



(12) 发明专利

(10) 授权公告号 CN 115065684 B

(45) 授权公告日 2022. 12. 02

(21) 申请号 202210984461.1

A63F 13/352 (2014.01)

(22) 申请日 2022.08.17

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 113079216 A, 2021.07.06

申请公布号 CN 115065684 A

CN 113209632 A, 2021.08.06

(43) 申请公布日 2022.09.16

US 2018176097 A1, 2018.06.21

(73) 专利权人 腾讯科技(深圳)有限公司

审查员 白生斌

地址 518057 广东省深圳市南山区高新区

科技中一路腾讯大厦35层

(72) 发明人 赵新达 刘京洋 龚志鹏 杨衍东

(74) 专利代理机构 广州三环专利商标代理有限公司

公司 44202

专利代理师 唐宇鑫

(51) Int. Cl.

H04L 67/10 (2022.01)

H04L 9/40 (2022.01)

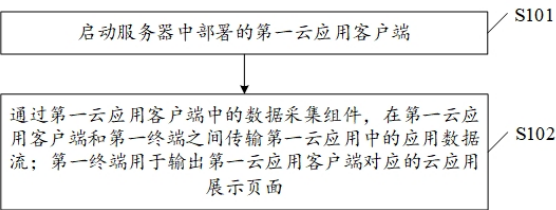
权利要求书3页 说明书25页 附图9页

(54) 发明名称

数据处理方法、装置、设备以及介质

(57) 摘要

本申请实施例提供了一种数据处理方法、装置、设备以及介质,该方法由服务器执行,服务器中包括多个云应用客户端,多个云应用客户端的数据传输过程是相互隔离的,多个云应用客户端包括第一云应用客户端,该方法包括:启动服务器中部署的第一云应用客户端;通过第一云应用客户端中的数据采集组件,在第一云应用客户端和第一终端之间传输第一云应用中的应用数据流;第一终端用于输出第一云应用客户端对应的云应用展示页面。采用本申请实施例,可以提高服务器的使用率,进而可以降低云应用的成本。



1. 一种数据处理方法,其特征在于,所述方法由服务器执行,所述服务器中包括多个云应用客户端,所述多个云应用客户端运行在所述服务器所提供的非虚拟化环境或同一个虚拟化环境中,所述多个云应用客户端的数据传输过程是相互隔离的,所述多个云应用客户端包括第一云应用客户端;所述方法包括:

启动所述服务器中部署的第一云应用客户端;

通过所述第一云应用客户端中的数据采集组件,在所述第一云应用客户端和第一终端之间直接传输第一云应用中的应用数据流;所述第一终端用于输出所述第一云应用客户端对应的云应用展示页面。

2. 根据权利要求1所述的方法,其特征在于,所述数据采集组件包括渲染抓取组件,所述应用数据流包括音视频数据;

所述通过所述第一云应用客户端中的数据采集组件,在所述第一云应用客户端和第一终端之间直接传输第一云应用中的应用数据流,包括:

通过所述第一云应用客户端中的所述渲染抓取组件,从所述第一云应用客户端对应的渲染缓冲区中获取图像数据;所述渲染缓冲区用于存储所述第一云应用客户端中渲染完成的数据;

通过所述第一云应用客户端中的所述渲染抓取组件,从所述第一云应用客户端对应的音频缓冲区中获取与所述图像数据相对应的音频数据;所述音频缓冲区用于存储所述第一云应用客户端中生成的音频;

基于所述图像数据和所述音频数据确定所述第一云应用客户端对应的音视频数据,将所述音视频数据直接传输至所述第一终端,以使所述第一终端输出所述音视频数据。

3. 根据权利要求2所述的方法,其特征在于,还包括:

通过所述第一云应用客户端所属的进程,调用所述服务器中的图形处理器驱动程序所提供的渲染接口;

基于所述渲染接口对所述第一云应用客户端中的页面展示数据进行渲染处理,得到所述第一云应用客户端对应的图像数据,将所述图像数据存储至所述渲染缓冲区。

4. 根据权利要求2所述的方法,其特征在于,所述通过所述第一云应用客户端中的所述渲染抓取组件,从所述第一云应用客户端对应的渲染缓冲区中获取图像数据,包括:

获取所述第一云应用客户端对应的渲染缓冲区的数据属性信息,根据所述数据属性信息,在所述第一云应用客户端的渲染抓取组件中创建纹理资源对象;

将所述第一云应用客户端对应的渲染缓冲区中的图像数据封装到所述纹理资源对象。

5. 根据权利要求2所述的方法,其特征在于,所述基于所述图像数据和所述音频数据确定所述第一云应用客户端对应的音视频数据,将所述音视频数据直接传输至所述第一终端,包括:

将所述图像数据和所述音频数据传输至所述第一云应用客户端中的数据压缩组件,通过所述数据压缩组件对所述图像数据和所述音频数据进行压缩处理,得到所述第一云应用客户端对应的音视频数据;

将所述音视频数据直接传输至所述第一终端,以使所述第一终端对所述音视频数据进行解压处理得到解压后的音视频数据,输出所述解压后的音视频数据。

6. 根据权利要求5所述的方法,其特征在于,所述图像数据位于所述渲染抓取组件中所

创建的纹理资源对象中；所述渲染抓取组件中还包括栅栏对象，所述栅栏对象用于测听所述纹理资源对象中的数据状态；

所述将所述图像数据和所述音频数据传输至所述第一云应用客户端中的数据压缩组件，通过所述数据压缩组件对所述图像数据和所述音频数据进行压缩处理，得到所述第一云应用客户端对应的音视频数据，包括：

将所述渲染抓取组件中的纹理资源对象设置为共享纹理资源，将所述渲染抓取组件中的栅栏对象设置为共享栅栏，将所述共享纹理资源和所述共享栅栏导入所述第一云应用客户端中的数据压缩组件；

当所述共享栅栏测听到所述共享纹理资源中的数据状态为已完成状态时，向所述数据压缩组件发送提醒信号；

在所述数据压缩组件中，基于所述提醒信号获取所述共享纹理资源对应的图像数据，以及获取所述渲染抓取组件中的所述音频数据，对所述图像数据和所述音频数据进行压缩处理，得到第一云应用客户端对应的音视频数据。

7. 根据权利要求2所述的方法，其特征在于，所述基于所述图像数据和所述音频数据确定所述第一云应用客户端对应的音视频数据，将所述音视频数据直接传输至所述第一终端，包括：

将所述图像数据和所述音频数据传输至数据压缩应用程序，通过所述数据压缩应用程序对所述图像数据和所述音频数据进行压缩处理，得到所述第一云应用客户端对应的音视频数据；所述数据压缩应用程序与所述第一云应用客户端在所述服务器中是互相独立的；

将所述音视频数据直接传输至所述第一终端，以使所述第一终端对所述音视频数据进行解压处理得到解压后的音视频数据，输出所述解压后的音视频数据。

8. 根据权利要求1所述的方法，其特征在于，所述数据采集组件包括操作处理组件，所述应用数据流包括对象操作数据，所述对象操作数据包括所述第一终端基于所述第一云应用客户端对应的云应用展示页面中的触发操作所生成的指令；

所述通过所述第一云应用客户端中的数据采集组件，在所述第一云应用客户端和第一终端之间直接传输云应用中的应用数据流，包括：

通过所述第一云应用客户端中的操作处理组件，接收所述第一终端传输的对象操作数据；

通过调用所述操作处理组件中的所述对象操作数据，执行所述第一云应用客户端中的业务逻辑。

9. 根据权利要求1所述的方法，其特征在于，所述启动服务器中部署的第一云应用客户端，包括：

接收所述第一终端发送的第一启动指令，基于所述第一启动指令中的应用标识信息，在所述服务器中部署所述第一云应用客户端；

基于所述第一启动指令中的对象信息，在所述服务器中启动所述第一云应用客户端。

10. 根据权利要求1所述的方法，其特征在于，所述多个云应用客户端还包括第二云应用客户端；

所述方法还包括：

当接收到第二终端发送的第二启动指令时，基于所述第二启动指令，在所述服务器中

创建并启动第二云应用客户端；

通过所述第二云应用客户端中的数据采集组件，在所述第二云应用客户端和第二终端之间直接传输第二云应用中的应用数据流；所述第二终端用于输出所述第二云应用客户端对应的云应用展示页面。

11. 一种数据处理装置，其特征在于，所述装置运行在服务器中，所述服务器中包括多个云应用客户端，所述多个云应用客户端运行在所述服务器所提供的非虚拟化环境或同一个虚拟化环境中，所述多个云应用客户端的数据传输过程是相互隔离的，所述多个云应用客户端包括第一云应用客户端；所述装置包括：

第一启动模块，用于启动所述服务器中部署的第一云应用客户端；

第一传输模块，用于通过所述第一云应用客户端中的数据采集组件，在所述第一云应用客户端和第一终端之间直接传输第一云应用中的应用数据流；所述第一终端用于输出所述第一云应用客户端对应的云应用展示页面。

12. 一种计算机设备，其特征在于，包括存储器和处理器；

所述存储器与所述处理器相连，所述存储器用于存储计算机程序，所述处理器用于调用所述计算机程序，以使得所述计算机设备执行权利要求1-10任一项所述的方法。

13. 一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有计算机程序，所述计算机程序适于由处理器加载并执行，以使得具有所述处理器的计算机设备执行权利要求1-10任一项所述的方法。

数据处理方法、装置、设备以及介质

技术领域

[0001] 本申请涉及云应用技术领域,尤其涉及一种数据处理方法、装置、设备以及介质。

背景技术

[0002] 云应用是以云计算为基础的应用,在云应用的运行模式下,所有应用程序都在服务器端运行,并将渲染完毕后的画面压缩后通过网络传送给用户。目前的云应用使用场景中,云应用通常运行在服务器端的虚拟化环境中,在服务器端渲染完成的结果最终都会传送到虚拟化环境中的显示设备所对应的显示缓冲区(framebuffer),进而由某个程序读取显示缓冲区中的内容传送给用户;用户针对云应用的操作数据可以传送到虚拟化环境中的输入设备,之后可以从输入设备中获取相应的操作数据注入云应用。

[0003] 通常情况下,一个虚拟化环境对应一个显示设备和一个输入设备,若存在多个用户在同一时间使用云应用,则需要在服务器端运行多个虚拟化环境才能实现各个用户所使用的云应用数据之间的相互隔离,即一个虚拟化环境只能运行一个云应用客户端,这样会造成云应用的使用成本过高。

发明内容

[0004] 本申请实施例提供一种数据处理方法、装置、设备以及介质,可以提高服务器的使用率,进而可以降低云应用的成本。

[0005] 本申请实施例一方面提供了一种数据处理方法,该方法由服务器执行,该服务器中包括多个云应用客户端,多个云应用客户端的数据传输过程是相互隔离的,多个云应用客户端中包括第一云应用客户端;该方法包括:

[0006] 启动服务器中部署的第一云应用客户端;

[0007] 通过第一云应用客户端中的数据采集组件,在第一云应用客户端和第一终端之间传输第一云应用中的应用数据流;第一终端用于输出第一云应用客户端对应的云应用展示页面。

[0008] 本申请实施例一方面提供了一种数据处理装置,该装置运行在服务器中,该服务器中包括多个云应用客户端,多个云应用客户端的数据传输过程是相互隔离的,多个云应用客户端中包括第一云应用客户端;该装置包括:

[0009] 第一启动模块,用于启动服务器中部署的第一云应用客户端;

[0010] 第一传输模块,用于通过第一云应用客户端中的数据采集组件,在第一云应用客户端和第一终端之间传输第一云应用中的应用数据流;第一终端用于输出第一云应用客户端对应的云应用展示页面。

[0011] 其中,数据采集组件包括渲染抓取组件,应用数据流包括音视频数据;

[0012] 第一传输模块包括:

[0013] 图像数据获取单元,用于通过第一云应用客户端中的渲染抓取组件,从第一云应用客户端对应的渲染缓冲区中获取音视频数据;渲染缓冲区用于存储第一云应用客户端中

渲染完成的数据；

[0014] 音频数据获取单元，用于通过第一云应用客户端中的渲染抓取组件，从第一云应用客户端对应的音频缓冲区中获取与图像数据相对应的音频数据；音频缓冲区用于存储第一云应用客户端中生成的音频；

[0015] 音视频数据传输单元，用于基于图像数据和音频数据确定第一云应用客户端对应的音视频数据，将音视频数据传输至第一终端，以使第一终端输出音视频数据。

[0016] 其中，该装置还包括：

[0017] 渲染接口调用模块，用于通过第一云应用客户端所属的进程，调用服务器中的图形处理器驱动程序所提供的渲染接口；

[0018] 渲染处理模块，用于基于渲染接口对第一云应用客户端中的页面展示数据进行渲染处理，得到第一云应用客户端对应的图像数据，将图像数据存储至渲染缓冲区。

[0019] 其中，图像数据获取单元具体用于：

[0020] 获取第一云应用客户端对应的渲染缓冲区的数据属性信息，根据数据属性信息，在第一云应用客户端的渲染抓取组件中创建纹理资源对象；

[0021] 将第一云应用客户端对应的渲染缓冲区中的图像数据封装到纹理资源对象。

[0022] 其中，音视频数据传输单元包括：

[0023] 第一压缩子单元，用于将图像数据和音频数据传输至第一云应用客户端中的数据压缩组件，通过数据压缩组件对图像数据和音频数据进行压缩处理，得到第一云应用客户端对应的音视频数据；

[0024] 压缩数据传输子单元，用于将音视频数据传输至第一终端，以使第一终端对音视频数据进行解压处理得到解压后的音视频数据，输出解压后的音视频数据。

[0025] 其中，图像数据位于渲染抓取组件中所创建的纹理资源对象中；渲染抓取组件中还包括栅栏对象，栅栏对象用于测听纹理资源对象中的数据状态；

[0026] 第一压缩子单元具体用于：

[0027] 将渲染抓取组件中的纹理资源对象设置为共享纹理资源，将渲染抓取组件中的栅栏对象设置为共享栅栏，将共享纹理资源和共享栅栏导入第一云应用客户端中的数据压缩组件；

[0028] 当共享栅栏测听到共享纹理资源中的数据状态为已完成状态时，向数据压缩组件发送提醒信号；

[0029] 在数据压缩组件中，基于提醒信号获取共享纹理资源对应的图像数据，以及获取渲染抓取组件中的音频数据，对图像数据和音频数据进行压缩处理，得到第一云应用客户端对应的音视频数据。

[0030] 其中，音视频数据传输单元包括：

[0031] 第二压缩子单元，用于将图像数据和音频数据传输至数据压缩应用程序，通过数据压缩应用程序对图像数据和音频数据进行压缩处理，得到第一云应用客户端对应的音视频数据；数据压缩应用程序与第一云应用客户端在服务器中是互相独立的；

[0032] 压缩数据传输子单元，用于将音视频数据传输至第一终端，以使第一终端对音视频数据进行解压处理得到解压后的音视频数据，输出解压后的音视频数据。

[0033] 其中，数据采集组件包括操作处理组件，应用数据流包括对象操作数据，对象操作

数据包括第一终端基于第一云应用客户端对应的云应用展示页面中的触发操作所生成的指令；

[0034] 第一传输模块包括：

[0035] 操作数据接收单元，用于通过第一云应用客户端中的操作处理组件，接收第一终端传输的对象操作数据；

[0036] 操作数据调用单元，用于通过调用操作处理组件中的对象操作数据，执行第一云应用客户端中的业务逻辑。

[0037] 其中，第一启动模块包括：

[0038] 指令接收单元，用于接收第一终端发送的第一启动指令，基于第一启动指令中的应用标识信息，在服务器中部署第一云应用客户端；

[0039] 云应用启动单元，用于基于第一启动指令中的对象信息，在服务器中启动第一云应用客户端。

[0040] 其中，多个云应用客户端还包括第二云应用客户端；

[0041] 该装置还包括：

[0042] 第二启动模块，用于当接收到第二终端发送的第二启动指令时，基于第二启动指令，在服务器中创建并启动第二云应用客户端；

[0043] 第二传输模块，用于通过第二云应用客户端中的数据采集组件，在第二云应用客户端和第二终端之间传输第二云应用中的应用数据流；第二终端用于输出第二云应用客户端对应的云应用展示页面。

[0044] 本申请实施例一方面提供了一种计算机设备，包括存储器和处理器，存储器与处理器相连，存储器用于存储计算机程序，处理器用于调用计算机程序，以使得该计算机设备执行本申请实施例中上述一方面提供的方法。

[0045] 本申请实施例一方面提供了一种计算机可读存储介质，计算机可读存储介质中存储有计算机程序，计算机程序适于由处理器加载并执行，以使得具有处理器的计算机设备执行本申请实施例中上述一方面提供的方法。

[0046] 根据本申请的一个方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。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，使得该计算机设备执行上述一方面提供的方法。

[0047] 本申请实施例中，可以在服务器端创建多个云应用客户端，多个云应用客户端的数据传输过程是相互隔离的，这多个云应用客户端中包括第一云应用客户端；可以启动服务器中部署的第一云应用客户端，通过第一云应用客户端中的数据采集组件，在第一云应用客户端与第一终端之间传输第一云应用中的应用数据流；该第一终端用于输出第一云应用客户端对应的云应用展示页面。可见，运行在服务器中的云应用客户端与第一终端之间可以直接进行数据传输，这样就可以就确保同一个服务端中的多个云应用客户端与其各自对应的终端设备之间的数据交互是独立的，互不影响，可以提高服务器的使用率，进而可以降低云应用的成本。

附图说明

[0048] 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。

[0049] 图1是本申请实施例提供的一种云应用的处理系统的架构图;

[0050] 图2是本申请实施例提供的一种云游戏的数据交互场景示意图;

[0051] 图3是本申请实施例提供的一种数据处理方法的流程示意图;

[0052] 图4是本申请实施例提供的一种云应用的数据交互示意图;

[0053] 图5是本申请实施例提供的另一种数据处理方法的流程示意图;

[0054] 图6是本申请实施例提供的一种云应用客户端内部集成的渲染抓取组件抓取图像数据的示意图;

[0055] 图7是本申请实施例提供的一种云应用客户端中的图像数据的压缩处理示意图;

[0056] 图8是本申请实施例提供的一种云应用客户端在服务器中的数据压缩结构示意图;

[0057] 图9是本申请实施例提供的一种云应用客户端内部集成的操作处理组件接收对象操作数据的示意图;

[0058] 图10是本申请实施例提供的一种数据处理装置的结构示意图;

[0059] 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

[0060] 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。

[0061] 本申请实施例涉及云计算(cloud computing)和云应用。其中,云计算是一种计算模式,它将计算任务分布在大量计算机构成的资源池上,使各种应用系统能够根据需要获取计算力、存储空间和信息服务。提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的,并且可以随时获取,按需使用,随时扩展,按使用付费。作为云计算的基础能力提供商,会建立云计算资源池(简称云平台,一般称为IaaS(Infrastructure as a Service,基础设施即服务)平台,在资源池中部署多种类型的虚拟资源,供外部客户选择使用。云计算资源池中主要包括:计算设备(为虚拟化机器,包含操作系统)、存储设备、网络设备。

[0062] 云应用作为云计算的子集,是云计算技术在应用层的体现,云应用的工作原理是把传统软件本地安装、本地运算的使用方式变为即取即用的服务,通过互联网或局域网连接并操控远程服务器集群,完成业务逻辑或运算任务的一种新型应用。云应用的优点是云应用的应用程序(如云应用客户端)运行在服务器端中,服务器端执行云应用的计算工作比如数据渲染,然后将云应用的计算结果传输给终端设备中的用户客户端进行显示,用户客户端可以采集用户的操作信息(也可以称为云应用的对象操作数据,或者可以称为云应用

的输入事件数据),将这些操作信息传输给服务器端中的云应用客户端,以实现服务器端对云应用的操控。其中,本申请实施例中所涉及的云应用客户端均为运行在服务器端的云应用实例,而用户客户端可以是指支持安装在终端设备中,且能够为用户提供对应的云应用体验服务的客户端,简单来说,用户客户端可以用于输出对应云应用客户端的云应用展示页面,也可以称为云应用用户客户端,后面不再对此进行解释;云应用可以包括云游戏、云教育、云安全、云会议、云呼叫以及云社交等等,云游戏作为云应用中的典型,近年来受到越来越多的关注。

[0063] 云游戏又可以称为游戏点播(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,游戏本身并不在用户所使用的游戏终端,游戏终端中仅运行用户客户端,真正的游戏应用程序(如云游戏客户端)是在服务器端中运行,并由服务器端将云游戏中的页面展示数据渲染为音视频码流,并将渲染完成的音视频码流传输给游戏终端中的用户客户端,由用户客户端对接收到的音视频码流进行显示。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用户输入事件数据并发送给云游戏客户端的能力即可。用户在体验云游戏时,其本质是在对云游戏的音视频码流进行操作,如通过触屏、键盘鼠标、摇杆等生成输入事件数据(或者称为对象操作数据,或者可以称为用户操作指令),然后通过网络传输到服务器端中的云游戏客户端,以达到操作云游戏的目的。

[0064] 其中,本申请所涉及的游戏终端可以是指玩家在体验云游戏时所使用的终端设备,即安装了与云游戏客户端相对应的用户客户端的终端设备,此处的玩家可以是指正在体验云游戏或者请求体验云游戏的用户;音视频码流可以包括云游戏客户端所生成的音频流和视频流,该音频流可以包括云游戏客户端在运行过程中所产生的持续的音频数据,视频流可以包括云游戏在运行过程中渲染完成的图像数据,如视频流可以认为是由云游戏客户端中渲染完成的一系列游戏画面(即渲染完成的图像数据)所构成的视频序列,那么此时的图像数据也可以认为是视频流中的视频帧。

[0065] 在云应用(例如,云游戏)的运行过程中,涉及服务器端中的云应用客户端与终端设备(例如,游戏终端)之间的通信连接(具体可以为云应用客户端与终端设备中的用户客户端之间的通信连接),云应用客户端与终端设备之间成功建立通信连接之后,在云应用客户端和终端设备之间可以传输云应用中的云应用数据流,如云应用数据流可以包括视频流(包括云应用客户端在运行过程中所产生的一系列图像数据)和音频流(包括云应用客户端在运行过程中产生的音频数据,为便于理解,此处的音频数据和前述图像数据可以统称为音视频数据),那么可以由该云应用客户端将视频流、音频流传输至终端设备;又比如云应用数据流可以包括终端设备所获取到的针对该云应用的对象操作数据,那么可以由该终端设备将对象操作数据传输至服务器端运行的云应用客户端。

[0066] 下面对本申请实施例涉及的基础概念进行解释说明:

[0067] 虚拟机:操作系统虚拟化的一种类型,操作系统运行在通过软件(例如,QEMU(一种虚拟操作系统模拟器))关联或模拟的硬件环境中,操作系统具有独立的内核态和用户态。

[0068] 容器:操作系统虚拟化的一种类型,多个操作系统通过内核态隔离机制,在内核态共用同一内核,在用户态保持相互独立。

[0069] 云应用实例:在服务器端,包含完整云应用功能的一组软件集合可以称为一个云应用实例;例如,包含完整云游戏功能的一组软件集合可以称为一个云游戏实例。

[0070] PC(personal computer,个人计算机)云游:Guest操作系统通常可以为Windows系统或者Linux系统,云游戏的应用程序运行在Windows系统或Linux系统中。其中,Guest操作系统可以是指虚拟机或容器内部运行的操作系统,虚拟机或容器可以运行于通用操作系统之上,此时的通用操作系统可以称为Host操作系统。

[0071] 渲染缓冲区:运行在服务器端的云应用客户端调用GPU驱动提供的渲染接口进行渲染操作,渲染完成后的图像数据所存放的区域可以称为渲染缓冲区,该渲染缓存区可以是指服务器为云应用客户端分配的GPU(Graphics Processing Unit,图形处理器)显存,是用来存储云应用客户端渲染完成的图像数据的部件,其作用类似于计算机的内存。

[0072] 请参见图1,图1是本申请实施例提供的一种云应用的处理系统的架构图。如图1所示,该云应用的处理系统可以包括终端设备10a、终端设备10c以及服务器10b等;图1所示的云应用的处理系统中终端设备和服务器的数量仅为举例说明,在实际应用场景中,可以根据需求来确定云应用的处理系统中的终端设备和服务器的具体数量,如终端设备和服务器的数量可以为一个或多个,本申请不对终端设备和服务器的数量进行限定。

[0073] 其中,服务器10b端可以运行云应用的应用程序(即云应用客户端),该服务器10b可以是独立的服务器,或者是多个服务器构成的服务器集群或者分布式系统,或者为提供云服务、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(Content Delivery Network,内容分发网络)、以及大数据和人工智能平台等基础云计算服务的云服务器,本申请不对服务器10b的类型进行限定。

[0074] 终端设备10a和终端设备10c中都可以包括与服务器10b中的云应用客户端相关联的用户客户端,该终端设备10a和终端设备10c均可以包括:智能手机(如Android手机、iOS手机等)、台式电脑、平板电脑、便携式个人计算机、移动互联网设备(Mobile Internet Devices,MID)以及可穿戴设备(例如智能手表、智能手环等)、车载设备等电子设备,本申请实施例不对云应用的处理系统中的终端设备的类型进行限定。

[0075] 如图1所示,服务器10b中可以运行一个或多个云应用客户端(此处的一个云应用客户端可以认为是一个云应用实例),一个应用客户端对应一个用户,即一个云应用客户端可以对应一个终端设备;服务器10b中所运行的一个或多个云应用客户端可以为同一个云应用,也可以为不同的云应用。例如,用户A和用户B在相同的时间体验云应用1时,此时可以在服务器10b中为用户A和用户B都创建一个云应用1实例;用户A和用户B在相同的时间体验不同的云应用(例如,用户A体验云应用1,用户B体验云应用2)时,此时可以在服务器10b中为用户A创建一个云应用1实例,为用户B创建一个云应用2实例。

[0076] 其中,终端设备10a和终端设备10c均可以是玩家所使用的电子设备,此处的玩家可以是指正在体验过云应用或者请求体验云应用的用户,一个终端设备中可以集成一个或多个用户客户端,每一个用户客户端都可以与服务器10b中对应的云应用客户端建立通信连接,用户客户端与其对应的云应用客户端之间可以通过该通信连接进行数据交互。如终端设备10a中的用户客户端可以基于该通信连接接收云应用客户端发送的音视频数据(例如,可以包括云应用客户端对应的图像数据和音频数据),并输出接收到的音视频数据;相应地,终端设备10a也可以将获取到的对象操作数据发送给对应的云应用客户端,以使服务

器端的云应用客户端可以基于注入的对象操作数据执行相应的业务逻辑。

[0077] 可以理解的是,在云应用场景下,云应用客户端均是运行在服务器端的,为了降低云应用的成本,所采用的方法通常是提高单个服务器中所运行的云应用实例的数量,一个云应用实例可以认为是一个云应用客户端,一个云应用客户端对应一个用户。本申请实施例中,图1所示的云应用的处理系统可以应用在单服务器的云应用并发运行场景(可以理解为单个服务器中同时运行多个云应用实例)中,本申请服务器端(例如,服务器10b)的多个云应用客户端可以运行在该服务器所提供的虚拟机,或容器,或其它类型的虚拟化环境中,也可以运行在该服务器所提供的非虚拟化环境中(如直接在服务器端的真实操作系统上运行),本申请对此不做限定。其中,服务器10b中运行的多个云应用客户端可以共享使用该服务器10b对应的物理GPU硬件;服务器10b中运行的每一个云应用客户端都可以通过其内部集成的特定功能组件(为便于理解,可以将此处的特定功能组件称为数据采集组件),与其对应的终端设备中的用户客户端之间直接进行数据传输。

[0078] 例如,若服务器端(例如,服务器10b)的多个云应用客户端均运行在该服务器所提供的同一个虚拟机中,那么该虚拟机中运行的各个云应用客户端中都各自集成了独立的数据采集组件,任一个云应用客户端内部集成的数据采集组件可以抓取该云应用客户端所生成的音视频数据,并将其直接传输至对应终端设备中的用户客户端,即云应用客户端所产生的音视频数据的传输过程无需与虚拟机对应显示设备、音频设备等产生交互操作;而终端设备中的用户客户端所产生的对象操作数据(可以称为输入事件数据,如用户在终端设备的用户客户端中输入的操作指令等),同样可以从用户客户端直接传输至对应云应用客户端内部集成的数据采集组件,云应用客户端中的其他组件(该云应用客户端中除数据采集组件之外的组件)需要调用对象操作数据时可以从数据采集组件中直接调用,即对象操作数据的传输无需与虚拟机对应的输入设备产生交互操作;这样可以减少云应用客户端与终端设备中的用户客户端之间的数据传输过程中的交互操作,进而可以提高性能和响应速度。

[0079] 需要说明的是,服务器端运行的云应用客户端通常对显示设备、音频设备、输入设备具有独占需求,在现有的云应用并发运行场景中,需要为服务器端的每个虚拟机分别模拟一个显示设备、一个音频设备以及一个输入设备(服务端本身通常是不具备显示设备、音频设备、输入设备的),即一个虚拟机对应一个显示设备、一个音频设备以及一个输入设备。例如,现有技术中的云应用客户端C1(可以认为是服务器端的任一虚拟机中所运行的云应用客户端)渲染完成的图像数据最终会合成到显示设备D1(运行该云应用客户端C1的虚拟机所对应的显示设备)对应的显示缓冲区(framebuffer),这样虚拟机中的独立抓取程序通过读取该显示缓冲区中的内容,才能够确保该独立抓取程序抓取到的内容是云应用客户端C1对应的图像数据,而不是其他云应用客户端(服务器端除云应用客户端C1之外的云应用客户端)对应的图像数据;也就是说,一个虚拟机对应一个模拟的显示设备,该虚拟机中包含独立抓取程序,该独立抓取程序通过读取当前虚拟机的显示设备所对应的显示缓冲区中的内容,才能够确保抓取到当前虚拟机中所运行的云应用客户端的图像数据,即现有技术中必须为虚拟机模拟一个显示设备,且一个虚拟机仅能运行一个云应用客户端,才能实现不同云应用客户端对应的图像数据之间的隔离。其中,显示缓存区可以用来驱动其对应的显示设备,该显示缓存区可以模仿显卡的功能,通过显示缓存区的读写直接对显

存进行操作。同样地,与前述显示设备的功能类似,现有技术方案中必须为虚拟机模拟一个音频设备,才能实现不同云应用客户端对应的音频数据之间的隔离,为每个虚拟机模拟一个输入设备才能实现不同云应用客户端对应的对象操作数据之间的隔离。

[0080] 本申请实施例所提供的技术方案中,服务器端运行的多个云应用客户端都可以由各自所在的进程执行渲染操作,渲染完成的图像数据会存放在渲染缓冲区,通过各个云应用客户端内部集成独立的数据采集组件,通过内部集成的数据采集组件可以直接从当前云应用客户端对应的渲染缓冲区读取渲染完成的图像数据,也可以直接抓取当前云应用客户端所产生的与图像数据相对应的音频数据,并将抓取到的图像数据、音频数据直接传输到终端设备进行输出。换言之,本申请实施例中的图像数据、音频数据均是由各个云应用客户端所在进程独立生成的,且该云应用客户端对应的图像数据、音频数据的抓取过程也是由该云应用客户端内部集成的数据采集组件来完成的,也就是说,图像数据、音频数据的生成过程和抓取过程均是在云应用客户端内部完成的,那么通过数据采集组件抓取到的图像数据、音频数据必然是当前云应用客户端对应的音视频数据,也不会存在数据混淆的情况。同样地,对于终端设备产生的对象操作数据,也可以直接传输至云应用客户端内部集成的数据采集组件,该数据采集组件中的对象操作数据可供云应用客户端中的其他组件进行调用。本申请实施例通过各个云应用客户端的内部集成的数据采集组件,就可以实现不同云应用客户端数据之间的相互隔离,这样就可以规避云应用客户端对显示设备、输入设备、音频设备等的独占需求,进而可以减少数据传输过程中的交互操作,减少数据传输路径,提高性能和响应速度;此外,由于各个云应用客户端与其各自对应的终端设备之间的数据传输,均可以由云应用客户端内部集成的数据采集组件来完成,那么通过本申请实施例所提供的技术方案,可以在服务器端所提供的一个虚拟机中,或者直接在服务器端的非虚拟化环境中就可以并发运行多个云应用客户端,进而可以避免在服务器端启动多个虚拟机所带来的CPU(Central Processing Unit,中央处理器)算力损耗,提高服务器的利用率,降低云应用的成本。

[0081] 为便于理解,下面以云应用是云游戏为例,对云应用的处理系统中的服务器10b和终端设备10a、终端设备10b之间的数据交互过程进行描述。请参见图2,图2是本申请实施例提供的一种云游戏的数据交互场景示意图。如图2所示的服务器10b可以为运行两个或两个以上云应用客户端的服务器,终端设备10a可以为用户A所使用的电子设备,该终端设备10a可以集成一个或多个与不同类型云游戏相关联的用户客户端,此处的用户客户端可以理解为安装在终端设备上的,且能够为用户提供对应的云游戏体验服务的客户端。例如,终端设备10a中的用户客户端21是与云游戏1相关联的客户端,那么该用户客户端21在终端设备10a中的图标可以为云游戏1的图标,该用户客户端21可以为用户提供云游戏1体验服务,即用户通过终端设备10a中的用户客户端21可以体验云游戏1。

[0082] 当用户A想要体验云游戏1时,可以对终端设备10a中的用户客户端21执行触发操作,此时的终端设备10a可以在联网状态下响应针对用户客户端21的启动操作,获取该用户客户端21中生成的启动指令,进而可以将该启动指令发送至服务器10b,这时可以在服务器10b中为用户A创建或分配一个云游戏1实例(即为用户A创建或分配一个云游戏1对应的云游戏客户端22),并在该服务器10b中运行该用户A对应的云游戏客户端22;与此同时,终端设备10a中的用户客户端21也会成功启动,即终端设备10a中的用户客户端21与服务器10b

中的云游戏客户端22保持相同的运行状态。应当理解,若服务器10b中已经预先部署了云游戏1实例,那么服务器10b在接收到用户客户端21的启动指令后,可以直接从服务器10b中为用户A分配一个云游戏1实例,并启动该云游戏1实例,这样可以提高云游戏1的启动时间,减少用户客户端21显示云游戏1页面的等待时间;若服务器10b中没有预先部署云游戏1实例,那么服务器10b在接收到用户客户端21的启动指令后,需要在服务器10b中为该用户A创建一个云游戏1实例,并启动该新创建的云游戏1实例。

[0083] 同理,如图2所示的终端设备10c可以为用户B所使用的电子设备,该终端设备10c同样可以集成一个或多个与不同类型云游戏相关联的用户客户端。例如,终端设备10c中的用户客户端23是与云游戏2相关联的客户端,那么该用户客户端23在终端设备10c中的图标可以为云游戏2的图标,当用户B想要体验云游戏2时,可以对终端设备10c中的用户客户端23执行触发操作,此时的终端设备10c可以在联网状态下响应针对用户客户端23的启动操作,获取该用户客户端23中生成的启动指令,进而可以将该启动指令发送至服务器10b,这时可以在服务器10b中为用户B创建或分配一个云游戏2实例(即为用户B创建或分配一个云游戏2对应的云游戏客户端24),并在该服务器10b中运行该用户B对应的云游戏客户端24;与此同时,终端设备10c中的用户客户端23也会成功启动,即终端设备10c中的用户客户端23与服务器10b中的云游戏客户端24保持相同的运行状态。换言之,服务器10b中可以运行多个云游戏客户端,如前述用户A对应的云游戏客户端22,以及用户B对应的云游戏客户端24。

[0084] 如图2所示,云游戏客户端22在服务器10b中运行时,可以执行云游戏1中的游戏逻辑,并通过该云游戏客户端22所在的进程调用GPU驱动程序提供的渲染接口执行渲染操作,渲染完成的结果可以如图2所示的游戏画面20a,云游戏客户端22内部集成的数据采集组件可以抓取渲染完成的游戏画面20a,并将该游戏画面20a直接发送给终端设备10a中的用户客户端21;终端设备10a中的用户终端21在接收到游戏画面20a后,可以对其进行展示。

[0085] 需要说明的是,云游戏客户端22实质上可以认为是服务器端包含完整云游戏功能的一组软件集合,其本身是静止的,该云游戏客户端22需要建立其对应的进程才能在服务器10b中运行,进程本身是动态的。换言之,在需要启动服务器10b中的云游戏客户端22时,可以在服务器10b中建立该云游戏客户端22对应的进程,并启动该云游戏客户端22所在的进程;也就是说,在服务器10b中运行云游戏客户端22的实质是在服务器10b中运行该云游戏客户端22所在的进程,该进程可以认为是云游戏客户端22在服务器10b中的基本执行实体。服务器10b中运行的操作系统可以包含GPU驱动程序,该GPU驱动程序可以为该服务器10b中运行的云游戏客户端提供渲染接口,例如,云游戏客户端22所在的进程需要调用该GPU驱动程序所提供的渲染接口执行渲染操作,以得到该云游戏客户端22的渲染结果(如游戏画面20a)。

[0086] 其中,该游戏画面20a中可以包括游戏控件20c,用户A可以通过触发该游戏控件20c来操作云游戏1,如该游戏控件20c可以用于操控云游戏1中的虚拟对象20d的移动、射击、隐身等功能。当用户A对游戏控件20c执行触发操作时,用户客户端21可以获取该用户A的操作数据(对象操作数据),并将用户A的操作数据直接传输至服务器10b中运行的云游戏客户端22内部集成的数据采集组件,云游戏客户端22可以直接从其内部集成的数据采集组件中调用该用户A的操作数据,继续执行渲染操作以生成新的游戏画面;云游戏客户端22渲

染完成的每一个游戏画面,都可以由该云游戏客户端22实时传送至终端设备10a中的用户客户端21进行显示;而用户客户端21获取到的每一个操作数据,都可以注入至服务器10b中运行的云游戏客户端22,以此来执行服务器10b中的云游戏客户端22与终端设备10a中的用户客户端21之间的数据交互。

[0087] 如图2所示,云游戏客户端24在服务器10b中运行时,可以执行云游戏2中的游戏逻辑,并通过该云游戏客户端24所在的进程调用GPU驱动程序(该GPU驱动程序为服务器10b所运行的操作系统中的驱动程序)提供的渲染接口执行渲染操作,渲染完成的结果可以如图2所示的游戏画面20b,云游戏客户端24内部集成的数据采集组件可以抓取渲染完成的游戏画面20b,并将该游戏画面20b直接发送给终端设备10c中的用户客户端23,并在用户客户端23中显示该游戏画面20b;终端设备10c中的用户终端23在接收到游戏画面20b后,可以对其进行展示。

[0088] 若该游戏画面20b中包括游戏控件20e,用户B可以通过触发该游戏控件20e来操作云游戏2。当用户A对游戏控件20e执行触发操作时,用户客户端23可以获取该用户B的操作数据,并将用户B的操作数据直接传输至服务器10b中运行的云游戏客户端24内部集成的数据采集组件,云游戏客户端24可以直接从其内部集成的数据采集组件中调用该用户B的操作数据,继续执行渲染操作以生成新的游戏画面。同样地,云游戏客户端24渲染完成的每一个游戏画面,都可以由该云游戏客户端24实时传送至终端设备10a中的用户客户端23进行显示;而用户客户端23获取到的每一个操作数据,都可以注入至服务器10b中运行的云游戏客户端24,以此来执行服务器10b中的云游戏客户端24与终端设备10a中的用户客户端23之间的数据交互。

[0089] 本申请实施例中,服务器10b中并发运行的云游戏客户端22(生成游戏画面20a的云游戏客户端)和云游戏客户端24(生成游戏画面20b的云游戏客户端)都可以通过各自的进程执行渲染操作,渲染而成的结果(例如,游戏画面20a、游戏画面20b)是互相独立、互不影响的,且各自渲染完成的结果都可以由各个云游戏客户端内部集成的数据采集组件传送至各自对应的用户客户端进行显示。换言之,云游戏客户端22与用户A所使用的终端设备10a可以通过云游戏客户端22内部集成的数据采集组件直接进行数据交互,云游戏客户端24与用户B所使用的终端设备10c可以通过云游戏客户端24内部集成的数据采集组件直接进行数据交互,其数据交互过程互相隔离。可见,本申请实施例所提供的技术方案中,通过各个云游戏客户端内部各自集成的数据采集组件就可以实现不同云游戏客户端数据之间的相互隔离,因此在服务器端运行的多个云游戏客户端无需通过配置多个虚拟机来进行数据隔离,可以避免在服务器端启动多个虚拟机所带来的CPU算力损耗,还可以提高服务器的利用率,降低云游戏的成本。

[0090] 请参见图3,图3是本申请实施例提供的一种数据处理方法的流程示意图。可以理解地,该数据处理方法由服务器(例如,图1所示的云应用的处理系统中的服务器10b)执行。其中,该数据处理方法可以包括以下步骤S101至步骤S102:

[0091] 步骤S101,启动服务器中部署的第一云应用客户端。

[0092] 本申请实施例中,服务器中可以创建多个云应用客户端(此处的云应用客户端可以理解为一个云应用实例,例如,图2所对应实施例中的云游戏客户端22和云游戏客户端24),该服务器中的多个云应用客户端的数据传输过程是相互隔离的。其中,服务器为多个

云应用提供的运行环境可以为虚拟化环境,也可以为非虚拟化环境;虚拟化环境可以包括虚拟机,容器,以及其余操作系统虚拟化的类型,本申请对此不做限定,此时服务器中的多个云应用客户端均运行在虚拟化环境的模拟操作系统中;非虚拟化环境可以是指服务器中的真实操作系统,此时服务器中的多个云应用客户端均运行在服务器所提供的真实操作系统中。

[0093] 为便于理解,本申请实施例以多个云应用客户端中的任意一个云应用客户端(本申请实施例可以称其为第一云应用客户端)为例进行详细说明,也就表示第一云应用客户端是运行在服务器中的,该第一云应用客户端可以认为是服务器中创建的第一云应用实例。本申请实施例将与第一云应用客户端进行数据交互的终端设备称为第一终端,该第一终端可以用于接收服务器端的第一云应用客户端所生成的音视频数据并进行输出;例如,第一终端可以用于输出第一云应用客户端对应的云应用展示页面,该云应用展示页面可以包括第一终端所接收到的第一云应用客户端的音视频数据;其中,该第一终端可以为前述图1所示的云应用的处理系统中的任意一个终端设备,本申请对此不做限定;音视频数据可以包括图像数据和音频数据中的至少一种数据类型,如音视频数据可以是指第一云应用客户端对应的音频数据,或者该音视频数据可以是指第一云应用客户端中渲染完成的图像数据,或者该音视频数据可以包括第一云应用客户端中产生的图像数据和音频数据,本申请对此不做限定;此处的图像数据可以是指第一云应用客户端中渲染完成的结果(例如,云游戏客户端中渲染完成的游戏画面)。

[0094] 具体的,第一终端安装有与第一云应用相关联的用户客户端,此处的用户客户端可以认为是安装在终端设备上的,且能够为用户提供第一云应用体验服务的客户端。当使用该第一终端的用户想要体验第一云应用时,该用户可以对第一终端中与第一云应用相关联的用户客户端执行触发操作,以启动该用户客户端;例如,用户通过点击第一终端中安装的用户客户端,进入该用户客户端的登录页面,进而可以在该登录页面中输入账号信息以登录第一云应用。

[0095] 当第一终端检测到用户针对该用户客户端的触发操作,可以生成针对第一云应用的第一启动指令,进而可以将该第一启动指令发送至服务器;其中,该第一启动指令可以包括应用标识信息和对象信息。该应用标识信息用于指示服务器需要创建或分配的云应用实例,如第一启动指令中的应用标识信息用于指示服务器创建或分配第一云应用实例(第一云应用客户端);应用标识信息可以为云应用ID(identity,身份标识)、或者其余可以唯一标识第一云应用的信息,本申请对此不做限定。对象信息可以用于指示服务器基于第一启动指令创建或分配的云应用实例所对应的数据交互对象,如第一启动指令中的对象信息用于指示服务器创建或分配的云应用实例所对应的数据交互对象为第一终端设备,更为具体地,数据交互对象为第一终端设备中的用户客户端;该对象信息可以包括第一终端的设备标识信息、使用该第一终端的用户标识(例如,账号信息等),本申请对此不做限定。

[0096] 服务器接收到第一终端发送的第一启动指令后,可以基于该第一启动指令中的应用标识信息,在该服务器中为第一终端创建或分配第一云应用客户端,即在服务器中部署第一云应用客户端;基于该第一启动指令中的对象信息,可以启动该第一云应用客户端,其本质上可以认为是启动服务器中为第一云应用客户端建立的进程。可以理解的是,在服务器端启动第一云应用客户端后,第一终端中的用户客户端也会成功启动第一云应用,且该

用户客户端中的第一云应用的进度状态与服务器中的第一云应用客户端的进度状态始终保持一致；其中，第一终端中的用户客户端与服务器中运行的第一云应用客户端之间需要进行实时数据传输，如第一云应用客户端中渲染完成的每一个图像数据都可以从第一云应用客户端传送至第一终端中的用户客户端，而用户在第一终端的用户客户端中每输入一个对象操作数据都可以传送至服务器中的第一云应用客户端，且第一云应用客户端与第一终端中的用户客户端可以直接进行数据交互，可以减少两者数据交互过程中的数据传输路径，减少数据交互过程中的响应时间，这样可以确保第一云应用客户端与用户客户端之间的进度状态始终保持一致，进而可以提升用户客户端所输出的音视频数据的流畅度，以及用户客户端中的操作灵敏度，有利于提高云应用的用户留存率。

[0097] 步骤S102,通过第一云应用客户端中的数据采集组件,在第一云应用客户端和第一终端之间传输第一云应用中的应用数据流;第一终端用于输出第一云应用客户端对应的云应用展示页面。

[0098] 具体的,由于服务器中可以运行多个云应用客户端,也就是该服务器中除了运行第一云应用客户端之外,还可以运行一个或多个云应用客户端,那么第一云应用客户端在与第一终端(具体可以为第一终端中与第一云应用相关联的用户客户端)进行数据交互时,需要保证其数据交互过程的隔离性。因此,本申请实施例在各个云应用客户端中集成了数据采集组件,通过该数据采集组件来实现服务器中的云应用客户端与其对应的终端设备直接进行数据交互,这样的数据交互方式可以规避云应用客户端对显示设备、音频设备以及输入设备的独占需求;换言之,多个云应用客户端不管是并发运行在服务器所提供的真实操作系统中,还是并发运行在服务器所提供的同一个虚拟化环境(例如,虚拟机)中,仍然可以确保每个云应用客户端与其对应的终端设备之间的数据交互过程是独立的,这样可以在各个云应用客户端正常运行的情形下,提高服务器的利用率。

[0099] 其中,在第一云应用客户端与第一终端之间的数据交互过程中,可以通过该第一云应用客户端中集成的数据采集组件,在第一云应用客户端与第一终端之间传输第一云应用中的应用数据流,如第一云应用客户端内部集成的数据采集组件可以抓取该第一云应用客户端中产生的音视频数据,并将抓取到的音视频数据传输至第一终端中的用户客户端,而用户客户端在获取到用户在该用户客户端中输入的对象操作数据后,可以将对象操作数据传送至第一云应用客户端内部集成的数据采集组件,随后第一云应用客户端中的其他组件(第一云应用客户端中除数据采集组件之外的组件)可以从数据采集组件中调用对象操作数据,此处的音视频数据和对象操作数据均属于第一云应用中的应用数据流。

[0100] 其中,第一云应用中的应用数据流可以包括音视频流和操作数据流(也可以称为输入数据流);操作数据流可以由第一终端基于云应用展示页面中的触发操作所生成的对象操作数据(也可以称为输入事件数据)组成;音视频流可以包括视频流和/或音频流,视频流可以由第一云应用客户端渲染生成的一系列图像数据组成,音频流可以由第一云应用客户端所生成的持续的音频数据组成,此处的音频数据和前述图像数据可以称为视频数据;换言之,音视频流可以由音视频数据组成。

[0101] 数据采集组件可以包括渲染抓取组件和操作处理组件,或者可以为集成了渲染抓取组件的功能和操作处理组件的功能的综合组件,本申请对此不做限定;可选地,渲染抓取组件可以包括图像抓取组件和音频抓取组件,或者可以为集成了图像抓取组件的功能和音

频抓取组件的功能的综合组件,本申请对此不做限定。当渲染抓取组件包括图像抓取组件和音频抓取组件时,图像抓取组件可以用于抓取第一云应用客户端中渲染完成的图像数据,音频抓取组件可以用于抓取第一云应用客户端生成的音频数据,抓取到的音频数据和图像数据可以一同传送至第一终端。当渲染抓取组件为集成了图像抓取组件的功能和音频抓取组件的功能的综合组件时,该渲染抓取组件可以用于抓取第一云应用客户端所产生的音视频数据(如音频数据和视频数据),并将抓取到的音视频数据传送至第一终端。操作处理组件可以用于接收第一终端的用户客户端抓取到的对象操作数据,并将接收到的对象操作数据注入至第一云应用客户端。

[0102] 可选地,云应用数据流可以包括第一云应用客户端所生成的音视频数据。服务器可以通过第一云应用客户端所属的进程,调用服务器端所运行操作系统中的GPU驱动程序所提供的渲染接口,基于该渲染接口对第一云应用客户端中的页面展示数据(可以认为是需要展示给用户的数据,例如云游戏中虚拟对象、游戏功能控件、游戏场景数据等)进行渲染处理,得到该第一云应用客户端对应的图像数据,该图像数据可以存储至第一云应用客户端对应的渲染缓冲区。其中,服务器中运行的每一个云应用客户端都可以对应一个或多个渲染缓冲区,该渲染缓冲区可以用于暂时存放其对应的云应用客户端渲染完成的图像数据。

[0103] 可选地,第一云应用客户端在服务器端的运行过程中,其自身还可以生成与渲染完成的图像数据相对应的音频数据,该音频数据可以存放在第一云应用客户端对应的音频缓冲区。其中,服务器可以为每一个会生成音频数据的云应用客户端分配独立的音频缓冲区,该音频缓冲区可以用于暂时存放其对应的云应用客户端中所生成的音频数据,该音频缓冲区可以认为是服务器端所运行的操作系统中的系统内存。若第一云应用客户端中生成了图像数据和音频数据,该图像数据和音频数据可以作为第一云应用客户端对应的音视频数据;若第一云应用客户端自身没有产生与图像数据相对应的音频数据,那么就可以将渲染完成的图像数据作为第一云应用客户端的音视频数据。

[0104] 在一个或多个实施例中,第一云应用客户端对应的渲染处理实质上可以理解为:服务器中的第一云应用客户端所在的进程可以通过调用GPU驱动程序提供的渲染接口进行渲染操作,渲染完成的结果可以为第一云应用客户端中一帧一帧的渲染画面,这些渲染画面可以作为第一云应用客户端对应的图像数据,渲染完成的图像数据可以存放在该第一云应用客户端对应的渲染缓冲区。可以理解的是,第一云应用客户端所在的进程每渲染完成一个渲染画面,就可以将其存放至渲染缓冲区,而数据采集组件(例如,渲染抓取组件)就可以从渲染缓冲区中抓取该渲染画面,并将该渲染画面作为第一云应用客户端的音视频数据传送至第一终端进行显示。

[0105] 可选地,若第一云应用客户端在运行过程中生成了图像数据相对应的音频数据,那么该音频数据可以存放在第一云应用客户端对应的音频缓冲区,第一云应用客户端内部集成的数据采集组件可以从音频缓冲区中抓取该音频数据,并将获取到的图像数据和音频数据作为该第一云应用客户端对应的音视频数据一同传输至第一终端。换言之,第一云应用客户端与第一终端之间可以通过数据采集组件来传输第一云应用客户端的音视频数据。

[0106] 可选地,云应用数据流可以包括第一终端所生成的对象操作数据。在第一终端中可以展示第一云应用客户端的展示页面(也可以称为云应用展示页面),该云应用展示页面

可以用于输出第一终端所接收到的音视频数据,用户可以对云应用展示页面所输出的音视频数据执行触发操作,如对云应用展示页面中所展示音视频数据中的功能控件执行触发操作(例如,点击、滑动、长按等接触式操作,或者语音、手势等非接触式操作,本申请对此不做限定),此时的第一终端可以响应针对该音视频数据的触发操作,生成针对第一云应用的对象操作数据,第一终端可以将对象操作数据直接传送至第一云应用客户端中的数据采集组件(例如,操作处理组件),由该数据采集组件将对象操作数据注入至第一云应用客户端,以执行与该对象操作数据相对应的业务逻辑。换言之,第一终端与第一云应用客户端之间可以通过数据采集组件来传输对象操作数据。

[0107] 可选地,假设服务器中运行的多个云应用客户端还包括第二云应用客户端,该第二云应用客户端可以认为是服务器中创建的第二云应用实例,本申请实施例将与该第二云应用客户端进行数据交互的终端设备称为第二终端,该第一终端可以用于接收服务器端的第二云应用客户端所生成的音视频数据,并在第二终端中输出所接收到的音视频数据,如第二终端可以用于输出第二云应用客户端对应的云应用展示页面;该第二终端也可以为前述图1所示的云应用的处理系统中的任意一个终端设备,本申请对此不做限定。

[0108] 其中,第二终端安装有与第二云应用相关联的用户客户端,此处的用户客户端可以认为是安装在终端设备上的,且能够为用户提供第二云应用体验服务的客户端。当使用该第二终端的用户想要体验第二云应用时,该用户可以对第二终端中与第二云应用相关联的用户客户端执行触发操作,此时的第二终端在检测到用户针对该用户客户端的触发操作后,可以生成针对第二云应用的第二启动指令,进而可以将该第二启动指令(可以包括第二云应用对应的应用标识信息,以及第二终端的设备标识信息、使用该第二终端的用户的账号信息等)发送至服务器。

[0109] 服务器接收到第二终端发送的第二启动指令后,可以基于该第二启动指令,在该服务器中为第二终端创建或分配第二云应用客户端,并启动服务器为第二终端所创建或分配的第二云应用客户端,如在创建或分配了第二云应用客户端后,可以为该第二云应用客户端建立对应的进程,并启动该第二云应用客户端所在的进程。可以理解的是,在服务器端启动第二云应用客户端后,第二终端中的用户客户端也会成功启动第二云应用,该用户客户端中的第二云应用的进度状态与服务器中的第二云应用客户端的进度状态始终保持一致。

[0110] 对于服务器中运行的第二云应用客户端内部同样可以集成数据采集组件,通过该第二云应用客户端内部集成的数据采集组件,可以在第二云应用客户端和第二终端之间传输第二云应用中的应用数据流;第二云应用客户端与第二终端之间的数据交互过程,与前述第一云应用客户端和第一终端之间的数据交互过程类似,此处不再赘述。应当理解,服务器中除了创建第一云应用客户端和第二云应用客户端之外,还可以继续创建其余云应用客户端,本申请对同一个服务器中所运行的云应用客户端的数量不做限定。前述涉及的第一云应用和第二云应用可以为同一个云应用,本申请对此不做限定。

[0111] 请参见图4,图4是本申请实施例提供的一种云应用的数据交互示意图。本申请实施例以服务器端并发运行2个云应用客户端(第一云应用客户端和第二云应用客户端)为例进行描述,如图4所示的第一云应用客户端与第一终端互为数据交互的对象,第二云应用客户端与第二终端互为数据交互的对象。

[0112] 服务器中运行的第一云应用客户端和第二云应用客户端,都可以通过各自所在的进程,分别调用GPU驱动程序所提供的渲染接口执行渲染操作,该GPU驱动程序为服务器端所运行的操作系统中的一种驱动程序。其中,第一云应用客户端在服务器端运行时,可以由该第一云应用客户端所在的进程直接调用GPU驱动程序所提供的渲染接口进行渲染操作,渲染完成的图像数据可以存放在服务器为第一云应用客户端所分配的渲染缓冲区中;第二云应用客户端在服务器端运行时,同样可以由第二云应用客户端所在的进程直接调用GPU驱动程序所提供的渲染接口进行渲染操作,渲染完成的图像数据可以存放在服务器为第二云应用客户端所分配的渲染缓冲区中,即服务器中运行的每一个云应用客户端的渲染操作都是在各自所在的进程中所完成的,不同云应用客户端之间的渲染操作互相独立,互不影响。需要说明的是,当不同云应用客户端各自所在的进程所调用的渲染接口为同一个GPU驱动程序所提供的渲染接口时,不同云应用客户端所在的进程调用渲染接口的调度可以由服务器端所运行的操作系统来完成,从而可以达到多个云应用客户端分别使用GPU资源的目的。换言之,基于多个云应用客户端各自所在进程分别调用渲染接口进行渲染操作,就可以实现服务器中的多个云应用客户端共享GPU硬件设备的需求,而无需创建其余额外的设备(例如,模拟的显示设备、模拟的音频设备以及模拟的输入设备等),可以最大程度地节约服务器资源。

[0113] 同理,服务器中运行的第一云应用客户端和第二云应用客户端,都可以通过各自所在的进程,生成与各自渲染完成的图像数据相对应的音频数据,生成的音频数据可以分别存放在各自云应用客户端对应的音频缓冲区,如第一云应用客户端所在的进程生成的音频数据存放在第一云应用客户端对应的音频缓冲区,第二云应用客户端所在的进程生成的音频数据存放在第二云应用客户端对应的音频缓冲区,这样可以规避各个云应用客户端对音频设备的独占需求,可以提高服务器的利用率。

[0114] 其中,由于服务器中运行的第一云应用客户端和第二云应用客户端中都各自集成了数据采集组件,因此可以通过各个云应用客户端中集成的数据采集组件,来保证多个云应用客户端之间数据流的隔离性。第一云应用客户端对应的渲染缓冲区中所存放的图像数据可以形成视频流,由第一云应用客户端内部集成的数据采集组件抓取后直接传送至第一终端,即由第一云应用客户端内部集成的数据采集组件抓取渲染缓冲区中的渲染结果并将其直接传送至第一终端。第一云应用客户端对应的音频缓冲区中所存放的音频数据可以形成音频流,由第一云应用客户端内部集成的数据采集组件抓取后直接传送至第一终端,即由第一云应用客户端内部集成的数据采集组件抓取音频缓冲区中的音频数据并直接传送至第一终端;第一云应用客户端内部集成的数据采集组件可以并行抓取图像数据和音频数据,并将图像数据和音频数据一同传送至第一终端。对于第一终端基于第一云应用客户端对应的展示页面中的触发操作所生成的输入事件数据流(如对象操作数据),在第一终端抓取后直接传送到第一云应用客户端进行后续处理,如输入事件数据流可以直接传送至第一云应用客户端内部集成的数据采集组件;也就是说,第一云应用客户端与第一终端之间的数据交互,规避了对显示设备、音频设备以及输入设备的独占需求,可以减少数据传输路径,提高性能和响应速度。

[0115] 其中,服务器中运行的第二云应用客户端与第二终端之间的数据交互过程,与前述第一云应用客户端和第一终端之间的数据交互过程相同,此处不再进行赘述。需要在服

务器中创建多少个云应用实例,就可以在该服务器中创建对应数量的云应用客户端,各个云应用客户端与其对应的终端设备之间的数据交互过程都可以独立进行,互不影响。

[0116] 本申请实施例中,通过云应用客户端内部集成的数据采集组件,在该云应用客户端与其对应的终端设备之间直接传输云应用数据流,可以规避云应用客户端对显示设备、音频设备以及输入设备的独占需求,这样就可以实现在一个服务器中并发运行多个云应用客户端,可以提高服务器的使用率,节省服务器资源,进而可以降低云应用的成本。

[0117] 请参见图5,图5是本申请实施例提供的另一种数据处理方法的流程示意图。可以理解地,该数据处理方法由服务器(例如,图1所示的云应用的处理系统中的服务器10b)执行。其中,该数据处理方法可以包括以下步骤S201至步骤S205:

[0118] 步骤S201,启动服务器中部署的第一云应用客户端。

[0119] 其中,步骤S201的具体实现过程可以参见图3所对应实施例中的步骤S101,此处不再进行赘述。

[0120] 步骤S202,通过第一云应用客户端中的渲染抓取组件,从第一云应用客户端对应的渲染缓冲区中获取图像数据;渲染缓冲区用于存储第一云应用客户端中渲染完成的数据。

[0121] 本申请实施例以第一云应用客户端内部集成的数据采集组件包括渲染抓取组件和操作处理组件为例,对第一云应用客户端与其对应的第一终端之间的数据交互过程进行描述。其中,对于服务器中运行的第一云应用客户端,其自身产生的音视频数据均可以由第一云应用客户端内部集成的渲染抓取组件来抓取得到,本申请实施例以音视频数据包括图像数据和音频数据为例,对音视频数据在第一云应用客户端与第一终端之间的传输过程进行描述。

[0122] 具体的,当第一云应用客户端所在的进程通过调用GPU驱动程序所提供的渲染接口进行渲染操作,并将渲染完成的结果暂时存放在该第一云应用客户端所对应的渲染缓冲区时,第一云应用客户端内部集成的渲染抓取组件可以用于从渲染缓冲区中抓取渲染完成的结果,即该渲染抓取组件可以用于抓取渲染缓冲区中存放的图像数据。举例来说,第一云应用客户端内部集成的渲染抓取组件可以对渲染缓冲区的数据存储状态进行实时测听,一旦测听到渲染缓冲区中存入了新的图像数据,该第一云应用客户端内部集成的渲染抓取组件就可以抓取该渲染缓冲区中新存入的图像数据;如渲染抓取组件可以通过某个函数(例如,该函数可以为现有的通用函数,也可以为自定义的定时器等方法,本申请对此不做限定)来测听渲染缓冲区的存储状态,当通过该函数测听到渲染缓冲区的存储状态发生变化时,表示渲染缓冲区中存放的数据发生了变更,此时的渲染抓取组件可以从渲染缓冲区中抓取变更后的图像数据。

[0123] 可以理解的是,第一云应用客户端对应的渲染缓冲区的数量可以为一个或多个,本申请对服务器中运行的各个云应用客户端所对应的渲染缓冲区的数量不做限定。当第一云应用客户端所在的进程可以按照预先设定的规则(可以根据实际需求进行预先设定,本申请对此不做限定),将渲染完成的图像数据存放在第一云应用客户端对应的一个或多个渲染缓冲区中,这一个或多个渲染缓冲区可以进行循环覆盖使用。

[0124] 举例来说,当第一云应用客户端对应的渲染缓冲区的数量为一个(该渲染缓冲区可以记为T1)时,第一云应用客户端所在的进程渲染完成的图像数据D1可以存放在渲染缓

缓冲区T1中,此时第一云应用客户端内部集成的渲染抓取组件可以抓取该渲染缓冲区T1中的图像数据D1,并将该图像数据D1传输至第一终端;当第一云应用客户端所在的进程继续执行渲染操作得到图像数据D2,该图像数据D2同样可以存放在渲染缓冲区T1中,此时的图像数据D2可以覆盖之前存放的图像数据D1,即渲染缓冲区T1中存放的始终是第一云应用客户端中渲染完成的最新图像数据。

[0125] 可选地,当第一云应用客户端对应的渲染缓冲区的数量为多个(如可以包括渲染缓冲区T1、渲染缓冲区T2、渲染缓冲区T3)时,第一云应用客户端所在的进程渲染完成的图像数据D1可以存放在渲染缓冲区T1中,此时第一云应用客户端内部集成的渲染抓取组件可以抓取该渲染缓冲区T1中的图像数据D1,并将该图像数据D1传输至第一终端;当第一云应用客户端所在的进程继续执行渲染操作得到图像数据D2,该图像数据D2可以存放在渲染缓冲区T2中,渲染抓取组件同样可以抓取该渲染缓冲区T2中的图像数据D2,并传输至第一终端;同理,后续渲染得到的图像数据D3可以存放在渲染缓冲区T3中,继续渲染得到的图像数据D4可以存放在渲染缓冲区T1中,此时的图像数据D4可以覆盖之前存放的图像数据D1,以此类推,第一云应用客户端中渲染完成的图像数据可以按照顺序依次存放至前述三个渲染缓冲区中;这样可以确保第一云应用客户端中每个渲染完成的图像数据都可以在渲染缓冲区中暂时存放一段时间(如三个渲染缓存区依次存入图像数据的间隔时间),可以防止渲染抓取组件出现丢失的情形。

[0126] 请参见图6,图6是本申请实施例提供的一种云应用客户端内部集成的渲染抓取组件抓取图像数据的示意图。如图6所示的服务器中并发运行的云应用客户端的数量为N(N可以为大于1的整数)个,如云应用客户端1,云应用客户端2,……,云应用客户端N。服务器中运行的每一个云应用客户端都可以被分配对应的渲染缓冲区,如云应用客户端1可以对应渲染缓冲区1,云应用客户端N可以对应渲染缓冲区N等;每个云应用客户端对应的渲染缓冲区中都可以存放每个云应用客户端各自所在的进程,分别调用GPU驱动程序提供的渲染接口进行渲染操作所得到的结果。每个云应用客户端分别集成有各自独立的渲染缓冲区抓取模块(可以认为是渲染抓取组件所包含的图像抓取组件),抓取到的图像数据分别传送至各自对应的终端设备中的用户客户端进行显示。例如,云应用客户端1中集成的渲染缓冲区抓取模块,在抓取到云应用客户端1中产生的图像数据D5后,可以将该图像数据D5传送至该第一云应用客户端1对应的终端设备进行显示;云应用客户端N中集成的渲染缓冲区抓取模块,在抓取到云应用客户端N中产生的图像数据D6后,可以将该图像数据D6传送至该第一云应用客户端N对应的终端设备进行显示等。

[0127] 可选地,第一云应用客户端内部集成的渲染抓取组件从渲染缓冲区抓取图像数据的过程可以包括:可以获取第一云应用客户端对应的渲染缓冲区的数据属性信息,根据该数据属性信息,可以在第一云应用客户端的渲染抓取组件中创建纹理资源对象(texture),进而可以将第一云应用客户端对应的渲染缓冲区中的图像数据封装到该纹理资源对象,通过将渲染缓冲区中的图像数据封装到纹理资源对象,以便于后续将该纹理资源对象设置为共享纹理资源,以供第一云应用客户端中的其余组件(例如,数据压缩组件)访问渲染缓冲区中的图像数据进行压缩处理,实现第一云应用客户端中的不同组件之间的数据共享。其中,该数据属性信息可以包括第一云应用客户端中的图像宽度、图像高度以及像素格式等信息;例如,第一云应用客户端内部集成的渲染抓取组件可以通过数据交换接口获取渲染

缓冲区的数据属性信息,基于该数据属性信息来创建纹理资源对象(texture),进而可以封装渲染缓冲区中的音视频数据到该纹理资源对象(texture),这样可以确保纹理资源对象与渲染缓冲区的图像数据一一对应,保持图像数据的真实性。

[0128] 步骤S203,通过第一云应用客户端中的渲染抓取组件,从第一云应用客户端对应的音频缓冲区中获取与图像数据相对应的音频数据;音频缓冲区用于存储第一云应用客户端中生成的音频。

[0129] 具体的,在服务器中启动第一云应用客户端时,该服务器可以为第一云应用客户端分配对应的音频缓冲区,第一云应用所在的进程可以生成与渲染完成的图像数据相对应的音频数据,并将其存放在第一云应用客户端对应的音频缓冲区。第一云应用客户端内部集成的渲染抓取组件(或者渲染抓取组件所包含的音频抓取组件)可以抓取音频缓冲区中的音频数据;其中,渲染抓取组件从音频缓冲区抓取音频数据的过程,与前述从渲染缓冲区中抓取图像数据的过程类似,如可以在渲染抓取组件中创建一个音频资源对象,并将音频缓冲区的音频数据封装到该音频资源对象等,此处对此不再进行赘述。

[0130] 步骤S204,将图像数据和音频数据传输至第一云应用客户端中的数据压缩组件,通过数据压缩组件对图像数据和音频数据进行压缩处理,得到第一云应用客户端对应的音视频数据。

[0131] 具体的,第一云应用客户端内部集成的渲染抓取组件在抓取到图像数据和音频数据后,可以将抓取到的图像数据和音频数据都传送至第一云应用客户端内部集成的数据压缩组件,通过该数据压缩组件对图像数据和音频数据分别进行压缩处理,可以得到压缩后的图像数据和压缩后的音频数据,压缩后的图像数据和压缩后的音频数据可以称为第一云应用客户端对应的音视频数据。

[0132] 其中,当第一云应用客户端内部集成的渲染抓取组件获取到图像数据(此处的图像数据可以认为是第一云应用客户端所产生的视频流中的视频帧)时,可以通过第一云应用客户端内部集成的数据压缩模块,对获取到的图像数据进行压缩处理,得到压缩后的图像数据,如可以将获取到的图像数据压缩为H264格式(一种视频压缩格式),或者可以将获取到的视频数据压缩为其他视频压缩格式,本申请对此不做限定;通过对渲染抓取组件获取到的图像数据进行压缩处理,可以减少传输的数据量,进而可以减轻带宽压力。

[0133] 当第一云应用客户端内部集成的渲染抓取组件获取到的音频数据时,也可以通过第一云应用客户端内部集成的数据压缩模块,对获取到的音频数据进行压缩处理,得到压缩后的音频数据,如可以将获取到的音频数据压缩为Opus格式(一种音频压缩格式),或者可以将获取到的音频数据压缩为其他音频压缩格式,本申请对此不做限定;通过对渲染抓取组件获取到的音频数据进行压缩处理,同样可以减少传输的数据量,进而可以减轻带宽压力。

[0134] 应当理解,服务器中运行的操作系统可以认为是该服务器端的多个云应用客户端的运行环境,该服务器中所运行的操作系统可以为Windows系统,也可以为Linux系统,或者还可以为其余类型的操作系统,本申请对此不做限定,也就是说,本申请实施例所涉及的多个云应用客户端在不同类型的操作系统中均可以正常运行,并不限于特定类型的操作系统。其中,服务器中运行的操作系统类型不同,那么第一云应用客户端内部集成的渲染抓取组件,将图像数据和音频数据传输至数据压缩组件所采用的函数可能会存在一些差异,即

不同类型的操作系统,渲染抓取组件可以采用不同的函数来将图像数据和音频数据传输至数据压缩组件。为便于理解,下面以服务器中运行的操作系统是Windows系统为例,对渲染抓取组件将图像数据传输至数据压缩组件的过程进行描述(音频数据从渲染抓取组件传输至数据压缩组件的过程,与图像数据从渲染抓取组件传输至数据压缩组件的过程类似,本申请实施例仅以图像数据为例进行说明)。

[0135] 在一个或多个实施例中,在第一云应用客户端内部,渲染抓取组件从渲染缓冲区抓取到图像数据之后,该图像数据可以封装在渲染抓取组件中所创建的纹理资源对象texture中,该渲染抓取组件中还可以创建栅栏对象(fence),该栅栏对象fence可以用于测听该纹理资源对象texture中的数据状态,或者可以将栅栏对象fence理解为是一种提醒机制,用于向数据压缩组件提醒纹理资源对象texture对应的音视频数据已完成渲染操作。

[0136] 其中,图像数据从渲染抓取组件传送至数据压缩组件的实现过程可以包括:可以将纹理资源对象texture设置为共享纹理资源texture,如可以通过调用CreateSharedHandle(Windows系统中的一种用于创建共享资源的句柄)方法,将纹理资源对象texture设置为共享纹理资源texture,即将纹理资源对象texture对应的图像数据设置为共享资源,以供数据压缩组件可以访问该共享纹理资源texture对应的图像数据,实现第一云应用客户端中的不同组件之间的资源共享,有利于减少图像数据在服务器端的数据处理时间;当然,通过共享纹理资源texture还可以实现不同程序之间的资源共享,如第一云应用客户端内部的渲染抓取组件与外部的独立程序之间的资源共享。同样地,也可以将栅栏对象fence设置为共享栅栏fence,如同样可以通过调用CreateSharedHandle方法,将栅栏对象fence设置为共享栅栏fence,以供数据压缩组件可以访问该共享栅栏fence,获取共享纹理资源texture对应的图像数据的渲染操作完成状态,这样可以使数据压缩组件可以及时获取渲染完成的图像数据,并及时将压缩后的图像数据传输至第一终端,以减少图像数据从第一云应用客户端传输至第一终端的时间,进一步确保第一终端中的进度状态与第一云应用客户端的进度状态保持一致。进而可以将共享纹理资源texture和共享栅栏fence导入第一云应用客户端中的数据压缩组件,如可以通过OpenSharedResource方法(Windows系统中的一种访问共享资源的方法)访问共享纹理资源texture,通过OpenSharedFence方法(Windows系统中的一种访问共享栅栏的方法)访问共享栅栏fence等。

[0137] 当共享栅栏fence测听到共享纹理资源texture中的数据状态为已完成状态时,可以向数据压缩组件发送提醒信号,在数据压缩组件中,基于该提醒信号获取共享纹理资源texture对应的图像数据,对该图像数据进行压缩处理,以得到压缩后的图像数据。换言之,共享纹理资源texture中每渲染完一个渲染画面,共享栅栏fence就会提醒数据压缩组件对渲染完的渲染画面进行压缩处理,得到压缩后的渲染画面。

[0138] 请参见图7,图7是本申请实施例提供的一种云应用客户端中的图像数据的压缩处理示意图。为便于理解,下面以PC云游(PC云游表示云游戏客户端运行在Windows系统中,即服务器中所运行的操作系统为Windows系统)为例,对图像数据在云应用客户端内部集成的渲染抓取组件和数据压缩组件中的处理过程进行描述;如图7所示,运行在服务器中的云游戏客户端(例如,第一云应用客户端)内部的渲染缓冲区抓取模块(渲染抓取组件)可以创建纹理资源对象texture,进而可以封装渲染缓冲区内容到纹理资源对象texture,即将渲染

缓冲区中的图像数据封装至纹理资源对象texture。

[0139] 进一步地,可以通过调用CreateSharedHandle方法,将纹理资源对象texture设置为共享纹理资源,将栅栏对象fence设置为共享栅栏,并在云游戏客户端内部的渲染缓冲区抓取模块中导出共享纹理资源texture和共享栅栏fence。对于云游戏客户端内部的音视频压缩模块(即数据压缩组件)而言,可以导入共享纹理资源texture和共享栅栏fence,如可以通过OpenSharedResource方法访问共享纹理资源texture,通过OpenSharedFence方法访问共享栅栏fence;该音视频压缩模块需要等待共享栅栏fence发送提醒信号,若接收到共享栅栏fence发送的提醒信号,则表示共享纹理资源texture对应的图像数据(如渲染画面)已经完成渲染操作,进而可以压缩编码共享纹理资源texture对应的图像,即通过音视频压缩组件对渲染完成的图像进行压缩编码处理,以得到压缩后的图像数据。

[0140] 步骤S205,将音视频数据传输至第一终端,以使第一终端对音视频数据进行解压处理得到解压后的音视频数据,输出解压后的音视频数据。

[0141] 具体的,通过数据压缩组件得到音视频数据后,可以将音视频数据传输至第一终端,具体可以传输至第一终端中的用户客户端;第一终端中的用户客户端接收到音视频数据后,可以对音视频数据进行解压处理,得到解压后的音视频数据,即压缩处理前的音视频数据,并在该用户客户端中输出音视频数据。通过在第一云应用客户端内部集成数据压缩组件,利用内部集成的数据压缩数据对第一云应用客户端中生成的图像数据和音频数据进行压缩编码处理,可以减少第一云应用客户端与外部程序之间的交互,进而减少音视频数据的传输路径以提高性能和响应速度。

[0142] 可选地,若第一云应用客户端内部没有集成数据压缩组件,则可以利用服务器运行的操作系统中的数据压缩应用程序,对第一云应用客户端内部集成的渲染抓取组件所获取到的图像数据和音频数据进行压缩处理,如可以将渲染抓取组件所获取到的图像数据和音频数据传输至数据压缩应用程序,通过数据压缩应用程序对图像数据和音频数据进行压缩处理,得到压缩后的图像数据和压缩后的音频数据,该压缩后的图像数据和压缩后的音频数据可以称为第一云应用客户端对应的音视频数据;该数据压缩应用程序与该第一云应用客户端在该服务器中是相互独立的程序。通过数据压缩应用程序获取到音视频数据,同样可以传输至第一终端,以使第一终端可以对接收到的音视频数据进行解压处理,以得到解压后的音视频数据,进而在第一终端中输出解压后的音视频数据;其中,第一云应用客户端所生成的图像数据和音频数据从渲染抓取组件传输至数据压缩应用程序的过程,与前述图像数据和音频数据从渲染抓取组件传输至数据压缩组件的过程类似,此处不再赘述。通过利用第一云应用客户端外部的数据压缩应用程序,对第一云应用客户端中的图像数据和音频数据进行压缩编码处理,即复用现有云应用方案中的数据压缩应用程序,可以减少云应用客户端的开发工作量。

[0143] 请参见图8,图8是本申请实施例提供的一种云应用客户端在服务器中的数据压缩结构示意图。下面以云游戏客户端为例对服务器中的数据压缩结构进行描述,如图8所示的服务器中运行了云游戏客户端1(如第一云应用客户端),与该云游戏客户端1进行数据交互的设备为终端设备1(如第一终端);云游戏客户端1在运行过程中,可以执行对应的游戏逻辑(如上述业务逻辑),且云游戏客户端1所在的进程可以通过调用渲染接口执行渲染操作,渲染完成的结果可以存放在渲染缓冲区中,该云游戏客户端1所在进程生成的与渲染完成

结果相对应的音频数据可以存放在音频缓冲区中；云游戏客户端1内部集成的渲染抓取模块(渲染抓取组件)可以从渲染缓冲区中抓取图像数据,从音频缓冲区中抓取音频数据。

[0144] 若云游戏客户端1内部集成了音视频压缩模块(数据压缩组件),则渲染缓冲区抓取模块获取到的音频数据、图像数据都可以传送至音视频压缩模块(数据压缩组件)进行压缩处理,并将压缩后音频数据、视频数据作为第一云应用客户端对应的音视频数据传输至终端设备1(如第一终端)。可选地,若云游戏客户端1内部没有集成音视频压缩模块,则渲染缓冲区抓取模块获取到的音频数据、图像数据可以传送至服务器所运行的操作系统中的音视频压缩应用程序(即数据压缩应用程序)进行压缩处理,并将压缩后音频数据、视频数据作为第一云应用客户端对应的音视频数据传输至终端设备1。

[0145] 在一个或多个实施例中,第一云应用客户端内部集成的数据采集组件可以包括操作处理组件(也可以称为输入事件处理模块),此时第一云应用客户端与第一终端之间传输的应用数据流可以包括第一终端所生成的对象操作数据,对象操作数据可以包括第一终端基于第一云应用客户端对应的云应用展示页面中的触发操作所生成的指令。第一终端对接收到的音视频数据进行输出时,这些音视频数据可以展示在第一终端的用户客户端中,如第一终端中的用户客户端可以输出第一云应用客户端对应的云应用展示页面,该云应用展示页面可以用于输出第一终端接收到的音视频数据。

[0146] 其中,云应用展示页面所输出的音视频数据可以包括不同类型的功能控件,当用户对云应用展示页面中的功能控件执行触发操作,第一终端可以生成该用户的操作数据(即对象操作数据),进而可以将该对象操作数据直接传输至第一云应用客户端中的操作处理组件;此时运行在服务器中的第一云应用客户端可以通过内部的操作处理组件,接收第一终端传送的对象操作数据,通过调用操作处理组件中的对象操作数据,执行第一云应用客户端中对应的业务逻辑。换言之,第一云应用客户端内部集成的操作处理组件中的对象操作数据,可供第一云应用客户端中的其余模块进行调用,以进行后续处理。

[0147] 请参见图9,图9是本申请实施例提供的一种云应用客户端内部集成的操作处理组件接收对象操作数据的示意图。下面仍然以云游戏客户端为例对云应用客户端内部集成的操作处理组件进行描述,如图9所示的服务器中运行了云游戏客户端1(如第一云应用客户端),与该云游戏客户端1进行数据交互的设备为终端设备1(如第一终端)。当终端设备1获取到云游戏客户端1对应的游戏展示页面(如云应用展示页面)中的输入事件数据(即对象操作数据)时,可以将获取到的输入事件数据传送到云游戏客户端1内部集成的输入事件处理模块(即操作处理组件),该输入事件处理模块中的输入事件数据可以供云游戏客户端1内部的其余模块进行调用。本申请实施例中,终端设备1所生成的输入事件数据可以直接传送到云游戏客户端1,可以减少输入事件数据的传输路径,可以提高云游戏客户端1的性能和响应速度。

[0148] 本申请实施例中,可以在一个服务器中并发运行多个云应用客户端,如多个云应用客户端中的第一云应用客户端可以通过其内部集成的渲染抓取组件,将第一云应用客户端所生成的音视频数据直接传输至对应的第一终端,而第一终端所产生的对象操作数据可以直接传送到第一云应用客户端内部集成的操作处理组件,即各个云应用客户端都可以与各自对应的终端设备直接进行数据交互,可以规避云应用客户端对显示设备、音频设备以及输入设备的独占需求,并减少云应用客户端与对应的终端设备之间的数据传输路径,进

而可以提升性能及响应速度;在同一个服务器中并发运行多个云应用客户端,可以节省服务器资源,提高服务器的利用率,进而可以降低云应用的成本;由于云应用客户端对应的数据流都不需要与模拟的显示设备、音频设备以及输入设备等产生交互操作,可以减少数据在服务器所运行操作系统中的显示设备、音频设备以及输入设备的中间转换,有利于终端设备中的云应用用户客户端中多点触控需求的实现。

[0149] 可以理解的是,在本申请的具体实施方式中,可能涉及到用户在各个云应用中的相关信息(例如,用户账号信息、用户所使用的终端设备信息等),当本申请以上实施例运用到具体产品或技术中时,需要获得用户的许可或同意,且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准。

[0150] 请参见图10,图10是本申请实施例提供的一种数据处理装置的结构示意图。如图10所示,该数据处理装置1可以包括第一启动模块11,第一传输模块12;

[0151] 第一启动模块11,用于启动服务器中部署的第一云应用客户端;

[0152] 第一传输模块12,用于通过第一云应用客户端中的数据采集组件,在第一云应用客户端和第一终端之间传输第一云应用中的应用数据流;第一终端用于输出第一云应用客户端对应的云应用展示页面。

[0153] 其中,第一启动模块11,第一传输模块12的具体功能实现方式可以参见图3所对应实施例中的步骤S101至步骤S102,此处不再进行赘述。

[0154] 在一个或多个实施例中,数据采集组件包括渲染抓取组件,应用数据流包括音视频数据;

[0155] 第一传输模块12包括:图像数据获取单元121,音频数据获取单元122,音视频数据传输单元123;

[0156] 图像数据获取单元121,用于通过第一云应用客户端中的渲染抓取组件,从第一云应用客户端对应的渲染缓冲区中获取图像数据;渲染缓冲区用于存储第一云应用客户端中渲染完成的数据;

[0157] 音频数据获取单元122,用于通过第一云应用客户端中的渲染抓取组件,从第一云应用客户端对应的音频缓冲区中获取与图像数据相对应的音频数据;音频缓冲区用于存储第一云应用客户端中生成的音频;

[0158] 音视频数据传输单元123,用于基于图像数据和音频数据确定第一云应用客户端对应的音视频数据,将音视频数据传输至第一终端,以使第一终端输出音视频数据。

[0159] 可选地,图像数据获取单元121可以具体用于:

[0160] 获取第一云应用客户端对应的渲染缓冲区的数据属性信息,根据数据属性信息,在第一云应用客户端的渲染抓取组件中创建纹理资源对象;

[0161] 将第一云应用客户端对应的渲染缓冲区中的图像数据封装到纹理资源对象。

[0162] 其中,图像数据获取单元121,音频数据获取单元122,音视频数据传输单元123的具体功能实现方式可以参见图5所对应实施例中的步骤S202至步骤S205,此处不再进行赘述。

[0163] 在一个或多个实施例中,该数据处理装置1还可以包括:渲染接口调用模块13,渲染处理模块14;

[0164] 渲染接口调用模块13,用于通过第一云应用客户端所属的进程,调用服务器中的

图形处理器驱动程序所提供的渲染接口；

[0165] 渲染处理模块14,用于基于渲染接口对第一云应用客户端中的页面展示数据进行渲染处理,得到第一云应用客户端对应的图像数据,将图像数据存储至渲染缓冲区。

[0166] 可选地,多个云应用客户端还可以包括第二云应用客户端;该数据处理装置1还可以包括:第二启动模块15,第二传输模块16;

[0167] 第二启动模块15,用于当接收到第二终端发送的第二启动指令时,基于第二启动指令,在服务器中创建并启动第二云应用客户端;

[0168] 第二传输模块16,用于通过第二云应用客户端中的数据收集组件,在第二云应用客户端和第二终端之间传输第二云应用中的应用数据流;第二终端用于输出第二云应用客户端对应的云应用展示页面。

[0169] 其中,渲染接口调用模块13,渲染处理模块14,第二启动模块15,第二传输模块16的具体功能实现方式可以参见图3所对应实施例中的步骤S102,此处不再进行赘述。

[0170] 在一个或多个实施例中,音视频数据传输单元123可以包括:第一压缩子单元1231,压缩数据传输子单元1232;

[0171] 第一压缩子单元1231,用于将图像数据和音频数据传输至第一云应用客户端中的数据压缩组件,通过数据压缩组件对图像数据和音频数据进行压缩处理,得到第一云应用客户端对应的音视频数据;

[0172] 压缩数据传输子单元1232,用于将音视频数据传输至第一终端,以使第一终端对音视频数据进行解压处理得到解压后的音视频数据,输出解压后的音视频数据。

[0173] 或者,音视频数据传输单元123可以包括:第二压缩子单元1233,压缩数据传输子单元1232;

[0174] 第二压缩子单元1233,用于将图像数据和音频数据传输至数据压缩应用程序,通过数据压缩应用程序对图像数据和音频数据进行压缩处理,得到第一云应用客户端对应的音视频数据;数据压缩应用程序与第一云应用客户端在服务器中是互相独立的;

[0175] 压缩数据传输子单元1232,用于将音视频数据传输至第一终端,以使第一终端对音视频数据进行解压处理得到解压后的音视频数据,输出解压后的音视频数据。

[0176] 其中,图像数据位于渲染抓取组件中所创建的纹理资源对象中;渲染抓取组件中还包括栅栏对象,栅栏对象用于测听纹理资源对象中的数据状态;

[0177] 第一压缩子单元1231具体用于:

[0178] 将渲染抓取组件中的纹理资源对象设置为共享纹理资源,将渲染抓取组件中的栅栏对象设置为共享栅栏,将共享纹理资源和共享栅栏导入第一云应用客户端中的数据压缩组件;

[0179] 当共享栅栏测听到共享纹理资源中的数据状态为已完成状态时,向数据压缩组件发送提醒信号;

[0180] 在数据压缩组件中,基于提醒信号获取共享纹理资源对应的图像数据,以及获取渲染抓取组件中的音频数据,对图像数据和音频数据进行压缩处理,得到第一云应用客户端对应的音视频数据。

[0181] 其中,第一压缩子单元1231,压缩数据传输子单元1232,第二压缩子单元1233的具体功能实现方式可以参见图5所对应实施例中的步骤S204至步骤S205,此处不再进行赘述。

[0182] 在一个或多个实施例中,数据采集组件包括操作处理组件,应用数据流包括对象操作数据,对象操作数据包括第一终端基于第一云应用客户端对应的云应用展示页面中的触发操作所生成的指令;

[0183] 第一传输模块12可以包括:操作数据接收单元124,操作数据调用单元125;

[0184] 操作数据接收单元124,用于通过第一云应用客户端中的操作处理组件,接收第一终端传输的对象操作数据;

[0185] 操作数据调用单元125,用于通过调用操作处理组件中的对象操作数据,执行第一云应用客户端中的业务逻辑。

[0186] 其中,操作数据接收单元124,操作数据调用单元125的具体功能实现方式可以参见图5所对应实施例的步骤S205中的相关描述,此处不再进行赘述。

[0187] 在一个或多个实施例中,第一启动模块11可以包括:指令接收单元111,云应用启动单元112;

[0188] 指令接收单元111,用于接收第一终端发送的第一启动指令,基于第一启动指令中的应用标识信息,在服务器中部署第一云应用客户端;

[0189] 云应用启动单元112,用于基于第一启动指令中的对象信息,在服务器中启动第一云应用客户端。

[0190] 其中,指令接收单元111,云应用启动单元112的具体功能实现方式可以参见图3所对应实施例中的步骤S102,此处不再进行赘述。

[0191] 本申请实施例中,可以在一个服务器中并发运行多个云应用客户端,如多个云应用客户端中的第一云应用客户端可以通过其内部集成的渲染抓取组件,将第一云应用客户端所生成的音视频数据直接传输至对应的第一终端,而第一终端所产生的对象操作数据可以直接传送至第一云应用客户端内部集成的操作处理组件,即各个云应用客户端都可以与各自对应的终端设备直接进行数据交互,可以规避云应用客户端对显示设备、音频设备以及输入设备的独占需求,并减少云应用客户端与对应的终端设备之间的数据传输路径,进而可以提升性能及响应速度;在同一个服务器中并发运行多个云应用客户端,可以节省服务器资源,提高服务器的利用率,进而可以降低云应用的成本;由于云应用客户端对应的数据流都不需要与模拟的显示设备、音频设备以及输入设备等产生交互操作,可以减少数据在服务器所运行操作系统中的显示设备、音频设备以及输入设备的中间转换,有利于终端设备中的云应用用户客户端中多点触控需求的实现。

[0192] 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可以为服务器,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,上述计算机设备1000还可以包括: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。可选的,网络接口1004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,也可以是非稳定的存储器(non-volatile memory),例如至少一个磁盘存储器。可选的,存储器1005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,作为一种计算机可读存储介质的存储器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管理应用程序。

[0193] 在如图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,以实现:

[0194] 启动所述服务器中部署的第一云应用客户端;

[0195] 通过所述第一云应用客户端中的数据采集组件,在所述第一云应用客户端和第一终端之间传输第一云应用中的应用数据流;所述第一终端用于输出所述第一云应用客户端对应的云应用展示页面。

[0196] 其中,服务器中包括多个云应用客户端,多个云应用客户端的数据传输过程是相互隔离的,多个云应用客户端包括第一云应用客户端。

[0197] 应当理解,本申请实施例中所描述的计算机设备1000可执行前文图3和图5任一个所对应实施例中对数据处理方法的描述,也可执行前文图10所对应实施例中对数据处理装置1的描述,在此不再赘述。另外,对采用相同方法的有益效果描述,也不再进行赘述。

[0198] 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可读存储介质,且计算机可读存储介质中存储有前文提及的数据处理装置1所执行的计算机程序,且计算机程序包括程序指令,当处理器执行程序指令时,能够执行前文图3和图5任一个所对应实施例中对数据处理方法的描述,因此,这里将不再进行赘述。另外,对采用相同方法的有益效果描述,也不再进行赘述。对于本申请所涉及的计算机可读存储介质实施例中未披露的技术细节,请参照本申请方法实施例的描述。作为示例,程序指令可被部署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。

[0199] 此外,需要说明的是:本申请实施例还提供了一种计算机程序产品或计算机程序,该计算机程序产品或者计算机程序可以包括计算机指令,该计算机指令可以存储在计算机可读存储介质中。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,处理器可以执行该计算机指令,使得该计算机设备执行前文图3和图5任一个所对应实施例中对数据处理方法的描述,因此,这里将不再进行赘述。另外,对采用相同方法的有益效果描述,也不再进行赘述。对于本申请所涉及的计算机程序产品或者计算机程序实施例中未披露的技术细节,请参照本申请方法实施例的描述。

[0200] 需要说明的是,对于前述的各个方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本申请并不受所描述的动作顺序的限制,因为依据本申请,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本申请所必须的。

[0201] 本申请实施例方法中的步骤可以根据实际需要进行顺序调整、合并和删减。

[0202] 本申请实施例装置中的模块可以根据实际需要进行合并、划分和删减。

[0203] 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,是可以通过计算机程序来指令相关的硬件来完成,计算机程序可存储于一计算机可读存储介质中,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,存储介质可为磁碟、光盘、只读存储器(Read-Only Memory,ROM)或随机存储器(Random Access Memory,RAM)等。

[0204] 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。

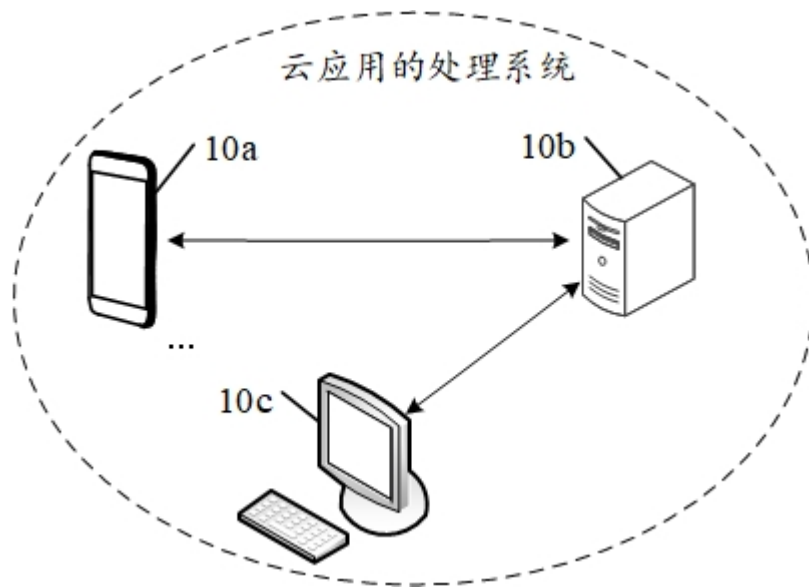


图1

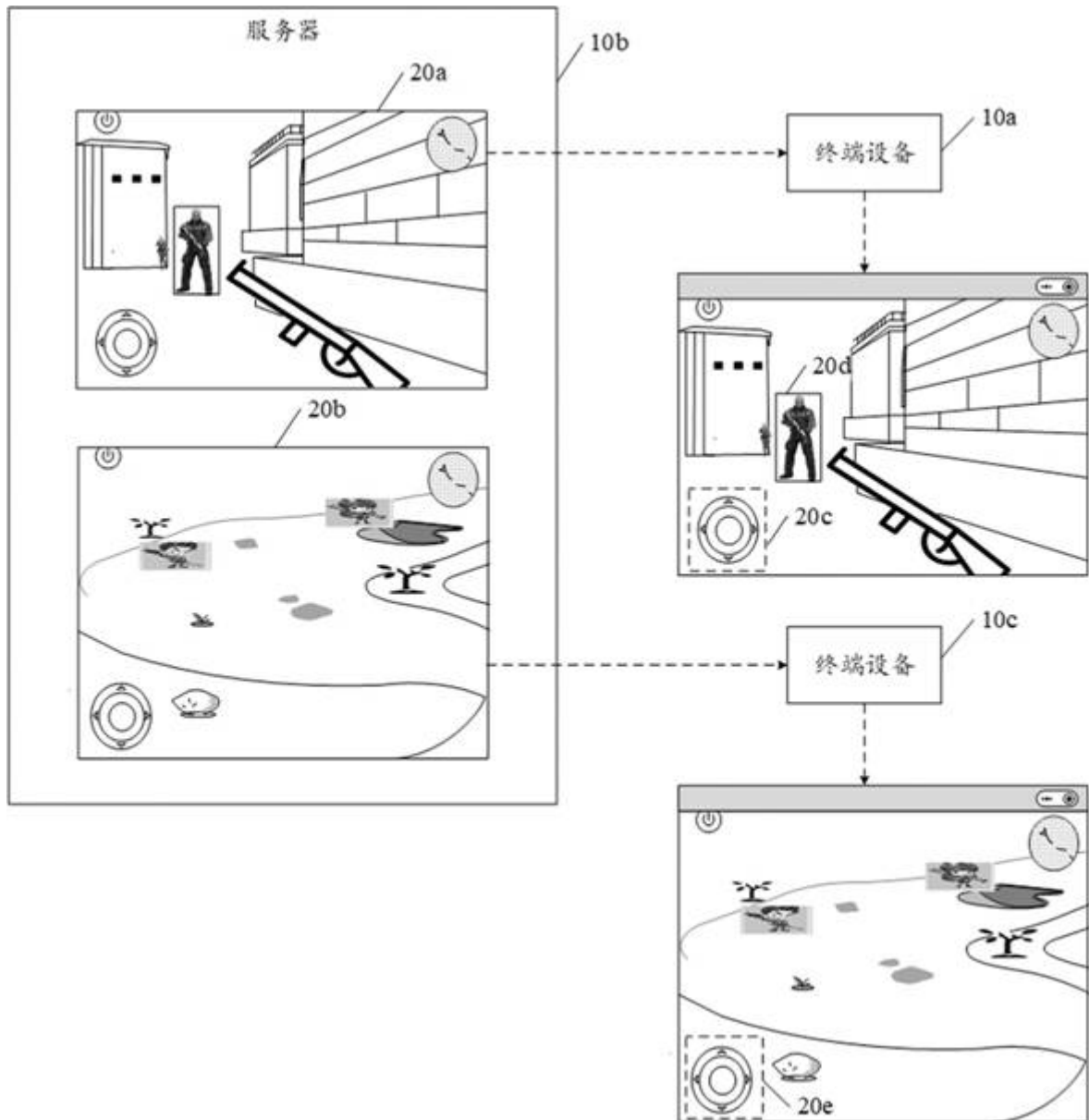


图2

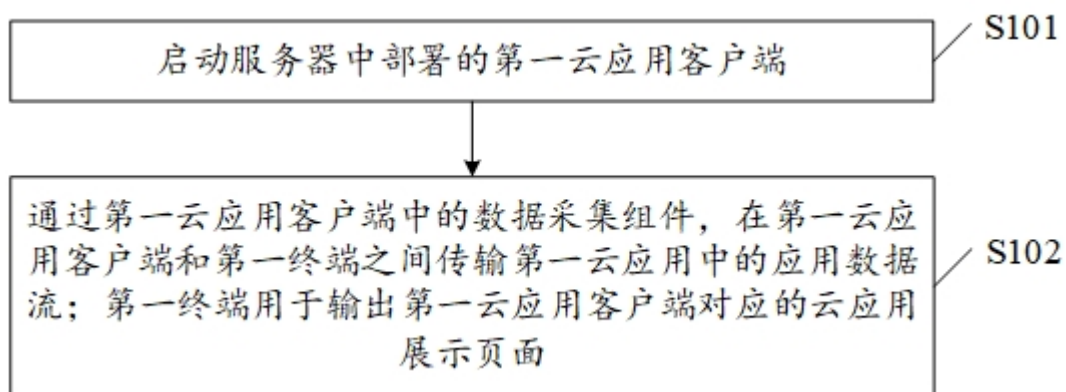


图3

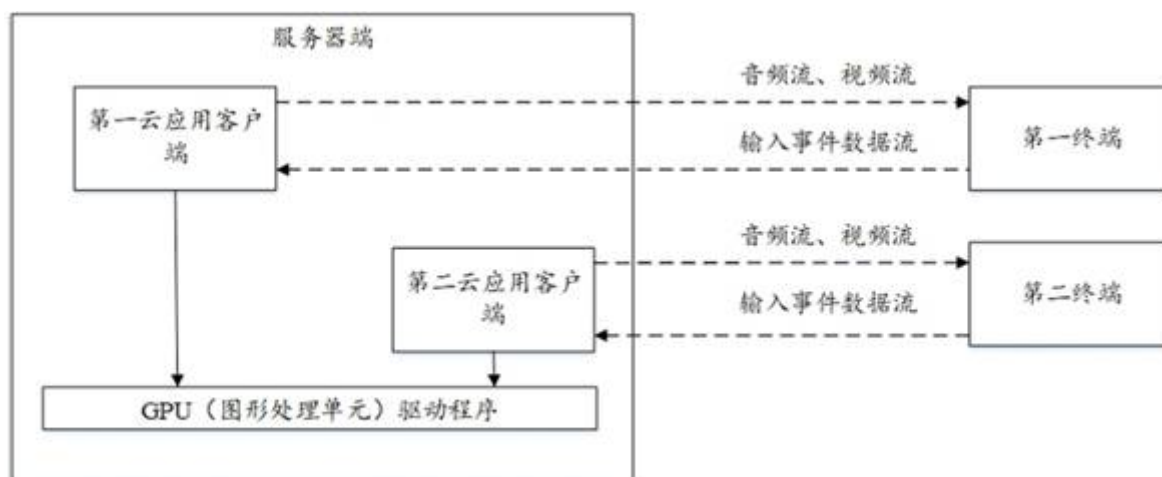


图4

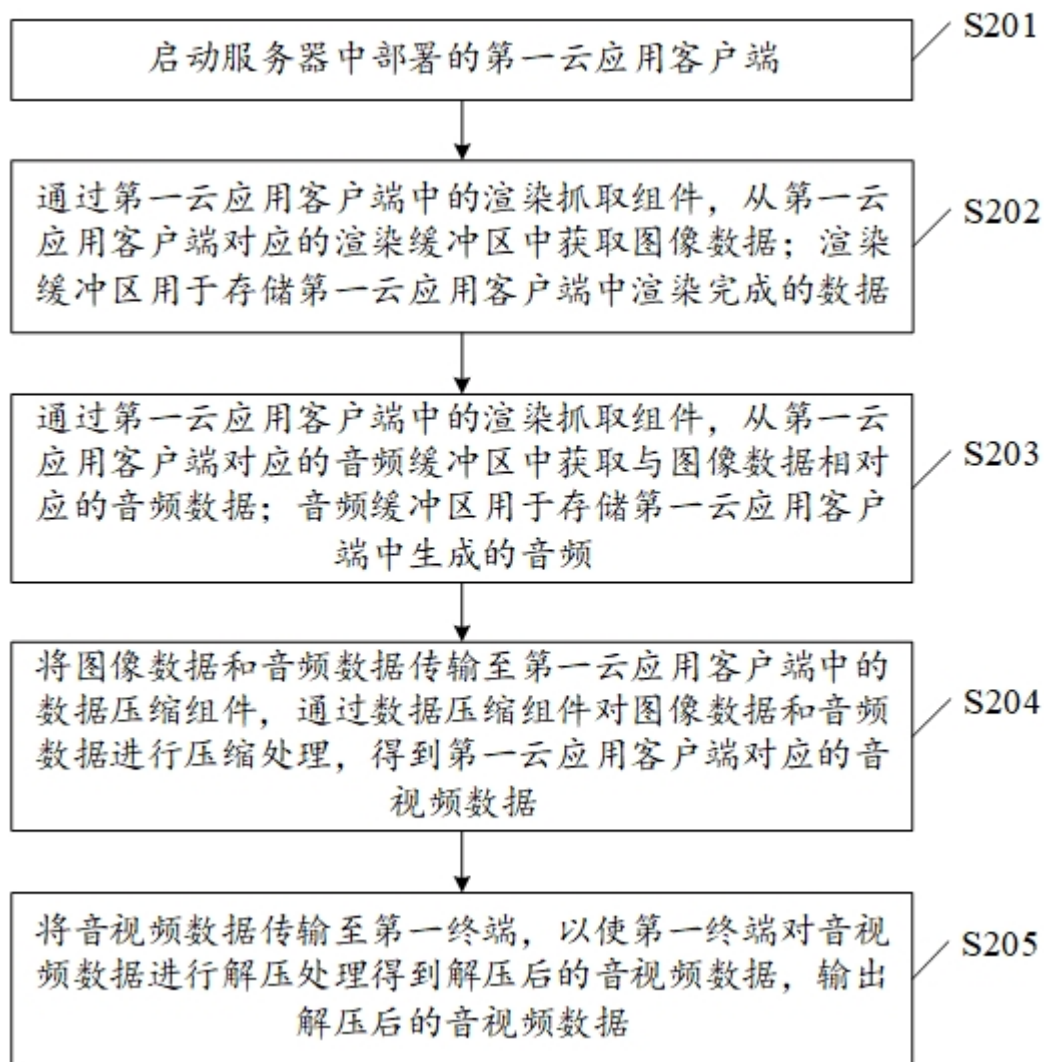


图5

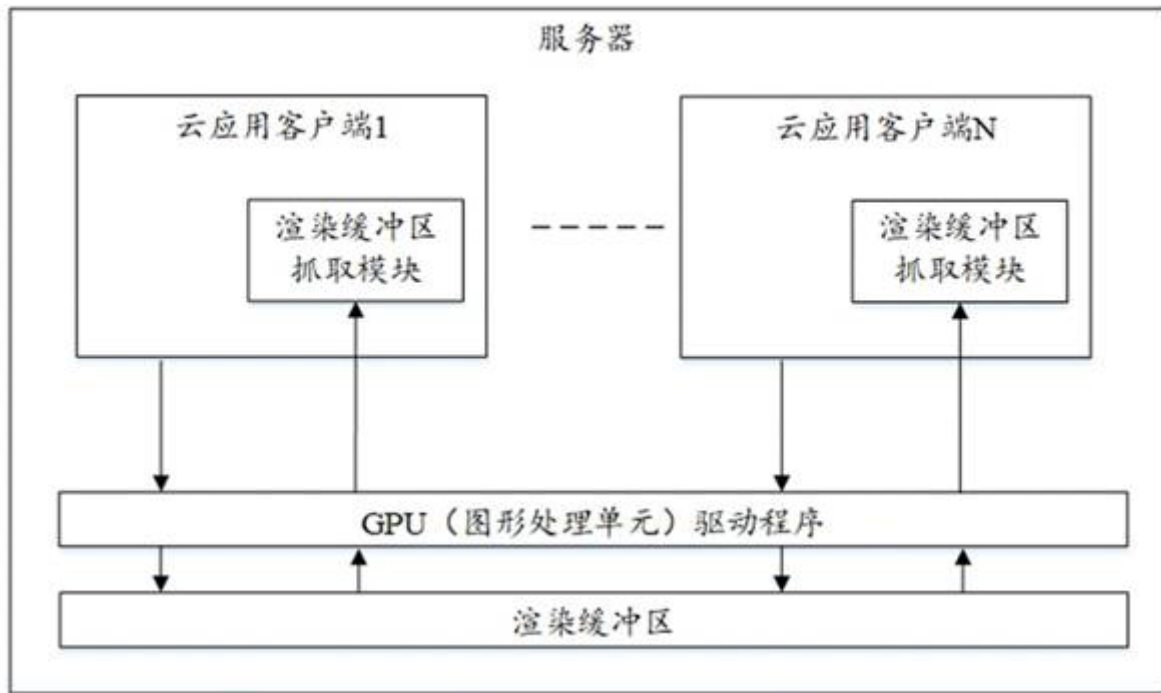


图6

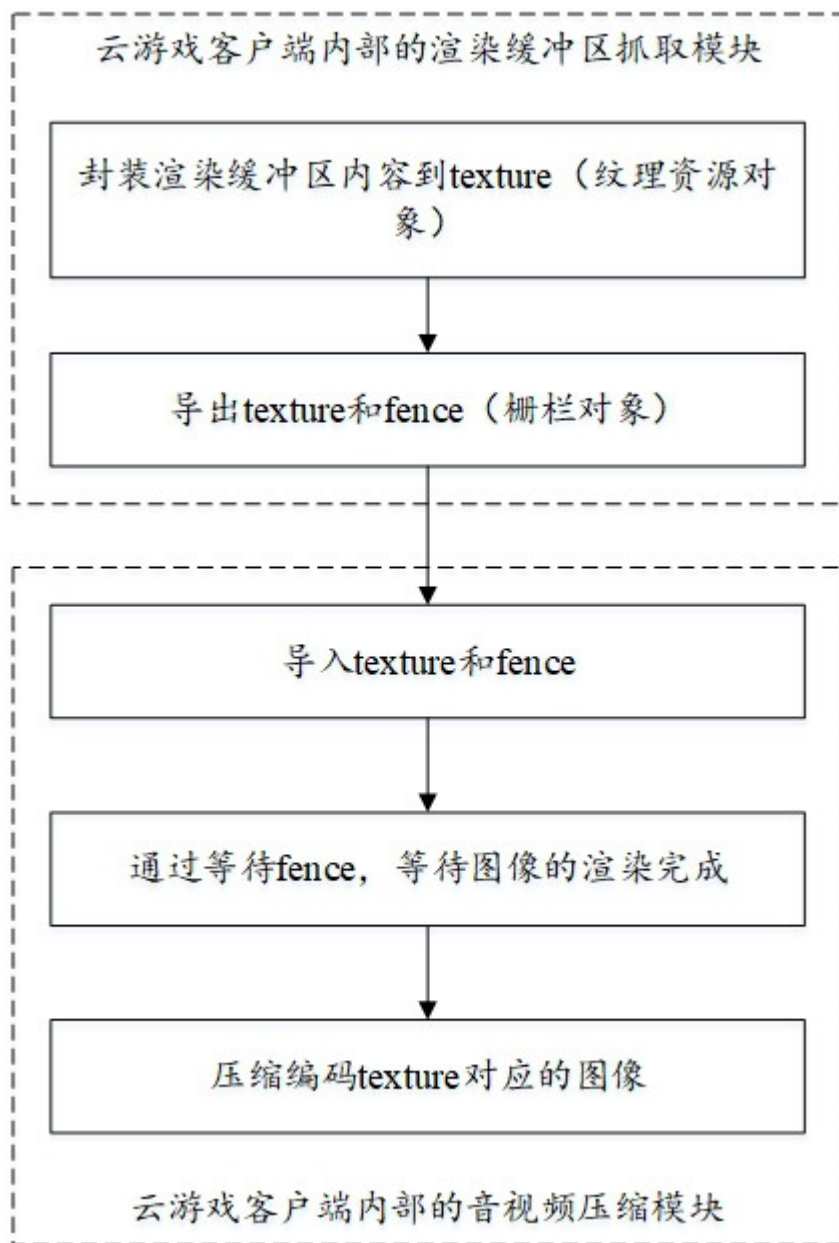


图7

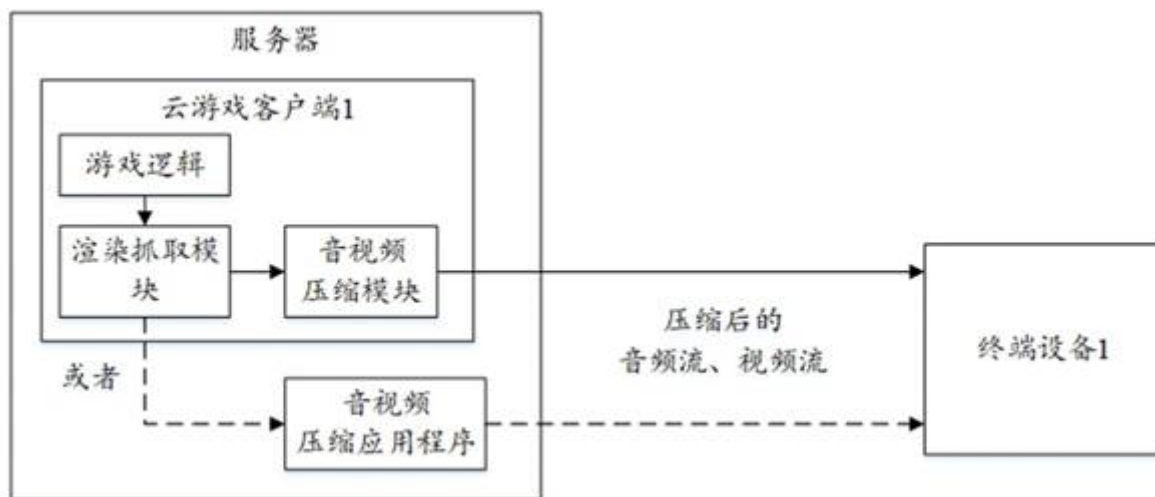


图8

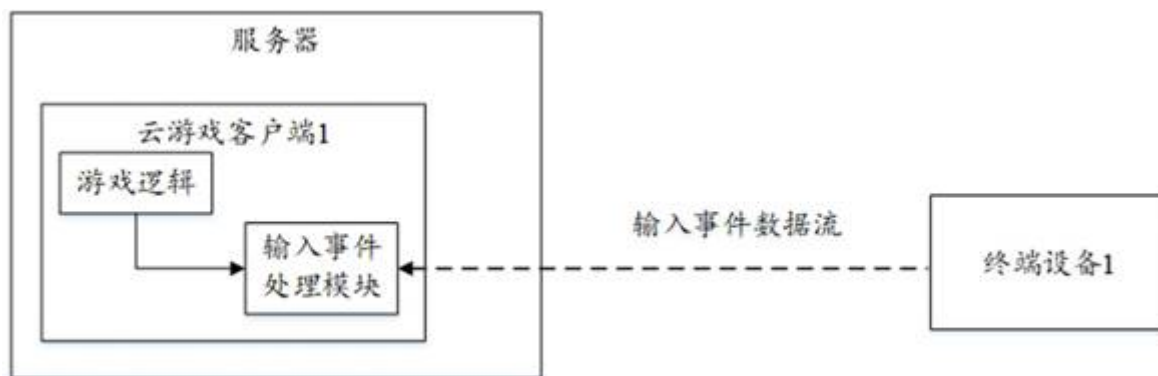


图9

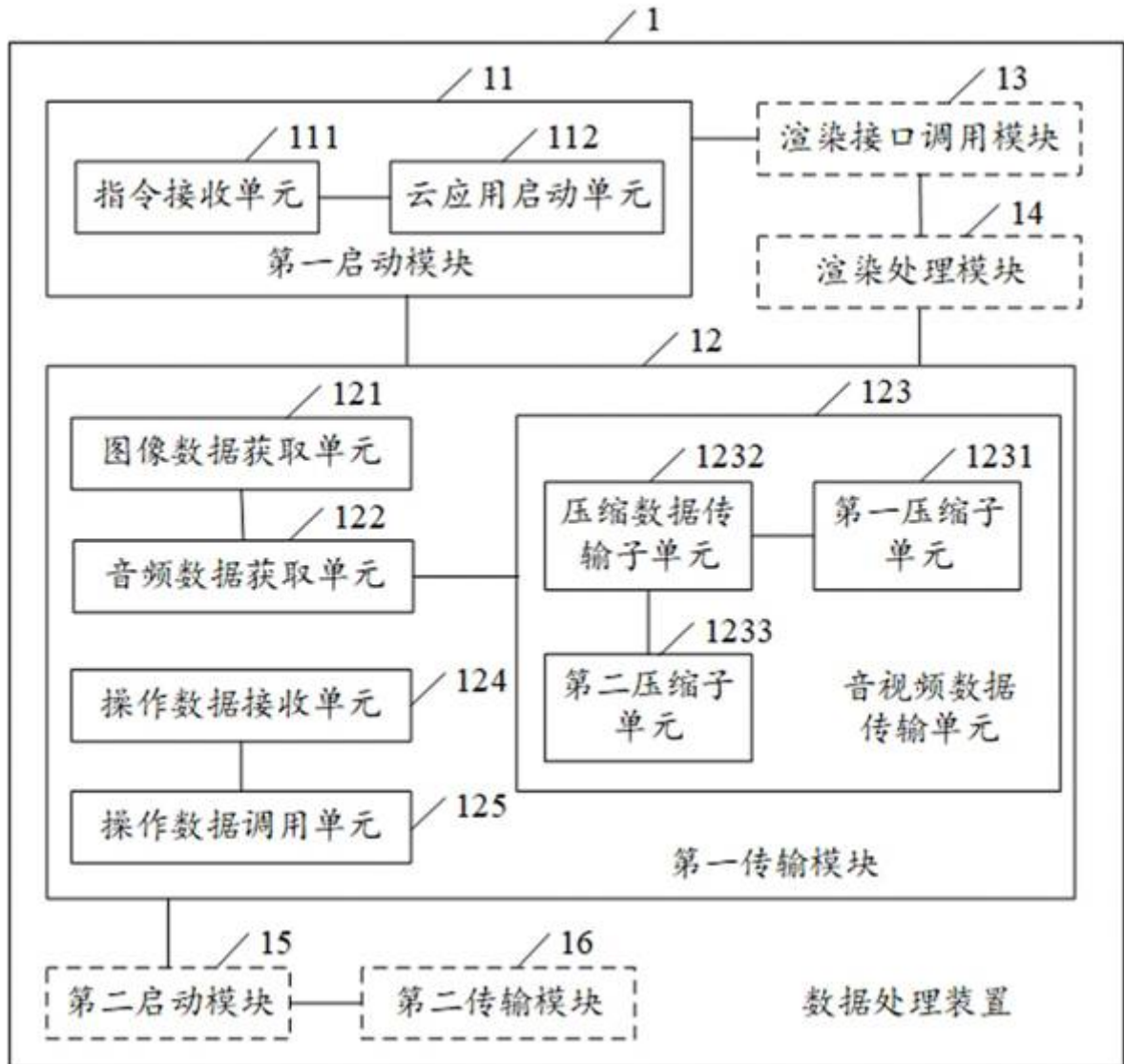


图10

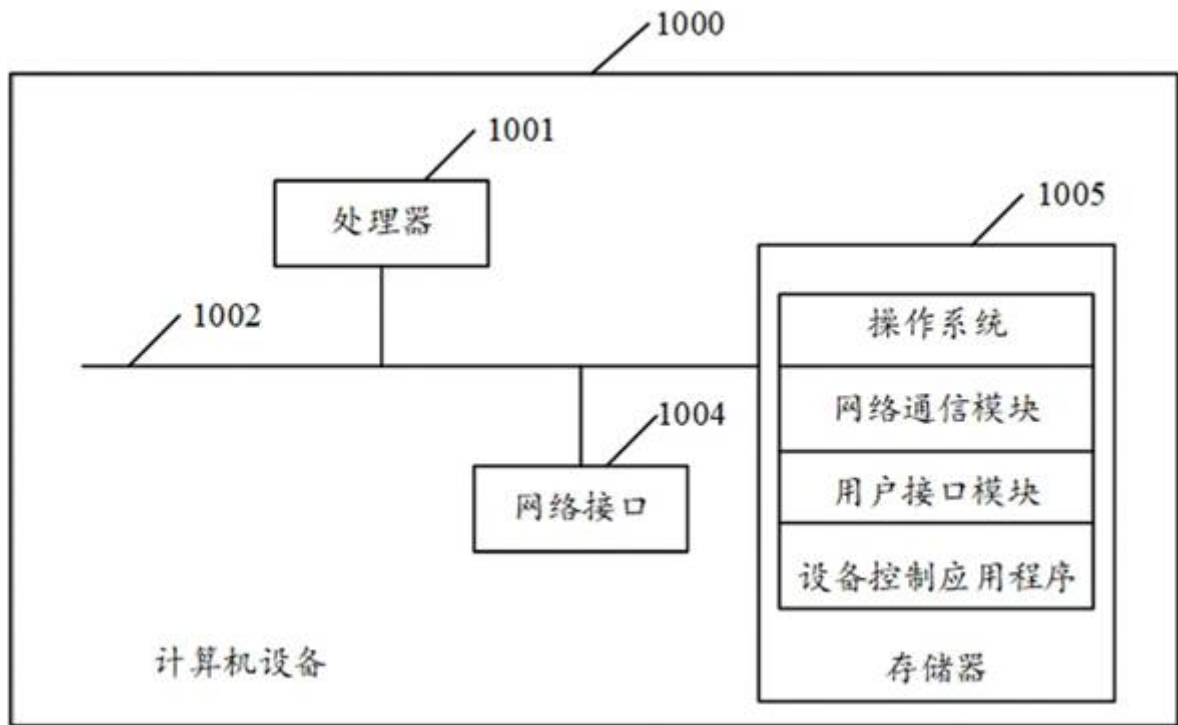


图11