



(12) 发明专利

(10) 授权公告号 CN 112870726 B

(45) 授权公告日 2023. 04. 18

(21) 申请号 202110277879.4

G06T 1/20 (2006.01)

(22) 申请日 2021.03.15

审查员 梅洲雄

(65) 同一申请的已公布的文献号

申请公布号 CN 112870726 A

(43) 申请公布日 2021.06.01

(73) 专利权人 腾讯科技(深圳)有限公司

地址 518000 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 张富春

(74) 专利代理机构 北京康信知识产权代理有限

责任公司 11240

专利代理师 周婷婷

(51) Int. Cl.

G06F 9/50 (2006.01)

A63F 13/77 (2014.01)

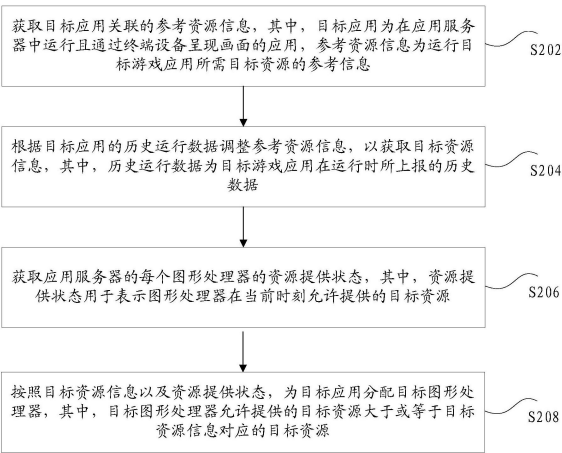
权利要求书3页 说明书21页 附图6页

(54) 发明名称

图形处理器的资源分配方法、装置和存储介质

(57) 摘要

本发明公开了一种云技术场景下的图形处理器的资源分配方法、装置和存储介质,还涉及云等技术。其中,该方法包括:获取目标应用关联的参考资源信息;根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息;获取应用服务器的每个图形处理器的资源提供状态,其中,资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源;按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,其中,目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源。本发明解决了图形处理器的资源利用率较低的技术问题。



1. 一种图形处理器的资源分配方法,其特征在于,包括:

获取目标应用关联的参考资源信息,其中,所述目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,所述参考资源信息为运行所述目标应用所需最大目标资源的参考信息;

根据所述目标应用的历史运行数据调整所述参考资源信息,以获取目标资源信息,其中,所述历史运行数据为所述目标应用在运行时所上报的历史数据;

所述根据所述目标应用的历史运行数据调整所述参考资源信息,以获取目标资源信息,包括:根据所述历史运行数据获取目标权重值,其中,所述历史运行数据包括动态运行数据和静态运行数据,所述静态运行数据为在所述目标应用运行过程中不发生实时变化的运行数据,所述动态运行数据为在所述目标应用运行过程中发生实时变化的运行数据,所述动态运行数包括运行卡顿数据,所述运行卡顿数据对应的卡顿率越高则提高对应的权重分配;整合计算所述目标权重值与所述参考资源信息,以获取所述目标资源信息;

获取所述应用服务器的每个图形处理器的资源提供状态,其中,所述资源提供状态用于表示所述图形处理器在当前时刻允许提供的所述目标资源;

按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器,其中,所述目标图形处理器允许提供的所述目标资源大于或等于所述目标资源信息对应的所述目标资源。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述目标应用的历史运行数据调整所述参考资源信息,以获取目标资源信息,包括:

在所述参考资源信息中携带有静态资源值的情况下,根据第一运行数据调整所述静态资源值,以获取目标资源值,其中,所述静态资源值为所述图形处理器在所述目标应用运行时所需提供的资源值,所述目标资源包括所述静态资源值,所述历史运行数据包括所述第一运行数据;在所述参考资源信息中携带有动态占用率的情况下,根据第二运行数据调整所述动态占用率,以获取目标占用率,其中,所述动态占用率为所述图形处理器在所述目标应用运行时所提升的资源占用率,所述目标资源包括所述动态占用率,所述历史运行数据包括所述第二运行数据。

3. 根据权利要求2所述的方法,其特征在于,所述按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器,包括:

在所述目标资源信息中携带有所述目标资源值的情况下,按照所述目标资源值以及所述资源提供状态对应的第一资源配置,为所述目标应用分配第一图形处理器,其中,所述第一资源配置用于表示所述图形处理器在所述当前时刻允许提供的资源值,所述第一图形处理器允许提供的资源值大于或等于所述目标资源值;

在所述目标资源信息中携带有所述目标占用率的情况下,按照所述目标资源值以及所述资源提供状态对应的第二资源配置,为所述目标应用分配第二图形处理器,其中,所述第二资源配置用于表示所述图形处理器所述当前时刻允许提升的资源占用率,所述第二图形处理器允许提升的资源占用率大于或等于所述目标占用率。

4. 根据权利要求3所述的方法,其特征在于,所述按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器,包括:

在所述目标资源信息中携带有目标资源值以及目标占用率的情况下,按照所述目标资

源值、所述目标占用率、所述第一资源配置以及所述第二资源配置,为所述目标应用分配第三图形处理器,其中,所述第三图形处理器允许提升的资源占用率大于或等于所述目标占用率,以及所述第三图形处理器允许提供的资源值大于或等于所述目标资源值。

5. 根据权利要求3或4所述的方法,其特征在于,在所述按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器之后,包括:

统计所述目标图形处理器在目标时间段内的N个实际占用率,其中,N为大于等于1的正整数,所述目标时间段包括所述当前时刻;

根据所述N个实际占用率调整所述目标图形处理器的第二资源配置。

6. 根据权利要求1所述的方法,其特征在于,所述根据所述历史运行数据获取目标权重值,包括:

将所述历史运行数据输入线性回归模型中,其中,所述线性回归模型使用具有线性关系的样本运行数据和样本权重值分别对初始回归模型进行训练后得到的模型;

获取所述线性回归模型输出的所述目标权重值。

7. 根据权利要求6所述的方法,其特征在于,在所述将所述历史运行数据输入线性回归模型中之前,包括:

在获取到目标反馈数据的情况下,更新所述线性回归模型,其中,所述目标反馈数据为所述目标应用在运行过程中,由于所述图形处理器提供的所述目标资源小于或等于所述目标运行时实际所需的所述目标资源而产生的反馈数据。

8. 一种图形处理器的资源分配装置,其特征在于,包括:

第一获取单元,用于获取目标应用关联的参考资源信息,其中,所述目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,所述参考资源信息为运行所述目标应用所需最大目标资源的参考信息;

调整单元,用于根据所述目标应用的历史运行数据调整所述参考资源信息,以获取目标资源信息,其中,所述历史运行数据为所述目标应用在运行时所上报的历史数据;

第二获取单元,用于获取所述应用服务器的每个图形处理器的资源提供状态,其中,所述资源提供状态用于表示所述图形处理器在当前时刻允许提供的所述目标资源;

分配单元,用于按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器,其中,所述目标图形处理器允许提供的所述目标资源大于或等于所述目标资源信息对应的所述目标资源;所述调整单元,包括:

第一调整模块,用于根据所述历史运行数据获取目标权重值,其中,所述历史运行数据包括动态运行数据和静态运行数据,所述静态运行数据为在所述目标应用运行过程中不发生实时变化的运行数据,所述动态运行数据为在所述目标应用运行过程中发生实时变化的运行数据,所述动态运行数包括运行卡顿数据,所述运行卡顿数据对应的卡顿率越高则提高对应的权重分配;

第二调整模块,用于整合计算所述目标权重值与所述参考资源信息,以获取所述目标资源信息。

9. 根据权利要求8所述的装置,其特征在于,所述调整单元,包括:

第一调整模块,用于在所述参考资源信息中携带有静态资源值的情况下,根据第一运行数据调整所述静态资源值,以获取目标资源值,其中,所述静态资源值为所述图形处理器

在所述目标应用运行时所需提供的资源值,所述目标资源包括所述静态资源值,所述历史运行数据包括所述第一运行数据;

第二调整模块,用于在所述参考资源信息中携带有动态占用率的情况下,根据第二运行数据调整所述动态占用率,以获取目标占用率,其中,所述动态占用率为所述图形处理器在所述目标应用运行时所提升的资源占用率,所述目标资源包括所述动态占用率,所述历史运行数据包括所述第二运行数据。

10. 根据权利要求9所述的装置,其特征在于,所述分配单元,包括:

第一分配模块,用于在所述目标资源信息中携带有所述目标资源值的情况下,按照所述目标资源值以及所述资源提供状态对应的第一资源配置,为所述目标应用分配第一图形处理器,其中,所述第一资源配置用于表示所述图形处理器在所述当前时刻允许提供的资源值,所述第一图形处理器允许提供的资源值大于或等于所述目标资源值;

第二分配模块,用于在所述目标资源信息中携带有所述目标占用率的情况下,按照所述目标资源值以及所述资源提供状态对应的第二资源配置,为所述目标应用分配第二图形处理器,其中,所述第二资源配置用于表示所述图形处理器在所述当前时刻允许提升的资源占用率,所述第二图形处理器允许提升的资源占用率大于或等于所述目标占用率。

11. 根据权利要求10所述的装置,其特征在于,所述分配单元,包括:

第三分配模块,用于在所述目标资源信息中携带有所述目标资源值以及所述目标占用率的情况下,按照所述目标资源值、所述目标占用率、所述第一资源配置以及所述第二资源配置,为所述目标应用分配第三图形处理器,其中,所述第三图形处理器允许提升的资源占用率大于或等于所述目标占用率,以及所述第三图形处理器允许提供的资源值大于或等于所述目标资源值。

12. 根据权利要求10或11所述的装置,其特征在于,包括:

统计模块,用于在所述按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器之后,统计所述目标图形处理器在目标时间段内的N个实际占用率,其中,N为大于等于1的正整数,

所述目标时间段包括所述当前时刻;

第三调整模块,用于在所述按照所述目标资源信息以及所述资源提供状态,为所述目标应用分配目标图形处理器之后,根据所述N个实际占用率调整所述目标图形处理器的第二资源配置。

13. 一种计算机可读的存储介质,其特征在于,所述计算机可读的存储介质包括存储的程序,其中,所述程序运行时执行上述权利要求1至7任一项中所述的方法。

14. 一种电子设备,包括存储器和处理器,其特征在于,所述存储器中存储有计算机程序,所述处理器被设置为通过所述计算机程序执行上述权利要求1至7任一项中所述的方法。

图形处理器的资源分配方法、装置和存储介质

技术领域

[0001] 本发明涉及计算机领域,具体而言,涉及一种图形处理器的资源分配方法、装置和存储介质。

背景技术

[0002] 近些年,云游戏的发展越发广泛。对于云游戏的用户来说,因云游戏摆脱了对硬件的依赖,用户可以得到高品质的游戏画面而无需购买高性能的计算机。但对于云游戏的运营商来说,云游戏还是会受限于云游戏服务器所能提供的相关资源。

[0003] 以作为云游戏所需关键资源的图形处理器(Graphics Processing Unit,简称GPU)资源为例说明,为了保障云游戏的最佳用户体验,一款云游戏部署在云游戏服务器之前会进行充分的测试,在各种场景观察游戏对GPU资源的消耗情况,然后选择云游戏对GPU的最大消耗作为游戏日常的资源调度基础。但是,云游戏的通常场景和GPU高消耗的场景并不是重合的,并且云游戏的实际在线人数以及用户所选择的GPU消耗等级也存在较大波动,进而使得现有技术中对GPU资源的分配不太合理,从而导致GPU资源利用率较低的技术问题出现。

[0004] 针对上述的问题,目前尚未提出有效的解决方案。

发明内容

[0005] 本发明实施例提供了一种图形处理器的资源分配方法、装置和存储介质,以至少解决图形处理器的资源利用率较低的技术问题。

[0006] 根据本发明实施例的一个方面,提供了一种图形处理器的资源分配方法,包括:获取目标应用关联的参考资源信息,其中,上述目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,上述参考资源信息为运行上述目标应用所需目标资源的参考信息;根据上述目标应用的历史运行数据调整上述参考资源信息,以获取目标资源信息,其中,上述历史运行数据为上述目标应用在运行时所上报的历史数据;获取上述应用服务器的每个图形处理器的资源提供状态,其中,上述资源提供状态用于表示上述图形处理器在当前时刻允许提供的上述目标资源;按照上述目标资源信息以及上述资源提供状态,为上述目标应用分配目标图形处理器,其中,上述目标图形处理器允许提供的上述目标资源大于或等于上述目标资源信息对应的上述目标资源。

[0007] 根据本发明实施例的另一方面,还提供了一种图形处理器的资源分配装置,包括:第一获取单元,用于获取目标应用关联的参考资源信息,其中,上述目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,上述参考资源信息为运行上述目标应用所需目标资源的参考信息;调整单元,用于根据上述目标应用的历史运行数据调整上述参考资源信息,以获取目标资源信息,其中,上述历史运行数据为上述目标应用在运行时所上报的历史数据;第二获取单元,用于获取上述应用服务器的每个图形处理器的资源提供状态,其中,上述资源提供状态用于表示上述图形处理器在当前时刻允许提供的上述目标资源;分

配单元,用于按照上述目标资源信息以及上述资源提供状态,为上述目标应用分配目标图形处理器,其中,上述目标图形处理器允许提供的上述目标资源大于或等于上述目标资源信息对应的上述目标资源。

[0008] 根据本发明实施例的又一方面,还提供了一种计算机可读的存储介质,该计算机可读的存储介质中存储有计算机程序,其中,该计算机程序被设置为运行时执行上述图形处理器的资源分配方法。

[0009] 根据本发明实施例的又一方面,还提供了一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其中,上述处理器通过计算机程序执行上述的图形处理器的资源分配方法。

[0010] 在本发明实施例中,获取目标应用关联的参考资源信息,其中,上述目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,上述参考资源信息为运行上述目标应用所需目标资源的参考信息;根据上述目标应用的历史运行数据调整上述参考资源信息,以获取目标资源信息,其中,上述历史运行数据为上述目标应用在运行时所上报的历史数据;获取上述应用服务器的每个图形处理器的资源提供状态,其中,上述资源提供状态用于表示上述图形处理器在当前时刻允许提供的上述目标资源;按照上述目标资源信息以及上述资源提供状态,为上述目标应用分配目标图形处理器,其中,上述目标图形处理器允许提供的上述目标资源大于或等于上述目标资源信息对应的上述目标资源,通过在为目标应用匹配的参考资源信息的基础上,根据目标应用的历史运行数据调整该参考资源信息,以获得与目标应用的运行情况之间契合度更高的目标资源信息,并通过契合度更高的目标资源值,为该目标应用分配更加合理的图形处理器,进而达到了对图形处理器的资源进行合理分配的目的,从而实现了提高图形处理器的资源利用率的技术效果,进而解决了图形处理器的资源利用率较低的技术问题。

附图说明

[0011] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0012] 图1是根据本发明实施例的一种可选的图形处理器的资源分配方法的应用环境的示意图;

[0013] 图2是根据本发明实施例的一种可选的图形处理器的资源分配方法的流程的示意图;

[0014] 图3是根据本发明实施例的一种可选的图形处理器的资源分配方法的示意图;

[0015] 图4是根据本发明实施例的另一种可选的图形处理器的资源分配方法的示意图;

[0016] 图5是根据本发明实施例的另一种可选的图形处理器的资源分配方法的示意图;

[0017] 图6是根据本发明实施例的另一种可选的图形处理器的资源分配方法的示意图;

[0018] 图7是根据本发明实施例的另一种可选的图形处理器的资源分配方法的示意图;

[0019] 图8是根据本发明实施例的一种可选的图形处理器的资源分配装置的示意图;

[0020] 图9是根据本发明实施例的另一种可选的图形处理器的资源分配装置的示意图;

[0021] 图10是根据本发明实施例的一种可选的电子设备的结构示意图。

具体实施方式

[0022] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明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发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,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,都应当属于本发明保护的范围。

[0023] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0024] 首先,为方便理解本发明实施例,下面对本发明中所涉及的部分术语或名词进行解释说明:

[0025] PCU (Peak concurrent users), 互联网术语,应用在网络游戏和其他互联网服务领域,意思是最高同时在线人数。

[0026] 图形处理器 (Graphics Processing Unit, 简称GPU), 又称显示核心、视觉处理器、显示芯片,是一种专门在个人电脑、工作站、游戏机和一些移动设备(如平板电脑、智能手机等)上做图像和图形相关运算工作的微处理器。

[0027] 线性回归 (Linear Regression, 简称LR) 模型,理解成一个线性方程。例如如果只有一个特征:也就是 $y=ax+b$,如果有两个特征也就是 $y=ax_1+bx_2+c$ 。

[0028] 云技术 (Cloud technology) 是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来,实现数据的计算、储存、处理和共享的一种托管技术。

[0029] 云技术 (Cloud technology) 基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源,如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用,将来每个物品都有可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0030] 云游戏 (Cloud gaming) 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,而是在云端服务器中运行,并由云端服务器将游戏场景渲染为视频音频流,通过网络传输给玩家游戏终端。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输入指令并发送给云端服务器的能力即可。

[0031] 人工智能 (Artificial Intelligence, AI) 是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能,感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。换句话说,人工智能是计算机科学的一个综合技术,它企图了解

智能的实质,并生产出一种新的能以人类智能相似的方式做出反应的智能机器。人工智能也就是研究各种智能机器的设计原理与实现方法,使机器具有感知、推理与决策的功能。

[0032] 人工智能技术是一门综合学科,涉及领域广泛,既有硬件层面的技术也有软件层面的技术。人工智能基础技术一般包括如传感器、专用人工智能芯片、云计算、分布式存储、大数据处理技术、操作/交互系统、机电一体化等技术。人工智能软件技术主要包括计算机视觉技术、语音处理技术、自然语言处理技术以及机器学习/深度学习等几大方向。

[0033] 机器学习(Machine Learning,ML)是一门多领域交叉学科,涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。专门研究计算机怎样模拟或实现人类的学习行为,以获取新的知识或技能,重新组织已有的知识结构使之不断改善自身的性能。机器学习是人工智能的核心,是使计算机具有智能的根本途径,其应用遍及人工智能的各个领域。机器学习和深度学习通常包括人工神经网络、置信网络、强化学习、迁移学习、归纳学习、式教学习等技术。

[0034] 随着人工智能技术研究和进步,人工智能技术在多个领域展开研究和应用,例如常见的智能家居、智能穿戴设备、虚拟助理、智能音箱、智能营销、无人驾驶、自动驾驶、无人机、机器人、智能医疗、智能客服等,相信随着技术的发展,人工智能技术将在更多的领域得到应用,并发挥越来越重要的价值。

[0035] 本申请实施例提供的方案涉及云技术的云游戏等技术、人工智能的机械学习等技术,具体通过如下实施例进行说明:

[0036] 根据本发明实施例的一个方面,提供了一种图形处理器的资源分配方法,可选地,作为一种可选的实施方式,上述图形处理器的资源分配方法可以但不限于应用于如图1所示的环境中。其中,可以但不限于包括用户设备102、网络110及服务器112,其中,该用户设备102上可以但不限于包括显示器108、处理器106及存储器104。

[0037] 具体过程可如下步骤:

[0038] 步骤S102,用户设备102通过检测虚拟按钮“登录”上的选中信号,以获取目标游戏应用1022的运行请求,其中,运行请求用于请求运行目标游戏应用1022;

[0039] 步骤S104-S106,用户设备102通过网络110将运行请求发送给服务器112,其中,运行请求中还携带有目标游戏应用1022的游戏标签;

[0040] 步骤S108,服务器112通过数据库114查找目标游戏应用1022关联的参考资源信息,以及目标游戏应用1022的历史运行数据,并通过处理引擎116根据历史运行数据将参考资源信进行处理,从而生成资源分配信息,其中,参考资源信息为测试阶段为目标游戏应用1022分配的初始资源信息,资源分配信息用于指示为目标游戏应用1022的运行所需分配的目标资源;

[0041] 步骤S110-S112,服务器112通过网络110将资源分配信息发送给用户设备102,用户设备102中的处理器106根据资源分配信息为目标游戏应用1022分配目标资源,并在分配好目标资源的情况下,将目标游戏应用1022的运行画面显示在显示器108中,以及将资源分配信息存储在存储器104中。

[0042] 除图1示出的示例之外,上述步骤可以由用户设备102独立完成,即由用户设备102执行资源分配信息的生成等步骤,从而减轻服务器的处理压力。该用户设备102包括但不限于手持设备(如手机)、笔记本电脑、台式电脑、车载设备等,本发明并不限制用户设备102的

具体实现方式。

[0043] 同理,除图1示出的示例之外,上述步骤也可以由服务器112独立完成,即由服务器112执行目标资源的分配等步骤,从而减轻对用户设备102的性能要求,以及提高资源分配的处理效率等。

[0044] 可选地,作为一种可选的实施方式,如图2所示,图形处理器的资源分配方法包括:

[0045] S202,获取目标应用关联的参考资源信息,其中,目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,参考资源信息为运行目标应用所需目标资源的参考信息;

[0046] S204,根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,其中,历史运行数据为目标应用在实际运行时所上报的历史数据;

[0047] S206,获取应用服务器的每个图形处理器的资源提供状态,其中,资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源;

[0048] S208,按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,其中,目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源。

[0049] 可选地,在本实施例中,图形处理器的资源分配方法可以但不限于应用在云游戏的GPU资源分配场景下,例如假设目标应用为目标游戏应用,应用服务器为云游戏服务器,则先在将目标游戏应用部署在云游戏服务器之前,对其进行充分的测试,以获取在各种场景下目标游戏应用的运行对GPU资源的消耗情况,然后选择游戏对GPU的最大消耗作为该目标游戏应用关联的参考资源信息。进一步,在获取到目标游戏应用关联的参考资源信息,且有目标游戏应用在实际运行中的历史运行数据指导的情况下,调整参考资源信息,以获取与目标游戏应用的实际运行更加契合的目标资源信息,再结合游戏应用服务器的每个图形处理器的资源提供状态,让每块GPU上都有合适的用户数,最终相比于传统的资源分配,在相同GPU资源的情况下,可以提高GPU资源的利用率,进而提升业务的PCU。

[0050] 可选地,在本实施例中,目标应用可以但不限于带有显示功能,且在应用服务器中运行且通过终端设备呈现画面,例如目标应用可以但不限于为视频应用、游戏应用、模拟器应用、漫画应用等。

[0051] 可选地,在本实施例中,参考资源信息可以但不限于为在将目标应用部署在应用服务器之前,工作人员通过产品测试而为目标应用的运行分配的资源信息,例如按照测试中目标应用所需要的最大目标资源来初步分配目标应用在实际运行所需的目标资源。

[0052] 可选地,在本实施例中,目标资源信息可以但不限于为与参考资源信息处于同一维度的资源信息,二者的区别可以但不限于在于参考资源信息可以但不限于表示目标游戏应用在实际运行所需的最大目标资源,而目标资源信息可以但不限于表示目标游戏应用在实际运行所需的实际目标资源。可选地,由于目标游戏应用在实际运行时只有很少情况下才会需要该最大目标资源,而大多数情况下所需的应为实际目标资源,进而使得该实际目标资源往往小于最大目标资源。

[0053] 可选地,在本实施例中,历史运行数据可以但不限于包括以下至少之一:动态运行数据、静态运行数据,其中,动态运行数据可以但不限于为在目标应用运行过程中发生实时变化的运行数据,例如GPU占用率、GPU空闲率、用户在线人数、剩余GPU资源、运行卡顿数据等;静态运行数据可以但不限于为在目标应用运行过程中一般不发生变化的运行数据,例

如应用服务器所关联机房的性能参数、出口宽带等。可选地,历史运行数据可以但不限于通过目标应用运行过程中所上报的日志获取,也可以但不限于通过检测目标应用的运行情况获取,在此不做限定。

[0054] 可选地,在本实施例中,目标资源可以但不限于用于衡量目标游戏应用运行时所需一种或多种维度的资源的量级指标。可选地,目标资源可以但不限于包括以下至少之一:静态资源、动态资源。其中,静态资源可以但不限于为GPU所能提供的固定资源值的静态衡量指标;动态资源可以但不限于为GPU所能提供的固定资源值的动态衡量指标。

[0055] 进一步以目标资源包括静态资源为例说明,可以但不限于预先将应用服务器的每块GPU的性能属性通过某种手段处理为更为直观的算力值,例如获取目标性能属性的第一GPU作为衡量,并假设该第一GPU可提供的最大算力值为100,再将该第一GPU分配给选取样本应用以运行,并获取该样本应用运行时该第一GPU的资源占用情况,如果该样本应用运行时该第一GPU的资源占用率为百分之五十,则可定义该样本应用运行时所需分配的算力值为50 ($100 \times 50\% = 50$);基于此,再选取另一不同性能属性的第二GPU,并将该第二GPU分配给上述样本应用,进而在该样本应用运行过程中,获取第二GPU的资源占用情况,如果该样本应用运行时该第二GPU的资源占用率为百分之百,则可定义该第二GPU可提供的最大算力值为50 ($50/100\% = 50$);同理,还可选取另一不同性能属性的第三GPU,如果该样本应用运行时该第二GPU的资源占用率为百分之二十,则可定义该第三GPU可提供的最大算力值为250 ($50/20\% = 250$)。

[0056] 再者以目标资源包括动态资源为例说明,可以但不限于用于表示GPU占用率,具体的,动态资源可以但不限于包括以下至少之一:在某一时刻的GPU实际的占用率、某一时间段内的全部GPU占用数据、某一段时间内GPU最高/低的占用率等。例如,在第一时刻,目标应用运行所需的动态资源为50,可以但不限于表示需为目标应用选取GPU占用率低于百分之五十的GPU,而GPU占用率低于百分之五十的GPU可以但不限于表示该GPU在某一时间段内最高的GPU占用率不得高于百分之五十。

[0057] 此外以目标资源包括静态资源以及动态资源为例说明,假设目标应用运行所需的静态资源为80,所需的动态资源为50,则可以但不限于需GPU剩余的算力值大于等于80,且GPU占用率小于等于50。或者说,在应用服务器的全部可分配图形处理器中,按照“GPU剩余的算力值大于等于80,且GPU占用率小于等于50”这一条件进行筛选,以并将满足该筛选条件的GPU分配给目标应用。可选地,在满足筛选条件的GPU数量大于等于2的情况下,以GPU剩余的算力值和/或GPU占用率进行优先级排序,以确定最佳的GPU分配方案,例如第一GPU的剩余算力值为80,第二GPU的剩余算力值为90,虽然二者皆满足算力值大于等于80的筛选条件,但第二GPU的剩余算力值更大,可以但不限于视为其优先级更高(或在资源紧张的情况下,可以但不限于设置GPU的剩余算力值越大其优先级越低,GPU占用率同理,且GPU的剩余算力值的优先级可以但不限于高于GPU占用率)。

[0058] 可选地,在本实施例中,图形处理器的资源提供状态可以但不限于随着图形处理器的资源分配和占用而随之动态变化,例如在图形处理器分配给多个正在运行目标应用的账户情况下,则该资源提供状态将发生上扬,用于表示图形处理器在当前时刻允许提供的目标资源减少。

[0059] 进一步图形处理器的资源提供状态还可以但不限于为以图形处理器可提供的目

标资源(静态资源和/或动态资源),例如在目标资源包括静态资源的情况下,图形处理器的资源提供状态可以但不限于为剩余可分配给目标应用的算力值;再例如在目标资源包括动态资源的情况下,图形处理器的资源提供状态可以但不限于为可因目标应用提升的GPU占用率。

[0060] 可选地,在本实施例中,上述图形处理器的资源分配方法可以但不限于应用在请求目标应用运行的过程中,例如目标应用的运行场景如图3所示,在目标应用关联的登录界面302检测到用于使用目标账号304运行目标应用的登录请求;进一步,响应该登录请求,获取目标应用关联的参考资源信息以及的历史运行数据,并由应用服务器根据历史运行数据将参考资源信息处理为目标资源信息;再者,基于目标资源信息以及应用服务器的每个图形处理器的资源提供状态,为目标应用选配最佳的图形处理器,即目标图形处理器。

[0061] 可选地,在本实施例中,可以但不限于利用实际运营的数据,由工作人员在经验中总结,设置动态调整参数,以完成对参考资源信息的调整,并在长期运营的过程中不断优化该动态调整参数。此外,还可以但不限于利用训练好的模型,将目标应用的历史运行数据作为该模型的输入,以及将用于调整参考资源信息的调整数据作为该模型的输出,从而完成对参考资源信息的调整。可选地,模型可以但不限于包括以下至少之一:LR模型、决策树模型、随机森林模型、梯度提升模型、多层感知机模型、朴素贝叶斯模型等。

[0062] 进一步以模型为例说明,如图4所示,把各种资源的实时上报作为机器学习的训练数据,导入到实时训练服务器群集(如服务器408、服务器412、服务器414、服务器416等)中,训练服务器群集通过接入节点404、服务器406、服务器416获取服务器机房、GPU当前利用率、CPU利用率、出口带宽、玩家数量、卡顿情况、游戏资源等多个维度的数据,并训练出一个相对简单,且可做到准实时模型更新的预测模型,例如LR模型。

[0063] 随着资源和环境的变化,预测模型会随时变更,其中的多项权重也在动态变化。预测模型更新后,会同步给模型推理服务。服务器414根据用户的请求参数,对资源的计算方式进行预测。并把预测的结果返回给服务器416。服务器416从预测结果中获得精确的权重值,然后在资源存储的资源表格410中进行搜索,并最终锁定合适的资源,返回给用户。

[0064] 最后,在极端情况下,可能会因为预测的精度问题,而产生对某个GPU分配用户过多的情况。通常的情况下,用户的平均GPU占用都不高,多个用户的并行使用不会发生问题。极端情况是两个或者以上的用户都处于游戏的高GPU消耗阶段,这个时候可能会出现GPU资源不足的情况。

[0065] 以目标账户402为例说明,目标账户402将因GPU资源不足而导致的卡顿情况上报给服务器408,并由服务器408汇总资源以及建立索引,以建立资源表格410,再根据卡顿情况获取卡顿数据,将该卡顿数据发送至服务器412,以更新模型;再由服务器414获取计算权重。

[0066] 需要说明的是,通过在为目标应用匹配的参考资源信息的基础上,根据目标应用的历史运行数据调整该参考资源信息,以获得与目标应用的运行情况之间契合度更高的目标资源信息,并通过契合度更高的目标资源值,为该目标应用分配更加合理的图形处理器。可选的,调整参考资源信息可以但不限于上调/下调参考资源信息,获取的目标资源信息对应的资源数据可以但不限于高于/低于参考资源信息。

[0067] 进一步举例说明,可选的例如获取的历史运行数据表示在按照参考资源信息运行

的目标应用经常出现卡顿情况,且假设该卡顿情况发生的原因是因为分配的GPU资源不足。基于此,可以但不限于上调参考资源信息,获取用于指示分配更多资源以克服卡顿情况的目标资源信息。

[0068] 再举例说明,可选的例如获取的历史运行数据表示在按照参考资源信息运行目标游戏应用时,GPU的实际占用率并未达到预期,例如图5所示的历史运行数据502,假设该历史运行数据502用于表示数据中心的所有GPU的实际资源使用情况的变化曲线,其中,横轴为时刻,纵轴为GPU实际占用率;

[0069] 进一步,通过统计图可以发现:每天的第一个用户使用高峰出现在12:50左右,每天的第二个用户使用高峰出现在21:20左右,GPU的实际使用情况的计算方法为:每15秒(每分钟采集四次)收集当前的GPU占用率,并且上报到数据服务器。该服务器汇总每分钟内的所有上报的占用率,选择占用率最高的一次数据上报作为当前分钟的占用率,然后,对整个数据中心的GPU占用情况取平均值,从而得到每分钟GPU实际占用率的曲线图,如图5所示。

[0070] 概括而言,统计每分钟上报最大的GPU占用率,每个GPU每天有1440条数据。累加每天每分钟的GPU占用率,除以GPU总数(取平均),得到一条曲线(即图5中所示曲线)。取曲线的最高点(平均所有GPU都最忙的某一分钟)。在这一分钟里,把GPU的占用率分成十个区间,统计每个区间内的GPU数量。

[0071] 再者,对最高峰21:20分的GPU占用情况进行分段统计,可以得到下述表(1)所示的统计结果:

[0072]

	高峰期		距高峰期15分钟内		距高峰期30分钟内		距高峰期60分钟内	
动态占用率分段	GPU卡数	GPU卡数占比	GPU卡数	GPU卡数占比	GPU卡数	GPU卡数占比	GPU卡数	GPU卡数占比
0-10	2	0.55%	1	0.28%	0	0%	0	0%
10-20	37	10.19%	19	5.23%	14	3.86%	10	2.75%
20-30	46	12.67%	42	11.57%	39	10.74%	21	5.79%
30-40	124	34.16%	127	34.99%	110	30.30%	104	28.65%
40-50	65	17.91%	64	17.63%	75	20.66%	67	18.46%
50-60	31	8.54%	36	9.92%	46	12.67%	58	15.98%
60-70	35	9.64%	32	8.82%	29	7.99%	39	10.74%
70-80	12	3.31%	22	6.06%	23	6.34%	28	7.71%
80-90	9	2.48%	12	3.31%	20	5.51%	20	5.51%
90-100	2	0.55%	8	2.20%	7	1.93%	16	4.41%

[0073] 表(1)

[0074] 其中,上述表(1)中的GPU卡可以但不限于为显卡,GPU卡数占比可以但不限于为动态占用率分段内的GPU卡数占GPU总卡数的比例。

[0075] 根据上述表(1)所示的统计数据可知:

[0076] 1、根据游戏所需的最大资源来进行分配,整体的GPU占用率并不高,存在浪费;

[0077] 2、在用户高峰期,75.48%的GPU的占用率都未超过50%;GPU占用率未超过50%,意味着负载翻倍,也仍然能够支持。

[0078] 3、瞬时的GPU占用率不代表一直以来的GPU占用率,因此选择一个时间窗进行统计:

[0079] 4、分别在15分钟,30分钟,60分钟三个时间窗口内,选择时间段内的GPU占用最大值来进行对比;

[0080] 5、选择最大的时间窗口1小时来计算:仍然有55.65%的GPU占用率小于50%

[0081] 6、在用户平均游戏时间不足一小时的情况下,假定所有GPU一小时内占用率小于50%的GPU卡上可以至少再分配进去一个用户,那么相比用户最高峰的PCU而言,整体的PCU提升至至少能提升25.4%。

[0082] 概括而言,在有历史运行数据指导的情况下,可以但不限于结合目标游戏应用所需的最大目标资源和GPU实际占用的目标资源,让每块GPU上都有合适的用户数,最终在相同资源的情况下,提升业务的PCU。

[0083] 通过本申请提供的实施例,获取目标应用关联的参考资源信息,其中,目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,参考资源信息为运行目标应用所需目标资源的参考信息;根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,其中,历史运行数据为目标应用在运行时所上报的历史数据;获取应用服务器的每个图形处理器的资源提供状态,其中,资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源;按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,其中,目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源,通过在为目标应用匹配的参考资源信息的基础上,根据目标应用的历史运行数据调整该参考资源信息,以获得与目标应用的运行情况之间契合度更高的目标资源信息,并通过契合度更高的目标资源值,为该目标应用分配更加合理的图形处理器,进而达到了对图形处理器的资源进行合理分配的目的,从而实现了提高图形处理器的资源利用率的技术效果。

[0084] 作为一种可选的方案,根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,包括:

[0085] S1,在参考资源信息中携带有静态资源值的情况下,根据第一运行数据调整静态资源值,以获取目标资源值,其中,静态资源值为图形处理器在目标应用运行时所需提供的资源值,目标资源包括静态资源值,历史运行数据包括第一运行数据;

[0086] S2,在参考资源信息中携带有动态占用率的情况下,根据第二运行数据调整动态占用率,以获取目标占用率,其中,动态占用率为图形处理器在目标应用运行时所提升的资源占用率,目标资源包括动态占用率,历史运行数据包括第二运行数据。

[0087] 可选地,在本实施例中,目标资源值可以但不限于为GPU的静态算力值,其中,该GPU的静态算力值可以但不限于是根据测试人员试玩各种后评估出来的一个整数。例如,根据云服务器的主流GPU类型和主流的匹配程度,将主流GPU的最大算力值设定为100,更昂滚的型号约为130算力值,旧一些型号的算力值为80。根据每个的最大资源占用率,及其在主流GPU上的并行使用情况,计算出所需的算力值,例如大型的3D所需算力值为80,而小型的RGP的算力值为20。

[0088] 可选地,在本实施例中,目标占用率可以但不限于为GPU占用率,以目标应用为例说明,用户如果玩目标应用,至少会对GPU造成的占用率提升即为该GPU占用率。通常情况下,GPU的静态算力值与GPU占用率之间具有关联关系,例如目标GPU的静态算力总值为100,已分配出去的算力值为80,剩余算力值为20,则理论上目标GPU的占用率为80% (80/100),但实际上目标GPU已分配出去的算力值不一定全部占用,其可能存在空闲的情况,例如已分配出去的80算力值当前实际只占用50,则目标GPU的当前占用率应为50% (50/100)。此外,还考虑到一些GPU占用异常的情况出现,例如目标GPU已被占用的算力值为50,其实际占用率应为50%,但因占用异常,使得目标GPU的当前占用率为80%。

[0089] 需要说明的是,参考资源信息中可以但不限于携带有一种或多种类型的目标资源,如静态资源值、动态占用率等。根据参考资源信息中资源的不同类型,以获取对应的运行数据,从而灵活调整参考资源信息,以获得针对程度更高的目标资源信息。

[0090] 进一步举例说明,可选的例如图6所示,根据历史运行数据602调整参考资源信息604,以获取目标资源信息606,具体的,根据第一运行数据6022调整静态资源值6042,以获取目标资源值6062;根据第二运行数据6024调整动态占用率6044,以获取目标占用率6064。

[0091] 通过本申请提供的实施例,在参考资源信息中携带有静态资源值的情况下,根据第一运行数据调整静态资源值,以获取目标资源值,其中,静态资源值为图形处理器在目标应用运行时所需提供的资源值,目标资源包括静态资源值,历史运行数据包括第一运行数据;在参考资源信息中携带有动态占用率的情况下,根据第二运行数据调整动态占用率,以获取目标占用率,其中,动态占用率为图形处理器在目标应用运行时所提升的资源占用率,目标资源包括动态占用率,历史运行数据包括第二运行数据,达到了灵活调整参考资源信息的目的,实现了提高参考资源信息的调整灵活度的效果。

[0092] 作为一种可选的方案,按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,包括:

[0093] S1,在目标资源信息中携带有目标资源值的情况下,按照目标资源值以及资源提供状态对应的第一资源配置,为目标应用分配第一图形处理器,其中,第一资源配置用于表示图形处理器在当前时刻允许提供的资源值,第一图形处理器允许提供的资源值大于或等于目标资源值;

[0094] S2,在目标资源信息中携带有目标占用率的情况下,按照目标资源值以及资源提供状态对应的第二资源配置,为目标应用分配第二图形处理器,其中,第二资源配置用于表示图形处理器当前时刻允许提升的资源占用率,第二图形处理器允许提升的资源占用率大于或等于目标占用率;或,

[0095] S3,在目标资源信息中携带有目标资源值以及目标占用率的情况下,按照目标资源值、目标占用率、第一资源配置以及第二资源配置,为目标应用分配第三图形处理器,其中,第三图形处理器允许提升的资源占用率大于或等于目标占用率,以及第三图形处理器允许提供的资源值大于或等于目标资源值。

[0096] 可选地,在本实施例中,可以但不限于将应用服务器的每个图形处理器的资源提供状态整理成一张带有索引功能的表格,进而在获取到目标资源值和/或目标占用率的情况下,可以但不限于将目标资源值和/或目标占用率作为索引条件,以在上述表格中查询出满足索引条件的目标图形处理器。

[0097] 进一步以下述表(2)中的GPU资源配置为例说明,可选地假设在目标资源信息中获取到为80的目标资源值,则筛选下述表(2)中剩余GPU静态算力(资源)值大于等于80的GPU作为目标图形处理器的候选处理器,如服务器“Ip:192.168.0.2Numa:0”对应的GPU;同理,假设在在目标资源信息中获取到为80的目标占用率,则筛选GPU资源配置中可提升动态占用率大于等于80的GPU作为目标图形处理器的候选处理器,如服务器“Ip:192.168.0.3Numa:1”对应的GPU。此外,还可以但不限于加入更多筛选条件,如CPU核数、软件版本、剩余宽带等,具体如下述表(2)所示。可选地,可以但不限于灵活搭配不同的筛选条件,如筛选剩余GPU静态算力(资源)值大于等于60,以及GPU资源配置中可提升动态占用率大于等于48的GPU作为目标图形处理器的候选处理器,如服务器“Ip:192.168.0.3Numa:0”对应的GPU。

[0098]	服务器	剩余 GPU 静态 算力值	GPU 实际利 用率	CPU 核 数	软件版 本	剩 余 带 宽
	Ip:192.168.0.2 Numa:0	100	1%	32	10.1.2	50gbps
	Ip:192.168.0.2 Numa:1	80	10%	20	10.1.3	48gbps
	Ip:192.168.0.3 Numa:0	60	48%	6	10.1.2	20gbps
	Ip:192.168.0.3 Numa:1	20	80%	9	10.1.2	30gbps

[0099] 表 (2)

[0100] 需要说明的是,在目标资源信息中携带有目标资源值以及目标占用率的情况下,可以但不限于将目标资源值、目标占用率共同作为筛选条件,并基于该筛选条件在第一资源配置以及第二资源配置中进行筛选,以获得满足该筛选条件的第三图形处理器。

[0101] 通过本申请提供的实施例,按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,包括以下至少之一:在目标资源信息中携带有目标资源值的情况下,按照目标资源值以及资源提供状态对应的第一资源配置,为目标应用分配第一图形处理器;在目标资源信息中携带有目标占用率的情况下,按照目标资源值以及资源提供状态对应的第二资源配置,为目标应用分配第二图形处理器;在目标资源信息中携带有目标资源值以及目标占用率的情况下,按照目标资源值、目标占用率、第一资源配置以及第二资源配置,为目标应用分配第三图形处理器,达到了准确为目标应用分配图形处理器的目的,实现了提高图形处理器的分配准确性的效果。

[0102] 作为一种可选的方案,在按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器之后,包括:

[0103] S1,统计目标图形处理器在目标时间段内的N个实际占用率,其中,N为大于等于1的正整数,目标时间段包括当前时刻;

[0104] S2,根据N个实际占用率调整目标图形处理器的第二资源配置。

[0105] 可选地,在本实施例中,目标时间段可以但不限于包括当前时刻之前的M个时刻,也可以但不限于包括当前时刻之后的P个时刻,其中,M、P皆为小于等于N的正整数。

[0106] 需要说明的是,考虑到图形处理器的实际占用率更能表现出图形处理器的分配情况,进而通过统计目标图形处理器在目标时间段内的N个实际占用率,确定出调整目标图形处理器的第二资源配置的方式。

[0107] 进一步举例说明,可选的例如目标GPU当前的第一资源配置为100,第二资源配置为0%,进而在为目标应用分配出为80的静态资源值的情况下,目标GPU的第一资源配置变为20,目标GPU的第二资源配置也应随之变化为80%。但在实际运行中,第一资源配置在无新的资源调整的情况下,通常是不会发生变化的,即为静态;而第二资源配置即使是在无新的资源调整的情况下,仍可发生变化,即为动态,例如在目标GPU已分配出80的静态资源值的情况下,统计当前时刻在内的目标时间段内的N个实际占用率,并对该N个实际占用率进行整合计算,例如平均计算、方差计算、取最大值计算等,以获取候选占用率,并将该候选占用率作为目标图形处理器新的第二资源配置。

[0108] 通过本申请提供的实施例,统计目标图形处理器在目标时间段内的N个实际占用率,其中,N为大于等于1的正整数,目标时间段包括当前时刻;根据N个实际占用率调整目标图形处理器的第二资源配置,达到了及时且准确地调整目标图形处理器的第二资源配置的目的,实现了提高图形处理器的资源配置的调整及时性的效果。

[0109] 作为一种可选的方案,根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,包括:

[0110] S1,根据历史运行数据获取目标权重值;

[0111] S2,整合计算目标权重值与参考资源信息,以获取目标资源信息。

[0112] 需要说明的是,在目标权重值小于1大于0的情况下,可以但不限于下调参考资源信息对应的目标资源,或者说目标资源信息对应的目标资源小于参考资源信息对应的目标资源;同理,在目标权重值大于1的情况下,可以但不限于上调参考资源信息对应的目标资源,或者说目标资源信息对应的目标资源大于参考资源信息对应的目标资源。

[0113] 进一步举例说明,可选的例如假设目标权重值为0.7(70%),且参考资源信息对应的资源值为80,则整合计算后的目标资源信息对应的资源值为56($80 \times 70\%$);再假设目标权重值为0.6(60%),且参考资源信息对应的占用率为80,则整合计算后的目标资源信息对应的占用率为48($80 \times 60\%$)。

[0114] 再举例说明,可选的例如假设目标权重值为1.1(110%),且参考资源信息对应的资源值为80,则整合计算后的目标资源信息对应的资源值为88($80 \times 110\%$)。

[0115] 通过本申请提供的实施例,根据历史运行数据获取目标权重值;整合计算目标权重值与参考资源信息,以获取目标资源信息,达到了利用参考资源信息从而高效计算目标资源信息的目的,实现了提高目标资源信息的获取效率的效果。

[0116] 作为一种可选的方案,根据历史运行数据获取目标权重值,包括:

[0117] S1,将历史运行数据输入线性回归模型中,其中,线性回归模型使用具有线性关系的样本运行数据和样本权重值分别对初始回归模型进行训练后得到的模型;

[0118] S2,获取线性回归模型输出的目标权重值。

[0119] 需要说明的是,将历史运行数据输入线性回归模型中,其中,线性回归模型使用具有线性关系的样本运行数据和样本权重值分别对初始回归模型进行训练后得到的模型;获取线性回归模型输出的目标权重值。可选地,由于LR模型相对简单,因此要做到准实时模型更新也变得容易。可选地,可以但不限于利用把一个用户的一条日志作为一次模型训练的输入。

[0120] 进一步举例说明,可选的例如将LR模型理解成一个线性方程,如果只有一个特征:

也就是 $y=ax+b$,如果有两个特征也就是 $y=ax_1+bx_2+c$,结合历史运行数据,假设将该线性方程应用在云游戏领域,则该线性方程可以但不限于为权重值=游戏* w_1 +区域* w_2 +运营商* w_3 +卡顿率* w_4 等等,并灵活调整,例如在卡顿率越高则提高对应的权重分配,服务器机房越差、出口带宽越窄,则提高对应的权重分配等。

[0121] 通过本申请提供的实施例,将历史运行数据输入线性回归模型中,其中,线性回归模型使用具有线性关系的样本运行数据和样本权重值分别对初始回归模型进行训练后得到的模型;获取线性回归模型输出的目标权重值,达到了快速且准确地获取目标权重值的目的,实现了提高目标权重值的获取效率的效果。

[0122] 作为一种可选的方案,在将历史运行数据输入线性回归模型中之前,包括:

[0123] 在获取到目标反馈数据的情况下,更新线性回归模型,其中,目标反馈数据为目标应用在运行过程中,由于图形处理器提供的目标资源小于或等于目标运行时实际所需的目标资源而产生的反馈数据。

[0124] 可选地,在本实施例中,目标反馈数据可以但不限于为目标应用在运行过程中,由于图形处理器提供的目标资源小于或等于目标运行时实际所需的目标资源而产生的反馈数据,除异常反馈数据的自动上报外,也可以是用户主动上报的反馈数据;

[0125] 进一步以用户主动上报的反馈数据为例说明,例如图7所示,在用户使用的客户端中,检测到由虚拟按键“卡顿上报”触发的反馈请求,该反馈请求用于反馈目标游戏应用702存在卡顿情况。

[0126] 需要说明的是,GPU资源不足必然导致用户的卡顿,用户发生卡顿会立即上报卡顿的情况。云游戏服务器会根据卡顿的情况动态调整视频的压缩率及其帧率,等到GPU资源不再紧张后再恢复到正常;因此对于用户而言,云游戏服务器GPU资源不足会在某几秒中出现画质降低和帧率降低,但是正常的游戏进程不会造成影响;

[0127] 配合GPU虚拟化技术,如果一个用户超过一定的时间仍然发生GPU资源不足,云游戏服务器会把玩家的分配的GPU换到另一张空闲的GPU上;

[0128] 最坏的情况,玩家不满意某几分钟内的游戏画质。这时玩家可以退出重新进入,由资源分配服务在资源充裕的服务器上重新分配资源。同是,产生的卡顿数据必然影响实时训练的LR模型的某些参数的权重,使得后续的资源分配方式变得宽裕一些,避免了连续出现资源不足的情况。

[0129] 通过本申请提供的实施例,在获取到目标反馈数据的情况下,更新线性回归模型,其中,目标反馈数据为目标应用在运行过程中,由于图形处理器提供的目标资源小于或等于目标运行时实际所需的目标资源而产生的反馈数据,达到了根据及时反馈的数据完成对线性回归模型的更新的目的,实现了提高线性回归模型的更新及时性的效果。

[0130] 需要说明的是,对于前述的各方法实施例,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,本发明并不受所描述的动作顺序的限制,因为依据本发明,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发明所必须的。

[0131] 根据本发明实施例的另一个方面,还提供了一种用于实施上述图形处理器的资源分配方法的图形处理器的资源分配装置。如图8所示,该装置包括:

[0132] 第一获取单元802,用于获取目标应用关联的参考资源信息,其中,目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,参考资源信息为运行目标应用所需目标资源的参考信息;

[0133] 调整单元804,用于根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,其中,历史运行数据为目标应用在运行时所上报的历史数据;

[0134] 第二获取单元806,用于获取应用服务器的每个图形处理器的资源提供状态,其中,资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源;

[0135] 分配单元808,用于按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,其中,目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源。

[0136] 可选地,在本实施例中,图形处理器的资源分配方法可以但不限于应用在云的GPU资源分配场景下,例如假设目标应用为目标应用,应用服务器为云服务器,则先在将目标应用部署在云服务器之前,对其进行充分的测试,以获取在各种场景下目标应用的运行对GPU资源的消耗情况,然后选择对GPU的最大消耗作为该目标应用关联的参考资源信息。进一步,在获取到目标应用关联的参考资源信息,且有目标应用在实际运行中的历史运行数据指导的情况下,调整参考资源信息,以获取与目标应用的实际运行更加契合的目标资源信息,再结合应用服务器的每个图形处理器的资源提供状态,让每块GPU上都有合适的用户数,最终相比于传统的资源分配,在相同GPU资源的情况下,可以提高GPU资源的利用率,进而提升业务的PCU。

[0137] 可选地,在本实施例中,目标应用可以但不限于带有显示功能,且在应用服务器中运行且通过终端设备呈现画面,例如目标应用可以但不限于为视频应用、应用、模拟器应用、漫画应用等。

[0138] 可选地,在本实施例中,参考资源信息可以但不限于为在将目标应用部署在应用服务器之前,工作人员通过产品测试而为目标应用的运行分配的资源信息,例如按照测试中目标应用所需要的最大目标资源来初步分配目标应用在实际运行所需的目标资源。

[0139] 可选地,在本实施例中,目标资源信息可以但不限于为与参考资源信息处于同一维度的资源信息,二者的区别可以但不限在于参考资源信息可以但不限于表示目标应用在实际运行所需的最大目标资源,而目标资源信息可以但不限于表示目标应用在实际运行所需的实际目标资源。可选地,由于目标应用在实际运行时只有很少情况下才会需要该最大目标资源,而大多数情况下所需的应为实际目标资源,进而使得该实际目标资源往往小于最大目标资源。

[0140] 可选地,在本实施例中,历史运行数据可以但不限于包括以下至少之一:动态运行数据、静态运行数据,其中,动态运行数据可以但不限于为在目标应用运行过程中发生实时变化的运行数据,例如GPU占用率、GPU空闲率、用户在线人数、剩余GPU资源、运行卡顿数据等;静态运行数据可以但不限于为在目标应用运行过程中一般不发生变化的运行数据,例如应用服务器所关联机房的性能参数、出口宽带等。可选地,历史运行数据可以但不限于通过目标应用运行过程中所上报的日志获取,也可以但不限于通过检测目标应用的运行情况获取,在此不做限定。

[0141] 可选地,在本实施例中,目标资源可以但不限于用于衡量目标应用运行时所需一种

或多种维度的资源的量级指标。可选地,目标资源可以但不限于包括以下至少之一:静态资源、动态资源。其中,静态资源可以但不限于为GPU所能提供的固定资源值的静态衡量指标;动态资源可以但不限于为GPU所能提供的固定资源值的动态衡量指标。

[0142] 进一步以目标资源包括静态资源为例说明,可以但不限于预先将应用服务器的每块GPU的性能属性通过某种手段处理为更为直观的算力值,例如获取目标性能属性的第一GPU作为衡量,并假设该第一GPU可提供的最大算力值为100,再将该第一GPU分配给选取样本应用以运行,并获取该样本应用运行时该第一GPU的资源占用情况,如果该样本应用运行时该第一GPU的资源占用率为百分之五十,则可定义该样本应用运行时所需分配的算力值为50 ($100 \times 50\% = 50$);基于此,再选取另一不同性能属性的第二GPU,并将该第二GPU分配给上述样本应用,进而在该样本应用运行过程中,获取第二GPU的资源占用情况,如果该样本应用运行时该第二GPU的资源占用率为百分之百,则可定义该第二GPU可提供的最大算力值为50 ($50/100\% = 50$);同理,还可选取另一不同性能属性的第三GPU,如果该样本应用运行时该第二GPU的资源占用率为百分之二十,则可定义该第三GPU可提供的最大算力值为250 ($50/20\% = 250$)。

[0143] 再者以目标资源包括动态资源为例说明,可以但不限于用于表示GPU占用率,具体的,动态资源可以但不限于包括以下至少之一:在某一时刻的GPU实际的占用率、某一时间段内的全部GPU占用数据、某一段时间内GPU最高/低的占用率等。例如,在第一时刻,目标应用运行所需的动态资源为50,可以但不限于表示需为目标应用选取GPU占用率低于百分之五十的GPU,而GPU占用率低于百分之五十的GPU可以但不限于表示该GPU在某一时间段内最高的GPU占用率不得高于百分之五十。

[0144] 此外以目标资源包括静态资源以及动态资源为例说明,假设目标应用运行所需的静态资源为80,所需的动态资源为50,则可以但不限于需GPU剩余的算力值大于等于80,且GPU占用率小于等于50。或者说,在应用服务器的全部可分配图形处理器中,按照“GPU剩余的算力值大于等于80,且GPU占用率小于等于50”这一条件进行筛选,以并将满足该筛选条件的GPU分配给目标应用。可选地,在满足筛选条件的GPU数量大于等于2的情况下,以GPU剩余的算力值和/或GPU占用率进行优先级排序,以确定最佳的GPU分配方案,例如第一GPU的剩余算力值为80,第二GPU的剩余算力值为90,虽然二者皆满足算力值大于等于80的筛选条件,但第二GPU的剩余算力值更大,可以但不限于视为其优先级更高(或在资源紧张的情况下,可以但不限于设置GPU的剩余算力值越大其优先级越低,GPU占用率同理,且GPU的剩余算力值的优先级可以但不限于高于GPU占用率)。

[0145] 可选地,在本实施例中,图形处理器的资源提供状态可以但不限于随着图形处理器的资源分配和占用而随之动态变化,例如在图形处理器分配给多个正在运行目标应用的账户情况下,则该资源提供状态将发生上扬,用于表示图形处理器在当前时刻允许提供的目标资源减少。

[0146] 进一步图形处理器的资源提供状态还可以但不限于为以图形处理器可提供的目标资源(静态资源和/或动态资源),例如在目标资源包括静态资源的情况下,图形处理器的资源提供状态可以但不限于为剩余可分配给目标应用的算力值;再例如在目标资源包括动态资源的情况下,图形处理器的资源提供状态可以但不限于为可因目标应用提升的GPU占用率。

[0147] 可选地,在本实施例中,上述图形处理器的资源分配方法可以但不限于应用在请求目标应用运行的过程中,例如目标应用的运行场景如图3所示,在目标应用关联的登录界面302检测到用于使用目标账号304运行目标应用的登录请求;进一步,响应该登录请求,获取目标应用关联的参考资源信息以及的历史运行数据,并由应用服务器根据历史运行数据将参考资源信息处理为目标资源信息;再者,基于目标资源信息以及应用服务器的每个图形处理器的资源提供状态,为目标应用选配最佳的图形处理器,即目标图形处理器。

[0148] 可选地,在本实施例中,可以但不限于利用实际运营的数据,由工作人员在经验中总结,设置动态调整参数,以完成对参考资源信息的调整,并在长期运营的过程中不断优化该动态调整参数。此外,还可以但不限于利用训练好的模型,将目标应用的历史运行数据作为该模型的输入,以及将用于调整参考资源信息的调整数据作为该模型的输出,从而完成对参考资源信息的调整。可选地,模型可以但不限于包括以下至少之一:LR模型、决策树模型、随机森林模型、梯度提升模型、多层感知机模型、朴素贝叶斯模型等。

[0149] 需要说明的是,通过在为目标应用匹配的参考资源信息的基础上,根据目标应用的历史运行数据调整该参考资源信息,以获得与目标应用的运行情况之间契合度更高的目标资源信息,并通过契合度更高的目标资源值,为该目标应用分配更加合理的图形处理器。可选的,调整参考资源信息可以但不限于上调/下调参考资源信息,获取的目标资源信息对应的资源数据可以但不限于高于/低于参考资源信息。

[0150] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此不再赘述。

[0151] 通过本申请提供的实施例,获取目标应用关联的参考资源信息,其中,目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,参考资源信息为运行目标应用所需目标资源的参考信息;根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,其中,历史运行数据为目标应用在运行时所上报的历史数据;获取应用服务器的每个图形处理器的资源提供状态,其中,资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源;按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,其中,目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源,通过在为目标应用匹配的参考资源信息的基础上,根据目标应用的历史运行数据调整该参考资源信息,以获得与目标应用的运行情况之间契合度更高的目标资源信息,并通过契合度更高的目标资源值,为该目标应用分配更加合理的图形处理器,进而达到了对图形处理器的资源进行合理分配的目的,从而实现了提高图形处理器的资源利用率的技术效果。

[0152] 作为一种可选的方案,例如图9所示,调整单元804,包括:

[0153] 第一调整模块902,用于在参考资源信息中携带有静态资源值的情况下,根据第一运行数据调整静态资源值,以获取目标资源值,其中,静态资源值为图形处理器在目标应用运行时所需提供的资源值,目标资源包括静态资源值,历史运行数据包括第一运行数据;

[0154] 第二调整模块904,用于在参考资源信息中携带有动态占用率的情况下,根据第二运行数据调整动态占用率,以获取目标占用率,其中,动态占用率为图形处理器在目标应用运行时所提升的资源占用率,目标资源包括动态占用率,历史运行数据包括第二运行数据。

[0155] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此不再赘述。

[0156] 作为一种可选的方案,分配单元808,包括:

[0157] 第一分配模块,用于在目标资源信息中携带有目标资源值的情况下,按照目标资源值以及资源提供状态对应的第一资源配置,为目标应用分配第一图形处理器,其中,第一资源配置用于表示图形处理器在当前时刻允许提供的资源值,第一图形处理器允许提供的资源值大于或等于目标资源值;

[0158] 第二分配模块,用于在目标资源信息中携带有目标占用率的情况下,按照目标资源值以及资源提供状态对应的第二资源配置,为目标应用分配第二图形处理器,其中,第二资源配置用于表示图形处理器当前时刻允许提升的资源占用率,第二图形处理器允许提升的资源占用率大于或等于目标占用率;或,

[0159] 第三分配模块,用于在目标资源信息中携带有目标资源值以及目标占用率的情况下,按照目标资源值、目标占用率、第一资源配置以及第二资源配置,为目标应用分配第三图形处理器,其中,第三图形处理器允许提升的资源占用率大于或等于目标占用率,以及第三图形处理器允许提供的资源值大于或等于目标资源值。

[0160] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此不再赘述。

[0161] 作为一种可选的方案,包括:

[0162] 统计模块,用于在按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器之后,统计目标图形处理器在目标时间段内的N个实际占用率,其中,N为大于等于1的正整数,目标时间段包括当前时刻;

[0163] 第三调整模块,用于在按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器之后,根据N个实际占用率调整目标图形处理器的第二资源配置。

[0164] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此不再赘述。

[0165] 作为一种可选的方案,调整单元804,包括:

[0166] 获取模块,用于根据历史运行数据获取目标权重值;

[0167] 计算模块,用于整合计算目标权重值与参考资源信息,以获取目标资源信息。

[0168] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此不再赘述。

[0169] 作为一种可选的方案,获取模块,包括:

[0170] 输入子模块,用于将历史运行数据输入线性回归模型中,其中,线性回归模型使用具有线性关系的样本运行数据和样本权重值分别对初始回归模型进行训练后得到的模型;

[0171] 获取子模块,用于获取线性回归模型输出的目标权重值。

[0172] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此不再赘述。

[0173] 作为一种可选的方案,包括:

[0174] 更新模块,用于在将历史运行数据输入线性回归模型中之前,在获取到目标反馈数据的情况下,更新线性回归模型,其中,目标反馈数据为目标应用在运行过程中,由于图形处理器提供的目标资源小于或等于目标运行时实际所需的资源而产生的反馈数据。

[0175] 具体实施例可以参考上述图形处理器的资源分配方法中所示示例,本示例中在此

不再赘述。

[0176] 根据本发明实施例的又一个方面,还提供了一种用于实施上述图形处理器的资源分配方法的电子设备,如图10所示,该电子设备包括存储器1002和处理器1004,该存储器1002中存储有计算机程序,该处理器1004被设置为通过计算机程序执行上述任一项方法实施例中的步骤。

[0177] 可选地,在本实施例中,上述电子设备可以位于计算机网络的多个网络设备中的至少一个网络设备。

[0178] 可选地,在本实施例中,上述处理器可以被设置为通过计算机程序执行以下步骤:

[0179] S1,获取目标应用关联的参考资源信息,其中,目标应用为在应用服务器中运行且通过终端设备呈现画面的应用,参考资源信息为运行目标应用所需目标资源的参考信息;

[0180] S2,根据目标应用的历史运行数据调整参考资源信息,以获取目标资源信息,其中,历史运行数据为目标应用在运行时所上报的历史数据;

[0181] S3,获取应用服务器的每个图形处理器的资源提供状态,其中,资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源;

[0182] S4,按照目标资源信息以及资源提供状态,为目标应用分配目标图形处理器,其中,目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源。

[0183] 可选地,本领域普通技术人员可以理解,图10所示的结构仅为示意,电子设备也可以是智能手机(如Android手机、iOS手机等)、平板电脑、掌上电脑以及移动互联网设备(Mobile Internet Devices,MID)、PAD等终端设备。图10其并不对上述电子设备的结构造成限定。例如,电子设备还可包括比图10中所示更多或者更少的组件(如网络接口等),或者具有与图10所示不同的配置。

[0184] 其中,存储器1002可用于存储软件程序以及模块,如本发明实施例中的图形处理器的资源分配方法和装置对应的程序指令/模块,处理器1004通过运行存储在存储器1002内的软件程序以及模块,从而执行各种功能应用以及数据处理,即实现上述的图形处理器的资源分配方法。存储器1002可包括高速随机存储器,还可以包括非易失性存储器,如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中,存储器1002可进一步包括相对于处理器1004远程设置的存储器,这些远程存储器可以通过网络连接至终端。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。其中,存储器1002具体可以但不限于用于存储参考资源信息、历史运行数据、资源提供状态、目标资源信息以及目标图形处理器等信息。作为一种示例,如图10所示,上述存储器1002中可以但不限于包括上述图形处理器的资源分配装置中的第一获取单元802、调整单元804、第二获取单元806及分配单元808。此外,还可以包括但不限于上述图形处理器的资源分配装置中的其他模块单元,本示例中不再赘述。

[0185] 可选地,上述的传输装置1006用于经由一个网络接收或者发送数据。上述的网络具体实例可包括有线网络及无线网络。在一个实例中,传输装置1006包括一个网络适配器(Network Interface Controller,NIC),其可通过网线与其他网络设备与路由器相连从而可与互联网或局域网进行通讯。在一个实例中,传输装置1006为射频(Radio Frequency, RF)模块,其用于通过无线方式与互联网进行通讯。

[0186] 此外,上述电子设备还包括:显示器1008,用于显示上述参考资源信息、历史运行

数据、资源提供状态、目标资源信息以及目标图形处理器等信息；和连接总线1010，用于连接上述电子设备中的各个模块部件。

[0187] 在其他实施例中，上述终端设备或者服务器可以是一个分布式系统中的一个节点，其中，该分布式系统可以为区块链系统，该区块链系统可以是由该多个节点通过网络通信的形式连接形成的分布式系统。其中，节点之间可以组成点对点 (Peer To Peer, 简称 P2P) 网络，任意形式的计算设备，比如服务器、终端等电子设备都可以通过加入该点对点网络而成为该区块链系统中的一个节点。

[0188] 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图形处理器的资源分配方法，其中，该计算机程序被设置为运行时执行上述任一项方法实施例中的步骤。

[0189] 可选地，在本实施例中，上述计算机可读的存储介质可以被设置为存储用于执行以下步骤的计算机程序：

[0190] S1，获取目标应用关联的参考资源信息，其中，目标应用为在应用服务器中运行且通过终端设备呈现画面的应用，参考资源信息为运行目标应用所需目标资源的参考信息；

[0191] S2，根据目标应用的历史运行数据调整参考资源信息，以获取目标资源信息，其中，历史运行数据为目标应用在运行时所上报的历史数据；

[0192] S3，获取应用服务器的每个图形处理器的资源提供状态，其中，资源提供状态用于表示图形处理器在当前时刻允许提供的目标资源；

[0193] S4，按照目标资源信息以及资源提供状态，为目标应用分配目标图形处理器，其中，目标图形处理器允许提供的目标资源大于或等于目标资源信息对应的目标资源。

[0194] 可选地，在本实施例中，本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通程序来指令终端设备相关的硬件来完成，该程序可以存储于一计算机可读存储介质中，存储介质可以包括：闪存盘、只读存储器 (Read-Only Memory, ROM)、随机存取器 (Random Access Memory, RAM)、磁盘或光盘等。

[0195] 上述本发明实施例序号仅仅为了描述，不代表实施例的优劣。

[0196] 上述实施例中的集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在上述计算机可读的存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或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存储介质中，包括若干指令用以使得一台或多台计算机设备 (可为个人计算机、服务器或者网络设备) 执行本发明各个实施例所述方法的全部或部分步骤。

[0197] 在本发明的上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其他实施例的相关描述。

[0198] 在本申请所提供的几个实施例中，应该理解到，所揭露的客户端，可通过其它的方式实现。其中，以上所描述的装置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，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之

间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0199] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者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者全部单元来实现本实施例方案的目的。

[0200] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,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,也可以采用软件功能单元的形式实现。

[0201] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

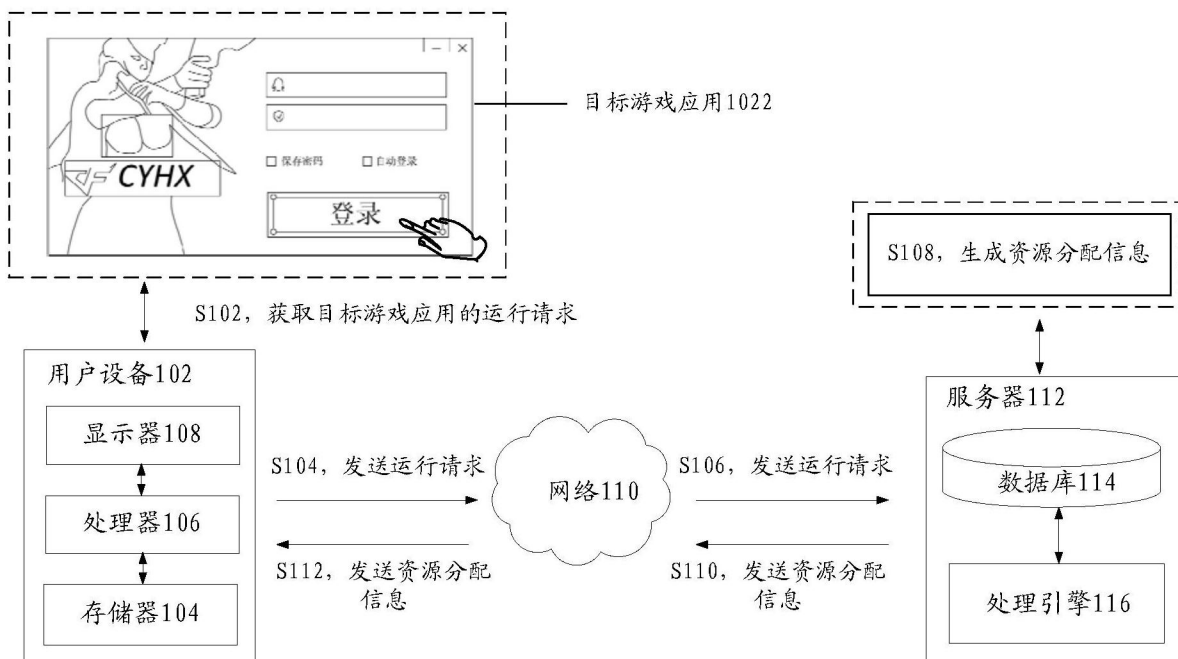


图1

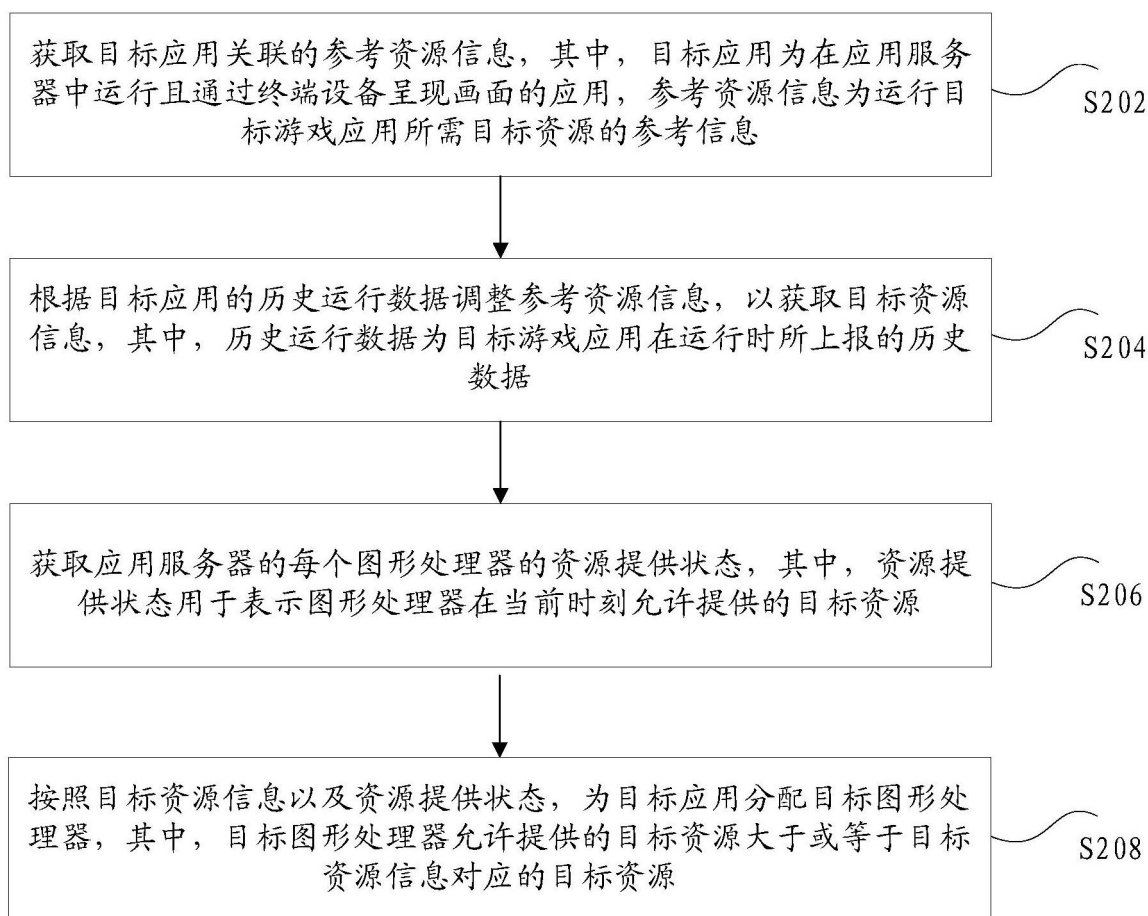


图2

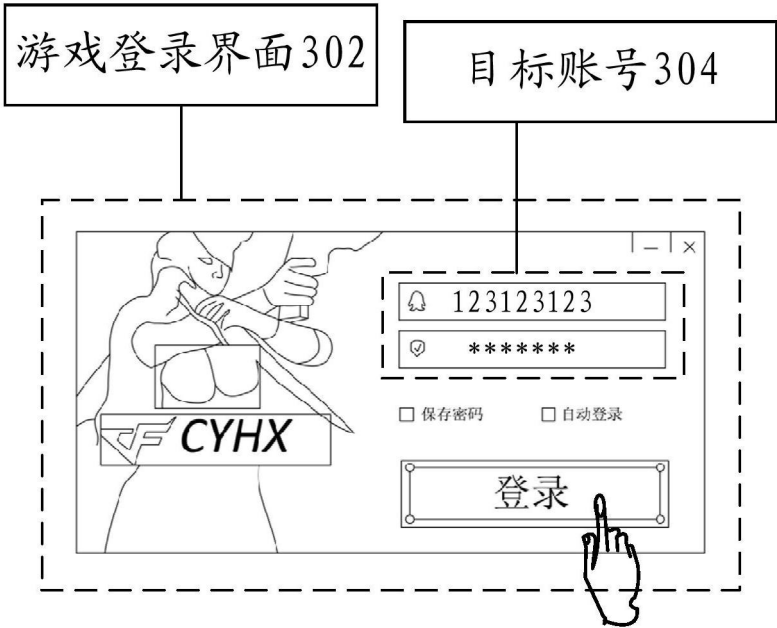


图3

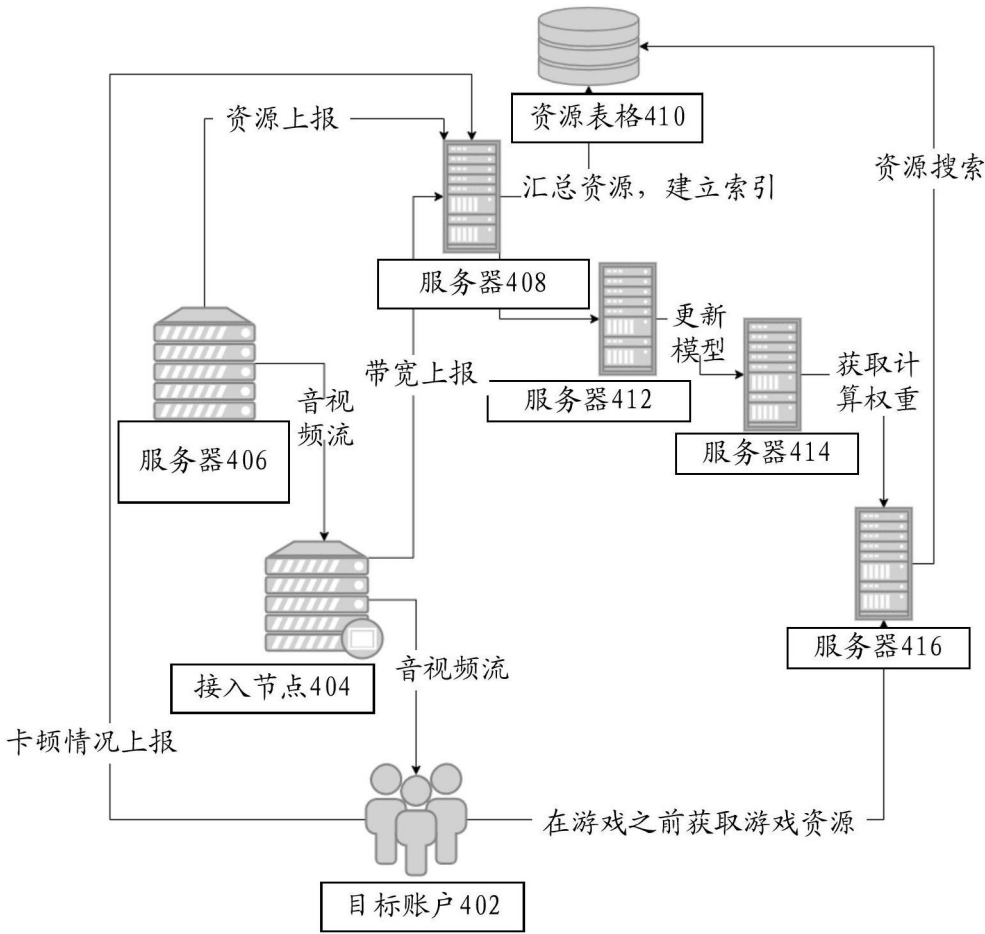


图4

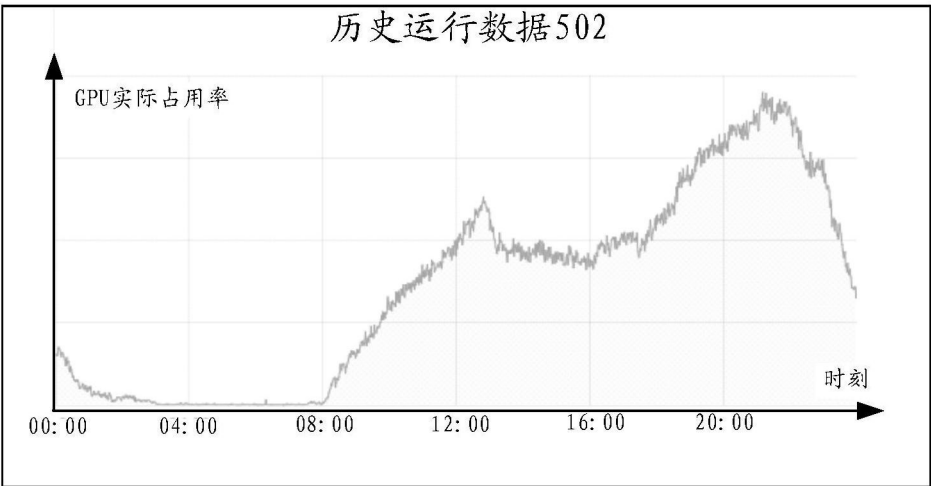


图5

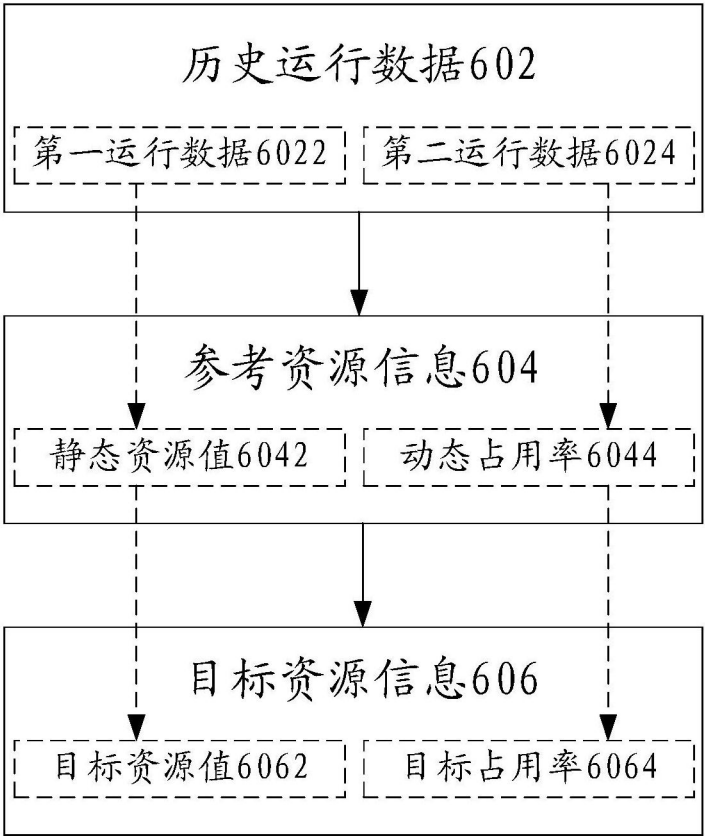


图6

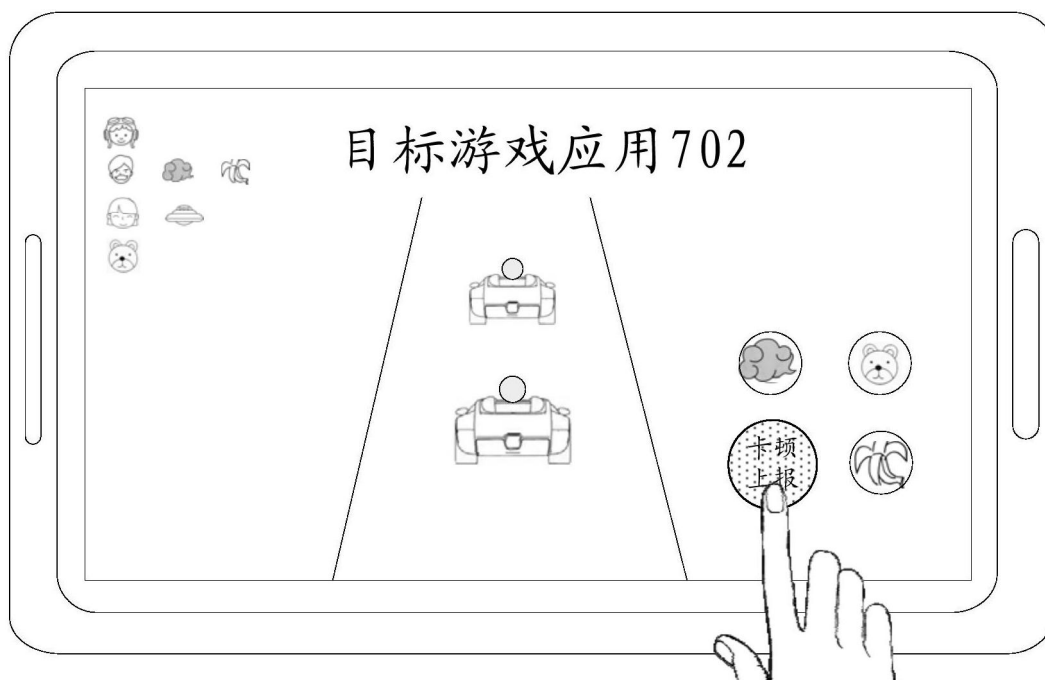


图7

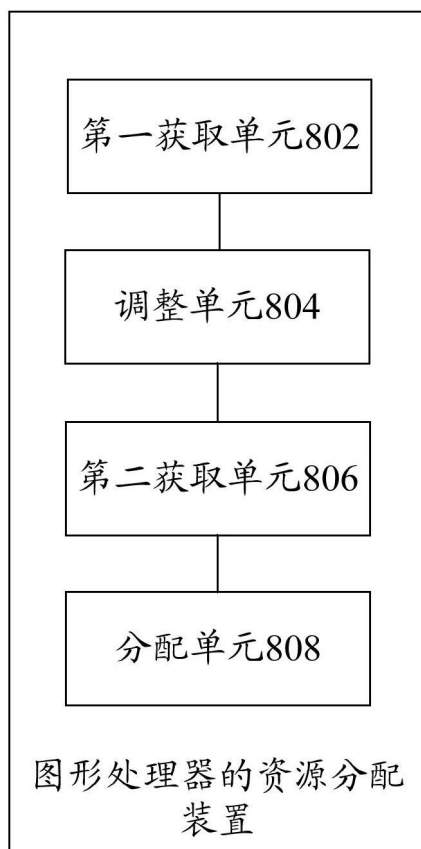


图8

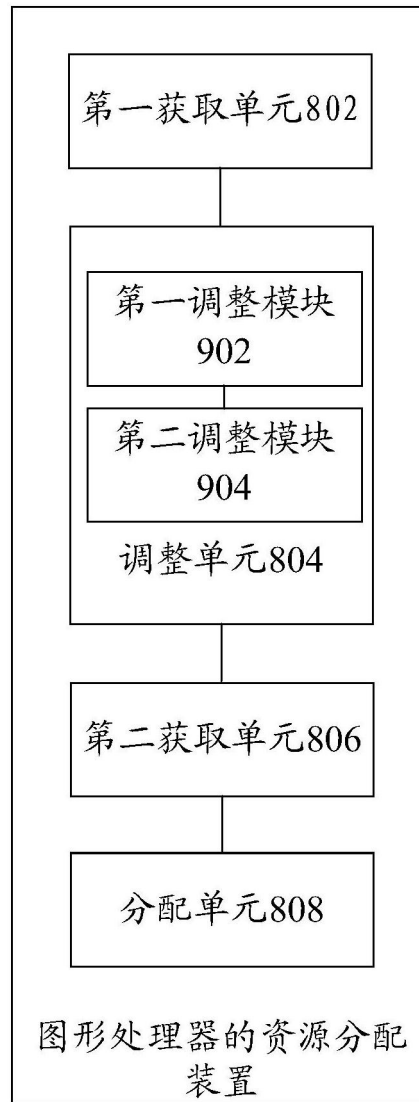


图9

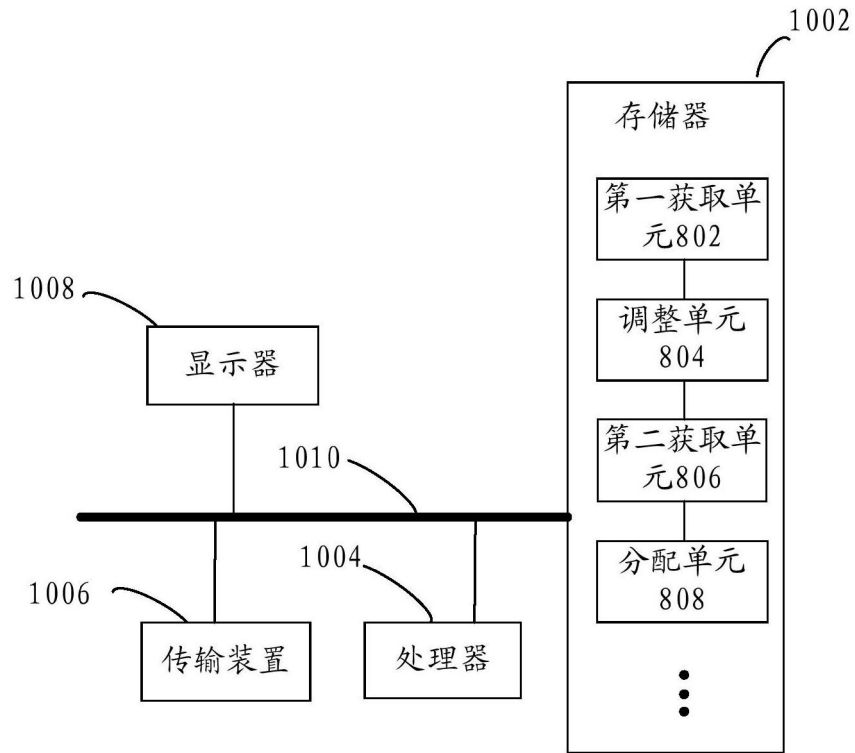


图10