          | 描述        |
|---------------------------------|-----------|
| "multipartSize(megabytes)": 5,  | 设置 5M 分片  |
| "multipartSize(megabytes)": 10, | 设置 10M 分片 |

**注意:**

对象 size 超过 multipartSize(megabytes), 工具采用分片上传。由于分片上传时, 限制最大分片个数为 10000, 即设置 5M 分片时最大支持 48.8G 的对象迁移; 设置 10M 分片时最大支持 97G 的对象迁移; 以此类推。该参数可填写范围是[5, 5120], 即 5M~5120M。用户可依据对象大小自定义设置。

但是, 当设置分块大小过大时, 需要降低工具整体的并发数, 避免内存溢出风险。

### 3.8 ratelimit

通过设置该参数可以限制工具的最大平均迁移速度, 防止迁移过程占用过多带宽。

| 参数填写示例          | 描述                |
|-----------------|-------------------|
| "ratelimit": 0, | 不限速               |
| "ratelimit": 5, | 全时限流, 整数, 单位 MB/s |

|                           |                                      |
|---------------------------|--------------------------------------|
| "ratelimit": {"7-22": 5}, | 分时限流，键值对。键：字符串，时间区间；值：整数，限速值，单位 MB/s |
|---------------------------|--------------------------------------|

#### 分时限流填写格式：

"ratelimit": {"start\_time-end\_time": rate\_limit}，其中 start\_time 取值范围[0, 23]；end\_time 取值范围[1, 24]；rate\_limit 通常小于迁移工具所在机器的网络带宽值。

#### 分时限流填写示例说明如下：

"ratelimit": {"7-8": 5}，限制 7 点到 8 点区间迁移速度最大为 5MB/s，其他时间不做限制。

"ratelimit": {"7-22": 5}，限制 7 点到 22 点区间迁移速度最大为 5MB/s，其他时间不做限制。

### 3.9 processNums/threadNums

| 参数填写示例            | 描述                              |
|-------------------|---------------------------------|
| "processNums":12, | 并行执行的进程数，建议不要超过 CPU 核心数的 1.5 倍。 |
| "threadNums":16,  | 每个进程内的并发线程数，建议不超过 20。           |

### 3.10 failRetryMode

#### 1.配置失败文件记录：

可以通过修改配置文件 migration.conf 中 recordFailedObject 字段进行配置。True 表示开启失败记录，在扫描完源存储桶和目的存储桶中所有对象以及迁移过程执行完成后，若出现错误，则在本地会存在相应记录错误的文件；False 表示不打开失败记录，工具不会记录迁移失败的对象。

#### 2.配置失败文件重试：

进行失败文件的重试需要用户配置 failRetryMode 和 retryFailFiles 两个字段。失败文件的记录保存在文件 srcType + "\_ZOS\_Migration\_Failure\_" + 时间 + ".txt"中，该文件记录了迁移过程中迁移失败的对象名和迁移错误原因；用户可以将 failRetryMode 字段配置为 True，将错误文件的名称填写到配置文件的 retryFailFiles 字段，重新运行工具即可重试迁移失败的对象。

| 参数填写示例   | 描述     |
|--|--------|
| "failRetryMode": "True",<br>"retryFailFiles": ["XXX_ZOS_Migration_Failure_XXXXXX.txt"] | 开启失败重试 |

### 3.11 ignoreGetObjTag

忽略迁移源端对象标签开关，可选参数为 True、False。

| 参数填写示例                     | 描述                 |
|----------------------------|--------------------|
| "ignoreGetObjTag": "False" | 迁移过程中会迁移源端对象的标签信息  |
| "ignoreGetObjTag": "True"  | 迁移过程中不会迁移源端对象的标签信息 |

### 3.12 s3AddressingStyle

当 srcType 设置为 S3 时，此参数生效，用于控制向 S3 存储桶发起请求时的 URL 格式。可选值为 path、virtual。

| 参数填写示例                         | 描述  |
|--------------------------------|---|
| "s3AddressingStyle": "path"    | 当 srcType 设置为 S3 时，使用路径风格的 URL 向 S3 存储桶发起请求，URL 格式为：<br>https://<endpoint>/<bucket-name>/<object-key> |
| "s3AddressingStyle": "virtual" | 当 srcType 设置为 S3 时，使用虚拟风格的 URL 向 S3 存储桶发起请求，URL 格式为：<br>https://<bucket-name>.<endpoint>/<object-key> |

## 4 迁移实践

以华为云 OBS 到天翼云 ZOS 迁移为例，其他云厂商迁移流程基本一致，仅**需修改 migration.conf 中对应的源端信息字段**即可。

### 4.1 Linux 系统

#### 1、工具安装

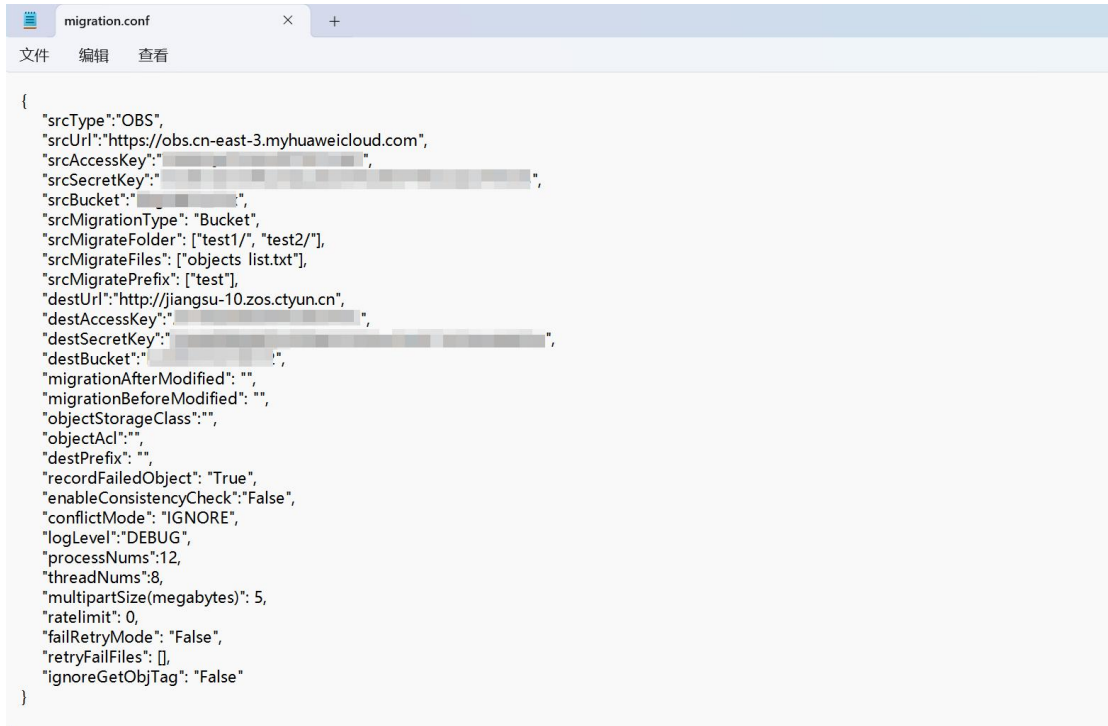
工具解压后：

```
[root@localhost zos-migration-tool]# chmod +x zsync
[root@localhost zos-migration-tool]# ls
migration.conf  zsync
[root@localhost zos-migration-tool]# █
```



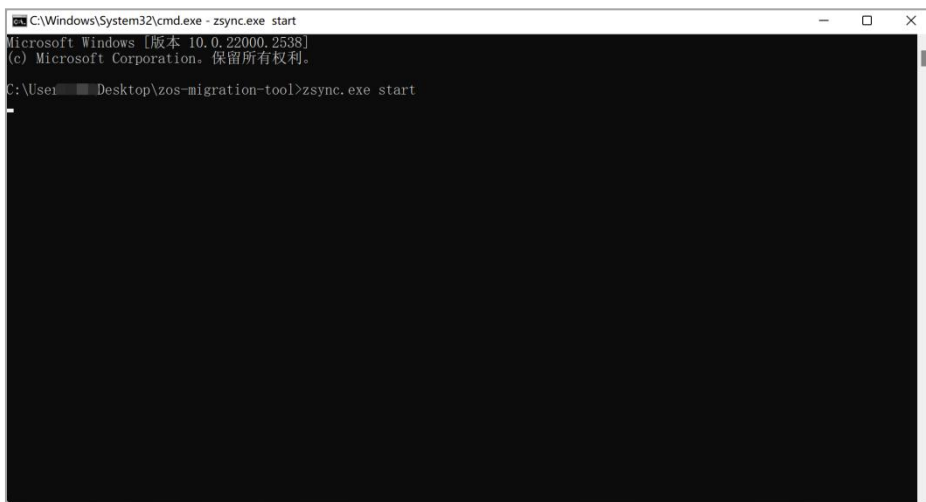


## 2、配置文件填写



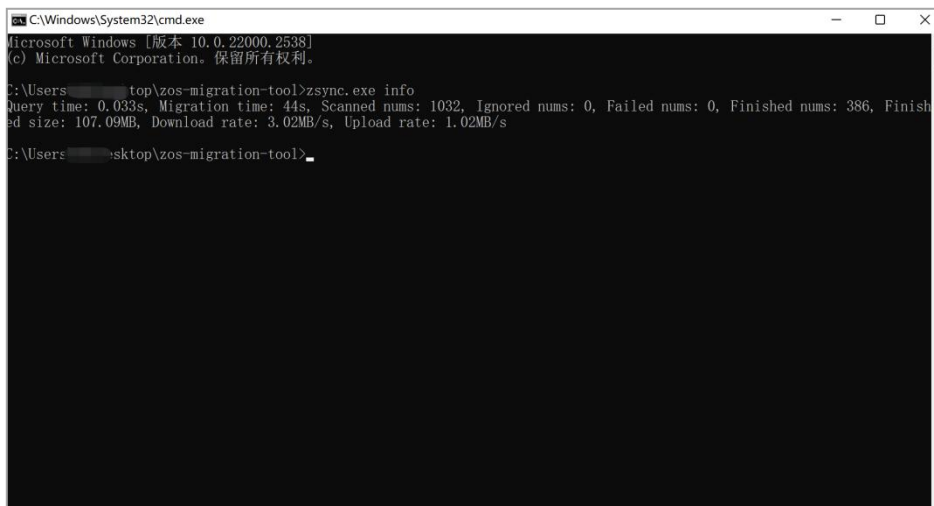
## 3、启动迁移

打开 cmd 窗口，使用 **cd** 命令切换到迁移工具包目录下，即包含 migration.conf 和 zsync.exe 的目录，然后执行 zsync.exe start 指令



## 4、查询迁移进度

打开新的 cmd 窗口，使用 **cd** 命令切换到迁移工具包目录下，即包含 migration.conf 和 zsync.exe 的目录，然后执行 zsync.exe info 指令



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\sktop\zsync-migration-tool>zsync.exe info
Query time: 0.033s, Migration time: 44s, Scanned nums: 1032, Ignored nums: 0, Failed nums: 0, Finished nums: 386, Finished size: 107.09MB, Download rate: 3.02MB/s, Upload rate: 1.02MB/s

C:\Users\sktop\zsync-migration-tool>
```

## 5、查看迁移结果



```
bucket: [bucket-f5e3] migration completed | migration start time: 2024-11-21 15:45:06 | migration end time: 2024-11-21 15:59:10 | scanned nums: 1032 | ignored nums: 0 | failed nums: 0 | finished nums: 1032 | migration size: 1.67GB
```

## 4.3 多任务迁移

多个任务排队依次执行，具体步骤如下：

### 步骤 1：创建目的桶

在目的端 ZOS 中创建与源端对象存储中对应的桶：

- 源端 bucket-a → 目的端 bucket-zos-a
- 源端 bucket-b → 目的端 bucket-zos-b
- 源端 bucket-c → 目的端 bucket-zos-c

### 步骤 2：配置文件创建

以 **migration.conf** 为模板，为每个桶分别创建迁移配置文件，命名格式如下：

- migration-a.conf
- migration-b.conf
- migration-c.conf

在每个配置文件中，请正确填写对应源桶与目标桶的名称及相关信息。（配置文件需要以.conf 为扩展名）

### 步骤 3：配置文件合规校验

在开始迁移前，请对配置文件进行合规性检查。

**Linux 系统执行：**

```
./zsync check-all
```

**Windows 系统执行：**

```
zsync.exe check-all
```

### 步骤 4：启动多桶迁移

check-all 校验通过后，启动多桶迁移任务。

**Linux 系统执行：**

```
./zsync start-all
```

为避免关闭终端导致 zsync 进程中断，可使用指令：

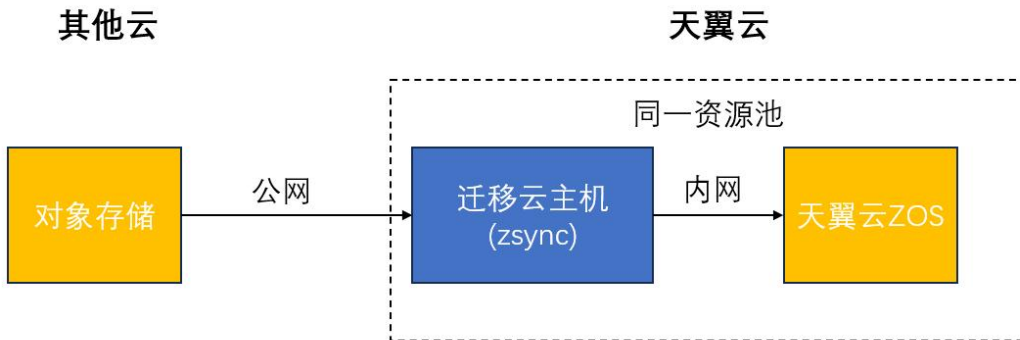
```
nohup ./zsync start-all &
```

将 zsync 进程放在后台执行，不受终端关闭影响。

**Windows 系统执行：**

```
zsync.exe start-all
```

## 4.4 数据迁移网络模型



通常搭建上图迁移网络模型开展数据迁移，即部署 zsync 工具的云主机和 ZOS 桶位于同一资源池可内网互通，zsync 通过公网从源端对象存储服务读取文件对象，然后通过内网将文件对象写入 ZOS 桶中。

**注意：**

数据迁移不限于上述网络模型，主要取决于迁移云主机和源端桶，以及 ZOS 桶的网络联通情况，即部署 zsync 的云主机和源端对象存储，以及 ZOS 网络互通即可。

## 5 约束与限制

| 限制项目   | 说明  |
|--------|---|
| 目的端    | 仅支持目的端为天翼云对象存储 ZOS。   |
| 源端     | 腾讯云 COS 作为源端时，直接使用 ap-beijing, ap-shanghai 类型的参数作为访问地址，而其他云厂商使用 https://<endpoint>作为访问地址。 |
| 迁移范围   | 工具仅支持单个任务迁移单个对象存储桶的数据。  |
| 迁移任务带宽 | <ul style="list-style-type: none"> <li>迁移进行时会占用源端和目的端的资源，若您的源端或目的端</li> </ul>             |

|         |   |
|---------|---|
|         | <p>有重要业务，请您提前评估，在空闲时间或避开业务高峰期执行迁移。</p> <ul style="list-style-type: none"> <li>实际的迁移速度和源端网络带宽、迁移带宽、文件大小均有关。对于小文件，由于 IOPS 较高的原因，会导致迁移速度有所较低。</li> </ul>   |
| 源端多版本   | 工具默认迁移源端多版本对象中的最新版本，不支持迁移历史版本的数据。   |
| 源端归档数据  | <ul style="list-style-type: none"> <li>归档数据需解冻后才能迁移。请您务必先全部解冻源端的归档数据，未解冻和解冻中的数据会迁移失败。</li> <li>请您根据待迁移的数据总量确保解冻时长，以防迁移期间数据再次变成冻结状态。</li> <li>归档存储类型需要用户进行数据取回操作（即：解冻操作），该操作可能会收取数据取回费用，具体计费规则可咨询您的源端存储服务提供商。</li> </ul> |
| 目的端存储类型 | 支持标准存储、低频存储、归档存储。   |
| 迁移单对象大小 | 默认情况下，系统支持最大 48.8GB 对象的迁移。若需迁移超过此大小的对象，请根据实际需求自定义设置分片大小。  |
| 目录对象的迁移 | <p>工具对源端的目录对象仅请求 <code>head_object</code> 操作，不请求 <code>get_object</code> 操作，工具会直接在目的端执行 <code>put_object</code> 操作来创建该目录。</p> <p>因此，若您的桶内有预迁移对象名以“/”结尾，则迁移工具会将其识别为目录并跳过。</p>  |
| 一致性校验   | <p>仅校验迁移前后对象的 <code>lastmodified</code> 和 <code>size</code>，无法保证迁移结果的正确性，故在迁移任务完成后，请您务必自行做好源端和目的端数据的正确性核验。</p> <p>同时，开启一致性校验会导致迁移效率降低，以及 API 调用次数增加。</p>  |

## 6 注意事项

- 1) **分片大小问题:** 工具支持自定义设置分片大小, 但是并发数固定, 分片设置过大可能会导致内存溢出。
- 2) **迁移中断问题:** 迁移过程中, 如果出现中断, 重新启动工具前, 可以设置 `conflictMode` 参数为 `IGNORE`, 已迁移过的同名文件会进行跳过, 节省迁移时间。
- 3) **迁移工具并发数如何调优:** 实际与服务器性能相关, 通常配置文件 `migration.conf` 进程数 `processNums` 设置不超过服务器 CPU 的核心数 1.5 倍。每个进程内的并发线程数 `threadNums`, 建议不超过 20。
- 4) **增量迁移问题:** 增量迁移无法实现动态的增量, 需要源桶和目的桶进行对比从而获得增量数据。
- 5) **各个源端云厂商对于同步迁移对象元数据的支持情况:**

| 云厂商 | 存储级别 | 对象标签 | 访问控制 | 自定义元数据 |
|-----|------|------|------|--------|
| S3  | √    | √    | √    | √      |
| OSS | √    | √    | √    | √      |
| COS | √    | √    | √    | √      |
| OBS | √    | ×    | √    | √      |
| OOS | √    | ×    | √    | ×      |

附: √ 表示支持, × 表示不支持

## 7 常见问题及处理

### 1) 使用迁移工具对目的端已有的数据是否会有影响？

迁移工具对目的端的影响取决于目的端与源端数据是否存在同名文件。以下分两种情况：

- **不存在同名文件：**迁移工具对目的端已有数据无影响，不会修改目的端已有数据，也不会删除目的端已有对象。
- **存在同名文件：**工具将按照配置文件 migration.conf 中的"conflictMode"参数来决定如何处理同名文件。
  - **IGNORE：**不覆盖，对同名文件，不进行任何判断，一律执行跳过，保留目的端文件。
  - **OVERWRITE：**全覆盖，对于同名文件，不进行任何判断，一律执行覆盖，源端文件直接覆盖目的端文件。
  - **COMPARE：**按最后修改时间，对于同名文件，优先判断二者的 lastmodified（即最后修改时间），仅当源端文件相较目的端修改时间更新时，源端文件覆盖目的端文件。若源端与目的端文件最后修改时间一致，则判断两者的文件大小，大小一致则执行跳过，大小不一致则执行覆盖。

### 2) 对象的存储类型，源端对象迁移至 ZOS 后的存储类型是什么？：

- 若配置文件 migration.conf 中参数"objectStorageClass"填写为 STANDARD（标准），STANDARD\_IA（低频），GLACIER（归档），则迁移后目的端数据的存储类型与配置文件中填写一致。
- 若配置文件 migration.conf 中参数"objectStorageClass"为空时，则工具会自动匹配源端，迁移后目的端 ZOS 桶内对象的访问控制与源端的对应关系如下表：

| 云厂商 | 源端对象存储级别 | (ZOS)目的端对象存储级别 |
|-----|----------|----------------|
| S3  | 标准存储     | 标准存储           |
|     | 低频存储     | 低频存储           |
|     | 归档存储或其他  | 归档存储           |
| OSS | 标准存储     | 标准存储           |
|     | 低频存储     | 低频存储           |
|     | 归档存储或其他  | 归档存储           |
| COS | 标准存储     | 标准存储           |

|     |         |      |
|-----|---------|------|
|     | 低频存储    | 低频存储 |
|     | 归档存储或其他 | 归档存储 |
| OBS | 标准存储    | 标准存储 |
|     | 低频存储    | 低频存储 |
|     | 归档存储或其他 | 归档存储 |
| OOS | 标准存储    | 标准存储 |
|     | 低频存储    | 低频存储 |
|     | 归档存储或其他 | 归档存储 |

**注意：**

源端归档类型的数据，需提前解冻后才能迁移。请您务必先全部解冻源端的归档数据，未解冻和解冻中的数据会迁移失败。

**3) 对象的访问控制权限，源端对象迁移到 ZOS 后的访问控制权限是什么？**

- 若配置文件 migration.conf 中参数"objectAcl"填写为"private"或"public-read"，则迁移后目的端数据的 ACL 配置与配置文件中填写一致。
- 若配置文件 migration.conf 中参数"objectAcl"为空时，工具会自动匹配源端，迁移后目的端（ZOS）的访问控制与源端的对应关系如下表：

| 云厂商 | 源端对象访问控制          | (ZOS)目的端对象访问控制    |
|-----|-------------------|-------------------|
| S3  | public-read       | public-read       |
|     | public-read-write | public-read-write |
|     | private 或其他       | private           |
| OSS | public-read       | public-read       |
|     | public-read-write | public-read-write |
|     | private 或其他       | private           |
| COS | public-read       | public-read       |
|     | public-read-write | public-read-write |
|     | private 或其他       | private           |
| OBS | public-read       | public-read       |
|     | public-read-write | public-read-write |
|     | private 或其他       | private           |
| OOS | public-read       | public-read       |
|     | public-read-write | public-read-write |
|     | private 或其他       | private           |

#### 4) 输入对象前缀时，实际并未迁移指定前缀的对象，是什么原因？

可能是前缀的输入有误。迁移服务支持源端指定对象前缀，该前缀为包含着对象所在目录路径的内容，而非仅对象名称的部分，格式为：文件夹/对象前缀。

例如：桶内的根目录有目录 A，目录 A 下有待迁移对象的名为 old-filetest.jpg 和 old-image.png，则实际待迁移对象的名称应为：A/old-filetest.jpg 和 A/old-image.png。指定前缀时，不能仅输入“old-”，而应该输入“A/old-”。

#### 5) 源端请求超时，是什么原因？如何处理比较好？

源端请求超时常见以下原因：

- 用户迁移机器公网访问源端对象存储不稳定，导致获取元数据和下载对象请求超时；
- 用户迁移机器带宽很小，工具并发参数设置很高，低带宽无法满足高并发的数据迁移，故部分源端获取元数据，下载对象请求超时

建议您迁移前进行迁移测试，针对以上原因提高带宽，降低工具并发数。也可直接按照推荐配置部署资源。

#### 6) 工具常见异常提示

启动迁移工具后，针对一些不规范或异常情况工具会进行异常提示，以下常见异常提示可帮助您快速确定异常原因：

| 错误提示  | 异常原因   |
|---|--|
| Error: the migration.conf configuration file is incorrect                               | 配置文件参数填写不合规，请检查配置文件参数。   |
| Error: check whether the network path has a proxy or whether the destination end is ZOS | 1、zos 端 url 填写错误；<br>2、目的端填写正确，但用户使用 nginx 做了代理转发，导致检测失败；<br>3、目的端填写正确，用户开启了代理网络工具，导致检测失败； |
| Source bucket error: Forbidden  | 源桶 ak, sk 错误，请检查参数。  |
| Source bucket error: Not Found  | 源端桶不存在，请确认源端桶及源端参数。  |

|                                     |                              |
|-------------------------------------|------------------------------|
| Destination bucket error: Forbidden | 目的端 zos 的 as,sk 信息错误, 请检查参数。 |
| Destination bucket error: Not Found | 目的端的 zos 桶不存在, 请确认源端桶及源端参数。  |