

媒体处理

最佳实践

文档版本 01
发布日期 2024-04-26



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 使用 MPC 对 OBS 中的音视频进行转码.....	1
2 H.264 和 H.265 高清低码帮助视频网站打造极致体验.....	7
3 MPC 的视频截图能力帮助视频网站客户快速构建媒体处理平台.....	12
4 MPC 的视频转封装能力帮助在线教育客户实现多终端播放视频.....	19
5 修订记录.....	22

1 使用 MPC 对 OBS 中的音视频进行转码

场景说明

您可能需要将已有的音视频文件引用到各种业务场景中，如产品官网、视频网站等，也可能需要在不同的终端进行播放，如Web端、移动端。为适用于多样化的场景，MPC提供了转码功能，支持改变音视频的编码格式、封装格式、分辨率、码率等参数，从而适用于不同的业务场景、播放终端及网络环境。

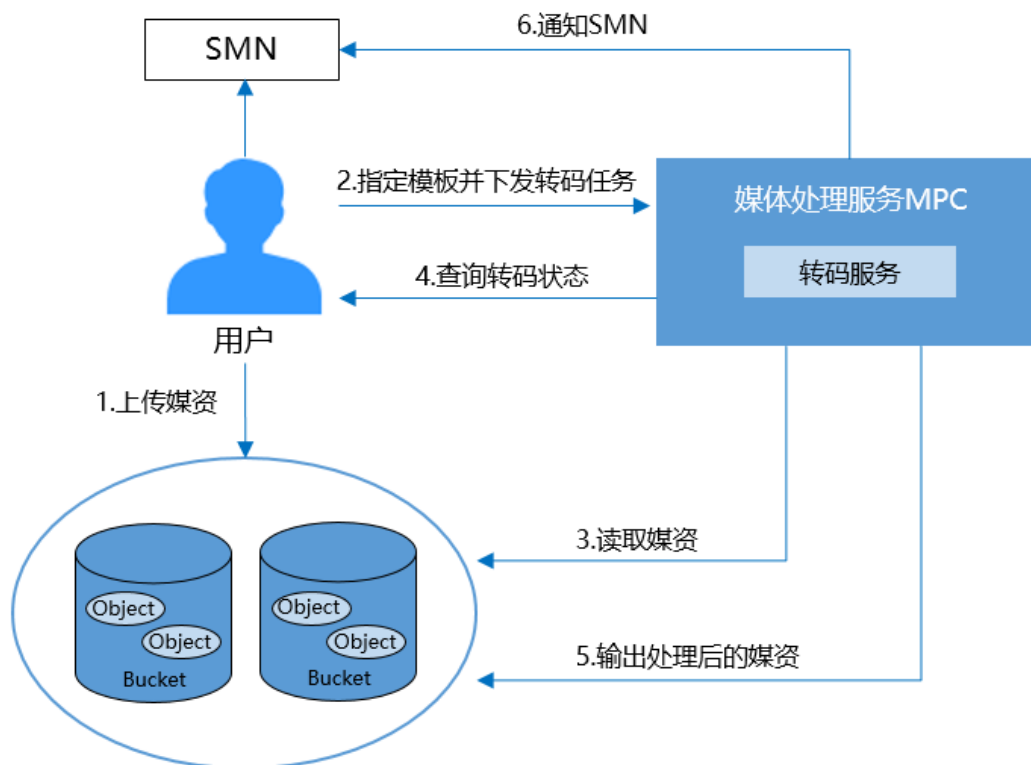
转码功能可以帮助您处理如下问题：

- 适配不同播放终端：您可以将原始音视频转码为更适应于各终端播放的格式，如MP4格式适用于网站播放的场景。
- 适配不同网络环境：您可以根据不同的网络带宽设置转码输出的码率，以适应不同的网络播放环境。
- 降低分发成本：支持压缩率更高的H.265编码格式，您也可以在转码时开启高清低码功能，在分辨率不变的情况下，码率下降20%左右，从而降低音视频的分发成本。
- HLS内容保护：您可以在转码时开启HLS加密，对转码的HLS格式音视频进行加密，从而防止音视频被盗用后进行二次分发。
- 版权保护：您可以为视频添加自己独有的logo等标识水印，以宣示该视频的版权所属。
- 音视频提取：支持转码提取音频文件，主要适用于一些纯音频的场景，如电台、音频APP等。
- 禁用音频：同时也支持输出无音频的视频文件，主要适用于需要禁用原音频的场景。

实现原理

在标准转码场景中，您可以通过使用系统默认的转码模板，对OBS桶内的音视频文件进行转码，并将转码后的文件输出至指定的OBS桶中。转码过程中支持查询转码状态，当转码任务完成后会通过SMN返回消息给您。

图 1-1 MPC 转码实现原理



流程说明如下所示：

1. 用户上传需要转码的音视频文件至OBS。
2. 用户指定输入/输出转码模板并下发转码任务。
3. MPC服务获取用户指定的音视频文件进行处理。
4. 转码过程定期查询转码的状态。
5. 转码任务完成后，将转码后的音视频文件输出至用户指定的OBS存储目录。
6. 通过SMN服务通知用户转码状态信息。
7. 用户订阅指定主题获取到转码相关信息。

转码前准备

- 原始音视频文件已上传到OBS桶中，且使用的MPC与OBS桶在同一区域，若还未上传，请参见[上传音视频文件](#)处理。
- 已将存储原始音视频和转码音视频的桶进行授权，允许MPC访问。若还未授权，请参见[获取云资源授权](#)处理。
- 若需要消息通知转码的任务执行状态，请先配置消息通知功能，具体操作请参见[配置消息通知](#)。
- 若您需要使用自定义模板/模板组进行转码，请先新建转码模板/模板组，具体操作请参见[自定义转码模板/自定义转码模板组](#)。

创建视频转码任务

您可以创建视频转码任务，按需求选择视频转码模板，对存储在OBS桶中的视频文件进行转码。

说明

支持转码的视频编码格式：H.264、H.265、MPEG-2、MPEG-4、MJPEG、VP6/7/8/9、WMV1/2/3、ProRes 422等，若源文件的编码格式不在如上列出的格式中，则会出现转码失败。

步骤1 登录**媒体处理控制台**。

步骤2 在左侧导航树中，选择“媒体处理 > 转码”，进入“转码”页面。

步骤3 单击“创建任务”。

在“创建任务”页面，配置视频转码参数。

步骤4 配置基本参数，如图1-2所示。

选择原始视频的存储桶和文件名，转码后输出文件的存储桶及输出路径。

图 1-2 创建转码任务

转码任务列表 / 创建任务

1. 启动转码任务前，请提前创建桶并上传待转码文件，同时授权：[去上传 去授权](#)
 2. 支持的转码格式有MP4、TS、MOV、MXF、MPG、FLV、WMV、AVI、MP3、ADTS、3GP、MKV、M3U8，更多格式即将上线，敬请期待！
 3. 若您选择使用一进多出模板进行转码，则产生的转码费用为多个输出文件的转码费用总和。[了解详情](#)

* 源文件所在桶: * 源文件名:

* 输出桶: 输出路径:

步骤5 选择符合您需求的视频转码模板，如图1-3所示。

图 1-3 转码模板

转码模板

一进一出模板 | 一进多出模板 | 自定义模板 | 自定义模板组

输出格式: 全部 HLS DASH MP4 MP3 ADTS DASH+HLS

视频编码格式: 全部 H.264 H.265

分辨率: 高低码:

输出格式	模板名	模板ID	分辨率/码率 (kbit/s)
<input type="radio"/> MP4	MP4_H.265_4K_低码	7000761	3840*2160/5600(4K)
<input type="radio"/> MP4	MP4_H.265_2K_低码	7000762	2560*1440/4900(2K)
<input type="radio"/> MP4	MP4_H.265_4K	7000605	3840*2160/5600(4K)
<input type="radio"/> MP4	MP4_H.265_2K	7000606	2560*1440/4900(2K)
<input type="radio"/> HLS	HLS_H.265_4K_低码	7000719	3840*2160/5600(4K)
<input type="radio"/> HLS	HLS_H.265_2K_低码	7000720	2560*1440/4900(2K)
<input type="radio"/> HLS	HLS_H.265_4K	7000563	3840*2160/5600(4K)
<input type="radio"/> HLS	HLS_H.265_2K	7000564	2560*1440/4900(2K)
<input type="radio"/> DASH	DASH_H.265_4K_低码	7000698	3840*2160/5600(4K)
<input type="radio"/> DASH	DASH_H.265_2K_低码	7000699	2560*1440/4900(2K)

总条数: 105 < 1 2 3 4 5 6 ... 11 >

媒体处理服务中预置了大量的一进一出和一进多出模板。由于系统模板预置了常见的清晰度、对应的码率、分辨率等参数，所以建议您优先使用系统预置模板。系统预置模板的相关参数，可以在媒体处理控制台的“全局设置 > 预置模板”页面查看。

须知

- 音频文件不能选择视频转码模板进行转码。
- 输入格式为GIF的文件，暂只支持转码输出格式为MP4。

步骤6 配置完成，单击“确定”，转码任务开始执行。

步骤7 可在转码任务列表中，查看转码任务执行情况。

当前仅支持查看60天内的转码详情。

图 1-4 转码任务列表



- 若转码成功，可在“转码输出”列单击“输出路径”，查看转码后的视频文件，并可在对象存储桶中下载或分享转码后的视频文件。
- 若转码失败，可通过“转码输出”列可查看失败原因，进行定位。

----结束

创建音频转码任务

您可以创建音频转码任务，按需求选择音频转码模板，对存储在OBS桶中的音频文件进行转码。音频转码的费用与视频转码是不同的，具体请参见[价格详情](#)。

说明

支持转码的音频编码格式：AAC、AC3、EAC3、HE-AAC、MP2、MP3、PCM (s161e, s16be, s241e, s24be, dvd)、WMA等。

若源文件的编码格式不在如上列出格式中，则会出现转码失败。

步骤1 登录[媒体处理控制台](#)。

步骤2 在左侧导航树中，选择“媒体处理 > 转码”，进入“转码”页面。

步骤3 单击“创建任务”。

在“创建任务”页面，配置音频转码参数。

步骤4 配置基本参数，如图1-5所示。

选择原始音频的存储桶和文件名，转码后输出文件的存储桶及输出路径。

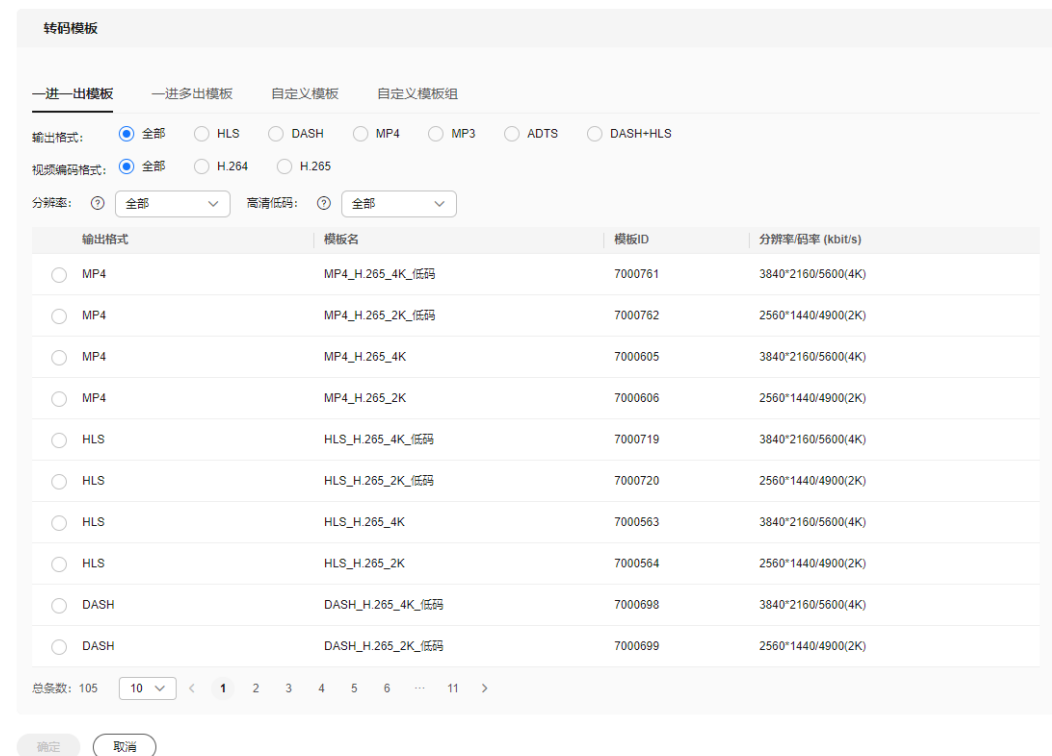
图 1-5 创建转码任务



步骤5 选择符合您需求的音频转码模板。

- 若选择系统预置的“一进一出模板”，输出格式请勾选“MP3”或者“ADTS”。
- 若选择“自定义模板”，请参考[创建音频转码模板](#)。

图 1-6 转码模板



步骤6 配置完成，单击“确定”，转码任务开始执行。

步骤7 可在转码任务列表中，查看转码任务执行情况。

当前仅支持查看60天内的转码详情。

图 1-7 转码任务列表

转码 满意度评价

说明：转码前请先完成以下步骤：[上传音视频](#) -> [获取云资源授权](#) -> [消息订阅](#) (可选) -> [转码设置](#) (可选)

创建任务

近30天

转码任务	转码状态	开始时间	结束时间	转码输入	转码输出	操作
任务：18011 模板：8000017	转码成功	2024/04/24 14:25:52 GMT+08:00	2024/04/24 14:25:55 GMT+08:00	桶名：xco 源文件名：.mp4	桶名：xco 输出路径：ma...	删除 取消 重试
任务：18010 模板：8000017	转码成功	2024/04/24 14:03:22 GMT+08:00	2024/04/24 14:03:26 GMT+08:00	桶名：xco 源文件名：.mp4	桶名：xco 输出路径：ma...	删除 取消 重试

----结束

2 H.264 和 H.265 高清低码帮助视频网站打造 极致体验

场景说明

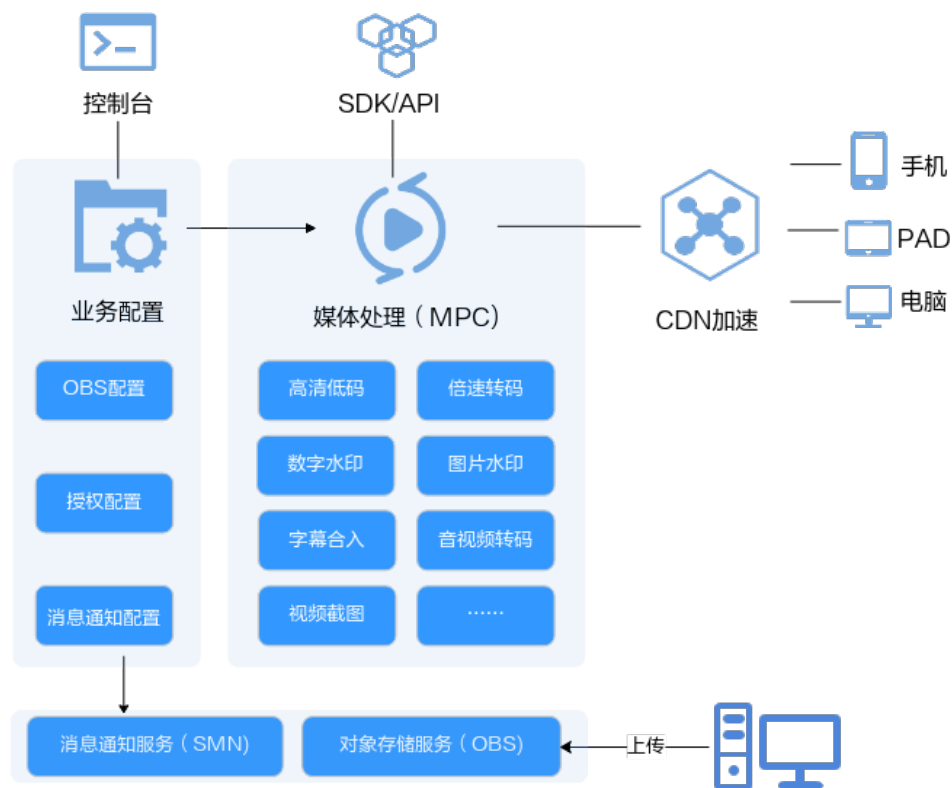
互联网视频网站需要在不同终端、不同带宽下，均能为客户提供良好的观看体验。MPC编解码算法，在优化原视频画质的同时，降低视频码率，实现保证视频清晰度的前提下，极大压缩视频体积。有效节约存储与带宽成本，降低视频卡顿率时延、失败率等指标，提升用户观看体验。

用户在创建转码任务时，可以根据使用场景启用MPC的高清低码能力，详情如下所示。

- [配置方式1](#)
- [配置方式2](#)
- [配置方式3](#)

实现原理

图 2-1 实现原理图



配置方式 1

访问 API Explorer 的 [媒资转码接口](#)，将 Body 体中的请求参数“PVC”的值设置为“true”，如下图所示。

图 2-2 媒体转码接口



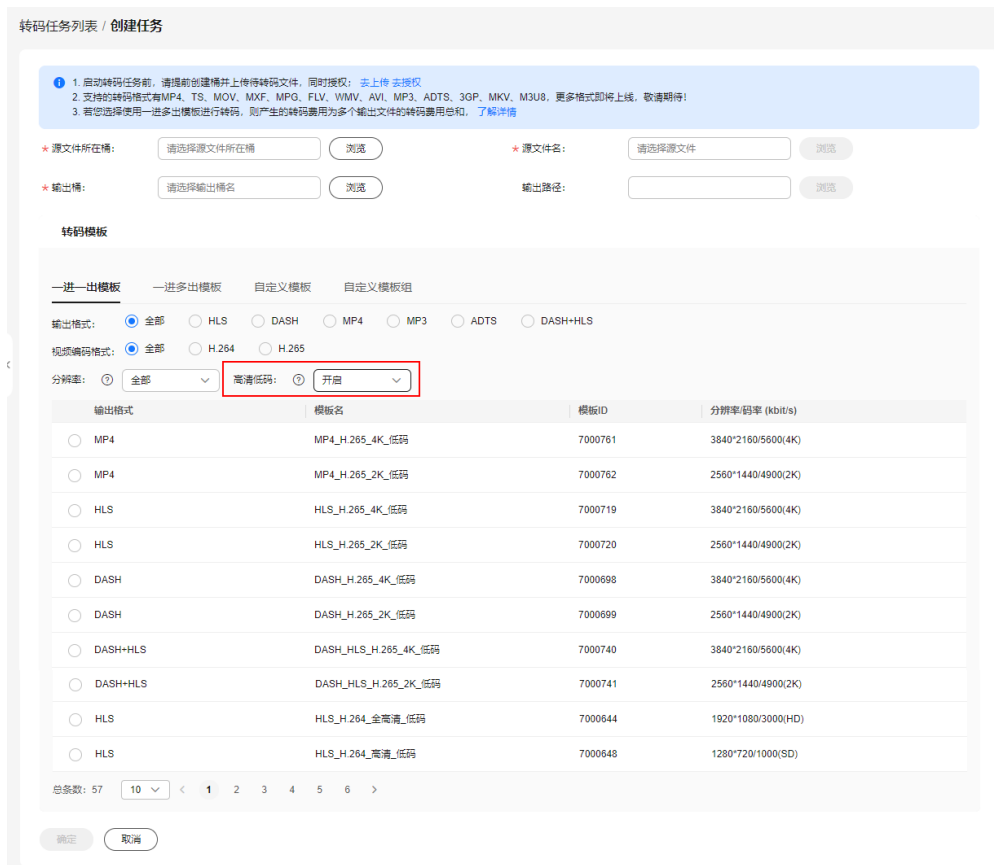
配置方式 2

- 步骤1** 登录[媒体处理控制台](#)。
- 步骤2** 在左侧导航树中，选择“媒体处理 > 转码”，进入“转码”页面。
- 步骤3** 单击“创建任务”，进入“创建任务”页面。
- 步骤4** 配置转码模板的“高清低码”参数，从下拉框中选择“开启”，再选择高清低码模板，如下图所示。

请注意：

- 如果创建高清低码模板，需要将请求参数“Common”中的“Pvc”的值设置为“true”，详情请参考[新建转码模板](#)。
- 如果创建高清低码模板组，需要将请求参数“Common”中的“Pvc”的值设置为“true”，详情请参考[新建转码模板组](#)。

图 2-3 高清低码模板



----结束

配置方式 3

接入MPC的SDK，在创建转码任务的请求体中，将请求参数“Common”中的“Pvc”的值设置为“true”，即“withPvc(true)”。

SDK集成方式，音视频转码功能的详细说明和样例代码，请参考[创建与查询转码任务](#)。

示例核心代码如下所示：

```
//设置转码输入视频地址和输出视频路径
ObsObjInfo input = new ObsObjInfo().withBucket("mpc-east-2").withLocation("cn-east-2").withObject("ok.mp4");
ObsObjInfo output = new ObsObjInfo().withBucket("mpc-east-2").withLocation("cn-east-2").withObject("output");

AvParameters avParameters = new AvParameters();
avParameters.setCommon(new Common().withPvc(true));
ArrayList<AvParameters> avParametersArrayList = new ArrayList<>();
avParametersArrayList.add(avParameters);
```

```
CreateTranscodingTaskRequest request
    = new CreateTranscodingTaskRequest().withBody(new CreateTranscodingReq()
        .withInput(input)
        .withOutput(output)
        .withAvParameters(avParametersArrayList)
    );
CreateTranscodingTaskResponse response = getMpcClient().createTranscodingTask(request);
System.out.println("CreateTranscodingTaskResponse=" + response);
```

3 MPC 的视频截图能力帮助视频网站客户快速构建媒体处理平台

场景说明

视频网站对视频画面截图有着多样化的业务诉求，包括视频封面、拖动预览、审核、海报、剧照等。媒体处理服务支持同步截图、异步截图，也支持指定时间点、固定时间间隔截图，有效帮助视频网站客户快速构建媒体处理平台。例如：拖动预览，视频播放过程中，鼠标悬浮在进度条上，可以根据预览画面，拖到进度到指定的位置。

实现原理

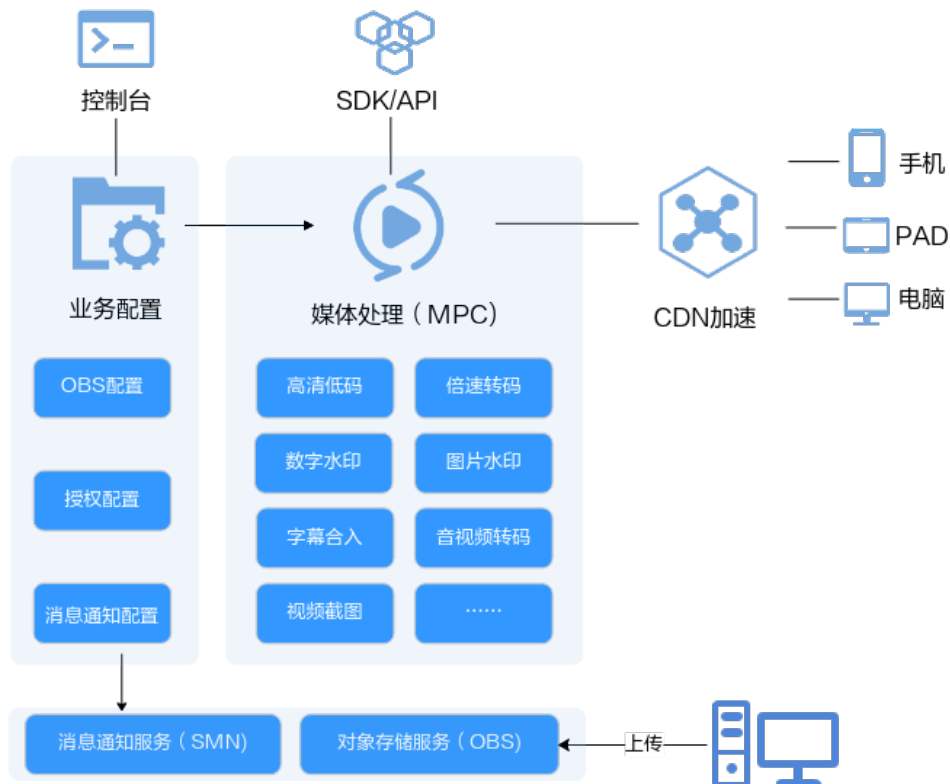
视频网站在构建媒体处理平台时，需要创建一个截图任务管理服务。该服务负责管理视频截图任务，包括创建、查询截图任务等操作。

视频网站截图任务管理服务通过SDK/API调用MPC的视频截图能力，首先MPC视频截图服务从OBS拉取原视频，再按照请求参数要求对原视频进行截图，完成截图后将文件放入OBS指定路径。视频网站的关联业务服务可以从“视频网站截图任务管理服务”获取到截图文件信息，并将截图应用于视频封面、视频播放过程中拖动预览画面、审核等场景中。

表 3-1 服务功能

服务名称	功能
视频网站截图任务管理服务	主要负责视频截图任务的管理，包括创建、查询截图任务等操作。
MPC视频截图服务	负责从OBS拉取原视频，按照任务要求对原视频进行截图，完成后将截图文件放入OBS指定路径。
OBS对象存储服务	用于客户上传和存储媒体文件。

图 3-1 实现原理图



开发时序图

图 3-2 创建截图任务

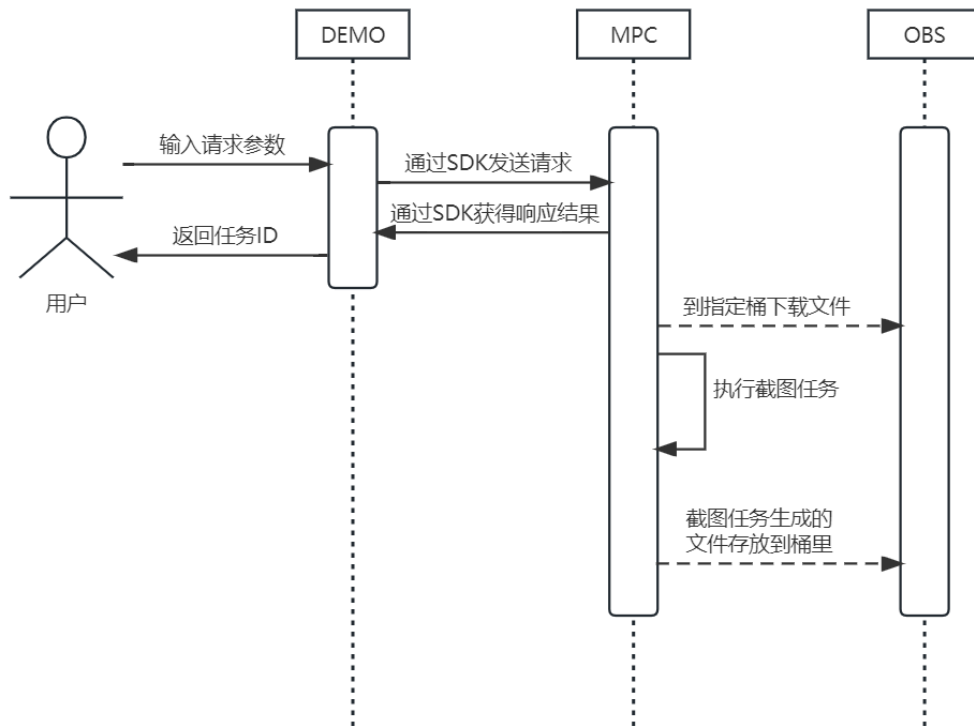


图 3-3 取消截图任务

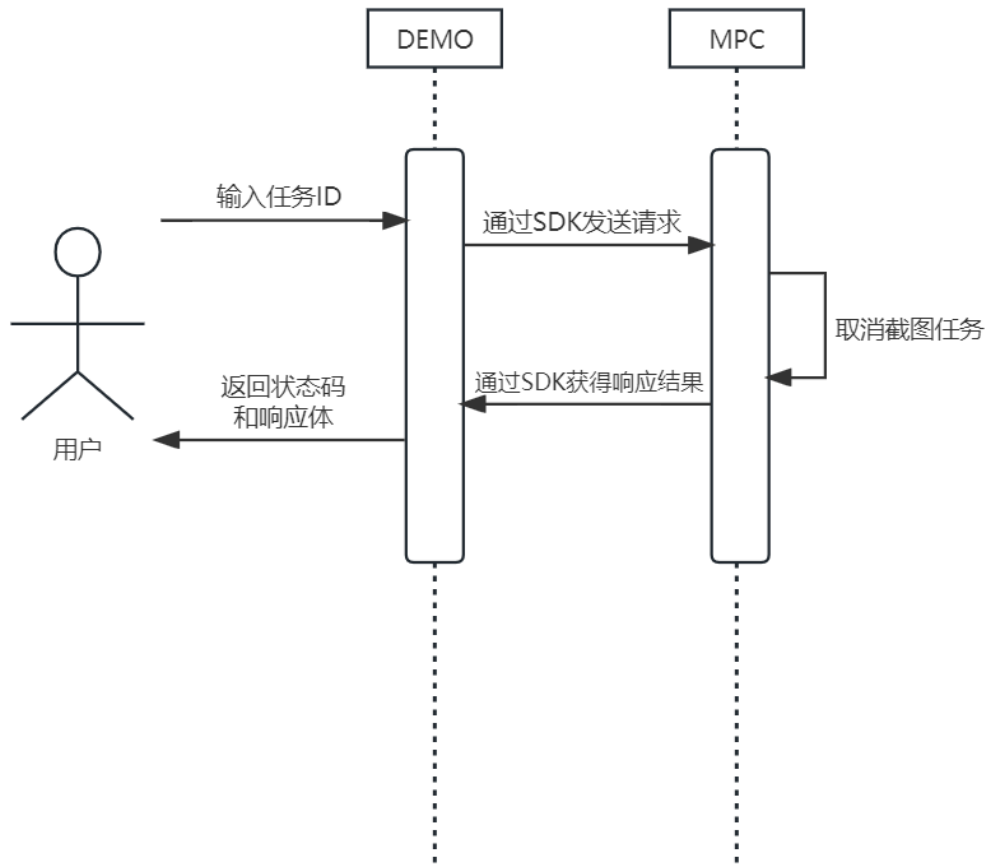
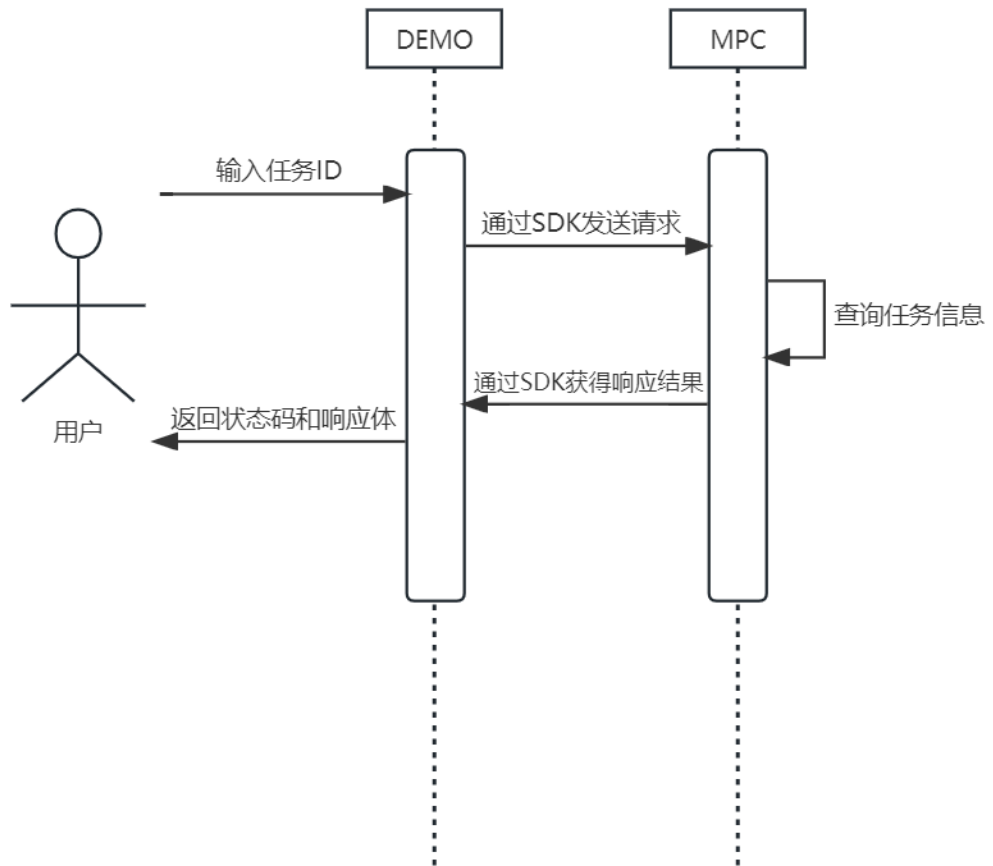


图 3-4 查询截图任务



操作步骤

步骤1 参考[上传音视频文件](#)，上传一个视频文件至OBS桶中。

步骤2 创建截图任务。

示例代码如下所示：

```
// 设置截图输入视频地址
ObsObjInfo input = new ObsObjInfo().withBucket("<example-bucket>").withLocation("<Region ID>").withObject("<example-path/input.mp4>");
ObsObjInfo output = new ObsObjInfo().withBucket("<example-bucket>").withLocation("<Region ID>").withObject("<example-path/output>");

// 创建截图请求
CreateThumbnailsTaskRequest request = new CreateThumbnailsTaskRequest();
CreateThumbReq body = new CreateThumbReq();
List<Integer> listThumbnailParaDots = new ArrayList<>();
listThumbnailParaDots.add(50000);
// 设置截图类型,此处按时间点截图
ThumbnailPara thumbnailParabody = new ThumbnailPara();
// 设置采样类型, 指定时间点截图
thumbnailParabody.withType(ThumbnailPara.TypeEnum.fromValue("DOTS"))
// 设置截图输出文件名
.withOutputFilename("photo")
// 设置采样截图的时间间隔值
.withTime(10)
// 设置采样类型为“TIME”模式的开始时间, 和“time”配合使用
```

```

        .withStartTime(100)
        // 采样类型为“TIME”模式的持续时间,和“time”、“start_time”配合使用,表示从视频文件的第
        “start_time”开始,持续时间为“duration”,每间隔“time”生成一张截图,单位:秒。
        .withDuration(1)
        // 指定时间截图的时间点数组
        .withDots(listThumbnailParaDots)
        // 设置截图文件格式
        .withFormat(1)
        // 设置截图的宽
        .withWidth(96)
        // 设置截图的高
        .withHeight(96);

        body.withThumbnailPara(thumbnailParabody);
        body.withOutput(output);
        body.withInput(input);
        request.withBody(body);

        // 发送截图请求
        CreateThumbnailsTaskResponse response = initMpcClient().createThumbnailsTask(request);
        logger.info(response.toString());    return response.getTaskId();
    
```

步骤3 查询截图任务。

示例代码如下所示:

```

        ListThumbnailsTaskRequest request = new ListThumbnailsTaskRequest();
        List<String> listRequestTaskId = new ArrayList<>();
        listRequestTaskId.add(taskId);
        request.withTaskId(listRequestTaskId);
        ListThumbnailsTaskResponse response = initMpcClient().listThumbnailsTask(request);
        logger.info(response.toString());
    
```

步骤4 查看运行结果。

创建截图任务,返回截图任务ID。

```
{"task_id": "1024"}
```

查询截图任务的状态和结果。

```

{
  "task_array": [
    {
      "task_id": 2528,
      "status": "SUCCEEDED",
      "create_time": 20201118121333,
      "end_time": 20201118121336,
      "input": {
        "bucket": "example-bucket",
        "location": "region01",
        "object": "example-input.ts"
      },
      "output": {
        "bucket": "example-bucket",
        "location": "region01",
        "object": "example-output/example-path"
      },
      "thumbnail_info": [ {
        "pic_name": "9.jpg"
      }, {
        "pic_name": "5.jpg"
      } ]
    }
  ],
  "is_truncated": 0,
  "total": 1
}
    
```

----结束

SDK 集成示例

视频截图功能的详细说明和示例代码，请参考[创建截图任务](#)。

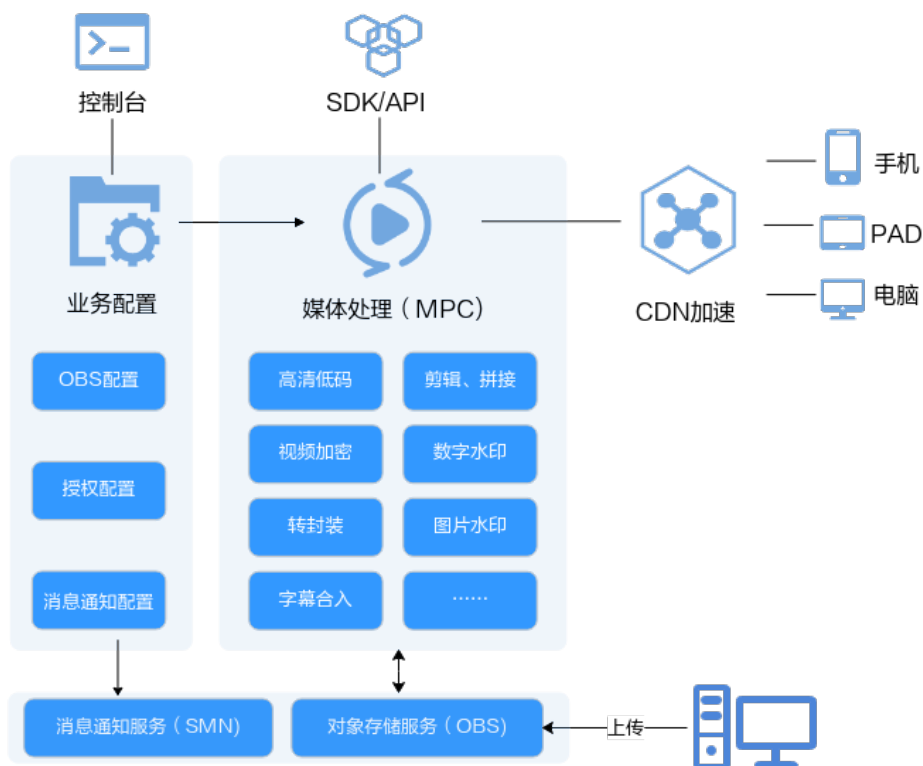
4 MPC 的视频转封装能力帮助在线教育客户实现多终端播放视频

场景说明

媒体处理服务MPC支持将各种主流视频封装格式转换为MP4和HLS格式，实现多终端兼容适配，满足在线教育行业在不同网络条件下的多端播放需求。例如：教育网客户可以在网站提供手机APP观看学习视频。

实现原理

图 4-1 实现原理图



使用限制

- 支持转封装的输入格式有：MP3、MP4、FLV、TS。
- 支持转封装的输出格式有：HLS、MP4。

操作步骤

步骤1 登录[媒体处理控制台](#)。

步骤2 在左侧导航树中，选择“媒体处理 > 转封装”，进入“转封装”页面。

步骤3 单击“创建任务”，进入“创建转封装任务”页面。

图 4-2 转封装任务

转封装 / 创建转封装任务

i 目前支持转封装的输入格式有MP3, MP4, FLV, TS, [了解更多](#)

* 源桶所在区域: cn-north-4

* 输入桶名:

* 源文件:

* 输出桶所在区域: cn-north-4

* 输出桶名:

输出路径:

文件名:

* 输出格式: HLS MP4

分片时长:

步骤4 配置转封装任务参数，如[表4-1](#)所示。

表 4-1 参数说明

参数	说明
源桶所在区域	源视频文件存储的对象存储桶所在的区域。

参数	说明
输入桶名	源视频文件存储的对象存储桶。
源文件	源视频文件的存储路径。
输出桶所在区域	转封装文件存储的对象存储桶所在的区域。
输出桶名	转封装文件存储的对象存储桶。
输出路径	转封装文件的存储路径。
文件名	转封装后的文件名称。
输出格式	转封装格式。暂只支持"HLS"和“MP4”格式。
分片时长	HLS分片的长度，当转封装格式为“HLS”时有效。 取值范围：[2,10]。 默认值：5。

步骤5 单击“确定”，完成创建。

步骤6 在任务列表中，查看执行情况。

当“任务状态”为“处理成功”时，您可以在“输出路径”中获取转封装后的文件。

----结束

SDK 集成示例

转封装功能的详细说明和示例代码，请参考[创建转封装任务](#)。

教育网站开发人员支持接入MPC的SDK核心代码，如下所示：

```

ObsObjInfo input = new ObsObjInfo().withBucket("mpc-
east-2").withLocation("region01").withObject("ok.flv");
ObsObjInfo output = new ObsObjInfo().withBucket("mpc-
east-2").withLocation("region01").withObject("output");
//创建转封装请求
CreateRemuxTaskRequest req = new CreateRemuxTaskRequest()
    .withBody(new CreateRemuxTaskReq().withInput(input).withOutput(output))
    // 设置转封装参数
    .withOutputParam(new RemuxOutputParam()
        //设置转封装格式
        .withFormat("HLS")
        //转成hls切片间隔
        .withSegmentDuration(5));
//发送转封装请求
CreateRemuxTaskResponse rsp = initMpcClient().createRemuxTask(req);
System.out.println(rsp.toString())
    
```


5 修订记录

发布日期	修订记录
2023-11-30	第二次正式发布。 本次更新说明如下： <ul style="list-style-type: none">新增“H.264和H.265高清低码帮助视频网站打造极致体验”章节新增“MPC的视频截图能力帮助视频网站客户快速构建媒体处理平台”章节新增“MPC的视频转封装能力帮助在线教育客户实现多终端播放视频”章节
2022-11-30	第一次正式发布。