



(21) 申请号 202311759393.X

G06F 9/50 (2006.01)

(22) 申请日 2023.12.20

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 109726169 A, 2019.05.07

申请公布号 CN 117435112 A

CN 109814903 A, 2019.05.28

(43) 申请公布日 2024.01.23

CN 111240631 A, 2020.06.05

(73) 专利权人 摩尔线程智能科技(成都)有限责任公司

CN 114972607 A, 2022.08.30

CN 116821040 A, 2023.09.29

WO 2018032696 A1, 2018.02.22

地址 610041 四川省成都市中国(四川)自由贸易试验区成都高新区天府大道中段1366号2栋E5座12层23-32

CN 113286005 A, 2021.08.20

CN 112465689 A, 2021.03.09

CN 110928695 A, 2020.03.27

(72) 发明人 请求不公布姓名 请求不公布姓名 请求不公布姓名

CN 114741044 A, 2022.07.12

CN 115202748 A, 2022.10.18

CN 116880937 A, 2023.10.13

(74) 专利代理机构 北京派特恩知识产权代理有限公司 11270

连成哲等. 基于ZYNQ的高清视频与图形叠加显示技术.《计算机技术与发展》.2022,第32卷(第4期), 74-79.

专利代理师 周艳 张颖玲

何小龙.dma-buf 由浅入深(一) —— 最简单的 dma-buf 驱动程序.《https://blog.csdn.net/hexiaolong2009/article/details/102596744》.2019, 1-6.

(51) Int. Cl.

审查员 杨欢

G06F 3/04845 (2022.01)

G06F 9/54 (2006.01)

G06T 1/20 (2006.01)

G06T 1/60 (2006.01)

G06F 3/06 (2006.01)

权利要求书3页 说明书16页 附图5页

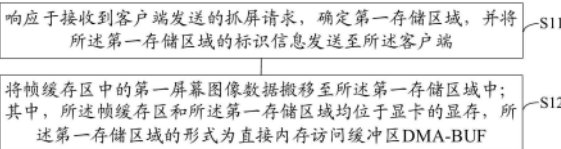
(54) 发明名称

数据处理方法、系统及装置、电子设备和存储介质

(57) 摘要

本公开提供了一种数据处理方法、系统及装置、电子设备和存储介质,以至少解决相关技术中在抓屏的过程中,需要通过CPU将GPU中的屏幕图像数据拷贝至系统内存中,存在CPU、内存等资源占用率高的问题,从而无法满足高分辨率、高帧率、实时性要求高等应用场景的需求。该方法应用于服务端中,所述方法包括:响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;将帧缓存区中的第一屏幕图像数据搬移至

所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。



1. 一种数据处理方法,其特征在于,应用于服务端中,所述方法包括:
响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;
将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;
其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF,所述第一存储区域是基于已创建的至少一块存储区域的状态信息确定的。
2. 根据权利要求1所述的方法,其特征在于,所述确定第一存储区域,包括:
获取已创建的至少一块存储区域的状态信息;
基于所述至少一块存储区域的状态信息,确定所述第一存储区域;其中,所述第一存储区域包括以下之一:所述至少一块存储区域中的一块存储区域、新的存储区域。
3. 根据权利要求2所述的方法,其特征在于,所述基于所述至少一块存储区域的状态信息,确定所述第一存储区域,包括以下至少之一:
在所述至少一块存储区域中的第二存储区域的状态信息为就绪状态的情况下,将所述第二存储区域作为所述第一存储区域;
在所述至少一块存储区域中的所有存储区域的状态信息均为非就绪状态的情况下,创建所述新的存储区域,将所述新的存储区域作为所述第一存储区域。
4. 根据权利要求1所述的方法,其特征在于,所述抓屏请求中包括抓屏区域和抓取格式;所述将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中,包括:
按照所述抓取格式对应的存储顺序,从所述帧缓存区中获取与所述抓屏区域匹配的第一屏幕图像数据;
按照所述抓取格式对应的存储顺序,将所述第一屏幕图像数据存储至所述第一存储区域中。
5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:
响应于检测到所述抓屏请求对应的第一屏幕图像数据满足第一预设条件的情况下,更新共享内存中所述抓屏请求对应的变化标识,以使得所述客户端在获取到所述更新后的变化标识后,发送新的抓屏请求至所述服务端中。
6. 根据权利要求1所述的方法,其特征在于,所述方法还包括:
响应于接收到客户端发送的连接请求,创建所述客户端对应的业务会话;其中,所述业务会话用于响应所述客户端发送的业务请求,所述业务请求包括抓屏请求。
7. 根据权利要求1至6中任一项所述的方法,其特征在于,所述方法还包括:
响应于接收到所述客户端发送的释放请求,基于所述释放请求中的目标标识信息,更新所述目标标识信息对应的存储区域的状态信息。
8. 一种数据处理方法,其特征在于,应用于客户端中,所述方法包括:
将抓屏请求发送至服务端中,以使得所述服务端响应于接收到所述抓屏请求,确定第一存储区域,并将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF,所述第一存储区域是基于已创建的至少一块存储区域的状态信息确定的;

响应于接收到所述服务端发送的所述第一存储区域的标识信息,存储所述第一存储区域的标识信息。

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:

从预设的共享内存中获取与所述抓屏请求对应的变化标识;

响应于所述变化标识满足第二预设条件的情况下,发送新的抓屏请求至服务端中,以使得所述服务端响应于接收到所述新的抓屏请求,确定第二存储区域,并将所述帧缓存区中的第三屏幕图像数据搬移至所述第二存储区域中;

响应于接收到所述服务端发送的所述第二存储区域的标识信息,存储所述第二存储区域的标识信息。

10. 根据权利要求8或9所述的方法,其特征在于,所述方法还包括:

响应于接收到所述服务端发送的就绪通知,调度所述显卡的视频处理单元生成所述第一存储区域的标识信息对应的码流;其中,所述就绪通知表征所述第一屏幕图像数据已搬移至所述第一存储区域中。

11. 根据权利要求10所述的方法,其特征在于,所述调度所述显卡的视频处理单元生成所述第一存储区域的标识信息对应的码流,包括:

将所述第一存储区域的标识信息透传至所述显卡的视频处理单元,以使得所述视频处理单元按照预设的编码方式,对所述第一存储区域中的第一屏幕图像数据进行编码,生成所述码流。

12. 一种数据处理系统,其特征在于,包括客户端和服务端,其中:

所述客户端,用于生成抓屏请求,将所述抓屏请求发送至服务端中;

所述服务端,用于响应于接收到所述抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF,所述第一存储区域是基于已创建的至少一块存储区域的状态信息确定的;

所述客户端,还用于接收所述第一存储区域的标识信息,并存储所述第一存储区域的标识信息。

13. 一种数据处理装置,其特征在于,应用于服务端中,所述装置包括:

确定模块,用于响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;

第一存储模块,用于将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF,所述第一存储区域是基于已创建的至少一块存储区域的状态信息确定的。

14. 一种数据处理装置,其特征在于,应用于客户端中,所述装置包括:

发送模块,用于将抓屏请求发送至服务端中,以使得所述服务端响应于接收到所述抓屏请求,确定第一存储区域,并将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF,所述第一存储区域是基于已创建的至少一块存储区域

的状态信息确定的；

第二存储模块,用于响应于接收到所述服务端发送的所述第一存储区域的标识信息,存储所述第一存储区域的标识信息。

15.一种电子设备,包括处理器和存储器,所述存储器存储有可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至11任一项所述方法。

16.一种计算机可读存储介质,其特征在于,其上存储有计算机程序,该计算机程序被处理器执行时实现权利要求1至11中任一项所述方法。

数据处理方法、系统及装置、电子设备和存储介质

技术领域

[0001] 本公开涉及但不限于计算机技术领域,尤其涉及一种数据处理方法、系统及装置、电子设备和存储介质。

背景技术

[0002] 相关技术中,在抓屏的过程中,需要先依赖CPU(Central Processing Unit,中央处理器),将Xorg(它是X窗口系统的一个实现,其中,X窗口系统是一种位图显示的视窗系统中记录的存储于GPU(Graphics Processing Unit,图形处理器)中的屏幕图像数据拷贝到系统内存中,存在CPU、内存等资源占用率高的问题,从而无法满足高分辨率、高帧率、实时性要求较高等应用场景的需求。

发明内容

[0003] 本公开实施例提供一种数据处理方法、系统及装置、电子设备和存储介质。

[0004] 本公开实施例的技术方案是这样实现的:

[0005] 本公开实施例提供一种数据处理方法,应用于服务端中,该方法包括:

[0006] 响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;

[0007] 将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;

[0008] 其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。

[0009] 本公开实施例提供一种数据处理方法,应用于客户端中,该方法包括:

[0010] 将抓屏请求发送至服务端中,以使得所述服务端响应于接收到所述抓屏请求,确定第一存储区域,并将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF;

[0011] 响应于接收到所述服务端发送的所述第一存储区域的标识信息,存储所述第一存储区域的标识信息。

[0012] 本公开实施例提供一种数据处理系统,该系统包括客户端及服务端,其中:

[0013] 所述客户端,用于生成抓屏请求,将所述抓屏请求发送至服务端中;

[0014] 所述服务端,用于响应于接收到所述抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF;

[0015] 所述客户端,还用于接收所述第一存储区域的标识信息,并存储所述第一存储区域的标识信息。

[0016] 本公开实施例提供一种数据处理装置,应用于服务端中,该装置包括:

[0017] 确定模块,用于响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;

[0018] 第一存储模块,用于将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。

[0019] 本公开实施例提供一种数据处理装置,应用于客户端中,该装置包括:

[0020] 发送模块,用于将抓屏请求发送至服务端中,以使得所述服务端响应于接收到所述抓屏请求,确定第一存储区域,并将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF;

[0021] 第二存储模块,用于响应于接收到所述服务端发送的所述第一存储区域的标识信息,存储所述第一存储区域的标识信息。

[0022] 本公开实施例提供一种电子设备,包括处理器和存储器,所述存储器存储有可在处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述方法。

[0023] 本公开实施例提供一种计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现上述方法。

[0024] 本公开实施例提供一种计算机程序产品,所述计算机程序产品包括存储了计算机程序的非瞬时性计算机可读存储介质,所述计算机程序被计算机读取并执行时,实现上述方法。

[0025] 在本公开实施例中,通过响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。这样,首先,通过在显卡上创建至少一块DMA-BUF,以支持至少一块屏幕的同步或异步抓取;其次,通过将抓取的屏幕数据从显存(即:帧缓存区)搬运到显存(即:DMA-BUF),一方面,由于显存带宽大于系统总线的带宽,提升了搬运效率,从而提升了抓取效率,另一方面,由于不需要CPU参与,降低了对CPU的占用率;最后,仅将DMA-BUF的标识信息发送至客户端中,而并不是发送整个屏幕图像数据,不仅减少了数据量,降低了对通信资源的占用,而且减少了数据拷贝带来的CPU、内存等资源的高消耗,从而在提高了系统的性能的同时还能够满足高分辨率、高帧率、实时性较高等应用场景的需求。

[0026] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,而非限制本公开。

附图说明

[0027] 此处的附图被并入说明书中并构成本说明书的一部分,这些附图示出了符合本公开的实施例,并与说明书一起用于说明本公开的技术方案。

[0028] 图1为本公开实施例提供的相关技术中利用XGetImage接口实现抓屏的示意图;

[0029] 图2为本公开实施例提供的相关技术中利用XShmCreateImage接口实现抓屏的示意图;

- [0030] 图3为本公开实施例提供的一种数据处理方法的实现流程示意图一；
- [0031] 图4为本公开实施例提供的一种数据处理方法的实现流程示意图二；
- [0032] 图5为本公开实施例提供的一种数据处理系统的组成结构示意图一；
- [0033] 图6为本公开实施例提供的一种数据处理方法的实现流程示意图三；
- [0034] 图7为本公开实施例提供的一种数据处理方法的实现流程示意图四；
- [0035] 图8为本公开实施例提供的一种数据处理方法的实现流程示意图五；
- [0036] 图9为本公开实施例提供的一种数据处理系统的组成结构示意图二；
- [0037] 图10为本公开实施例提供的一种数据处理装置的组成结构示意图一；
- [0038] 图11为本公开实施例提供的一种数据处理装置的组成结构示意图二；
- [0039] 图12为本公开实施例提供的一种电子设备的硬件实体示意图。

具体实施方式

[0040] 为了使本公开的目的、技术方案和优点更加清楚,下面将结合附图对本公开作进一步地详细描述,所描述的实施例不应视为对本公开的限制,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它实施例,都属于本公开保护的范围。

[0041] 在以下的描述中,涉及到“一些实施例”,其描述了所有可能实施例的子集,但是可以理解,“一些实施例”可以是所有可能实施例的相同子集或不同子集,并且可以在不冲突的情况下相互结合。

[0042] 在以下的描述中,所涉及的术语“第一\第二\第三”仅仅是区别类似的对象,不代表针对对象的特定排序,可以理解地,“第一\第二\第三”在允许的情况下可以互换特定的顺序或先后次序,以使这里描述的本公开实施例能够以除了在这里图示或描述的以外的顺序实施。

[0043] 除非另有定义,本文所使用的所有的技术和科学术语与属于本公开的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本公开实施例的目的,不是旨在限制本公开。

[0044] 相关技术中,XGetImage是X11协议(又称之为X 窗口系统)提供的一个抓取屏幕或者窗口数据的接口,该接口的实现主要包括数据抓取和数据传输这两部分。图1为本公开实施例提供的相关技术中利用XGetImage接口实现抓屏的示意图,如图1所示:

[0045] 对于数据抓取部分,服务端101(X server)通常基于readpixels函数将数据从显存中的帧缓冲区102(Framebuffer)拷贝到内存中。这样,不管是使用CPU搬运还是使用GPU加速,都会受到系统总线带宽的限制,同时占用大量总线带宽导致系统卡顿。

[0046] 对于数据传输部分,通常是通过Xlib(Xlib是一个用c语言编写的X Window System协议的客户端库)的跨进程通信接口,服务端101将抓取得到的图像拷贝至客户端103(Client),随着图像分辨率的增加,系统负载及延时也越大。同时对于需要使用GPU进行编码、渲染等场景,需要将数据从内存再次搬运到显存中。

[0047] XShmCreateImage也是X11协议提供的一个抓取屏幕或者窗口数据的接口,该接口是对XGetImage的优化改进。图2为本公开实施例提供的相关技术中利用XShmCreateImage接口实现抓屏的示意图,如图2所示:

[0048] 对于数据抓取部分,服务端111(X server)通常基于readpixels函数将数据从显

存中的帧缓冲区112 (Framebuffer) 拷贝到内存中。这样,不管是使用CPU搬运还是使用GPU加速,都会受到系统总线带宽的限制,同时占用大量总线带宽导致系统卡顿。

[0049] 对于数据传输部分,采用共享内存113 (Shared Memory, SM), 其中,共享内存113中存储有抓取的图像,服务端111将抓取的图像共享给客户端114,以减少一次数据的拷贝。但是对于桌面流化等场景,需要使用GPU进行编码、渲染等,需要将数据从内存再次搬运到显存中,会增加桌面流化处理管线/流水线的整体延时,尤其对于高分辨率的图像(例如,分辨率为4K(3840*2160)的云桌面),实时性更差。

[0050] 本公开实施例提供一种数据处理方法,首先,通过在显卡上创建至少一块DMA-BUF,以支持至少一块屏幕的同步或异步抓取;其次,通过将抓取的屏幕数据从显存(即:帧缓存区)搬运到显存(即:DMA-BUF),一方面,由于显存带宽大于系统总线的带宽,提升了搬运效率,从而提升了抓取效率,另一方面,由于不需要CPU参与,降低了对CPU的占用率;最后,仅将DMA-BUF的标识信息发送至客户端中,而并不是发送整个屏幕图像数据,不仅减少了数据量,降低了对通信资源的占用,而且减少了数据拷贝带来的CPU、内存等资源的高消耗,从而在提高了系统的性能的同时还能够满足高分辨率、高帧率、实时性较高等应用场景的需求。

[0051] 下面,将结合本公开实施例中的附图,对本公开实施例中的技术方案进行清楚、完整地描述。

[0052] 图3为本公开实施例提供的一种数据处理方法的实现流程示意图一,应用于服务端中,如图3所示,所述方法包括步骤S11至步骤S12,其中:

[0053] 步骤S11、响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端。

[0054] 这里,服务端可以至少包括Xorg中的X server,主要用于搬运屏幕图像数据,传递屏幕图像数据、屏幕图像数据的标识信息等。其中,屏幕图像数据可以是本地屏幕图像数据,也可以是远程屏幕图像数据。

[0055] 客户端可以是电子设备的某一应用进程。该客户端用于接收抓屏请求,传递抓屏请求、接收并存储屏幕图像数据对应的标识信息、码流等。其中,电子设备可以为笔记本电脑,平板电脑,台式计算机,机顶盒,移动设备(例如,移动电话,便携式音乐播放器,个人数字助理,专用消息设备,便携式游戏设备)等各种类型的终端,也可以实施为服务器。服务器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、内容分发网络(Content Delivery Network, CDN)、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。在实施时,该客户端和服务端可以位于同一电子设备,也可以位于不同的电子设备中。

[0056] 抓屏请求用于对屏幕图像进行抓取。在实施时,该抓屏请求可以是任意合适的方式生成的请求。例如,按键、手势、语音、其它电子设备发送的等。比如,通过按压实体按键生成的抓屏请求。又比如,在操作界面上输入的手势生成的抓屏请求。还比如,接收其它电子设备发送的抓屏请求。在实施时,本领域技术人员可以根据实际需求自主确定抓屏请求的生成方式,本公开实施例不作限定。

[0057] 在一些实施方式中,该抓屏请求中可以包括但不限于屏幕标识、抓屏区域、抓屏格

式、是否抓取鼠标等。其中,屏幕标识表征某一屏幕,在实施时,由于部分电子设备包括至少两个桌面,那么通过该屏幕标识,便可以指定某一屏幕。抓屏区域是指某一屏幕中的某一区域,在实施时,该抓屏区域的默认值可以是整个屏幕。抓屏格式是指抓取的屏幕图像数据的格式,屏幕图像数据的格式可以是任意合适的格式,例如,RGB (Red Green Blue,红绿蓝)、YUV (Luminance Chrominance,明亮度及色度)、sRGB (standard Red Green Blue,标准红绿蓝)等。

[0058] 第一存储区域位于显卡的显存中,该第一存储区域的形式为直接内存访问缓存区DMA-BUF。显卡可以是任意合适的显卡。在实施时,该显卡至少具有GPU、显存等。DMA-BUF是系统内核中一个独立的子系统,可以让不同设备、子系统之间进行内存共享。它的本质是缓冲区与文件的结合,即,该DMA-BUF既是一块物理缓冲区,又是一个文件。在一些实施方式中,该第一存储区域可以是显卡中已创建的某一存储区域,也可以是新建的存储区域。每一存储区域的形式均为DMA-BUF。

[0059] 在一些实施方式中,所述步骤S11中的“确定第一存储区域”包括步骤S111至步骤S112,其中:

[0060] 步骤S111、获取已创建的至少一块存储区域的状态信息。

[0061] 这里,存储区域的状态信息可以包括但不限于就绪状态、未就绪状态等。其中,就绪状态表征当前存储区域可以被使用。在实施时,可以在显存上创建至少一块存储区域,每一存储区域用于存储屏幕图像数据。

[0062] 步骤S112、基于所述至少一块存储区域的状态信息,确定所述第一存储区域;其中,所述第一存储区域包括以下之一:所述至少一块存储区域中的一块存储区域、新的存储区域。

[0063] 这里,可以通过缓冲池的方式来管理已创建的多块存储区域,该缓冲池既可以是对已创建的多块存储区域进行管理,还具有增加存储区域、回收存储区域等能力。在实施时,可以对已创建的多个存储区域进行遍历,若某一块存储区域的状态信息为就绪状态,则可以将该存储区域作为该第一存储区域;若已创建的多个存储区域的状态信息均为非就绪状态,那么,可以先创建新的存储区域,将该新的存储区域作为该第一存储区域。

[0064] 在一些实施方式中,所述步骤S112包括步骤S1121至步骤S1122,其中:

[0065] 步骤S1121、在所述至少一块存储区域中的第二存储区域的状态信息为就绪状态的情况下,将所述第二存储区域作为所述第一存储区域。

[0066] 这里,可以对所有已创建的存储区域进行遍历,若某一存储区域的状态信息为就绪状态,则将该存储区域作为第二存储区域,停止继续遍历。例如,当前已创建了两块存储区域,即:存储区域0~存储区域1,按照正序依次遍历存储区域0~存储区域1,首先遍历存储区域0,若存储区域0的状态信息为就绪状态,则将存储区域0作为该第二存储区域,并停止继续遍历;若存储区域0的状态信息为非就绪状态,则继续遍历存储区域1,若存储区域1的状态信息为就绪状态,则将存储区域1作为该第二存储区域,并停止遍历,若存储区域1的状态信息也为非就绪状态,则停止遍历。

[0067] 在一些实施方式中,已创建的存储区域并不会在数据传输完成后销毁,而是在下一次传输时可以重用,这样,降低了因反复创建带来的系统开销。

[0068] 步骤S1122、在所述至少一块存储区域中的所有存储区域的状态信息均为非就绪

状态的情况下,创建所述新的存储区域,将所述新的存储区域作为所述第一存储区域。

[0069] 这里,若所有已创建的存储区域均为非就绪状态,此时,需要创建新的存储区域。例如,若当前已创建了存储区域0~存储区域1均为非就绪状态,此时,创建新的存储区域,将该新的存储区域作为第一存储区域。

[0070] 存储区域的标识信息可以包括但不限于编号、文件描述符等。其中,文件描述符在形式上是一个非负整数,其实质是一个索引值。在实施时,通过该标识信息,便可以得到该存储区域。

[0071] 客户端可以按照预设的存储方式存储该存储区域的标识信息。其中,存储方式可以是任意合适的存储方式,例如,数组、队列等。

[0072] 在一些实施方式中,服务端可以处理至少一个客户端发送的抓屏请求,在实施时,服务端也可以按照该存储方式存储该存储区域的标识信息,以支持至少一帧屏幕图像数据、至少一个屏幕的异步/同步抓取。

[0073] 步骤S12、将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。

[0074] 这里,帧缓存区(FrameBuffer,FB)是系统内核将显存进行抽象后的一块缓存,用于存储待显示的图像。在实施时,通过对FB进行读取,可以获取当前屏幕的待显示的图像。在一些实施方式中,FB的数量可以为至少一个,在实施时,不同的屏幕可以对应不同的FB。该帧缓存区与该抓屏请求适配。在一些实施方式中,该服务端调度GPU将第一屏幕图像数据搬移至该第一存储区域中。

[0075] 在一些实施方式中,所述抓屏请求中包括抓屏区域和抓取格式,所述步骤S12包括步骤S121至步骤S122,其中:

[0076] 步骤S121、按照所述抓取格式对应的存储顺序,从所述帧缓存区中获取与所述抓屏区域匹配的第一屏幕图像数据。

[0077] 这里,抓屏区域可以是某一屏幕中的一个小区域,也可以是整个屏幕。在实施时,可以根据该抓屏区域,获取对应的第一屏幕图像数据。例如,当前屏幕大小为**1920*1080**,抓屏区域可以是左上角为(100,100)、右下角为(800,600)所形成的区域,那么则将该抓屏区域对应的图像数据作为该第一屏幕图像数据。这样,通过抓屏请求指定抓取某一屏幕、或某一屏幕中的某一区域,相对于仅能抓取整个屏幕而言,提升了抓屏的灵活性。

[0078] 抓取格式可以是任意合适的格式,例如,RGB、YUV、NV12等。在一些实施方式中,该抓取格式与帧缓存区中的存储的屏幕图像数据的格式可以相同,也可以不同。在实施时,可以按照该抓取格式对应的存储顺序,从该帧缓存区中读出该第一屏幕图像数据。存储顺序可以包括但不限于stride、twiddle等。其中,stride是指按照“从左到右、从上到下”的线性顺序进行存储,twiddle是指按照“zigzag”的Z字型顺序进行存储。

[0079] 步骤S122、按照所述抓取格式对应的存储顺序,将所述第一屏幕图像数据存储至所述第一存储区域中。

[0080] 这里,按照该抓取格式对应的存储顺序,将第一屏幕图像数据存储至第一存储区域。例如,若帧缓存区中的屏幕图像数据的格式为RGB,抓取格式为YUV,此时,可以按照YUV对应的存储顺序,从帧缓存区中读取该第一屏幕图像数据,并按照YUV对应的存储顺序,将

该第一屏幕图像数据存储至第一存储区域。这样,通过抓屏请求设定抓屏格式,相较于仅能获取源格式的数据而言,提升了抓屏的灵活性,同时在数据搬运的过程中,利用GPU通过对应的存储顺序,读写数据以完成数据格式的转换,不仅提升了数据转换的效率,而且由于不需要CPU的参与,降低了对CPU的占用。

[0081] 在一些实施方式中,当第一屏幕图像数据成功搬运至第一存储区域后,发送就绪通知至客户端,以使得客户端可以从第一存储区域中获取第一屏幕图像数据、调度GPU的VPU (Video ProcessUnit, 视频处理单元) 进行硬编码、或调度GPU进行二次渲染等。其中,VPU是一种全新的视频处理平台核心引擎,具有硬解码功能以及减少CPU负荷的能力。

[0082] 在本公开实施例中,首先,通过在显卡上创建至少一块DMA-BUF,以支持至少一块屏幕的同步或异步抓取;其次,通过将抓取的屏幕数据从显存(即:帧缓存区)搬运到显存(即:DMA-BUF),一方面,由于显存带宽大于系统总线的带宽,提升了搬运效率,从而提升了抓取效率,另一方面,由于不需要CPU参与,降低了对CPU的占用率;最后,仅将DMA-BUF的标识信息发送至客户端中,而并不是发送整个屏幕图像数据,不仅减少了数据量,降低了对通信资源的占用,而且减少了数据拷贝带来的CPU、内存等资源的高消耗,从而在提高了系统的性能的同时还能够满足高分辨率、高帧率、实时性较高等应用场景的需求。

[0083] 在一些实施方式中,所述方法还包括步骤S13,其中:

[0084] 步骤S13、响应于检测到所述抓屏请求对应的第一屏幕图像数据满足第一预设条件的情况下,更新共享内存中所述抓屏请求对应的变化标识,以使得所述客户端在获取到所述更新后的变化标识后,发送新的抓屏请求至所述服务端中。

[0085] 这里,第一预设条件可以是任意合适的条件,例如,该抓屏请求中的抓屏区域内的屏幕图像数据(对应于第一屏幕图像数据)发生变化、该抓屏区域内的某一区域的屏幕图像数据发生变化等。

[0086] 共享内存是指系统内存中的一片存储空间,用于存储至少一个变化标识,变化标识用于表征某一屏幕的抓屏区域中的内容是否发生变化等。在实施时,可以通过任意合适的方式来表示该变化标识。例如,当变化标识为“0”时,表征对应的抓屏区域的内容未发生变化,当变化标识为“1”时,表征对应的抓屏区域的内容发生变化。又例如,当变化标识为“FALSE”时,表征对应的抓屏区域的内容未发生变化,当变化标识为“TRUE”时,表征对应的抓屏区域的内容发生变化。

[0087] 在实施时,由于屏幕的显示内容在一段时间内可能是不变的,因此,可以在屏幕的显示内容发生变化时,再去抓取屏幕,那么,服务端可以监测该抓屏区域的内容是否发生变化,若发生变化,则进行记录,并更新共享内存中对应的变化标识,以使得客户端通过读取共享内存,得到更新后的变化标识后,再发送新的抓屏请求至服务端中,以得到新的屏幕图像数据。

[0088] 在本公开实施方式中,通过响应于检测到所述抓屏请求对应的第一屏幕图像数据满足第一预设条件的情况下,更新共享内存中所述抓屏请求对应的变化标识,以使得所述客户端在获取到所述更新后的变化标识后,发送新的抓屏请求至所述服务端中。这样,仅在抓屏区域的内容发生变化的情况下再发送抓屏请求来抓取屏幕内容,不仅减少了不必要的抓取,而且还降低了搬运重复内容的可能性,从而有效的降低了对带宽的占用。

[0089] 在一些实施方式中,所述方法还包括步骤S14,其中:

[0090] 步骤S14、响应于接收到客户端发送的连接请求,创建所述客户端对应的业务会话。

[0091] 这里,业务会话(session)用于响应客户端发送的业务请求。其中,业务请求可以包括但不限于抓屏请求、会话连接与关闭等。在实施时,在DDX(Device Dependent X,设备相关X,是Xorg处理硬件/操作系统的一部分)初始化时,会创建一个跨进程通信的节点socket,并将其注册为Xorg handler1(用于处理各种类型的回调函数),此后服务端会在这个节点上侦听连接请求。当接收到客户端发送的连接请求时,会创建一个业务会话,该业务会话将会注册成一个独立的Xorghandler,该Xorg handler用于响应这个客户端的业务请求。

[0092] 在本公开实施方式中,通过响应于接收到客户端发送的连接请求,创建所述客户端对应的业务会话。这样,利用创建的业务会话与客户端进行业务通信,实现了业务通信与连接通信的解耦。

[0093] 在一些实施方式中,所述方法还包括步骤S15,其中:

[0094] 步骤S15、响应于接收到所述客户端发送的释放请求,基于所述释放请求中的目标标识信息,更新所述目标标识信息对应的存储区域的状态信息。

[0095] 这里,释放请求用于释放某一存储区域,即:该存储区域的屏幕图像数据可以删除。在实施时,该释放请求中至少包括某一存储区域的标识信息(目标标识信息),这样,服务端根据该标识信息,将对应的存储区域的状态信息更新为就绪状态,以便于存储区域的重用。

[0096] 在本公开实施方式中,通过响应于接收到所述客户端发送的释放请求,基于所述释放请求中的目标标识信息,更新所述目标标识信息对应的存储区域的状态信息。这样,通过及时释放对DMA-BUF的占用,以便于DMA-BUF的重用,从而降低了因未及时释放而创建新的DMA-BUF带来的系统开销。

[0097] 图4为本公开实施例提供的一种数据处理方法的实现流程示意图二,应用于客户端中,如图4所示,所述方法包括步骤S21至步骤S22,其中:

[0098] 步骤S21、将抓屏请求发送至服务端中,以使得所述服务端响应于接收到所述抓屏请求,确定第一存储区域,并将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。

[0099] 这里,客户端通过对应的业务会话,将抓屏请求发送至服务端中。在实施时,抓屏请求的生成方式可以参见上述步骤S11中的具体实施方式。

[0100] 服务端确定第一存储区域的方式、及将屏幕图像数据搬移至第一存储区域的过程可以参见上述步骤S11至步骤S12的具体实施方式。

[0101] 步骤S22、响应于接收到所述服务端发送的所述第一存储区域的标识信息,存储所述第一存储区域的标识信息。

[0102] 这里,可以按照数组、队列等方式存储第一存储区域的标识信息。在实施时,客户端可以发送至少一个抓屏请求至服务端中,由于不同的抓屏请求中抓屏区域、屏幕、抓取格式等可以相同,也可以不同,因此,服务端可能返回不同的存储区域的标识信息,那么,客户端可以存储服务端发送的至少一个存储区域的标识信息,这样,待接收到某一存储区域的

数据搬运完成后,便可以根据该标识信息去取数据、或二次使用(例如,硬编码、渲染)等。

[0103] 在本公开实施例中,首先,服务端通过在显卡上创建至少一块DMA-BUF,以支持至少一块屏幕的同步或异步抓取;其次,客户端将抓屏请求发送至服务端中,以使得服务端将抓取的屏幕数据从显存(即:帧缓存区)搬运到显存(即:DMA-BUF),一方面,由于显存带宽大于系统总线的带宽,提升了搬运效率,从而提升了抓取效率,另一方面,由于不需要CPU参与,降低了对CPU的占用率;最后,接收服务端发送的DMA-BUF的标识信息,而并不是发送整个屏幕图像数据,不仅减少了数据量,降低了对通信资源的占用,而且减少了数据拷贝带来的CPU、内存等资源的高消耗,从而在提高了系统的性能的同时还能够满足高分辨率、高帧率、实时性较高等应用场景的需求。

[0104] 在一些实施方式中,所述方法还包括步骤S23至步骤S25,其中:

[0105] 步骤S23、从预设的共享内存中获取与所述抓屏请求对应的变化标识。

[0106] 这里,共享内存是指系统内存中的一片存储空间。在实施时,可以预先创建该共享内存,用于存储至少一个变化标识,变化标识用于表征某一屏幕的抓屏区域(对应某一抓屏请求)中的内容是否发生变化等。

[0107] 步骤S24、响应于所述变化标识满足第二预设条件的情况下,发送新的抓屏请求至服务端中,以使得所述服务端响应于接收到所述新的抓屏请求,确定第二存储区域,并将所述帧缓存区中的第三屏幕图像数据搬移至所述第二存储区域中。

[0108] 这里,第二预设条件可以是任意合适的条件,例如,为预设值、为预设标志等。比如,当变化标识为“1”(第二预设条件)时,表征抓屏区域的内容发生了改变,此时,需要发送新的抓屏请求至服务端中,以得到新的屏幕图像数据。其中,新的抓屏请求的内容与原抓屏请求的内容的可以相同,也可以不同。

[0109] 在实施时,服务端确定第二存储区域的方式、及搬运屏幕图像数据的过程可以参见前述步骤S11至步骤S12的具体实施方式。

[0110] 步骤S25、响应于接收到所述服务端发送的所述第二存储区域的标识信息,存储所述第二存储区域的标识信息。

[0111] 这里,可以按照数组、队列等方式存储第二存储区域的标识信息。

[0112] 在本公开实施方式中,通过共享内存中的变化标识来确定抓屏区域的内容是否发生改变,仅在发生改变后才发送新的抓屏请求来获取新的屏幕内容,不仅减少了不必要的抓取,而且也降低了服务端搬运重复内容的可能性,从而有效的降低了对带宽的占用。

[0113] 在一些实施方式中,所述方法还包括步骤S26,其中:

[0114] 步骤S26、响应于接收到所述服务端发送的就绪通知,调度所述显卡的视频处理单元生成所述第一存储区域的标识信息对应的码流。

[0115] 这里,就绪通知表征第一屏幕图像数据已存储至第一存储区域中,此时,客户端可以调度VPU进行编码,以生成对应的码流。在实施时,该显卡还包括VPU。

[0116] 在一些实施方式中,所述步骤S26中的“调度所述显卡的视频处理单元生成所述第一存储区域的标识信息对应的码流”,包括步骤S261,其中:

[0117] 步骤S261、将所述第一存储区域的标识信息透传至所述显卡的视频处理单元,以使得所述视频处理单元按照预设的编码方式,对所述第一存储区域中的第一屏幕图像数据进行编码,生成所述码流。

[0118] 这里,将标识信息透传至VPU中,那么,VPU接收到该标识信息后,根据该标识信息,直接访问位于显存中第一存储区域的屏幕图像数据,并按照该编码策略,对该屏幕图像数据进行编码,生成该屏幕图像数据对应的码流。其中,编码策略可以包括但不限于编码方式、编码格式等。其中,编码方式可以包括但不限于硬编码。编码格式可以是任意合适的格式。例如,对于视频而言,编码格式可以包括但不限于MPEG2、MPEG4、H.264、AVS+、H.265等。

[0119] 在本公开实施方式中,通过响应于接收到所述服务端发送的就绪通知,调度所述显卡的视频处理单元生成所述第一存储区域的标识信息对应的码流;其中,所述就绪通知表征所述第一屏幕图像数据已搬移至所述第一存储区域中。这样,通过视频处理单元直接使用显存中的屏幕图像数据,实现了数据的零拷贝,不仅使得CPU占用较低,而且码流生成效率高,从而可以将其应用于云桌面渲染、实时直播、数字孪生等实时性要求较高的场景中。

[0120] 在一些实施方式中,所述方法还包括步骤S27,其中:

[0121] 步骤S27、将释放请求发送至服务端中,以使得所述服务端基于所述释放请求中的目标标识信息,更新所述目标标识信息对应的存储区域的状态信息。

[0122] 这里,释放请求用于释放某一存储区域,即:该存储区域的屏幕图像数据可以删除,此时,该存储区域可以被复用。

[0123] 释放请求的生成方式可以是任意合适的方式,例如,按键、手势、语音、其它电子设备发送的、自动等。例如,当客户端收到VPU编码后的码流后,自动生成该释放请求。

[0124] 在本公开实施方式中,通过将释放请求发送至服务端中,以使得所述服务端基于所述释放请求中的目标标识信息,更新所述目标标识信息对应的存储区域的状态信息。这样,客户端及时释放对DMA-BUF的占用,以便于服务端对该DMA-BUF的重用,从而降低了因未及时释放而创建新的DMA-BUF带来的系统开销。

[0125] 图5为本公开实施例提供的一种数据处理系统的组成结构示意图一,如图5所示,所述系统30包括客户端31和服务端32,其中:

[0126] 所述客户端31,用于生成抓屏请求,将所述抓屏请求发送至服务端中;

[0127] 所述服务端32,用于响应于接收到所述抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF;

[0128] 所述客户端31,还用于接收所述第一存储区域的标识信息,并存储所述第一存储区域的标识信息。

[0129] 这里,客户端31可以是电子设备的某一应用进程。

[0130] 服务端32可以包括Xorg。在实施时,可以先通过跨进程通信(Domain Socket),服务端32建立与客户端31之间的连接,在连接成功后,服务端32创建并初始化对应的业务会话,以响应客户端31的业务请求,例如,抓屏请求。在实施时,服务端32确定第一存储区域的方式、及将屏幕图像数据搬运至第一存储区域的过程可以参见上述步骤S11至步骤S12的具体实施方式。

[0131] 图6为本公开实施例提供的一种数据处理方法的实现流程示意图三,应用于服务

端中,如图6所示,该方法包括步骤S301至步骤S303,其中:

[0132] 步骤S301、初始化完成后,创建通信节点;

[0133] 这里,在Xorg的DDX中构建了一个服务端(cap server),并注册一个公开的Domain socket节点(即:通信节点)提供连接服务。该服务端可以支持多个屏幕的抓取,也支持多个客户端的连接。在实施时,由于Screen(屏幕)是对显示设备的抽象,一张显卡可以包括至少一个屏幕,因此,可以通过Xorg的DIX(Device Independent X,与设备无关的X)中的AddScreen()函数来添加屏幕,并返回对应屏幕的标识信息。这样,以便于根据屏幕的标识信息,获取对应的屏幕的显示内容。

[0134] 步骤S302、在该通信节点上侦听连接请求;

[0135] 步骤S303、响应于接收到客户端发送的连接请求,创建并初始化对应的业务会话,并返回步骤S302。

[0136] 图7为本公开实施例提供的一种数据处理方法的实现流程图四,应用于服务端中,如图7所示,该方法包括步骤S310至步骤S315,其中:

[0137] 步骤S310、接收到客户端发送的抓屏请求;

[0138] 步骤S311、判断抓屏请求是否超时,若是,则进入步骤S314,反之,则进入步骤S312;

[0139] 步骤S312、基于该抓屏请求,确定第一存储区域,并将第一存储区域的标识信息发送至客户端中;

[0140] 这里,对所有已创建的存储区域进行遍历,若某一存储区域的状态信息为就绪状态,则将该存储区域作为第一存储区域,停止遍历;反之,若所有的存储区域均为未就绪状态,则创建一个新的存储区域,作为该第一存储区域。

[0141] 步骤S313、将帧缓存区中的第一屏幕图像数据搬移至第一存储区域中;

[0142] 这里,首先,根据抓屏请求中的屏幕标识,从多块屏幕中确定目标屏幕,以从多个帧缓存区中确定目标屏幕对应的目标帧缓存区;其次,根据抓屏请求中的抓屏区域,从该目标帧缓存区中获取抓屏区域和抓取格式对应的第一屏幕图像数据;最后,按照抓取格式对应的存储顺序,将第一屏幕图像数据搬运至第一存储区域中。

[0143] 步骤S314、判断与客户端的连接是否健康,若是,则进入步骤S310,反之,则进入步骤S315;

[0144] 步骤S315、关闭业务会话。

[0145] 图8为本公开实施例提供的一种数据处理方法的实现流程图五,应用于客户端中,如图8所示,该方法包括步骤S320至步骤S325,其中:

[0146] 步骤S320、建立与服务端的业务连接;

[0147] 步骤S321、将抓屏请求发送至服务端中;

[0148] 这里,抓屏请求中可以包括屏幕标识、抓取区域、抓取格式、是否抓取鼠标数据等。

[0149] 步骤S322、判断是否接收到数据,若是,则进入步骤S323,反之,进入步骤S324;

[0150] 这里,接收的数据可以是错误码、DMA-BUF的标识信息等。

[0151] 步骤S323、判断数据是否为错误码,若是,则进入步骤S325,反之,则进入步骤S321;

[0152] 步骤S324、判断是否超时,若是,则进入步骤S325,反之,则进入步骤S322;

[0153] 步骤S325、关闭业务会话。

[0154] 图9为本公开实施例提供的一种数据处理系统的组成结构示意图二,如图9所示,该系统包括客户端31、服务端32及显存33,显存33中包括至少一个已创建的存储区域331、及至少一个屏幕对应的帧缓存区332,其中:

[0155] 当客户端31与服务端32建立连接后,服务端32会创建一个业务会话322,以响应客户端31的业务请求;

[0156] 客户端31中的业务会话311与服务端32的业务会话322建立连接后,将抓屏请求发送至服务端32中;

[0157] 服务端32根据该抓屏请求中的屏幕标识,从多个屏幕321中确定目标屏幕,其中,不同的屏幕321对应不同的帧缓存区332;

[0158] 服务端32根据多个存储区域331的状态信息,确定第一存储区域,并将该第一存储区域的标识信息发送至客户端31中;其中,该标识信息可以是服务端32中某一数组中的数组序号(index),该数组对应的数组序号中存储了第一存储区域的文件标识符fd;

[0159] 客户端31接收并存储该第一存储区域的标识信息,在实施时,可以按照数组的方式存储该标识信息;

[0160] 服务端32根据该抓屏请求中的抓取区域和抓取格式,从与屏幕标识对应的帧缓存区332中获取第一屏幕图像数据;

[0161] 服务端32将第一屏幕图像数据搬运至第一存储区域,并发送一个就绪通知至客户端31;

[0162] 客户端31根据该就绪通知,可以从第一存储区域中获取第一屏幕图像数据、调度GPU的VPU进行硬编码、或调度GPU进行二次渲染等。

[0163] 在一些实施方式中,由于屏幕的内容在一段时间内可能并不会改变,那么为了减少不必要的抓取,服务端在检测到屏幕的内容发生改变的情况下,更新共享内存的变化标识,以使得客户端在获取该更新后的变化标识后,重新发送新的抓屏请求来获取变化后的屏幕内容,这样,在流化场景中,可以大大降低对带宽的占用。

[0164] 表1为本公开实施例提供的一种在同一硬件中,采用不同抓取方案的抓取耗时结果,如表1所示:

[0165]	方案/分辨率	4K	2K	1080P
	XGetImage	137.633ms	61.864ms	33.954 ms
	XShmCreateImage	48.496 ms	21.892 ms	12.328 ms
	本公开	1.289 ms	0.680 ms	0.461 ms

[0166] 在1080P/2K/4K等多种分辨率中,通过采用本公开的实施方案,即:利用GPU,将数据直接从显存搬运到显存,而不需要将数据从显存复制到内存,大大降低了抓屏的耗时。

[0167] 在本公开实施例中,首先,通过在显卡上创建至少一块DMA-BUF,以支持至少一块屏幕的同步或异步抓取;其次,通过将抓取的屏幕数据从显存(即:帧缓存区)搬运到显存(即:DMA-BUF),一方面,由于显存带宽大于系统总线的带宽,提升了搬运效率,从而提升了抓取效率,另一方面,由于不需要CPU参与,降低了对CPU的占用率;然后,仅将DMA-BUF的标识信息发送至客户端中,而并不是发送整个屏幕图像数据,不仅减少了数据量,降低了对通

信资源的占用,而且减少了数据拷贝带来的CPU、内存等资源的高消耗,从而在提高了系统的性能的同时还能够满足高分辨率、高帧率、实时性较高等应用场景的需求;最后,仅在抓屏区域的内容发生变化的情况下再发送抓屏请求来抓取屏幕内容,不仅减少了不必要的抓取,而且还降低了搬运重复内容的可能性,从而有效的降低了对带宽的占用。

[0168] 基于上述实施例,本公开实施例提供一种数据处理装置,应用于服务端中,图10为本公开实施例提供的一种数据处理装置的组成结构示意图一,如图10所示,所述数据处理装置40包括确定模块41、及第一存储模块42,其中:

[0169] 所述确定模块41,用于响应于接收到客户端发送的抓屏请求,确定第一存储区域,并将所述第一存储区域的标识信息发送至所述客户端;

[0170] 所述第一存储模块42,用于将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF。

[0171] 在一些实施方式中,所述第一存储模块42,还用于:获取已创建的至少一块存储区域的状态信息;基于所述至少一块存储区域的状态信息,确定所述第一存储区域;其中,所述第一存储区域包括以下之一:所述至少一块存储区域中的一块存储区域、新的存储区域。

[0172] 在一些实施方式中,所述第一存储模块42,还用于以下至少之一:在所述至少一块存储区域中的第二存储区域的状态信息为就绪状态的情况下,将所述第二存储区域作为所述第一存储区域;在所述至少一块存储区域中的所有存储区域的状态信息均为非就绪状态的情况下,创建所述新的存储区域,将所述新的存储区域作为所述第一存储区域。

[0173] 在一些实施方式中,所述抓屏请求中包括抓屏区域和抓取格式;所述第一存储模块42,还用于:按照所述抓取格式对应的存储顺序,从所述帧缓存区中获取与所述抓屏区域匹配的第一屏幕图像数据;按照所述抓取格式对应的存储顺序,将所述第一屏幕图像数据存储至所述第一存储区域中。

[0174] 在一些实施方式中,所述装置还包括第一更新模块,所述第一更新模块,用于:响应于检测到所述抓屏请求对应的第一屏幕图像数据满足第一预设条件的情况下,更新共享内存中所述抓屏请求对应的变化标识,以使得所述客户端在获取到所述更新后的变化标识后,发送新的抓屏请求至所述服务端中。

[0175] 在一些实施方式中,所述装置还包括创建模块,所述创建模块,用于:响应于接收到客户端发送的连接请求,创建所述客户端对应的业务会话;其中,所述业务会话用于响应所述客户端发送的业务请求,所述业务请求包括抓屏请求。

[0176] 在一些实施方式中,所述装置还包括第二更新模块,所述第二更新模块,用于:响应于接收到所述客户端发送的释放请求,基于所述释放请求中的目标标识信息,更新所述目标标识信息对应的存储区域的状态信息。

[0177] 以上装置实施例的描述,与上述方法实施例的描述是类似的,具有同方法实施例相似的有益效果。对于本公开装置实施例中未披露的技术细节,请参照本公开方法实施例的描述而理解。

[0178] 基于上述实施例,本公开实施例提供一种数据处理装置,应用于客户端中,图11为本公开实施例提供的一种数据处理装置的组成结构示意图二,如图11所示,所述数据处理装置50包括发送模块51、及第二存储模块52,其中:

[0179] 所述发送模块51,用于将抓屏请求发送至服务端中,以使得所述服务端响应于接收到所述抓屏请求,确定第一存储区域,并将帧缓存区中的第一屏幕图像数据搬移至所述第一存储区域中;其中,所述帧缓存区和所述第一存储区域均位于显卡的显存,所述第一存储区域的形式为直接内存访问缓冲区DMA-BUF;

[0180] 所述第二存储模块52,用于响应于接收到所述服务端发送的所述第一存储区域的标识信息,存储所述第一存储区域的标识信息。

[0181] 在一些实施方式中,所述装置还包括读取模块,所述读取模块,用于:从预设的共享内存中获取与所述抓屏请求对应的变化标识;所述发送模块51,还用于:响应于所述变化标识满足第二预设条件的情况下,发送新的抓屏请求至服务端中,以使得所述服务端响应于接收到所述新的抓屏请求,确定第二存储区域,并将所述帧缓存区中的第三屏幕图像数据搬移至所述第二存储区域中;所述第二存储模块52,还用于:响应于接收到所述服务端发送的所述第二存储区域的标识信息,存储所述第二存储区域的标识信息。

[0182] 在一些实施方式中,所述装置还包括调度模块,所述调度模块,用于:响应于接收到所述服务端发送的就绪通知,调度所述显卡的视频处理单元生成所述第一存储区域的标识信息对应的码流;其中,所述就绪通知表征所述第一屏幕图像数据已存储至所述第一存储区域中。

[0183] 在一些实施方式中,所述调度模块,还用于:将所述第一存储区域的标识信息透传至所述显卡的视频处理单元,以使得所述视频处理单元按照预设的编码方式,对所述第一存储区域中的第一屏幕图像数据进行编码,生成所述码流。

[0184] 以上装置实施例的描述,与上述方法实施例的描述是类似的,具有同方法实施例相似的有益效果。对于本公开装置实施例中未披露的技术细节,请参照本公开方法实施例的描述而理解。

[0185] 需要说明的是,本公开实施例中,如果以软件功能模块的形式实现上述方法,并作为独立的产品销售或使用,也可以存储在一个计算机可读取存储介质中。基于这样的理解,本公开实施例的技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来,该软件产品存储在一个存储介质中,包括若干指令用以使得一台电子设备(可以是个人计算机、服务器、或者网络设备等)执行本公开各个实施例所述方法的全部或部分。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read Only Memory,ROM)、磁碟或者光盘等各种可以存储程序代码的介质。这样,本公开实施例不限制于任何特定的硬件和软件结合。

[0186] 本公开实施例提供一种电子设备,包括存储器和处理器,所述存储器存储有可在处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述方法。

[0187] 本公开实施例提供一种计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现上述方法。所述计算机可读存储介质可以是瞬时性的,也可以是非瞬时性的。

[0188] 本公开实施例提供一种计算机程序产品,所述计算机程序产品包括存储了计算机程序的非瞬时性计算机可读存储介质,所述计算机程序被计算机读取并执行时,实现上述方法中的部分或全部步骤。该计算机程序产品可以具体通过硬件、软件或其结合的方式实现。在一个可选实施例中,所述计算机程序产品具体体现为计算机存储介质,在另一个可选

实施例, 计算机程序产品具体体现为软件产品, 例如软件开发包 (Software Development Kit, SDK) 等等。

[0189] 需要说明的是, 图12为本公开实施例提供的一种电子设备的硬件实体示意图, 如图12所示, 该电子设备600的硬件实体包括: 处理器601、通信接口602和存储器603, 其中:

[0190] 处理器601通常控制电子设备600的总体操作。

[0191] 通信接口602可以使电子设备通过网络与其他终端或服务器通信。

[0192] 存储器603配置为存储由处理器601可执行的指令和应用, 还可以缓存待处理器601以及电子设备600中各模块待处理或已经处理的数据 (例如, 图像数据、音频数据、语音通信数据和视频通信数据), 可以通过闪存 (FLASH) 或随机访问存储器 (Random Access Memory, RAM) 实现。处理器601、通信接口602和存储器603之间可以通过总线604进行数据传输。

[0193] 这里需要指出的是: 以上存储介质和设备实施例的描述, 与上述方法实施例的描述是类似的, 具有同方法实施例相似的有益效果。对于本公开存储介质和设备实施例中未披露的技术细节, 请参照本公开方法实施例的描述而理解。

[0194] 应理解, 说明书通篇中提到的“一个实施例”或“一实施例”意味着与实施例有关的特定特征、结构或特性包括在本公开的至少一个实施例中。因此, 在整个说明书各处出现的“在一个实施例中”或“在一实施例中”未必一定指相同的实施例。此外, 这些特定的特征、结构或特性可以任意适合的方式结合在一个或多个实施例中。应理解, 在本公开的各种实施例中, 上述各过程的序号的大小并不意味着执行顺序的先后, 各过程的执行顺序应以其功能和内在逻辑确定, 而不应对本公开实施例的实施过程构成任何限定。上述本公开实施例序号仅仅为了描述, 不代表实施例的优劣。

[0195] 需要说明的是, 在本文中, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下, 由语句“包括一个……”限定的要素, 并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0196] 在本公开所提供的几个实施例中, 应该理解到, 所揭露的设备和方法, 可以通过其它的方式实现。以上所描述的设备实施例仅仅是示意性的, 例如, 所述单元的划分, 仅仅为一种逻辑功能划分, 实际实现时可以有另外的划分方式, 如: 多个单元或组件可以结合, 或可以集成到另一个系统, 或一些特征可以忽略, 或不执行。另外, 所显示或讨论的各组成部分相互之间的耦合、或直接耦合、或通信连接可以是通过一些接口, 设备或单元的间接耦合或通信连接, 可以是电性的、机械的或其它形式的。

[0197] 上述作为分离部件说明的单元可以是、或也可以不是物理上分开的, 作为单元显示的部件可以是、或也可以不是物理单元; 既可以位于一个地方, 也可以分布到多个网络单元上; 可以根据实际的需要选择其中的部分或全部单元来实现本实施例方案的目的。

[0198] 另外, 在本公开实施例中的各功能单元可以全部集成在一个处理单元中, 也可以是各单元分别单独作为一个单元, 也可以两个或两个以上单元集成在一个单元中; 上述集成的单元既可以采用硬件的形式实现, 也可以采用硬件加软件功能单元的形式实现。

[0199] 本领域普通技术人员可以理解: 实现上述方法实施例的全部或部分步骤可以通过

程序指令相关的硬件来完成,前述的程序可以存储于计算机可读取存储介质中,该程序在执行时,执行包括上述方法实施例的步骤;而前述的存储介质包括:移动存储设备、只读存储器(Read Only Memory,ROM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0200] 或者,本公开上述集成的单元如果以软件功能模块的形式实现并作为独立的产品销售或使用时,也可以存储在一个计算机可读取存储介质中。基于这样的理解,本公开的技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台电子设备(可以是个人计算机、服务器、或者网络设备等)执行本公开各个实施例所述方法的全部或部分。而前述的存储介质包括:移动存储设备、ROM、磁碟或者光盘等各种可以存储程序代码的介质。

[0201] 以上所述,仅为本公开的实施方式,但本公开的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本公开揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本公开的保护范围之内。

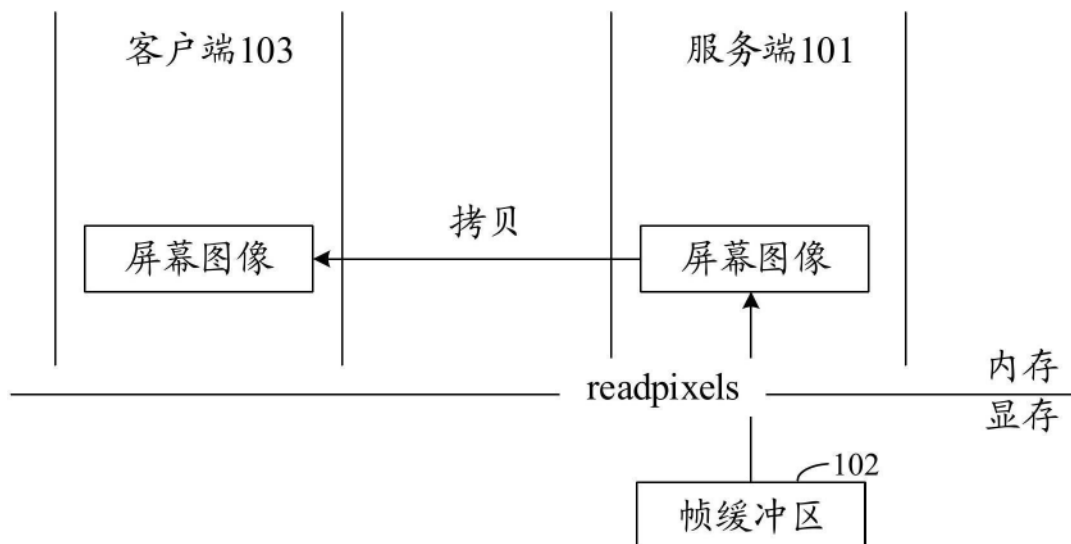


图 1

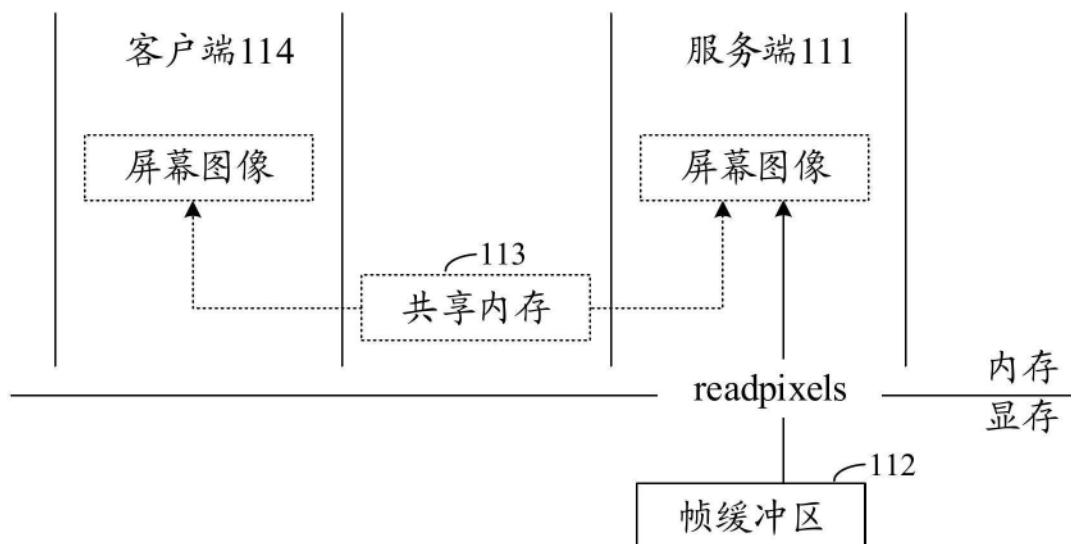


图 2

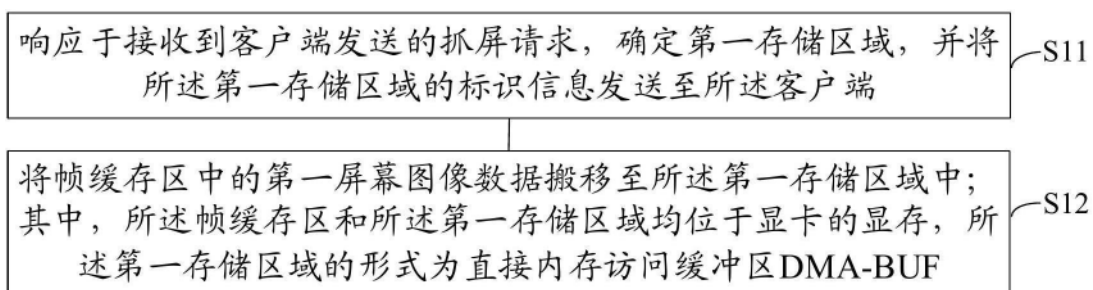


图 3

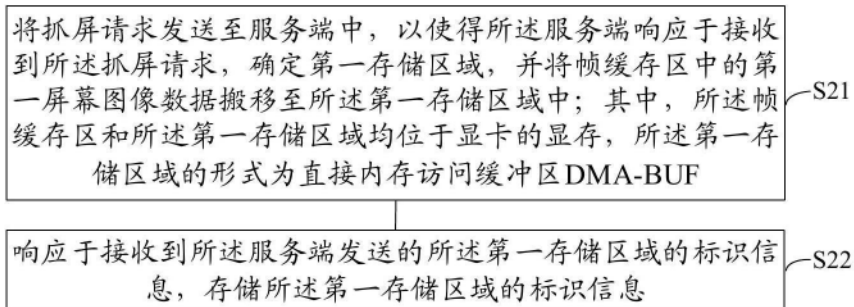


图 4

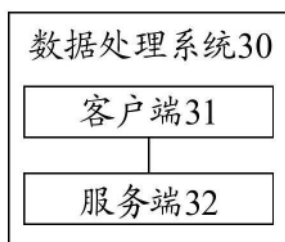


图 5

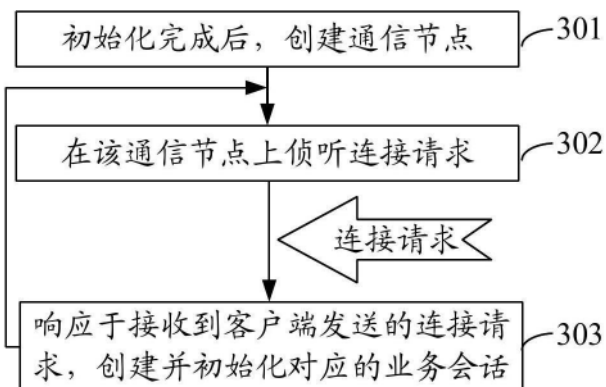


图 6

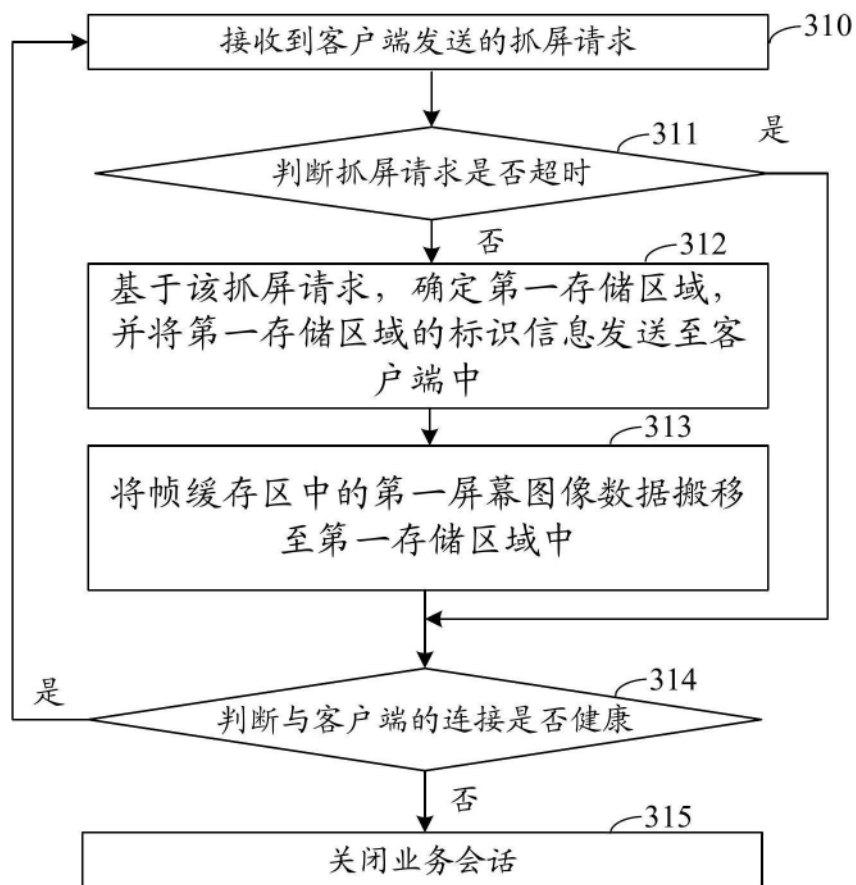


图 7

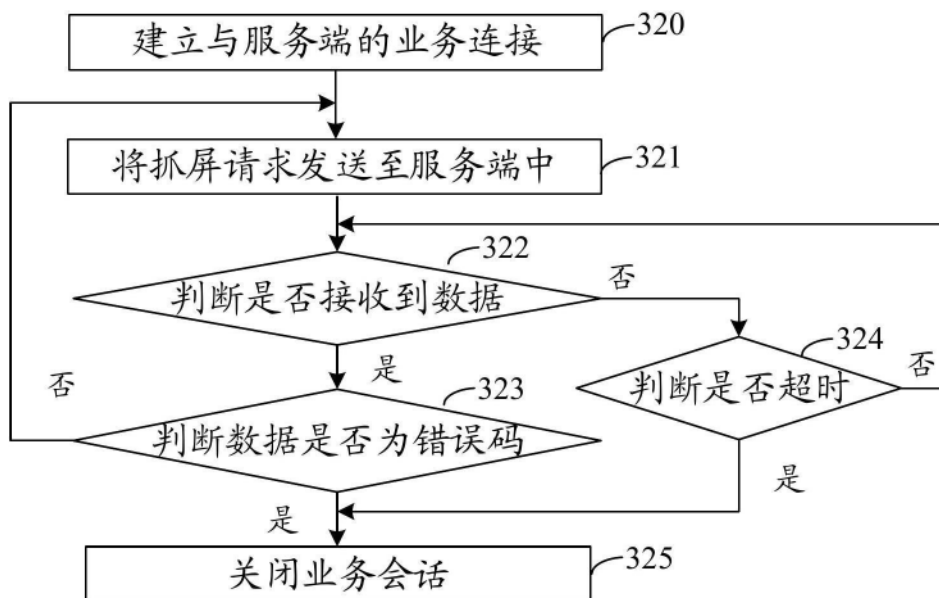


图 8

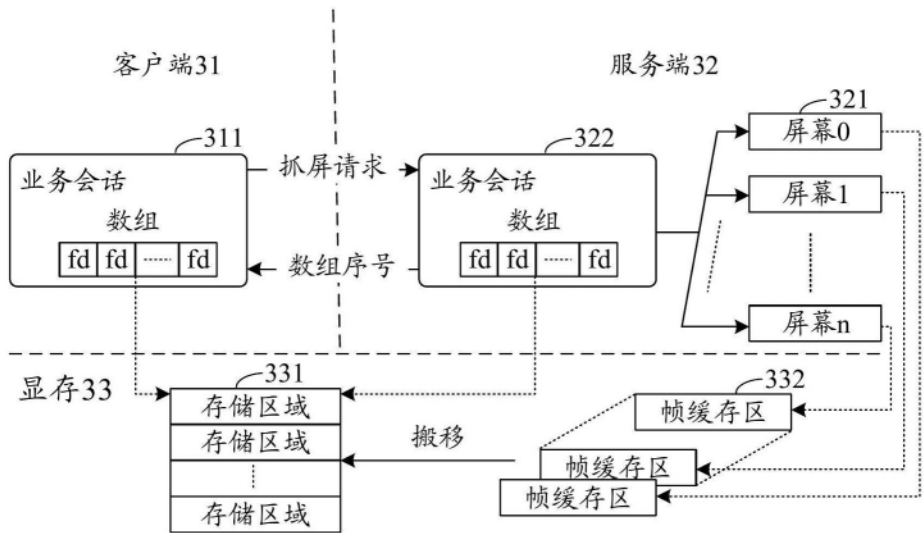


图 9

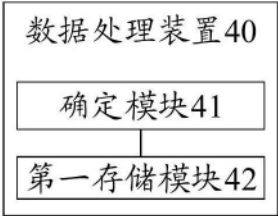


图 10

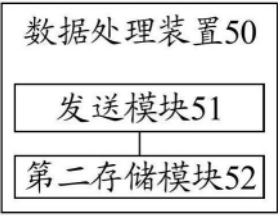


图 11

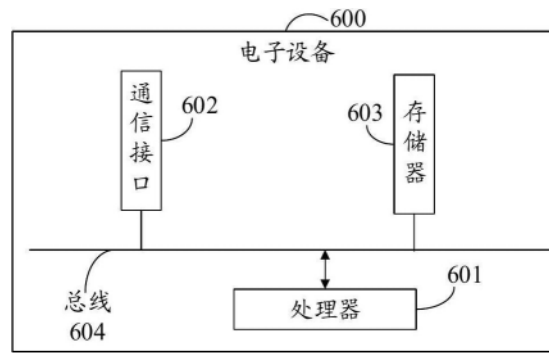


图 12