

一种游戏数据处理方法、装置及存储介质

本申请要求于 2021 年 08 月 24 日提交中国专利局，申请号为 202110974053.3，申请名称为“一种游戏数据处理方法、装置及存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机计算领域，具体涉及一种游戏数据处理方法、装置及存储介质。

背景技术

目前，在传统的游戏直播场景下，主播用户可以在移动终端上操作并运行二维游戏（即 2D 游戏），并可以通过该移动终端对当前运行的 2D 游戏进行直播录制，进而可以将由该移动终端所实时录制的音视频流作为直播推流进行推送。

显然，在传统的游戏直播场景下，实时录制的音视频流均是在移动终端侧所得到的，以至于在该传统的游戏直播场景下，需要依赖于该主播用户所使用的移动终端的性能，来将编码后的直播流直接推送到远程的直播流。在传统的游戏直播场景下，由于传统的 2D 游戏与云游戏，分别对移动终端具有不同的性能要求，以至于无法直接依赖于传统的游戏直播方式来实现云游戏的直播。因此，在云游戏的直播场景下，如何在不同终端之间实现数据传输时的音视频同步是一个亟待解决的技术问题。

发明内容

根据本申请提供的各种实施例，提供一种游戏数据处理方法、装置及存储介质。

一方面，提供了一种游戏数据处理方法，由计算机设备执行，方法包括：

接收云游戏客户端发送的针对目标云游戏的云游戏直播请求，基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接；

基于直播服务连接获取目标云游戏的游戏图像数据和游戏音频数据，记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳；图像采集时间戳和音频采集时间戳为同一参考系下的时间戳；

在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据；混音音频数据对应的混音时间戳与音频采集时间戳保持一致；及

基于混音音频数据、音频采集时间戳、游戏图像数据和图像采集时间戳，得到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

另一方面，提供了一种游戏数据处理装置，装置包括：

直播请求接收模块，用于接收云游戏客户端发送的针对目标云游戏的云游戏直播请求，基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接；

数据获取模块，用于基于直播服务连接获取目标云游戏的游戏图像数据和游戏音频数据，记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳；图像采集时间戳和音频采集时间戳为同一参考系下的时间戳；

混音处理模块，用于在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据；混音音频数据对应的混音

时间戳与音频采集时间戳保持一致；及

直播流下发模块，用于基于混音音频数据、音频采集时间戳、游戏图像数据和图像采集时间戳，得到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

另一方面，提供了一种计算机设备，包括存储器和一个或多个处理器，所述存储器存储有计算机可读指令，所述计算机可读指令被所述处理器执行时，使得所述一个或多个处理器执行上述游戏数据处理方法的步骤。

一个或多个非易失性可读存储介质，其上存储有计算机可读指令，所述计算机可读指令被一个或多个处理器执行时，使得所述一个或多个处理器实现上述游戏数据处理方法的步骤。

一种计算机程序产品，包括计算机可读指令，所述计算机可读指令被处理器执行时实现上述游戏数据处理方法的步骤。

本申请的一个或多个实施例的细节在下面的附图和描述中提出。本申请的其他特征、目的和优点将从说明书、附图以及权利要求书变得明显。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图1是本申请实施例提供的一种网络架构的结构示意图；

图2是本申请实施例提供的一种在云游戏场景下进行直播推流的场景示意图；

图3是本申请实施例提供的一种游戏数据处理方法的流程示意图；

图4是本申请实施例提供的一种在云游戏场景下针对目标云游戏进行云直播的场景示意图；

图5是本申请实施例提供的一种通过媒体编码进程抓取游戏画面和音频的场景示意图；

图6是本申请实施例提供的一种媒体转发模块的场景示意图；

图7是本申请实施例提供的一种优先级队列的场景示意图；

图8是本申请实施例提供的一种游戏数据处理方法的流程示意图；

图9是本申请实施例提供的一种通过媒体转发模块对收到的音频帧和视频帧进行平滑处理的场景示意图；

图10是本申请实施例提供的一种游戏数据处理装置的结构示意图；

图11是本申请实施例提供的一种计算机设备的结构示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

本申请实施例涉及云技术（cloud technology），云计算（cloud computing）和云游戏（cloud gaming）。云游戏又可称为游戏点播（gaming on demand），是一种以云计算技术为基础的在线游戏技术。云游戏技术使图形处理与数据运算能力相对有限的轻端设备（thin client）能运行高品质游戏。在云游戏场景下，游戏并不在玩家游戏终端（也可以称之为游戏对象对应的游戏终端），而是在云服务器中运行，并由云服务器将游戏场景渲染为视频音频流（即音

视频流），通过网络传输给玩家游戏终端。比如，在本申请实施例中，该云服务器所采用的渲染技术属于一种容器内的渲染技术，这里的容器是指在云服务器中所虚拟得到的云虚拟容器，即本申请实施例可以将云虚拟容器中所渲染得到的视频音频流通过网络传输给玩家游戏终端。这里的玩家游戏终端无需拥有强大的图形运算与数据处理能力，仅需拥有基本的流媒体播放能力与获取玩家（即游戏用户）输入指令并发送给云服务器的能力即可。

在云游戏场景下，玩家（即前述游戏对象）可以通过玩家游戏终端中所运行的游戏客户端与云服务器之间进行数据交互，这样，当云服务器在云端的云虚拟容器中完成对该游戏客户端中的游戏数据的渲染操作之后，可以进一步对渲染所得到的游戏画面和音频进行编码操作，以将编码后的音视频流下发给玩家游戏终端进行解码操作。音视频流可以包括音频流或视频流中的至少一种。

图1是本申请实施例提供的一种网络架构的结构示意图。如图1所示，网络架构可以应用于云游戏直播场景下的数据处理系统。该数据处理系统具体可以包括图1所示的云服务器1000、主播终端集群、直播服务器2000以及观众终端集群。其中，主播终端集群具体可以包括一个或者多个主播终端，多个是指至少两个，主播终端可以为上述具有直播功能的玩家游戏终端。如图1所示，多个主播终端具体可以包括主播终端3000a、主播终端3000b、主播终端3000c、...、主播终端3000n；如图1所示，主播终端3000a、主播终端3000b、主播终端3000c、...、主播终端3000n可以分别与云服务器1000进行网络连接，以便于每个主播终端可以通过该网络连接与云服务器1000之间进行数据交互。

当某个主播终端中运行有具有直播功能的云游戏客户端，且游戏对象使用该云游戏客户端对某款云游戏（即目标云游戏）进行游戏互动时，可以在该云服务器1000侧对该目标云游戏的游戏画面进行渲染，并可以将渲染完成后的游戏画面进行压缩处理，以将编码得到的游戏视频流作为待返回给该主播终端的游戏编码流。游戏对象例如为游戏用户。压缩处理例如为编码处理。

若游戏对象通过具有直播功能的云游戏客户端，向云服务器1000发送针对目标云游戏的云游戏直播请求，则可以将当前访问云游戏客户端的游戏对象统称为主播对象，并可以基于云游戏直播请求在云服务器1000侧与直播服务器2000建立直播服务连接，进而可以通过直播服务连接在云端实现云端推流。如图1所示，云服务器1000可以将云端编码得到的直播数据流推流至直播服务器2000，以使直播服务器2000可以对接收到的直播数据流进行解码处理，以将解码得到的直播数据流所对应的音视频序列进行缓存，以便于后续可以由直播服务器2000所编码得到的转码视频流，下发给位于该虚拟直播间内的观众用户所对应的观众终端。对于不同的观众终端而言，转码视频流是由直播服务器2000基于虚拟直播间内不同观众终端的终端屏幕信息所确定的。终端屏幕信息包括但不限于是终端屏幕的尺寸大小或分辨率中的至少一个。直播服务器2000可以根据不同观众终端的终端屏幕信息，对解码得到的音视频序列进行适配处理，以将适配处理后所得到的转码视频流下发给这些观众终端，以使这些观众终端在对接收到的转码视频流进行解码处理之后，可以在直播显示界面上显示不同尺寸或者不同分辨率的云游戏直播画面。音视频序列包括但不限于是在云服务器1000侧进行混音处理后的音视频序列。

其中，观众终端集群具体可以包括一个或者多个观众终端，多个是指至少两个。如图1所示，多个观众终端具体可以包括观众终端4000a、观众终端4000b、观众终端4000c、...、观众终端4000n；如图1所示，观众终端4000a、观众终端4000b、观众终端4000c、...、观众终端4000n可以分别与直播服务器2000进行网络连接，以便于每个观众终端在运行有观众客户端时，可以通过该网络连接与直播服务器2000之间进行数据交互。比如，这些观众终端在虚拟直播间中请求播放上述主播对象当前正在直播的目标云游戏的直播视频时，可以从该直播服务器2000上拉取主播对象当前正在直播的目标云游戏的转码视频流。其中，这里的观众客户端为在观众终端所运行的直播客户端。

其中，如图1所示的云服务器1000和直播服务器2000均可以是独立的物理服务器，也

可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据或人工智能平台等基础云计算服务的云服务器。

其中，为便于理解，本申请实施例可以在图 1 所示的主播终端集群中选择一个主播终端作为目标主播终端，例如，本申请实施例可以将图 1 所示的主播终端 3000a 作为目标直播终端，该目标主播终端可以集成有具备游戏数据直播功能的云游戏客户端。该云游戏客户端上集成有一种或者多种云游戏，这里的云游戏可以包括但不限于竞技类云游戏、酷跑类云游戏、棋牌类云游戏或博射击类云游戏中的至少一种。

其中，目标主播终端具体可以包括：智能手机、平板电脑、笔记本电脑、桌上型电脑、可穿戴设备、智能家居等携带视频数据处理功能的智能终端。智能家居包括智能电视，视频数据处理功能包括视频数据播放功能。

图 2 是本申请实施例提供的一种在云游戏场景下进行直播推流的场景示意图。其中，为便于理解，这里以图 2 所示的主播终端 A 为上述目标主播终端为例，以阐述该主播终端 A 与云服务器 B 进行数据交互的数据过程。其中，该主播终端 A 上运行有具有直播功能的云游戏客户端。如图 2 所示，当某个游戏对象需要通过该主播终端 A 玩云游戏时，该主播终端 A 可以响应该游戏对象针对前述云游戏客户端的触发操作，以在主播终端 A 上输出包含 N 个云游戏的游戏显示界面，这里的 N 为正整数。游戏对象可以在游戏显示界面上所显示的 N 个云游戏中选择自己感兴趣的云游戏作为目标云游戏，主播终端 A 可以响应针对这 N 个云游戏中的目标云游戏的触发操作，向云服务器 B 发送针对目标云游戏的游戏访问请求。游戏对象例如为用户 U1。

云服务器 B 可以为上述图 1 所对应实施例中的云服务器 1000。如图 2 所示，云服务器 B 可以包含图 2 所示的用于提供云游戏服务的云游戏边缘集群或用于提供云推流服务的云游戏中央集群中的至少一个。

云游戏边缘集群可以包含一个或者多个云游戏边缘服务器，多个是指至少两个。以云游戏边缘集群包含一个云游戏边缘服务器为例，云游戏边缘服务器可以包含但不限于图 2 所示的多个云服务模块，多个云服务模块具体可以包含云游戏实例模块 21a、媒体编码模块 21b 以及媒体传输模块 21c。云游戏边缘服务器中的一个云服务模块可以用于运行图 2 所示的云服务器 B 的一个进程。

云游戏实例模块 21a 所提供的游戏进程，可以用于在云端运行目标云游戏的游戏实例，游戏实例所对应的游戏进程运行在云服务器 B 为主播终端 A 的客户端环境系统所配置的云虚拟容器中。运行在云虚拟容器中的云游戏实例是以云计算为基础的游戏方式。在云游戏的运行模式下，所有游戏都在云服务器 B 端运行，并可以将渲染完毕后的游戏画面压缩后通过网络传送给用户 U1。比如，主播终端 A 上所运行的云游戏客户端可以实时检测用户的触发事件，如触屏事件、键盘鼠标事件或摇杆事件等中的至少一个，然后通过网络将检测到的相应触发事件的指令传输给图 2 所示的云服务器 B，以达到远程操作云虚拟容器中的游戏进程运行云游戏实例的目的。运行在云虚拟容器中的云游戏实例是指目标云游戏所对应的游戏实例。

媒体编码模块 21b 所提供的媒体编码进程可以用于抓取云游戏实例模块 21a 所运行的云游戏实例的游戏画面和音频，并对抓取到的游戏画面和音频进行音视频编码处理，将音视频编码处理后所得到的音视频流传输到图 2 所示的媒体传输模块 21c。云游戏实例例如为游戏进程所对应的游戏实例，比如，云游戏实例可以为 Android 实例。

在本申请实施例中，媒体编码模块 21b 所对应的媒体编码进程和云游戏实例模块 21a 所对应的游戏进程可以运行在同一云虚拟容器中，即本申请实施例可以在运行云游戏实例的云虚拟容器中进行容器内的数据渲染，进而可以从根源上降低游戏画面渲染时的渲染时延。此外，媒体编码模块 21b 也可以接收由图 2 所示的媒体传输模块 21c 所转发的用户 U1 通过主播终端 A 所发送的语音输入操作或触摸操作等，并可以将接收到的语音输入操作或触摸操作等注入到云游戏实例中。

以媒体传输模块 21c 所转发的操作为用户 U1 通过主播终端 A 所执行的触摸操作为例，图 2 所示的媒体编码模块 21b 在将该触摸操作在 T1 时刻注入到云游戏实例模块 21a 中的云游戏实例中时，媒体编码模块 21b 还可以用于在该 T1 时刻的下一时刻抓取到云游戏实例针对触摸操作进行响应时的游戏画面和音频。本申请实施例中，可以将由该媒体编码模块 21b 所定时抓取到的游戏画面统称为游戏图像数据，并将由该媒体编码模块 21b 所定时抓取到的音频统称为游戏音频数据。T1 时刻的下一时刻例如为 T2 时刻。游戏画面例如为游戏视频帧数据，音频例如为游戏背景音或者背景音频帧数据中的至少一种。

媒体传输模块 21c 可以用于接收用户 U1 发送的云游戏访问请求，并可以基于接收到的云游戏访问请求对该用户 U1 进行鉴权，在鉴权成功的情况下，允许用户 U1 通过主播终端 A 访问云服务器 B，并在云服务器 B 与用户 U1 所对应的云游戏客户端之间建立游戏服务连接。本申请实施例中，可以将该媒体传输模块 21c 所对应的进程统称为云服务器 B 中的媒体传输进程，媒体传输进程可以运行在独立于云虚拟容器的另一虚拟容器中，本申请实施例中，可以将用于进行数据转发处理和混音处理的另一虚拟容器统称为辅助虚拟容器。

当媒体传输模块 21c 所对应的媒体传输进程接收到接收媒体编码模块 21b 所对应的媒体编码进程发送的音视频流时，发送的音视频流例如为音视频流 100a，可以基于主播终端 A 的终端屏幕信息，对音视频流 100a 进行音视频编码，编码得到的音视频流可以统称为游戏编码流，可以将游戏编码流通过网络传输到用户 U1 所对应的主播终端 A。编码得到的音视频流例如为音视频流 100b，音视频流 100b 中携带媒体编码模块 21b 按照云游戏编码标准所编码得到的一路视频流。其中，主播终端 A 在接收到云服务器 B 通过媒体传输模块 21c 所返回的游戏编码流时，可以对其进行解码处理，以得到该游戏编码流所对应的音视频序列，进而可以在主播终端 A 的云游戏显示界面上显示用户 U1 所直播的目标云游戏的云游戏画面。

用户 U1 在玩上目标云游戏的过程中，可以通过运行云游戏客户端的主播终端 A 实时检测用户 U1 的语音输入操作，将语音输入操作所对应的用户 U1 的语音数据通过游戏服务连接发送给媒体传输模块 21c。当媒体传输模块 21c 接收到用户 U1 的语音数据时，用户 U1 的语音数据可以统称为主播对象的待处理音频数据，可以通过媒体传输模块 21c 所对应的媒体传输进程将待处理音频数据转发到图 2 所示的媒体编码模块 21b，以使媒体编码模块 21b 将待处理音频数据注入到云游戏实例中。

在云游戏直播场景下，媒体编码模块 21b 还可以按照云直播编码标准开启另一路视频编码，以按照云直播编码标准所指示的固定帧率、帧组长度等编码得到另一路视频流，该另一路视频流为不同于按照云游戏编码标准所编码得到的视频流。比如，这里的帧组长度用于描述两个 I 帧之间的间隔，比如，帧组长度可以为 120 帧，这里的固定帧率可以为 60 帧，这意味着在每 2 秒就会出现一次 I 帧。

如图 2 所示，在云游戏直播场景下，媒体传输模块 21c 还可以在接收到用户 U1 的语音数据（即前述待处理音频数据）时，将待处理音频数据和由媒体编码模块 21b 所抓取的音频（即上述游戏音频数据）进行实时混音处理，以得到混音音频数据。其中，由于用于采集前述待处理音频数据的主播终端 A 与用于采集该游戏音频数据的云服务器 B 可能使用的是不同参考系，以至于会导致不同参考系下所采集到的游戏音频数据的时间戳与待处理音频数据的时间戳之间存在一定的时间间隔。所以，为确保后续在观众终端所播放的音视频的同步，本申请实施例提出，可以在云游戏直播场景下，将媒体编码模块 21b 所抓取的音频（即媒体编码模块 21b 按照固定音频采集时间间隔所定时采集到的游戏音频数据）的时间戳（即音频采集时间戳）作为参考时间戳，进而使得混音处理后所得到的混音音频数据的混音时间戳与前述游戏音频数据的音频采集时间戳保持一致。

云服务器 B 可以通过媒体传输模块 21c 将混音处理后的混音音频数据、混音时间戳（即前述音频采集时间戳）、游戏图像数据或图像采集时间戳等中的至少一个，转发给图 2 所示的媒体转发模块 22a。如图 2 所示，媒体转发模块 22a 可以部署在用于提供云推流服务的云游戏中央集群中，云游戏中央集群可以包含一个或者多个云游戏中央服务器，多个是指至少两

个,每个云游戏中央服务器均可以集成部署有媒体转发模块 22a,进而可以通过媒体转发模块 22a 将媒体传输模块 21c 所发送的音视频缓存至优先级队列,以便于后续可以按照前述固定时间间隔从媒体转发模块 22a 所对应的优先级队列中进行音视频包的获取,进而可以将获取到的音视频包进行音视频编码处理,以编码得到的目标云游戏的直播数据流。

云服务器 B 可以通过媒体转发模块 22a 将直播数据流推流至直播服务器 C,以使直播服务器 C 在接收到与主播对象处于同一虚拟直播间中的观众用户的直播拉流请求的情况下,实时将直播数据流下发至图 2 所示的观众终端,例如观众终端 D1 或观众终端 D2 中的至少一个。各观众终端中所运行的直播客户端,均可以基于直播数据流中的音频采集时间戳和图像采集时间戳,对混音音频数据和游戏图像数据进行同步播放,进而可以在观众终端提升混音后的音视频的同步播放效果。

如图 2 所示,直播服务器 C 可以包含图 2 所示的直播边缘服务器 201a、直播源站服务器 202a 以及直播转发服务器 203a。直播边缘服务器 201a 可以用于接收推送来的直播数据流(支持 rtmp,hls,webrtc 等协议),并可以将直播数据流转发到直播源站服务器 202a。直播源站服务器 202a 可以为源站集群中的一个服务器,直播源站服务器 202a 可以用于缓存最近一段时间所收到的直播数据流,并可以用来对直播数据流进行以下后处理,比如转码处理、录制处理或鉴黄处理中的至少一种处理。此外,直播源站服务器 202a 也可以接收直播转发服务器 203a 转发的观众终端所发送的用于请求拉流的命令。直播源站服务器 202a 例如为 OriginServer。直播转发服务器 203a 可以为布署在全国各地的边缘节点,例如为 CDN 节点,直播转发服务器 203a 可以用来缓存一些热门的媒体文件,还可以用来转发直播源站服务器 202a 进行后处理所得到的媒体流,并可以将媒体流下发各观众终端所对应的观众对象,观众对象可以为与主播对象处于同一虚拟直播间的观众用户(player)。

此外,当某个观众用户请求观看某个云游戏的直播视频,但直播转发服务器 203a 中没有缓存该云游戏的直播视频时,直播转发服务器 203a 可以向直播源站服务器 202a 进行拉流,以将拉流到的直播视频缓存到本地,以下发到该观众用户对应的观众终端。云服务器获取游戏图像数据和游戏音频数据,并对游戏音频数据和待处理音频数据进行混音处理的具体过程可以参见如下图 3 至图 8 所对应的实施例。

在一些实施例中如图 3 所示,提供了一种游戏数据处理方法的流程示意图。如图 3 所示,该方法可以由计算机设备执行,计算机设备可以为云服务器,云服务器可以为图 2 所对应实施例中的云服务器 B。计算机设备也可以为玩家游戏终端,玩家游戏终端可以为图 2 所对应实施例中的主播终端 A。本申请实施例所涉及的方法可以由玩家游戏终端执行,也可以由该玩家游戏终端所对应的云服务器执行,还可以由玩家游戏终端和云服务器共同执行。本实施例以该方法由云服务器执行为例,阐述在云服务器中对游戏音频数据和待处理音频数据进行混音处理的具体过程。其中,该方法至少可以包括以下步骤 S101-步骤 S104:

步骤 S101,接收云游戏客户端发送的针对目标云游戏的云游戏直播请求,基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接。

具体的,计算机设备(例如,云服务器)可以在云游戏的直播推流场景(即在云端对目标云游戏进行直播的云游戏场景)下,接收云游戏客户端发送的针对目标云游戏的云游戏直播请求;

其中,云游戏直播请求是云游戏客户端响应于主播对象针对直播指示控件执行的触发操作所生成的;进一步的,计算机设备(例如,云服务器)可以基于云游戏直播请求中所携带的游戏启动标识,向与云游戏客户端相关联的直播服务器发送直播认证请求,以使直播服务器可以基于直播认证请求,为云游戏客户端配置用于直播目标云游戏的直播推流地址;进一步的,计算机设备(例如,云服务器)可以接收直播服务器返回的直播推流地址,在将直播推流地址返回给云游戏客户端时,触发建立云服务器与直播服务器之间的直播服务连接。

图 4 是本申请实施例提供的一种在云游戏场景下针对目标云游戏进行云直播的场景示意图。其中,如图 4 所示的游戏显示界面 400a 为用户 U1 当前正在玩的目标云游戏的显示界面。

目标云游戏为用户 U1 在图 4 所示的游戏终端 41a 中所运行的云游戏客户端上所选择的自己感兴趣的云游戏。在云游戏直播场景下，用户 U1 可以为需要对目标云游戏进行直播的主播对象，主播对象可以理解为通过游戏启动标识访问云游戏客户端的游戏对象即图 4 所示的用户 U1。游戏启动标识例如可以为云游戏账号。

在一些实施例中，如图 4 所示，云服务器 42a 可以接收用户 U1 发送的云游戏直播请求（即开始直播云游戏的信息流），也可以接收用户 U1 发送的云游戏结束请求。如图 4 所示，当用户 U1 需要退出云游戏时，可以触发用于退出云游戏的退出控件，游戏终端 41a 可以响应针对退出控件的触发操作，生成能够向云服务器 42a 发送的用于结束当前正在直播的云游戏的云游戏结束请求。当云服务器 42a 接收到针对目标云游戏发送的云游戏结束请求（即结束直播云游戏的信息流）时，可以中断与云直播服务器 43a 进行数据交互的直播服务连接。

步骤 S102，基于直播服务连接获取目标云游戏的游戏图像数据和游戏音频数据，记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳。

其中，云服务器在通过媒体编码进程所指示的数据采集组件，实时抓取到游戏画面和音频的情况下，为确保在云游戏的直播推流场景下提供稳定的帧率，本申请实施例中，可以将实时抓取到的游戏画面（即游戏视频帧数据）和音频（即游戏背景音），缓存到云内存（即 buffer）中，以便于后续可以从云内存中通过定时器以固定采集间隔抓取相应的数据。本申请实施例中，可以在采集端将通过定时器所采集到的游戏画面（即缓存在云内存中的游戏视频帧数据）统称为游戏图像数据，并将定时器所记录到的用于采集该游戏图像数据的时间戳统称为图像采集时间戳。

本申请实施例中，可以在采集端将通过定时器所采集到的游戏背景音统称为游戏音频数据，并将定时器所记录到的用于采集该游戏音频数据的时间戳统称为音频采集时间戳。为确保后续在观众终端所播放的游戏背景音与游戏画面的音画同步的效果，需要确保图像采集时间戳和音频采集时间戳为同一参考系下的时间戳，以避免数据在云端的多个云服务模块中进行传输所造成的传输延迟。

其中，可以理解的是，由于直播需要稳定的帧率，所以，本申请实施例提出可以预先将通过上述数据采集组件所抓取到的游戏画面和音频缓存到云内存中，以便于后续可以通过定时器，从云内存中以固定时间间隔抓取相应帧率的音视频数据，以进行编码。具体的，如图 5 所示，云服务器可以通过该媒体编码进程 51b 所指示的定时器，从上述云内存所存储的游戏视频帧数据中采集与第一采集时间间隔（例如，每 33ms）所指示的第一帧率（例如，30 帧/s）相匹配的图像帧，并可以将采集到的图像帧作为目标云游戏的游戏图像数据，并可以在定时器所指示的参考系下，记录游戏图像数据对应的图像采集时间戳；进一步的，云服务器还可以通过定时器，从云内存所存储的背景音频帧数据中采集与第二采集时间间隔（例如，10ms）所指示的第二帧率（例如，100 帧/s）相匹配的音频帧，将采集到的音频帧作为目标云游戏的游戏音频数据，并在定时器所指示的同一参考系下，记录游戏音频数据对应的音频采集时间戳。第一采集时间间隔、第一帧率、第二采集时间间隔以及第二帧率可以预设或根据需要设置，例如，第一采集时间间隔为 33ms，第一帧率为 30 帧/s，第二采集时间间隔为 10ms，第二帧率为 100 帧/s。

其中，应当理解，由于编解码设备对视频帧的编码性能的要求，会高于编解码设备对音频帧的编码性能的要求，且用户对音频的感知需求较大。所以，为避免定时器采集到过多的视频帧，以及出现音频的卡顿，本申请实施例提出可以在进行音视频的采集过程中，设定该第二采集时间间隔记录小于第一采集时间间隔，这样，该第二采集时间间隔所指示的第二帧率就会大于第一采集时间间隔所指示的第一帧率。基于此，本申请实施例通过在音视频的采集过程中，设置不同的帧率可以从根源上减少在单位采集时间间隔内（即一个第一采集时间间隔内）采集到过多的视频帧的情况，从而可以减少由于过多的视频帧所造成的视频编码性能的过渡消耗。

步骤 S103，在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数

据和待处理音频数据进行混音处理，得到混音音频数据。

其中，混音音频数据对应的混音时间戳与音频采集时间戳保持一致；音频采集时间戳也可以称为采集音频时间戳。

具体的，云服务器在获取到云游戏客户端发送的主播对象的待处理音频数据时，可以对待处理音频数据进行解码处理，得到主播对象的待处理音频帧以及待处理音频帧的录制时间戳；云服务器可以基于游戏音频数据对应的第二采集时间间隔，获取游戏音频数据中的音频帧，并可以将获取到的音频帧作为参考音频帧，且根据游戏音频数据对应的采集音频时间戳，确定参考音频帧的混音时间戳；云服务器可以基于混音时间戳调整录制时间戳，并可以基于调整后的录制时间戳，对参考音频帧和待处理音频帧进行混音处理，得到与采集音频时间戳保持一致的混音时间戳对应的混音音频帧，将得到的混音音频帧和混音音频帧对应的混音时间戳作为目标云游戏的混音音频数据。

步骤 S104，基于混音音频数据、音频采集时间戳、游戏图像数据和图像采集时间戳，得到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

其中，云服务器包含基于云游戏直播请求所启动的媒体转发进程；媒体转发进程用于保持云服务器与直播服务器之间的直播服务连接；媒体转发进程可以为图 5 所对应实施例中的媒体转发进程 52a。

在一些实施例中，云服务器可以获取混音音频数据，从混音音频数据中获取混音音频帧以及混音音频帧对应的音频采集时间戳，基于音频采集时间戳对混音音频帧进行排序处理，将排序处理后的混音音频帧，写入与媒体转发进程相关联的音频缓存队列；云服务器可以获取游戏图像数据，从游戏图像数据中获取图像帧以及图像帧对应的图像采集时间戳，基于图像采集时间戳对图像帧进行排序处理，将排序处理后的图像帧，写入与媒体转发进程相关联的视频缓存队列；云服务器可以获取与媒体转发进程相关联的平滑获取线程，通过平滑获取线程从视频缓存队列中，获取与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，通过平滑获取线程从音频缓存队列中，获取与第二采集时间间隔所指示的第二帧率相匹配的音频数据包；云服务器可以将视频数据包和音频数据包进行打包处理，得到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

在本申请实施例中，云服务器可以在接收到云游戏客户端发送的针对目标云游戏的云游戏直播请求时，基于云游戏直播请求，建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接；云服务器可以基于直播服务连接，获取目标云游戏的游戏图像数据和游戏音频数据，记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳；其中，为提高数据传输延迟过程中音视频同步的准确性，本申请实施例中，在云游戏直播场景下，确保图像采集时间戳和音频采集时间戳为同一参考系下的时间戳，进而可以基于同一参考系下的这些时间戳实现音视频同步的可靠性；进一步的，计算机设备在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据；在云游戏直播场景下，主播对象的声音是由云游戏客户端录制得到的，游戏背景音是由云服务器在云端抓取得到的，考虑到云游戏客户端与云服务器之间存在数据传输，且观众用户所能感知到的是游戏画面和游戏背景音的同步，所以，在对主播对象的声音（即待处理音频数据）和游戏背景音进行混音的过程中，可以将游戏背景音的音频采集时间戳作为参考，以使得混音后所得到的混音音频数据对应的混音时间戳与音频采集时间戳保持一致；进一步的，计算机设备可以基于混音音频数据、音频采集时间戳、游戏图像数据和图像采集时间戳，得到目标云游戏对应的直播数据流，进而可以将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端，以使直播客户端基于直播数据流中的音频采集时间戳和图像采集时间戳，对混音音频数据和游戏图像数据进行同步播放。由此可见，在云游戏直播场景下，云游戏客户端并不会直接向直播服务器推送直播数据流，而是通过该云服务器来向直播服务器进行直播数据流的推送，从而可以利用云服务器强大的计算能

力，降低对云游戏客户端侧的性能消耗。另外，通过在云服务器侧进行基于音频采集时间戳和图像采集时间戳进行音视频的同步，不仅可以解决直播推流过程中因抖动、乱序等所造成的音视频不同步的现象，还可以实现多路混音的同步及卡顿问题，进而可以实现音视频同步的可靠性，这样，当在直播客户端中基于直播数据流中的音频采集时间戳和图像采集时间戳，对混音音频数据和游戏图像数据进行同步播放时，可以提升混音后的音视频的同步播放效果。

在一些实施例中，在接收云游戏客户端发送的针对目标云游戏的云游戏直播请求之前，该方法还包括：在获取到云游戏客户端针对目标云游戏发送的游戏访问请求时，基于游戏访问请求对云游戏客户端对应的游戏启动标识进行鉴权处理，得到游戏启动标识对应的鉴权结果；游戏访问请求中携带云游戏客户端所在的客户端环境系统；若鉴权结果指示云游戏客户端具有通过游戏启动标识访问目标云游戏的权限，则建立云服务器与云游戏客户端之间的游戏服务连接，且基于游戏服务连接为云游戏客户端所在的客户端环境系统配置用于运行目标云游戏的云虚拟容器；在云虚拟容器中启动目标云游戏的游戏进程，通过游戏进程运行目标云游戏。

其中，玩家游戏终端可以用于作为该云游戏客户端的前端，云服务器可以用于作为云游戏客户端的后台，当云服务器获取到玩家游戏终端中的游戏对象发送的游戏访问请求之后，可以提取该游戏访问请求中所携带的云游戏客户端对应的游戏启动标识，对游戏启动标识进行鉴权处理，得到游戏启动标识对应的鉴权结果。若云服务器确认鉴权成功，即游戏用户所对应的云游戏客户端具有通过游戏启动标识访问目标云游戏的权限时，则可以建立云服务器与云游戏客户端之间的游戏服务连接，且基于游戏服务连接为云游戏客户端所在的客户端环境系统配置用于运行目标云游戏的云虚拟容器。

其中，游戏访问请求中还可以携带云游戏客户端所在的客户端环境系统；客户端环境系统可以包含但不限于安卓系统，比如，客户端环境系统还可以包含在玩家游戏终端中所运行的鸿蒙系统或苹果系统等。

其中，云服务器可以用于运行多个云虚拟容器，每个云虚拟容器均为一个容器实例。每个容器实例中，均可以运行与相应玩家游戏终端中的客户端环境系统相匹配的远程移动游戏运行环境。在云服务器中为每个玩家用户终端所虚拟出的远程移动游戏运行环境可以统称为云游戏环境系统，例如，云游戏环境系统可以为在云服务器中为每个用户终端所虚拟出的远程的安卓系统，任意一个云虚拟容器，均可以用于为对应玩家游戏终端提供基础的安卓运行环境。比如，云服务器在为主播终端 A 配置好用于运行目标云游戏的云虚拟容器时，可以在云虚拟容器中启动目标云游戏的游戏进程，通过游戏进程运行目标云游戏。

其中，当游戏用户在玩家游戏终端中触发目标云游戏的游戏图标时，可以在云虚拟容器中启动与游戏图标相关联的游戏进程，进而可以在云虚拟容器中为游戏进程分配相应的帧缓冲区，以利用容器内的渲染技术实现内部渲染，以渲染得到用于显示在玩家游戏终端中的游戏显示界面。游戏显示界面中可以包含具有操控游戏角色执行相应业务功能的业务控件。

其中，云游戏客户端对应的游戏显示界面，还可以包含用于指示主播对象进行云游戏直播的直播指示控件。主播对象可以为通过游戏启动标识访问（即接入）目标云游戏的游戏对象。

本实施例中，在鉴权结果指示云游戏客户端具有通过游戏启动标识访问目标云游戏的权限的情况下，建立云服务器与云游戏客户端之间的游戏服务连接，从而提高了建立的游戏服务连接的安全性。

在一些实施例中，云游戏客户端对应的游戏显示界面上包含用于指示主播对象进行云游戏直播的直播指示控件；基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接的步骤包括：接收云游戏客户端发送的针对目标云游戏的云游戏直播请求；云游戏直播请求，是云游戏客户端响应于主播对象针对直播指示控件执行的触发操作所生成的。

具体地，如图 4 所示，游戏终端 41a 可以根据用户 U1 的直播需求，在游戏显示界面 400a

上输出包含用于指示该主播对象进行云游戏直播的直播指示控件的子页面，比如，这里的子页面可以为独立于该游戏显示界面 400a 的悬浮窗口；如图 4 所示，用户 U1 可以在该悬浮窗口上针对用于开启直播的“直播指示控件”执行触发操作。此时，该游戏终端 41a 可以响应于该用户 U1 针对直播指示控件执行的触发操作，生成图 4 所示的云游戏直播请求，该云游戏直播请求中携带在该用户 U1 访问该云游戏客户端的云游戏账号，这里的云游戏账号可以为该用户 U1 通过实名认证后在云服务器 42a 中所注册得到的账号。

本实施例中，游戏显示界面上包含直播指示控件，从而可以指示主播对象进行云游戏直播，提高了交互效率，从而提高了资源利用率。

在一些实施例中，云游戏直播请求中所携带游戏启动标识；基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接包括：基于游戏启动标识向与云游戏客户端相关联的直播服务器发送直播认证请求，以使直播服务器基于直播认证请求，为云游戏客户端配置用于直播目标云游戏的直播推流地址；接收直播服务器返回的直播推流地址，在将直播推流地址返回给云游戏客户端时，触发建立云服务器与直播服务器之间的直播服务连接。

具体地，如图 4 所示，云服务器 42a 在接收到运行有云游戏客户端的游戏终端 41a 发送的云游戏直播请求的情况下，可以进一步基于该云游戏直播请求中所携带的云游戏账号，生成图 4 所示的直播认证请求，并可以将该直播认证请求发送给图 4 所示的直播服务器 43a，以使直播服务器 43a 可以基于该直播认证请求在直播系统中，查找是否存在与该云游戏账号绑定的直播账号。如果存在，则可以基于获取到的该直播认证请求，为游戏终端 41a 中所运行的云游戏客户端配置图 4 所示的直播推流地址。

其中，应当理解，云服务器 42a 和直播服务器 43a 均可以用于存储直播账号与云游戏账号之间的绑定关系。可以理解的是，这里的直播账号为该用户 U1 通过某个直播客户端，在与该直播客户端相关联的该直播服务器 43a 中所注册的用于进行直播的账号。

如图 4 所示，云服务器 42a 可以接收直播服务器 43a 所返回的直播推流地址，并可以将该直播推流地址返回给游戏终端 41a 中所运行的云游戏客户端，以在该云游戏客户端中对当前运行的目标云游戏进行直播处理。应当理解，在本申请实施例中，云服务器 42a 还可以在将直播推流地址返回给云游戏客户端时，触发建立云服务器 42a 与直播服务器 43a 之间的直播服务连接，以便于后续在用户 U1 对目标云游戏进行直播的过程中，可以使得云服务器 42a 基于该直播服务连接与云直播服务器 43a 进行数据交互。

本实施例中，云服务器在将直播推流地址返回给云游戏客户端时，触发建立云服务器与直播服务器之间的直播服务连接，从而使得云服务器基于直播服务连接与云直播服务器进行数据交互，提高了数据交互的效率。

在一些实施例中，目标云游戏是由云虚拟容器中的游戏进程运行的；云虚拟容器是由云服务器为云游戏客户端所在的客户端环境系统所配置的；基于直播服务连接获取目标云游戏的图像数据和游戏音频数据，记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳，包括：基于直播服务连接，在云虚拟容器中启动与游戏进程相关联的媒体编码进程；通过媒体编码进程所指示的数据采集组件，抓取与目标云游戏相关联的游戏背景音频帧数据和游戏视频帧数据，并可以将抓取到的背景音频帧数据和游戏视频帧数据，存放至与数据采集组件相关联的云缓存空间；云缓存空间为独立于云虚拟容器的云内存；通过媒体编码进程所指示的定时器，从云内存所存储的游戏视频帧数据中采集与第一采集时间间隔所指示的第一帧率相匹配的图像帧，并可以将采集到的图像帧作为目标云游戏的图像数据，并在定时器所指示的参考系下，记录游戏图像数据对应的图像采集时间戳；通过定时器，从云内存所存储的背景音频帧数据中采集与第二采集时间间隔所指示的第二帧率相匹配的音频帧，将采集到的音频帧作为目标云游戏的音频数据，并在定时器所指示的同一参考系下，记录游戏音频数据对应的音频采集时间戳；第二采集时间间隔记录小于第一采集时间间隔，且第二帧率大于第一帧率。

其中，游戏进程所运行的目标云游戏可以为上述图 2 所对应实施例中的云游戏实例模块 21a 所运行的云游戏的云游戏实例。媒体编码进程可以为图 2 中的媒体编码模块 21b 提供的与目标云游戏相关联的关联进程。

如图 5 所示，提供了一种通过媒体编码进程抓取游戏画面和音频的场景示意图。如图 5 所示的云服务器可以为上述图 4 所对应实施例中的云服务器 42a，且图 5 所示的主播终端可以为上述图 4 所对应实施例中的游戏终端 41a。

其中，如图 5 所示的云游戏实例 51a 所对应的游戏进程可以运行在由该云服务器为云游戏客户端所在的客户端环境系统所配置的云虚拟容器中。如图 5 所示，当云服务器与直播服务器之间建立直播服务连接时，该云服务器可以基于该直播服务连接在运行云游戏实例 51a 的云虚拟容器中一并启动与游戏进程相关联的媒体编码进程 51b。其中，应当理解，这里的媒体编码进程 51b 可以为上述图 2 所对应实施例中的媒体编码模块 21b 所提供的与目标云游戏相关联的关联进程。

以客户端环境系统为安卓系统为例。如图 5 所示，这里媒体编码进程 51b 可以用于通过数据采集组件（例如，Android VirtualDisplay，即 Android 系统提供的一个组件）实时抓取云游戏实例 51a 的游戏画面和音频，这里的游戏画面指抓取到的目标云游戏的游戏视频帧数据，这里的音频指目标云游戏的背景音频帧数据，即该背景音频帧数据为实时抓取到的云游戏中的游戏背景音。数据采集组件可以用于抓取云端所提供的虚拟屏幕或者虚拟窗口的画面，即这里的虚拟屏幕或者虚拟窗口，可以理解为图 5 所示的主播终端内的安卓窗口的物理屏幕的拷贝。通过数据采集组件抓取到的游戏画面和音频，可以存储在与该云虚拟容器相关联的云内存（即云缓存空间）中，即无需在云内存中将其渲染绘制出来。

在本申请实施例中，媒体编码进程 51b 可以进行两路编码操作，一路编码操作为按照云游戏编码标准，对抓取到的游戏画面和音频进行音视频编码，以编码得到图 5 所示的游戏编码流 1'。另一路编码操作为按照云直播编码标准，对定时采集到的具有相应帧率的游戏画面和音频进行音视频编码，以编码得到图 5 所示的直播数据流 2'。这里之所以需要进行两路编码操作，是因为当媒体传输模块出现传输中断时，会在中断许可时长内暂时中断主播终端侧的游戏服务连接，而并未中断直播服务连接，即观众终端仍可以在中断许可时长内看到主播终端所推流过来的直播视频。这样可以从根本上减少发生避免如果玩家出现因为网络异常，中断许可时长内暂时断开游戏服务连接，导致直播也会中断的现象。

如图 5 所示，云服务器还可以将编码得到的游戏编码流 1'，作为用户 U1 针对主播终端所执行的触摸事件流的响应信息。云服务器还可以将编码得到的直播数据流 2' 推流给图 5 所示的直播服务器。

本实施例中，将实时抓取到的游戏画面和音频（即游戏背景音），缓存到云内存中，从云内存中通过定时器以固定采集间隔抓取相应的数据，从而可以在云游戏的直播推流场景下提供稳定的帧率，提高了直播流畅性。

在一些实施例中，该方法还包括：在云虚拟容器中运行与媒体编码进程相关联的第一编码线程；获取第一编码线程所指示的云游戏编码标准，通过云游戏编码标准对抓取到的背景音频帧数据进行音频编码，得到目标云游戏的第一音频编码流；通过云游戏编码标准对抓取到的游戏视频帧数据进行视频编码，得到目标云游戏的第一视频编码流；将第一音频编码流和第一视频编码流作为目标云游戏的游戏编码流，将游戏编码流返回给云游戏客户端，以使云游戏客户端对游戏编码流进行解码处理，得到用于在云游戏客户端中进行同步播放的背景音频帧数据和游戏视频帧数据。

其中，以图 5 为例，由于云游戏编码标准无需在玩家玩目标云游戏的过程中，为玩家提供固定帧率和固定帧组长度的音视频数据包。所以，云服务器可以在接收到主播对象针对云游戏客户端执行的触摸事件流时，响应于该触摸事件流，在云虚拟容器中运行与该媒体编码进程 51b 相关联的第一编码线程，并可以通过该第一编码线程所指示的云游戏编码标准，对实时抓取到的前述背景音频帧数据进行音频编码，以编码得到目标云游戏的第一音频编码流，

与此同时，云服务器还可以通过云游戏编码标准，对抓取到的游戏视频帧数据进行视频编码，以得到目标云游戏的第一视频编码流。所以，对于图 5 的游戏编码流 1' 而言，可以包含目标云游戏的第一音频编码流和目标云游戏的第一视频编码流。如图 5 所示，主播终端在接收到该游戏编码流 1' 时，可以基于快速对该游戏编码流 1' 进行解码处理，以得到用于在云游戏客户端中进行同步播放的背景音频帧数据和游戏视频帧数据。该游戏视频帧数据所对应的游戏画面，可以为显示在图 5 所示的游戏显示界面 500a 内的直播显示区域中的画面。

本实施例中，通过云游戏编码标准对抓取到的背景音频帧数据进行音频编码，得到目标云游戏的第一音频编码流；通过云游戏编码标准对抓取到的游戏视频帧数据进行视频编码，得到目标云游戏的第一视频编码流，提高了编码效率。

在一些实施例中，云服务器用于运行游戏进程、媒体编码进程和媒体传输进程；媒体传输进程运行在独立于云虚拟容器的辅助虚拟容器中；该方法还包括：在云虚拟容器中运行与媒体编码进程相关联的第二编码线程；获取第二编码线程所指示的云直播编码标准，通过云直播编码标准对与第二帧率相关联的游戏音频数据进行音频编码，得到目标云游戏的第二音频编码流；通过云直播编码标准对与第一帧率相关联的游戏图像数据进行视频编码，得到目标云游戏的第二视频编码流；将第二音频编码流和第二视频编码流作为目标云游戏的待处理编码流，将待处理编码流转发给媒体传输进程，以使媒体传输进程基于待处理编码流在辅助云虚拟容器中执行在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据的步骤。

其中，如图 5 所示的云服务器，可以用于运行图 5 所示的云游戏实例 51a 所对应的游戏进程、媒体编码进程 51b 和媒体传输进程 51c；其中，应当理解，这里的媒体传输进程可以运行在独立于云虚拟容器的辅助虚拟容器中。

可以理解的是，在云游戏直播场景（即上述云游戏的直播推流场景）下，云服务器还可以在云虚拟容器中运行与该媒体编码进程 51b 相关联的第二编码线程；此时，云服务器可以获取第二编码线程所指示的云直播编码标准，并可以通过云直播编码标准，对与第二帧率相关联的游戏音频数据进行音频编码，以得到目标云游戏的第二音频编码流；与此同时，云服务器还可以通过云直播编码标准，对与第一帧率相关联的游戏图像数据进行视频编码，以得到目标云游戏的第二视频编码流；这意味着图 5 所示的直播数据流 2' 至少可以包含该第二音频编码流和第二视频编码流。

比如，云服务器可以将第二音频编码流和第二视频编码流，作为目标云游戏的待处理编码流，进而可以将待处理编码流转发给图 5 所示的媒体传输进程 51c，以使该媒体传输进程 51c 可以基于待处理编码流，在上述辅助云虚拟容器中执行下述步骤 S103，以通过该媒体传输进程 51c 对游戏音频数据和接收到的待处理音频数据进行混音处理，进而可以将混音处理后的混音音频数据、游戏图像数据等转发给图 5 所示的媒体转发进程 52a，进而可以通过媒体转发进程 52a 对混音音频数据、游戏图像数据等进行音视频编码处理，以得到用于推送给直播服务器的直播数据流 2'。

其中，可以理解的是，这里的媒体转发进程，可以为上述图 2 所对应实施例中的媒体转发模块 22a 所提供的与目标云游戏相关联的其他关联进程。可以理解的是，如上述图 2 所示，媒体转发模块 22a 可以在收到 start-streaming 请求（即上述云游戏直播请求）后，向上述图 2 所示的媒体编码模块 21b 请求开启新的一路视频编码（即需要按直播视频的编码要求（即上述云直播编码标准），比如直播的视频要求有固定的帧组长度、固定的帧率），然后，媒体转发模块 22a 可以开启实时混音（例如可以对游戏的声音和主播的声音进行混音），以将（媒体编码模块 21b 新开的那路）游戏视频流和（混音后）音频流，转发到上述媒体转发模块 22a。start-streaming 请求是指云游戏直播请求，直播视频的编码要求是指云直播编码标准，比如直播的视频要求有固定的帧组长度、固定的帧率。

本实施例中，通过云直播编码标准对与第二帧率相关联的游戏音频数据进行音频编码，得到目标云游戏的第二音频编码流；通过云直播编码标准对与第一帧率相关联的游戏图像数

据进行视频编码，得到目标云游戏的第二视频编码流，提高了编码效率。

在一些实施例中，在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据，包括：在获取到云游戏客户端发送的主播对象的待处理音频数据时，对待处理音频数据进行解码处理，得到主播对象的待处理音频帧以及待处理音频帧的录制时间戳；基于游戏音频数据对应的第二采集时间间隔，获取游戏音频数据中的音频帧，将获取到的音频帧作为参考音频帧，且根据游戏音频数据对应的采集音频时间戳，确定参考音频帧的混音时间戳；基于混音时间戳调整录制时间戳，基于调整后的录制时间戳对参考音频帧和待处理音频帧进行混音处理，得到与采集时间戳保持一致的混音时间戳对应的混音音频帧，将得到的混音音频帧和混音音频帧对应的混音时间戳作为目标云游戏的混音音频数据。

在本申请实施例，用户可以感知在观众终端所播放的音画是否同步，机器无法感知是否同步的。所以，在云服务器侧进行云端推流的过程中，可以记录采集到的每个音频帧和视频帧对应的采集时间戳，在进行直播推流的时，可以把相应的采集时间戳也一并推送至直播服务器。基于此，在播放端例如观众终端 D1 和观众终端 D2，就可以在拉流拿到最近的音视频视频帧时，快速根据音频帧和视频帧的采集时间戳做排序和同步，例如视频向音频同步，进而可以使得在直播客户端中实现音视频的同步播放效果。本申请实施例中，着重考虑在云服务器侧进行云端推流时，就预先记录好采集到的音视频帧的采集时间戳，这是音画同步的关键之处，这是因为在多媒体数据（即音视频帧）的传输过程中，每个音视频帧在中转的各个模块中都可能产生延迟，若在媒体转发模块来计算采集时间戳势必是不精确的，所以，本申请实施例提出需要在采集端（比如，媒体编码模块）中预先记录好每个音视频的采集时间戳。

其中，对于运行有直播客户端的播放端而言，播放器对收到的音视频帧的时间戳应满足如下要求：

1) 音频的采集时间戳需要单调递增；2) 视频的采集时间戳需要单调递增；3) 音频和视频的采集时间戳之间的差距不能太大，比如需要小于 1000ms。

同理，对于云服务器侧的采集端而言，采集端的时间戳应满足如下条件：

1) 采集端的音视频帧的到达间隔基本精确。

比如，按上述采集时间间隔，音频可以每 10ms 采集一个音频包，视频可以每 33ms 采集一个视频包，以保证采集到的音视频帧的帧率的相对稳定。

2) 采集端的音视频帧的采集时间戳需要使用同一参考系。

例如，不能音频使用媒体编码模块的时间，而视频使用媒体传输模块的时间。为便于理解，这里以音频和视频均以使用媒体编码模块的时间为例，以记录得到采集到的每个视频帧的视频采集时间戳和每个音频帧的音频采集时间戳。

3) 保证交叉输出时的音视频间隔基本精确。

保证编码输出到推流到直播服务器，它们的间隔是稳定的，不然会产生卡顿。

其中，在本申请实施例中，云服务器可以通过媒体传输进程，对目标云游戏的声音（即游戏音频数据）和主播的声音（即待处理音频数据）进行混音，但是考虑到用于采集主播的声音的终端（即主播终端）的时间戳不同于用于采集游戏音频数据的时间戳，即两者的时间戳是不一样的，所以，在本申请实施中，云服务器需要选择使用游戏音频数据所对应的音频采集时间戳作为混音后的时间戳，这是因为观众主要关注游戏画面和游戏背景音的同步。

本实施例中，基于混音时间戳调整录制时间戳，基于调整后的录制时间戳对参考音频帧和待处理音频帧进行混音处理，得到与采集时间戳保持一致的混音时间戳对应的混音音频帧，将得到的混音音频帧和混音音频帧对应的混音时间戳作为目标云游戏的混音音频数据，从而使得游戏画面和游戏背景音同步。

在一些实施例中，云服务器在进行混音处理的过程中，可以统计获取到的音频帧的帧数，若音频帧的帧数为 M 帧，且 M 为不为零的正整数，则用 M 表征待进行混音处理的次数；云服务器可以基于待进行混音处理的次数，在执行基于混音时间戳调整录制时间戳，基于调整

后的录制时间戳对参考音频帧和待处理音频帧进行混音处理的步骤时，对待处理音频帧的音量进行增强处理，且对参考音频帧的音量进行减弱处理。其中，若音频帧的帧数为空值或者待处理音频帧的帧数为空值，则云服务器还可以不对参考音频帧和待处理音频帧进行混音处理。在进行混音的过程中，可以考虑如下几点：

1) 若在直播中需要突出主播的声音，则可以在混音时先调低游戏背景音，进而可以在调高主播的声音之后，再进行混音。

2) 可以使用 WebRTC AudioMixer 混音组件进行混音处理，且该混音组件（即 WebRTC AudioMixer）在用于进行混音时有个特点：从每个音频流取数据时若所有 audio source 都为空，则混音出来的音频帧是静音数据，而不是返回混音失败。由于静音数据推流到远程会造成播放卡顿。所以在混音前需判断是否所有参与进行混音的音频帧都为空，若是，则不混音。audio source 是指游戏背景音对应的音频帧和直播的声音对应的音频帧。

3) 因为音频有固定的采样率的，所以混音的次数不能多、也不能少。多的话声音会乱掉，少的话会有卡顿。所以，这里建议以游戏背景音所对应的固定采集间隔比如每 10ms，来触发进行混音，进而可以避免在一个时间点，因为网络抖动可能造成某路音频流的数据没有来。比如，这样可以避免在 T1 时间点只混到了其中一路，而在另一个时间点例如 T2 时间点，又只混到了另一路的现象。

基于此，本申请实施例提出可以在混音处理时，选择一个音频流作为主流，主流的输入频率需要是稳定的，比如每秒 100 个包，进而可以按主流的帧数进行混音。由于游戏背景音的包是稳定输出的，所以可以用它来作为主流，以上述游戏背景音所对应的固定采集间隔比如每 10ms，来触发混音。这样，在进行混音前，可以统计获取到的主流的帧数，如果是 0（即空值），则不进行混音操作。反之，如果是 M（即不为零的正整数），则可以进行 M 次混音操作。

本实施例中，统计获取到的音频帧的帧数，得到待进行混音处理的次数，从而准确的确定出了待进行混音处理的次数。

在一些实施例中，该方法还包括：若音频帧的帧数为空值或者待处理音频帧的帧数为空值，则不对参考音频帧和待处理音频帧进行混音处理。

具体地，在进行混音前，可以统计获取到的主流的帧数，如果是 0（即空值），则不进行混音操作。

本实施例中，若音频帧的帧数为空值或者待处理音频帧的帧数为空值，则不对参考音频帧和待处理音频帧进行混音处理，提高了混音处理的合理性。

在一些实施例中，云服务器可以获取混音音频数据，从混音音频数据中获取混音音频帧以及混音音频帧对应的音频采集时间戳，基于音频采集时间戳对混音音频帧进行排序处理，将排序处理后的混音音频帧，写入与媒体转发进程相关联的音频缓存队列；云服务器可以获取游戏图像数据，从游戏图像数据中获取图像帧以及图像帧对应的图像采集时间戳，基于图像采集时间戳对图像帧进行排序处理，将排序处理后的图像帧，写入与媒体转发进程相关联的视频缓存队列；云服务器可以获取与媒体转发进程相关联的平滑获取线程，通过平滑获取线程从视频缓存队列中，获取与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，通过平滑获取线程从音频缓存队列中，获取与第二采集时间间隔所指示的第二帧率相匹配的音频数据包；云服务器可以将视频数据包和音频数据包进行打包处理，得到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

本实施例中，在云服务器侧的媒体转发模块中采用优先级队列+平滑获取线程，提升了在观众终端进行音视频播放的流畅性。

在一些实施例中，获取与媒体转发进程相关联的平滑获取线程，通过平滑获取线程从视频缓存队列中采集与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，并通过平滑获取线程从音频缓存队列中采集与第二采集时间间隔所指示的第二帧率相匹配的音频数据包，包括：云服务器可以获取与媒体转发进程相关联的平滑获取线程，获取与平滑获取线

程具有相同采集时间间隔的数据采集组件，基于数据采集组件的采集时间间隔，确定与游戏图像数据相关联的第一采集时间间隔和与游戏音频数据相关联的第二采集时间间隔；云服务器可以通过平滑获取线程，确定与第一采集时间间隔相匹配的平滑视频获取时间戳，从视频缓存队列中，获取图像采集时间戳小于平滑视频获取时间戳的缓存视频帧，将获取到的缓存视频帧和缓存视频帧对应的图像采集时间戳进行封装处理，得到与第一采集时间间隔所指示的第一帧率相匹配的视频数据包；云服务器可以通过平滑获取线程，确定与第二采集时间间隔相匹配的平滑音频获取时间戳，从音频缓存队列中，获取音频采集时间戳小于平滑音频获取时间戳的缓存音频帧，将获取到的缓存音频帧和缓存音频帧对应的音频采集时间戳进行封装处理，得到与第二采集时间间隔所指示的第二帧率相匹配的音频数据包。

图6是本申请实施例提供的一种媒体转发模块的场景示意图。图6中的媒体转发模块可以为图5所对应实施例中的媒体转发进程所对应的云服务模块。

由于在云服务器内部进行多媒体数据传输的过程中，会流经多个模块，且每个模块的处理速度是不同的，比如，每个模块在分别用于处理音频数据和视频数据等多媒体数据时的速度可能存在不同，从而导致在多媒体数据的传输过程中，可能会有丢包、乱序到达、传输阻塞的问题。基于此，若图6所示的媒体转发模块在收到其他模块传来的直播数据流后，直接将接收到的直播数据流推到直播服务器，则可能造成直播客户端所在的观众终端的卡顿。

本申请实施例，为了提升在观众终端进行音视频播放的流畅性，在云服务器侧的媒体转发模块中采用优先级队列+平滑获取线程（即 **buffer + pacing**）的思路来优化这个问题。即云服务器中的媒体转发模块在收到音频帧和视频帧时，无需马上发送出去，比如，可以按照接收到的帧率将其存储至图6所述的优先级队列。

具体的，如图6所示，媒体转发模块可以暂时将收到的音频帧进行音频处理之后，存入到图6所示的优先级队列，并可以暂时将收到的视频帧进行视频处理之后，存入到图6所示的优先级队列。优先级队列可以包含用于存储音频帧的音频缓存队列和用于存储视频帧的视频缓存队列。其中，在本申请实施例中，写入优先级队列中的音频帧和视频帧，均需要按照采集时间戳进行排序处理，以便于后续可以通过平滑获取线程，从优先级队列中可以优先取采集时间戳较小的数据帧，以尽可能地减少内部传输或处理过程中的抖动和乱序问题。

其中，如图6所示，媒体转发模块中的另一个 **pacing** 线程即平滑获取线程，可以在固定间隔内从优先级队列中尝试获取音视频包，可以将固定间隔内所获取到的带有相应采集时间戳的音视频包即图6所示的直播数据流，平滑地推送到直播服务器。比如，对于视频数据包而言，可以通过平滑获取线程，从视频缓存队列中获取与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，通过平滑获取线程，从音频缓存队列中获取与第二采集时间间隔所指示的第二帧率相匹配的音频数据包，这样，通过 **buffer + pacing** 的平滑策略可以从一定程度上解决接收到的音频帧和视频帧的帧率所出现的抖动，进而可以从根源上确保需要转发出去的音频数据包和视频数据包的稳定性和平滑度。

通常情况下，在缓存时长内，当前推入优先级队列的音视频包的时间戳，大概率会比之前推入优先级队列的音视频包的时间戳大，但是，不排除偶尔会因为传输抖动而造成一定的乱序问题，例如，当前推入优先级队列的音视频包的时间戳比之前推入优先级队列的音视频包的时间戳小的现象。

图7是本申请实施例提供的一种优先级队列的场景示意图。如图7所示的优先级队列采用双向链表的方式，来确保每次推入队列的音视频包的时间戳，大概率会比之前推入优先级队列的音视频包的时间戳大。比如，如图7所示的优先级队列的队头节点可以用于存储采集时间戳较小的音视频数据包，且图7所示的优先级队列的队尾节点，可以用于存储采集时间戳较大的音视频数据包。对头节点和队尾节点以及其他的一些中间节点均存储在云服务器的云内存中。

其中，在出队列时，媒体转发模块可以通过 **pacing** 线程，基于双向链接所指示的路径，例如，时间戳由小到大的路径，向后移动队头节点的指针。

本实施例中，在云服务器侧的媒体转发模块中采用优先级队列+平滑获取线程，提升了在观众终端进行音视频播放的流畅性。

在一些实施例中，在将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端之后，该方法还包括：确定直播数据流的最大音频采集时间戳；若通过平滑获取线程记录到待发送音频数据包的音频采集时间戳小于最大音频采集时间戳，则丢弃待发送音频数据包待发送音频数据包，是指待发送给直播服务器的下一个音频数据包。

其中，云服务器可以直播数据流中获取具有最大音频采集时间戳的音频数据包作为辅助音频数据包。

本实施例中，若通过平滑获取线程记录到待发送音频数据包的音频采集时间戳小于最大音频采集时间戳，则丢弃待发送音频数据包待发送音频数据包，从而可以减少因网络抖动所引起的乱序问题，提升音视频同步播放的流畅性。

在一些实施例中，在将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端之后，该方法还包括：确定直播数据流的最大图像采集时间戳；若通过平滑获取线程记录到待发送视频数据包的视频采集时间戳小于最大视频采集时间戳，则丢弃待发送视频数据包，待发送视频数据包，是指待发送给直播服务器的下一个视频数据包。

其中，云服务器可以直播数据流中获取具有最大图像采集时间戳的视频数据包作为辅助视频数据包。

本实施例中，若通过平滑获取线程记录到待发送视频数据包的视频采集时间戳小于最大视频采集时间戳，则丢弃待发送视频数据包，从而可以减少因网络抖动所引起的乱序问题，提升音视频同步播放的流畅性。

图8是本申请实施例提供的一种游戏数据处理方法的流程示意图。如图8所示，该计算机设备可以为上述图3所对应实施例中的云服务器。其中，该方法至少可以包括以下步骤S201-步骤S210：

步骤S201，在获取到云游戏客户端针对目标云游戏发送的游戏访问请求时，基于游戏访问请求对云游戏客户端对应的游戏启动标识进行鉴权处理，得到游戏启动标识对应的鉴权结果。

步骤S202，若鉴权结果指示云游戏客户端具有通过游戏启动标识访问目标云游戏的权限，则建立云服务器与云游戏客户端之间的游戏服务连接，且基于游戏服务连接为云游戏客户端所在的客户端环境系统配置用于运行目标云游戏的云虚拟容器；

步骤S203，在云虚拟容器中启动目标云游戏的游戏进程，通过游戏进程运行目标云游戏。

步骤S204，接收云游戏客户端发送的针对目标云游戏的云游戏直播请求，基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接。

具体的，云服务器可以接收云游戏客户端发送的针对目标云游戏的云游戏直播请求；其中，云游戏直播请求，是云游戏客户端响应于主播对象针对直播指示控件执行的触发操作所生成的；云服务器可以基于云游戏直播请求中所携带的游戏启动标识，向与云游戏客户端相关联的直播服务器发送直播认证请求，直播服务器基于直播认证请求，为云游戏客户端配置用于直播目标云游戏的直播推流地址；云服务器可以接收直播服务器返回的直播推流地址，在将直播推流地址返回给云游戏客户端时，触发建立云服务器与直播服务器之间的直播服务连接。

步骤S205，基于直播服务连接获取目标云游戏的游戏图像数据和游戏音频数据，记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳；

其中，图像采集时间戳和音频采集时间戳为同一参考系下的时间戳；

步骤S206，在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据；

其中，混音音频数据对应的混音时间戳与音频采集时间戳保持一致。

步骤S207，基于混音音频数据、音频采集时间戳、游戏图像数据和图像采集时间戳，得

到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端，以使直播客户端基于直播数据流中的音频采集时间戳和图像采集时间戳，对混音音频数据和游戏图像数据进行同步播放；

步骤 S208，在直播数据流中获取具有最大音频采集时间戳的音频数据包作为辅助音频数据包，且在直播数据流中获取具有最大图像采集时间戳的视频数据包作为辅助视频数据包；

步骤 S209，若通过平滑获取线程记录到待发送给直播服务器的下一个音频数据包的音频采集时间戳小于已发送的辅助音频数据包的最大音频采集时间戳，则丢弃待发送给直播服务器的下一个音频数据包；

步骤 S210，若通过平滑获取线程记录到待发送给直播服务器的下一个视频数据包的视频采集时间戳小于已发送的辅助视频数据包的最大视频采集时间戳，则丢弃待发送给直播服务器的下一个视频数据包。

在本申请实施例中，云服务器还可以在执行完上述步骤 S207 之后，继续执行前述步骤 S208-步骤 S210，进而可以减少因网络抖动所引起的乱序问题，提升音视频同步播放的流畅性。

由于在云服务器内部进行多媒体数据传输的过程中，会流经多个云服务模块，且每个云服务模块的处理速度是不同的，比如，每个云服务模块在分别用于处理音频数据和视频数据等多媒体数据时的速度可能存在不同，从而导致在多媒体数据的传输过程中，可能会有丢包、乱序到达、传输拥塞的问题。基于此，若云服务器中的媒体转发模块在收到其他云服务模块传来的直播数据流后，直接将接收到的直播数据流推到直播服务器，势必会造成直播客户端所在的观众终端的卡顿。

图 9 是一种通过媒体转发模块，对收到的音频帧和视频帧进行平滑处理的场景示意图。如图 9 所示，媒体转发模块可以用于接收由其他云服务模块（即上述媒体传输模块）转发的混音处理后的音频数据流、和由第二编码线程所编码得到的视频数据流。具体的，曲线 91a 可以用于表征媒体转发模块所接收到的、混音处理后的音频数据流中的音频帧的接收帧率，比如，音频帧的接收帧率可以为第二采集时间间隔所指示的第二帧率，例如，100 帧/s。同理，曲线 92a 可以用于表征媒体转发模块所接收到的、由第二编码线程所编码得到的视频数据流中的视频帧的接收帧率，比如，视频帧的接收帧率可以为第一采集时间间隔所指示的第一帧率，例如，30 帧/s。

如图 9 所示，为了提升在观众终端进行音视频播放的流程性，本申请实施例中，提出在云服务器侧使用优先级队列+平滑获取线程的思路来优化这个问题。即云服务器中的媒体转发模块在收到音频帧和视频帧时，无需马上发送出去，而是暂时将收到的音频帧和视频帧存入到优先级队列中，优先级队列可以包含用于存储音频帧的音频缓存队列、和用于存储视频帧的视频缓存队列，另一个 pacing 线程，则可以在固定间隔内从优先级队列中尝试获取音视频包，将在固定间隔内所获取到的带有相应采集时间戳的音视频包，平滑地发送到直播服务器。

具体的，曲线 91b 可以用于表征媒体转发模块按照与采集端相同的采集时间间隔、从优先级队列中尝试获取到的音频包的发送帧率，比如，待发送的音频帧的发送帧率可以为第二采集时间间隔所指示的第二帧率。曲线 92b 可以用于表征媒体转发模块按照与采集端相同的采集时间间隔、从优先级队列中尝试获取到的视频包的发送帧率，比如，此时，待发送的视频帧的发送帧率可以为上述第一采集时间间隔所指示的第一帧率。由此可见，对于图 9 所示的媒体转发模块而言，在经过优先级队列+平滑获取线程策略，来对接收到的音视频流进行平滑处理之后，可以对接收到的存在抖动的音视频帧的曲线进行平滑处理，进而可以在云游戏直播场景下，提升在观众终端播放音视频的稳定性和流畅性。

通常情况下，在缓存时长内，当前推入优先级队列的音视频包的时间戳，大概率会比之前推入优先级队列的音视频包的时间戳大，但是，不排除偶尔会因为传输抖动而造成一定的乱序问题，例如，当前推入优先级队列的音视频包的时间戳，比之前推入优先级队列的音视频包的时间戳小的现象。

基于此,考虑到上述优先级队列可用于缓存最近一段时间的帧数据,例如,1~2秒的缓存时长内的帧数据,所以,若云服务器内部之间的各个模块因为网络传输问题等出现乱序抖动,且乱序抖动的抖动时长超过了优先级队列的缓存时长,则可以通过 pacing 线程所记录的最后一个音视频包的时间戳来快速判断是否需要丢弃下一次采集到的音视频包。

比如,本申请实施例所公开的云服务器,在通过 pacing 线程从优先级队列中获取到一个或者多个音视频包,并向直播服务器发送完获取到的每个音视频包后,记录发送的这些音视频包中的最后一个音频包的时间戳和视频包的时间戳,这样,如果 pacing 线程当前从优先级队列中获取到下一个音视频包的时间戳,小于前一个音视频包的时间戳(即前述记录的最后一个音频包的时间戳和视频包的时间戳),则可以丢弃当前获取到的下一个音视频包的时间戳。

在本申请实施例中,考虑到在云游戏直播场景下,云游戏客户端并不会直接向直播服务器推送直播数据流,而是通过该云服务器来向直播服务器进行直播数据流的推送,从而可以利用云服务器强大的计算能力,从根源上降低对云游戏客户端侧的性能消耗。另外,通过在云服务器侧进行基于音频采集时间戳和图像采集时间戳进行音视频的同步,不仅可以解决直播推流过程中因抖动、乱序等所造成的音视频不同步的现象,还可以实现多路混音的同步及卡顿问题,进而可以实现音视频同步的可靠性,这样,当在直播客户端中基于直播数据流中的音频采集时间戳和图像采集时间戳,对混音音频数据和游戏图像数据进行同步播放时,可以提升混音后的音视频的同步播放效果。

进一步地,请参见图 10,图 10 是本申请实施例提供的一种游戏数据处理装置的结构示意图。游戏数据处理装置 1 可以应用于计算机设备,该计算机设备可以为上述图 2 所对应实施例中的云服务器 B。进一步地,游戏数据处理装置 1 可以包括:直播请求接收模块 10,数据获取模块 20,混音处理模块 30 和直播流下发模块 40;可选的,在直播请求接收模块 10 接收云游戏客户端发送的针对目标云游戏的云游戏直播请求之前,游戏数据处理装置 1 还包括:访问请求获取模块 50,游戏服务建立模块 60 和云游戏运行模块 70;

直播请求接收模块 10,用于接收云游戏客户端发送的针对目标云游戏的云游戏直播请求,基于云游戏直播请求建立云服务器与云游戏客户端相关联的直播服务器之间的直播服务连接。

数据获取模块 20,用于基于直播服务连接获取目标云游戏的游戏图像数据和游戏音频数据,记录游戏图像数据对应的图像采集时间戳和游戏音频数据对应的音频采集时间戳;图像采集时间戳和音频采集时间戳为同一参考系下的时间戳。

混音处理模块 30,用于在获取到云游戏客户端发送的主播对象的待处理音频数据时,对游戏音频数据和待处理音频数据进行混音处理,得到混音音频数据;混音音频数据对应的混音时间戳与音频采集时间戳保持一致。

直播流下发模块 40,用于基于混音音频数据、音频采集时间戳、游戏图像数据和图像采集时间戳,得到目标云游戏对应的直播数据流,将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

在一些实施例中,该装置还包括:

访问请求获取模块 50,用于在获取到云游戏客户端针对目标云游戏发送的游戏访问请求时,基于游戏访问请求对云游戏客户端对应的游戏启动标识进行鉴权处理,得到游戏启动标识对应的鉴权结果;游戏访问请求中携带云游戏客户端所在的客户端环境系统。

游戏服务建立模块 60,用于若鉴权结果指示云游戏客户端具有通过游戏启动标识访问目标云游戏的权限,则建立云服务器与云游戏客户端之间的游戏服务连接,且基于游戏服务连接为云游戏客户端所在的客户端环境系统配置用于运行目标云游戏的云虚拟容器。

云游戏运行模块 70,用于在云虚拟容器中启动目标云游戏的游戏进程,通过游戏进程运行目标云游戏。

在一些实施例中,云游戏客户端对应的游戏显示界面上包含用于指示主播对象进行云游

戏直播的直播指示控件；直播请求接收模块 10 包括：直播请求接收单元 101，认证请求接收单元 102，服务连接建立单元 103；

直播请求接收单元 101，用于接收云游戏客户端发送的针对目标云游戏的云游戏直播请求；云游戏直播请求是云游戏客户端响应于主播对象针对直播指示控件执行的触发操作所生成的。

在一些实施例中，直播请求接收模块 10 包括：认证请求接收单元 102，服务连接建立单元 103；云游戏直播请求中所携带所述游戏启动标识；

认证请求接收单元 102，用于基于游戏启动标识向与云游戏客户端相关联的直播服务器发送直播认证请求，以使直播服务器基于直播认证请求，为云游戏客户端配置用于直播目标云游戏的直播推流地址。

服务连接建立单元 103，用于接收直播服务器返回的直播推流地址，在将直播推流地址返回给云游戏客户端时，触发建立云服务器与直播服务器之间的直播服务连接。

在一些实施例中，目标云游戏是由云虚拟容器中的游戏进程运行的；云虚拟容器是由云服务器为云游戏客户端所在的客户端环境系统所配置的；数据获取模块 20 包括：编码进程启动单元 201，数据抓取单元 202，图像数据采集单元 203 和音频数据采集单元 204。

编码进程启动单元 201，用于基于直播服务连接在云虚拟容器中启动与游戏进程相关联的媒体编码进程。

数据抓取单元 202，用于通过媒体编码进程所指示的数据采集组件，抓取与目标云游戏相关联的背景音频帧数据和游戏视频帧数据，将抓取到的背景音频帧数据和游戏视频帧数据，存放至与数据采集组件相关联的云缓存空间；云缓存空间为独立于云虚拟容器的云内存。

图像数据采集单元 203，用于通过媒体编码进程所指示的定时器，从云内存所存储的游戏视频帧数据中，采集与第一采集时间间隔所指示的第一帧率相匹配的图像帧，将采集到的图像帧作为目标云游戏的游戏图像数据，并在定时器所指示的参考系下，记录游戏图像数据对应的图像采集时间戳。

音频数据采集单元 204，用于通过定时器，从云内存所存储的背景音频帧数据中，采集与第二采集时间间隔所指示的第二帧率相匹配的音频帧，将采集到的音频帧作为目标云游戏的游戏音频数据，并在定时器所指示的同一参考系下，记录游戏音频数据对应的音频采集时间戳；第二采集时间间隔记录小于第一采集时间间隔，且第二帧率大于第一帧率。

在一些实施例中，数据获取模块 20 还包括：第一编码线程运行单元 205，第一音频编码单元 206，第一视频编码单元 207，游戏流返回单元 208，第二编码线程运行单元 209，第二音频编码单元 210，第二视频编码单元 211 和编码流转发单元 212；

第一编码线程运行单元 205，用于在云虚拟容器中运行与媒体编码进程相关联的第一编码线程。

第一音频编码单元 206，用于获取第一编码线程所指示的云游戏编码标准，通过云游戏编码标准对抓取到的背景音频帧数据进行音频编码，得到目标云游戏的第一音频编码流。

第一视频编码单元 207，用于通过云游戏编码标准对抓取到的游戏视频帧数据进行视频编码，得到目标云游戏的第一视频编码流。

游戏流返回单元 208，用于将第一音频编码流和第一视频编码流作为目标云游戏的游戏编码流，将游戏编码流返回给云游戏客户端，以使云游戏客户端对游戏编码流进行解码处理，得到用于在云游戏客户端中进行同步播放的背景音频帧数据和游戏视频帧数据。

在一些实施例中，数据获取模块 20 还包括：第二编码线程运行单元 209，第二音频编码单元 210，第二视频编码单元 211 和编码流转发单元 212；云服务器用于运行游戏进程、媒体编码进程和媒体传输进程；媒体传输进程运行在独立于云虚拟容器的辅助虚拟容器中；

第二编码线程运行单元 209，用于在云虚拟容器中运行与媒体编码进程相关联的第二编码线程。

第二音频编码单元 210，用于获取第二编码线程所指示的云直播编码标准，通过云直播

编码标准对与第二帧率相关联的游戏音频数据进行音频编码，得到目标云游戏的第二音频编码流。

第二视频编码单元 211，用于通过云直播编码标准对与第一帧率相关联的游戏图像数据进行视频编码，得到目标云游戏的第二视频编码流。

编码流转发单元 212，用于将第二音频编码流和第二视频编码流作为目标云游戏的待处理编码流，将待处理编码流转发给媒体传输进程，以使媒体传输进程通知混音处理模块 30 基于待处理编码流在辅助云虚拟容器中执行在获取到云游戏客户端发送的主播对象的待处理音频数据时，对游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据的步骤。

在一些实施例中，混音处理模块 30 包括：音频解码单元 301，参考音频帧获取单元 302，混音处理单元 303。

音频解码单元 301，用于在获取到云游戏客户端发送的主播对象的待处理音频数据时，对待处理音频数据进行解码处理，得到主播对象的待处理音频帧以及待处理音频帧的录制时间戳。

参考音频帧获取单元 302，用于基于游戏音频数据对应的第二采集时间间隔，获取游戏音频数据中的音频帧，将获取到的音频帧作为参考音频帧，且根据游戏音频数据对应的采集音频时间戳，确定参考音频帧的混音时间戳。

混音处理单元 303，用于基于混音时间戳调整录制时间戳，基于调整后的录制时间戳对参考音频帧和待处理音频帧进行混音处理，得到与采集时间戳保持一致的混音时间戳对应的混音音频帧，将得到的混音音频帧和混音音频帧对应的混音时间戳作为目标云游戏的混音音频数据。

在一些实施例中，混音处理模块 30 还包括：音频帧统计单元 304，音量调整单元 305。

音频帧统计单元 304，用于统计获取到的音频帧的帧数，若音频帧的帧数为 M 帧； M 不为零的正整数，且 M 用于表征待进行混音处理的次数。

音量调整单元 305，用于基于待进行混音处理的次数，在执行基于混音时间戳调整录制时间戳，基于调整后的录制时间戳对参考音频帧和待处理音频帧进行混音处理的步骤时，对待处理音频帧的音量进行增强处理，且对参考音频帧的音量进行减弱处理。

在一些实施例中，混音处理模块 30 还包括：空值检测单元 306。

空值检测单元 306，用于若音频帧的帧数为空值或者待处理音频帧的帧数为空值，则不对参考音频帧和待处理音频帧进行混音处理。

在一些实施例中，云服务器包含基于云游戏直播请求所启动的媒体转发进程；媒体转发进程用于保持云服务器与直播服务器之间的直播服务连接；直播流下发模块 40 包括：音频排序处理单元 401，图像排序处理单元 402，数据包获取单元 403 和打包下发单元 404。

音频排序处理单元 401，用于获取混音音频数据，从混音音频数据中获取混音音频帧以及混音音频帧对应的音频采集时间戳，基于音频采集时间戳对混音音频帧进行排序处理，将排序处理后的混音音频帧写入与媒体转发进程相关联的音频缓存队列。

图像排序处理单元 402，用于获取游戏图像数据，从游戏图像数据中获取图像帧以及图像帧对应的图像采集时间戳，基于图像采集时间戳对图像帧进行排序处理，将排序处理后的图像帧写入与媒体转发进程相关联的视频缓存队列。

数据包获取单元 403，用于获取与媒体转发进程相关联的平滑获取线程，通过平滑获取线程从视频缓存队列中获取与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，并通过平滑获取线程从音频缓存队列中获取与第二采集时间间隔所指示的第二帧率相匹配的音频数据包。

打包下发单元 404，用于将视频数据包和音频数据包进行打包处理，得到目标云游戏对应的直播数据流，将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端。

在一些实施例中，数据包获取单元 403 包括：时间间隔获取子单元 4031，视频帧获取子单元 4032 和音频帧获取子单元 4033。

时间间隔获取子单元 4031, 用于获取与媒体转发进程相关联的平滑获取线程, 且获取与平滑获取线程具有相同采集时间间隔的数据采集组件, 基于数据采集组件的采集时间间隔确定与游戏图像数据相关联的第一采集时间间隔和与游戏音频数据相关联的第二采集时间间隔。

视频帧获取子单元 4032, 用于通过平滑获取线程确定第一采集时间间隔相匹配的平滑视频获取时间戳, 从视频缓存队列中获取图像采集时间戳小于平滑视频获取时间戳的缓存视频帧, 将获取到的缓存视频帧和缓存视频帧对应的图像采集时间戳进行封装处理, 得到与第一采集时间间隔所指示的第一帧率相匹配的视频数据包。

音频帧获取子单元 4033, 用于通过平滑获取线程确定第二采集时间间隔相匹配的平滑音频获取时间戳, 从视频缓存队列中获取音频采集时间戳小于平滑音频获取时间戳的缓存音频帧, 将获取到的缓存音频帧和缓存音频帧对应的音频采集时间戳进行封装处理, 得到与第二采集时间间隔所指示的第二帧率相匹配的音频数据包。

在一些实施例中, 直播流下发模块 40 还包括: 辅助包获取单元 405, 音频包丢弃单元 406 和视频包丢弃单元 407。

在直播流下发模块 40 将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端之后, 辅助包获取单元 405, 用于确定直播数据流的最大音频采集时间戳。

音频包丢弃单元 406, 用于若通过平滑获取线程记录到待发送音频数据包的音频采集时间戳小于最大音频采集时间戳, 则丢弃待发送音频数据包待发送音频数据包, 是指待发送给直播服务器的下一个音频数据包。

在一些实施例中, 直播流下发模块 40 还包括: 视频包丢弃单元 407。在直播流下发模块 40 将直播数据流通过直播服务器下发至与云游戏客户端相关联的直播客户端之后, 辅助包获取单元 405, 还用于确定直播数据流的最大图像采集时间戳;

视频包丢弃单元 407, 用于若通过平滑获取线程记录到待发送视频数据包的视频采集时间戳小于最大视频采集时间戳, 则丢弃待发送视频数据包, 待发送视频数据包, 是指待发送给直播服务器的下一个视频数据包。进一步地, 请参见图 11, 图 11 是本申请实施例提供的一种计算机设备的结构示意图。如图 11 所示, 该计算机设备 1000 可以为云服务器, 该云服务器可以为上述图 2 所对应实施例中的云服务器 B。该计算机设备 1000 可以包括: 处理器 1001, 网络接口 1004 和存储器 1005, 此外, 计算机设备 1000 还可以包括: 用户接口 1003, 和至少一个通信总线 1002。其中, 通信总线 1002 用于实现这些组件之间的连接通信。其中, 可选用户接口 1003 还可以包括标准的有线接口、无线接口。网络接口 1004 可选的可以包括标准的有线接口、无线接口(如 WI-FI 接口)。存储器 1005 可以是高速 RAM 存储器, 也可以是非不稳定的存储器(non-volatile memory), 例如至少一个磁盘存储器。存储器 1005 可选的还可以是至少一个位于远离前述处理器 1001 的存储装置。如图 11 所示, 作为一种计算机存储介质的存储器 1005 中可以包括操作系统、网络通信模块、用户接口模块以及设备控制应用程序。

其中, 计算机设备 1000 中的网络接口 1004 还可以与上述图 1 所对应实施例中的主播终端集群中的任一主播终端进行网络连接。在图 11 所示的计算机设备 1000 中, 网络接口 1004 可提供网络通讯功能; 而用户接口 1003 主要用于为用户提供输入的接口; 而处理器 1001 可以用于调用存储器 1005 中存储的设备控制应用程序, 以执行前文图 3 或图 8 所对应实施例中对游戏数据处理方法的描述, 也可执行前文图 10 所对应实施例中对游戏数据处理装置 1 的描述。

此外, 这里需要指出的是: 本申请实施例还提供了一种计算机存储介质, 且计算机存储介质中存储有前文提及的游戏数据处理装置 1 所执行的计算机可读指令, 且计算机可读指令包括程序指令, 当处理器执行程序指令时, 能够执行前文图 3 或图 8 所对应实施例中对游戏数据处理方法的描述。作为示例, 程序指令可被部署为在一个计算设备上执行, 或者在位于一个地点的多个计算设备上执行, 又或者, 在分布在多个地点且通过通信网络互连的多个计

算设备上执行，分布在多个地点且通过通信网络互连的多个计算设备可以组成区块链系统。

一种计算机设备，包括存储器和一个或多个处理器，所述存储器存储有计算机可读指令，所述计算机可读指令被所述处理器执行时，使得所述一个或多个处理器执行上述电竞数据处理方法的步骤。

一个或多个非易失性可读存储介质，其上存储有计算机可读指令，所述计算机可读指令被一个或多个处理器执行时，使得所述一个或多个处理器实现上述电竞数据处理方法的步骤。

在一些实施例中，还提供了一种计算机程序产品，包括计算机可读指令，计算机可读指令被处理器执行时实现上述各方法实施例中的步骤。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备可执行前文图 3 或者图 8 所对应实施例中对游戏数据处理方法的描述。

本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机可读指令来指令相关的硬件来完成，的程序可存储于一计算机可读取存储介质中，该程序在执行时，可包括如上述各方法的实施例的流程。其中，的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory, ROM)或随机存储记忆体(Random Access Memory, RAM)等。

以上所揭露的仅为本申请较佳实施例而已，当然不能以此来限定本申请之权利范围，因此依本申请权利要求所作的等同变化，仍属本申请所涵盖的范围。

权利要求书

1、一种游戏数据处理方法，其特征在于，由计算机设备执行，包括：

接收云游戏客户端发送的针对目标云游戏的云游戏直播请求，基于所述云游戏直播请求建立云服务器与所述云游戏客户端相关联的直播服务器之间的直播服务连接；

基于所述直播服务连接获取所述目标云游戏的游戏图像数据和游戏音频数据，记录所述游戏图像数据对应的图像采集时间戳和所述游戏音频数据对应的音频采集时间戳；所述图像采集时间戳和所述音频采集时间戳为同一参考系下的时间戳；

在获取到所述云游戏客户端发送的主播对象的待处理音频数据时，对所述游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据；所述混音音频数据对应的混音时间戳与所述音频采集时间戳保持一致；及

基于所述混音音频数据、所述音频采集时间戳、所述游戏图像数据和所述图像采集时间戳，得到所述目标云游戏对应的直播数据流，将所述直播数据流通过所述直播服务器下发至与所述云游戏客户端相关联的直播客户端。

2、根据权利要求 1 所述的方法，其特征在于，在所述接收云游戏客户端发送的针对目标云游戏的云游戏直播请求之前，所述方法还包括：

在获取到云游戏客户端针对目标云游戏发送的游戏访问请求时，基于所述游戏访问请求对所述云游戏客户端对应的游戏启动标识进行鉴权处理，得到所述游戏启动标识对应的鉴权结果；所述游戏访问请求中携带所述云游戏客户端所在的客户端环境系统；

若所述鉴权结果指示所述云游戏客户端具有通过所述游戏启动标识访问所述目标云游戏的权限，则建立所述云服务器与所述云游戏客户端之间的游戏服务连接，且基于所述游戏服务连接为所述云游戏客户端所在的客户端环境系统配置用于运行所述目标云游戏的云虚拟容器；及

在所述云虚拟容器中启动所述目标云游戏的游戏进程，通过所述游戏进程运行所述目标云游戏。

3、根据权利要求 2 所述的方法，其特征在于，所述云游戏客户端对应的游戏显示界面上包含用于指示所述主播对象进行云游戏直播的直播指示控件；

所述基于所述云游戏直播请求建立云服务器与所述云游戏客户端相关联的直播服务器之间的直播服务连接，包括：

接收云游戏客户端发送的针对所述目标云游戏的云游戏直播请求；所述云游戏直播请求，是所述云游戏客户端响应于所述主播对象针对所述直播指示控件执行的触发操作所生成的。

4、根据权利要求 2 所述的方法，其特征在于，所述云游戏直播请求中所携带所述游戏启动标识；

所述基于所述云游戏直播请求建立云服务器与所述云游戏客户端相关联的直播服务器之间的直播服务连接包括：

基于所述游戏启动标识向与所述云游戏客户端相关联的直播服务器发送直播认证请求，以使所述直播服务器基于所述直播认证请求，为所述云游戏客户端配置用于直播所述目标云游戏的直播推流地址；及

接收所述直播服务器返回的所述直播推流地址，在将所述直播推流地址返回给所述云游戏客户端时，触发建立所述云服务器与所述直播服务器之间的直播服务连接。

5、根据权利要求 1 所述的方法，其特征在于，所述目标云游戏是由云虚拟容器中的游戏进程运行的；所述云虚拟容器是由所述云服务器为所述云游戏客户端所在的客户端环境系统所配置的；

所述基于所述直播服务连接获取所述目标云游戏的游戏图像数据和游戏音频数据，记录所述游戏图像数据对应的图像采集时间戳和所述游戏音频数据对应的音频采集时间戳，包括：

基于所述直播服务连接，在所述云虚拟容器中启动与所述游戏进程相关联的媒体编码进程；

通过所述媒体编码进程所指示的数据采集组件，抓取与所述目标云游戏相关联的背景音频帧数据和游戏视频帧数据，将抓取到的所述背景音频帧数据和所述游戏视频帧数据，存放至与所述数据采集组件相关联的云缓存空间；所述云缓存空间为独立于所述云虚拟容器的云内存；

通过所述媒体编码进程所指示的定时器，从所述云内存所存储的所述游戏视频帧数据中，采集与第一采集时间间隔所指示的第一帧率相匹配的图像帧，将采集到的图像帧作为所述目标云游戏的游戏图像数据，并在所述定时器所指示的参考系下，记录所述游戏图像数据对应的图像采集时间戳；及

通过所述定时器，从所述云内存所存储的所述背景音频帧数据中，采集与第二采集时间间隔所指示的第二帧率相匹配的音频帧，将采集到的音频帧作为所述目标云游戏的游戏音频数据，并在所述定时器所指示的同一参考系下，记录所述游戏音频数据对应的音频采集时间戳；所述第二采集时间间隔记录小于所述第一采集时间间隔，且所述第二帧率大于所述第一帧率。

6、根据权利要求5所述的方法，其特征在于，所述方法还包括：

在所述云虚拟容器中运行与所述媒体编码进程相关联的第一编码线程；

获取所述第一编码线程所指示的云游戏编码标准，通过所述云游戏编码标准对抓取到的所述背景音频帧数据进行音频编码，得到所述目标云游戏的第一音频编码流；

通过所述云游戏编码标准对抓取到的所述游戏视频帧数据进行视频编码，得到所述目标云游戏的第一视频编码流；及

将所述第一音频编码流和所述第一视频编码流作为所述目标云游戏的游戏编码流，将所述游戏编码流返回给所述云游戏客户端，以使所述云游戏客户端对所述游戏编码流进行解码处理，得到用于在所述云游戏客户端中进行同步播放的所述背景音频帧数据和所述游戏视频帧数据。

7、根据权利要求5所述的方法，其特征在于，所述云服务器用于运行所述游戏进程、所述媒体编码进程和媒体传输进程；所述媒体传输进程运行在独立于所述云虚拟容器的辅助虚拟容器中；

所述方法还包括：

在所述云虚拟容器中运行与所述媒体编码进程相关联的第二编码线程；

获取所述第二编码线程所指示的云直播编码标准，通过所述云直播编码标准对与所述第二帧率相关联的所述游戏音频数据进行音频编码，得到所述目标云游戏的第二音频编码流；

通过所述云直播编码标准对与所述第一帧率相关联的所述游戏图像数据进行视频编码，得到所述目标云游戏的第二视频编码流；及

将所述第二音频编码流和所述第二视频编码流作为所述目标云游戏的待处理编码流，将所述待处理编码流转发给所述媒体传输进程，以使所述媒体传输进程基于所述待处理编码流在所述辅助云虚拟容器中执行所述在获取到所述云游戏客户端发送的主播对象的待处理音频数据时，对所述游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据的步骤。

8、根据权利要求1所述的方法，其特征在于，所述在获取到所述云游戏客户端发送的主播对象的待处理音频数据时，对所述游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据，包括：

在获取到所述云游戏客户端发送的主播对象的待处理音频数据时，对所述待处理音频数据进行解码处理，得到所述主播对象的待处理音频帧以及所述待处理音频帧的录制时间戳；

基于所述游戏音频数据对应的第二采集时间间隔，获取所述游戏音频数据中的音频帧，将获取到的音频帧作为参考音频帧，且根据所述游戏音频数据对应的采集音频时间戳，确定所述参考音频帧的混音时间戳；及

基于所述混音时间戳调整所述录制时间戳，基于调整后的录制时间戳对所述参考音频帧和待处理音频帧进行混音处理，得到与所述采集时间戳保持一致的混音时间戳对应的混音音频帧，将得到的混音音频帧和所述混音音频帧对应的混音时间戳作为所述目标云游戏的混音音频数据。

9、根据权利要求8所述的方法，其特征在于，所述方法还包括：

统计获取到的音频帧的帧数，若所述音频帧的帧数为M帧，且M为不为零的正整数，则用M表征待进行混音处理的次数；

基于所述待进行混音处理的次数，在执行所述基于所述混音时间戳调整所述录制时间戳，基于调整后的录制时间戳对所述参考音频帧和待处理音频帧进行混音处理的步骤时，对所述待处理音频帧的音量进行增强处理，且对所述参考音频帧的音量进行减弱处理。

10、根据权利要求9所述的方法，其特征在于，所述方法还包括：

若所述音频帧的帧数为空值或者所述待处理音频帧的帧数为空值，则不对所述参考音频帧和待处理音频帧进行混音处理。

11、根据权利要求1所述的方法，其特征在于，所述云服务器包含基于所述云游戏直播请求所启动的媒体转发进程；所述媒体转发进程用于保持所述云服务器与所述直播服务器之间的所述直播服务连接；

所述基于所述混音音频数据、所述音频采集时间戳、所述游戏图像数据和所述图像采集时间戳，得到所述目标云游戏对应的直播数据流，将所述直播数据流通过所述直播服务器下发至与所述云游戏客户端相关联的直播客户端，包括：

获取所述混音音频数据，从所述混音音频数据中获取混音音频帧以及所述混音音频帧对应的所述音频采集时间戳，基于所述音频采集时间戳对所述混音音频帧进行排序处理，将排序处理后的混音音频帧，写入与所述媒体转发进程相关联的音频缓存队列；

获取所述游戏图像数据，从所述游戏图像数据中获取图像帧以及所述图像帧对应的所述图像采集时间戳，基于所述图像采集时间戳对所述图像帧进行排序处理，将排序处理后的图像帧，写入与所述媒体转发进程相关联的视频缓存队列；

获取与媒体转发进程相关联的平滑获取线程，通过所述平滑获取线程从所述视频缓存队列中获取与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，并通过所述平滑获取线程从所述音频缓存队列中获取与第二采集时间间隔所指示的第二帧率相匹配的音频数据包；及

将所述视频数据包和所述音频数据包进行打包处理，得到所述目标云游戏对应的直播数据流，将所述直播数据流通过所述直播服务器下发至与所述云游戏客户端相关联的直播客户端。

12、根据权利要求11所述的方法，其特征在于，所述获取与媒体转发进程相关联的平滑获取线程，通过所述平滑获取线程从所述视频缓存队列中采集与第一采集时间间隔所指示的第一帧率相匹配的视频数据包，并通过所述平滑获取线程从所述音频缓存队列中采集与第二采集时间间隔所指示的第二帧率相匹配的音频数据包，包括：

获取与媒体转发进程相关联的平滑获取线程，且获取与所述平滑获取线程具有相同采集时间间隔的数据采集组件，基于所述数据采集组件的采集时间间隔确定与所述游戏图像数据相关联的第一采集时间间隔和与所述游戏音频数据相关联的第二采集时间间隔；

通过所述平滑获取线程确定与所述第一采集时间间隔相匹配的平滑视频获取时间戳，从所述视频缓存队列中获取图像采集时间戳小于所述平滑视频获取时间戳的缓存视频帧，将获取到的缓存视频帧和所述缓存视频帧对应的图像采集时间戳进行封装处理，得到与所述第一

采集时间间隔所指示的第一帧率相匹配的视频数据包；及

通过所述平滑获取线程确定与所述第二采集时间间隔相匹配的平滑音频获取时间戳，从所述音频缓存队列中获取音频采集时间戳小于所述平滑音频获取时间戳的缓存音频帧，将获取到的缓存音频帧和所述缓存音频帧对应的音频采集时间戳进行封装处理，得到与所述第二采集时间间隔所指示的第二帧率相匹配的音频数据包。

13、根据权利要求 12 所述的方法，其特征在于，在所述将所述直播数据流通过所述直播服务器下发至与所述云游戏客户端相关联的直播客户端之后，所述方法还包括：

确定所述直播数据流的最大音频采集时间戳；及

若通过所述平滑获取线程记录到待发送音频数据包的音频采集时间戳小于所述最大音频采集时间戳，则丢弃所述待发送音频数据包所述待发送音频数据包，是指待发送给所述直播服务器的下一个音频数据包。

14、根据权利要求 12 所述的方法，其特征在于，在所述将所述直播数据流通过所述直播服务器下发至与所述云游戏客户端相关联的直播客户端之后，所述方法还包括：

确定所述直播数据流的最大图像采集时间戳；及

若通过所述平滑获取线程记录到待发送视频数据包的视频采集时间戳小于所述最大视频采集时间戳，则丢弃所述待发送视频数据包，所述待发送视频数据包，是指待发送给所述直播服务器的下一个视频数据包。

15、一种游戏数据处理装置，其特征在于，包括：

直播请求接收模块，用于接收云游戏客户端发送的针对目标云游戏的云游戏直播请求，基于所述云游戏直播请求建立云服务器与所述云游戏客户端相关联的直播服务器之间的直播服务连接；

数据获取模块，用于基于所述直播服务连接获取所述目标云游戏的游戏图像数据和游戏音频数据，记录所述游戏图像数据对应的图像采集时间戳和所述游戏音频数据对应的音频采集时间戳；所述图像采集时间戳和所述音频采集时间戳为同一参考系下的时间戳；

混音处理模块，用于在获取到所述云游戏客户端发送的主播对象的待处理音频数据时，对所述游戏音频数据和待处理音频数据进行混音处理，得到混音音频数据；所述混音音频数据对应的混音时间戳与所述音频采集时间戳保持一致；及

直播流下发模块，用于基于所述混音音频数据、所述音频采集时间戳、所述游戏图像数据和所述图像采集时间戳，得到所述目标云游戏对应的直播数据流，将所述直播数据流通过所述直播服务器下发至与所述云游戏客户端相关联的直播客户端。

16、一种计算机设备，包括存储器和一个或多个处理器，所述存储器存储有计算机可读指令，其特征在于，所述计算机可读指令被所述处理器执行时，使得所述一个或多个处理器执行权利要求 1 至 14 中任一项所述的方法的步骤。

17、一个或多个非易失性可读存储介质，存储有计算机可读指令，其特征在于，所述计算机可读指令被一个或多个处理器执行时，使得所述一个或多个处理器实现权利要求 1 至 14 中任一项所述的方法的步骤。

18、一种计算机程序产品，包括计算机可读指令，其特征在于，所述计算机可读指令被处理器执行时实现权利要求 1 至 14 中任一项所述的方法的步骤。