



(12)发明专利申请

(10)申请公布号 CN 110955504 A
(43)申请公布日 2020.04.03

(21)申请号 201911001408.X

(22)申请日 2019.10.21

(71)申请人 量子云未来(北京)信息科技有限公司

地址 100021 北京市海淀区云会里金雅园
过街楼六层6235

(72)发明人 李甫

(74)专利代理机构 北京知呱呱知识产权代理有限公司 11577

代理人 盛明星

(51)Int.Cl.

G06F 9/48(2006.01)

G06F 9/50(2006.01)

G06T 13/20(2011.01)

G06T 15/00(2011.01)

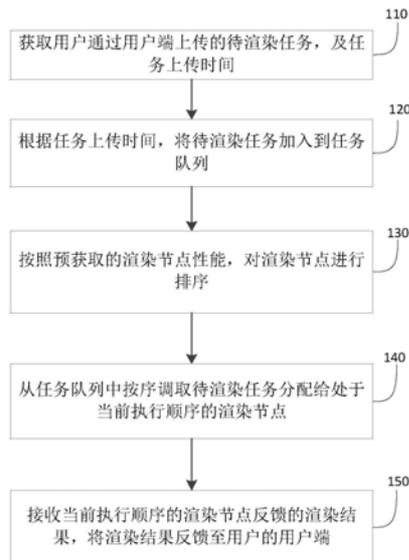
权利要求书2页 说明书7页 附图2页

(54)发明名称

智能分配渲染任务的方法、服务器、系统及存储介质

(57)摘要

本发明实施例涉及计算机技术领域,具体公开了一种智能分配渲染任务的方法、服务器、系统及存储介质,该方法包括:获取用户通过用户上传的待渲染任务,及任务上传时间;根据任务上传时间,将待渲染任务加入到任务队列;按照预获取的渲染节点性能,对渲染节点进行排序;从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点;接收当前执行顺序的渲染节点反馈的渲染结果,将渲染结果反馈至用户的用户端。通过上述方式,可以在保证其他工作正常进行的前提下,降低图像渲染的工作时长,大大提升图像渲染工作效率。



1. 一种智能分配渲染任务的方法,其特征在于,所述方法由智能分配渲染任务的服务器执行,包括:

获取用户通过用户上传的待渲染任务,及任务上传时间;

根据所述任务上传时间,将所述待渲染任务加入到任务队列;

按照预获取的渲染节点性能,对渲染节点进行排序;

从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点,以便处于当前执行顺序的渲染节点按照预设规则对所述待渲染任务进行渲染,所述预设规则至少包括渲染执行启动时间和用户渲染需求;

接收所述当前执行顺序的渲染节点反馈的渲染结果,将所述渲染结果反馈至所述用户的用户端,所述渲染节点为已授权的任一终端设备。

2. 根据权利要求1所述的方法,其特征在于,所述渲染节点性能根据所述渲染节点的硬件配置以及当前所述渲染节点的空闲资源综合确定。

3. 根据权利要求1所述的方法,其特征在于,当用户通过用户上传的待渲染任务包括至少两个子任务时,将每一个子任务分别分配给一个渲染节点执行渲染。

4. 根据权利要求1-3任一项所述的方法,其特征在于,所述从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点之前,所述方法还包括:

接收用户为待渲染任务选择的第一执行区域的操作指令;

当根据所述操作指令,确定所述第一执行区域内的第一渲染节点符合执行启动时间和用户渲染需求时,将所述待渲染任务分配至所述第一渲染节点;

或者,

当根据所述操作指令,确定所述第一执行区域内没有任何渲染节点符合执行启动时间,和/或用户渲染需求时,自动为所述用户分配至符合执行启动时间和用户渲染需求的第二执行区域,并规定执行渲染任务的第二渲染节点,其中所述执行区域包括至少两个,每一个执行区域均包含至少一个渲染节点。

5. 根据权利要求4所述的方法,其特征在于,所述将所述待渲染任务分配至第一渲染节点,或者第二渲染节点之后,所述方法还包括:为所述待渲染任务确定最佳传输路径,以便以最短时间将所述待渲染任务传输至少所述第一渲染节点或第二渲染节点。

6. 根据权利要求5所述的方法,其特征在于,每一个执行区域均分别包含至少一个中转存储节点,所述待渲染任务及任务上传时间存储至预设中转存储节点中,所述预设中转存储节点为所述待渲染任务执行区域中的一个中转存储节点,且所述中转存储节点和所述用户的用户端建立通信连接后,传输数据时间最短;

所述为所述待渲染任务确定最佳传输路径,具体包括:

根据所述预设中转存储节点、所述待渲染任务执行区域中的其他中转存储节点以及执行搜索渲染任务的渲染节点之间的通信关系,确定所述最佳传输路径。

7. 一种智能分配渲染任务的服务器,其特征在于,所述服务器包括:

接收单元,用于获取用户通过用户上传的待渲染任务,及任务上传时间;

处理单元,用于根据所述任务上传时间,将所述待渲染任务加入到任务队列;

按照预获取的渲染节点性能,对渲染节点进行排序;

从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点,以便处于当

前执行顺序的渲染节点按照预设规则对所述待渲染任务进行渲染,所述预设规则至少包括渲染执行启动时间和用户渲染需求;

所述接收单元,还用于接收所述当前执行顺序的渲染节点反馈的渲染结果;

发送单元,用于将所述渲染结果反馈至所述用户的用户端,所述渲染节点为已授权的任一终端设备。

8.一种智能分配渲染任务的系统,其特征在于,所述系统包括:用户端、智能分配渲染任务的服务器和渲染节点;

所述用户端用于根据用户操作指令,将待渲染任务上传至智能分配渲染任务的服务器;

所述智能分配渲染任务的服务器用于执行如权利要求1-6任一项所述的方法;

所述渲染节点用于按照预设规则,对所述待渲染任务进行渲染;

将渲染结果反馈至所述智能分配渲染任务的服务器。

9.一种计算机存储介质,其特征在于,所述计算机存储介质中包含一个或多个程序指令,所述一个或多个程序指令用于被智能分配渲染任务的服务器执行如权利要求1-6任一项所述的方法。

智能分配渲染任务的方法、服务器、系统及存储介质

技术领域

[0001] 本发明实施例涉及计算机技术领域，具体涉及一种智能分配渲染任务的方法、服务器、系统及存储介质。

背景技术

[0002] 图像渲染是将三维的光能传递处理转换为一个二维图像的过程。在图像渲染前，需要准备好三维几何模型信息，三维动画定义信息和材质信息。如果三维动画比较复杂，占用内存较多，那么图像渲染过程将会耗时较长，耽误其他工作的正常进行。例如电影特效中的动画将会有很多帧构成，在图像渲染时，需要在对一帧图像进行渲染完成后，再对下一帧图像进行渲染，工作耗时必然很长。

[0003] 因此，如何才能能够在保证其他工作正常进行的前提下，降低图像渲染的工作时长，提升图像渲染工作效率成为本申请亟待解决的技术问题。

发明内容

[0004] 为此，本发明实施例提供一种智能分配渲染任务的方法、服务器、系统及存储介质，以解决现有技术中由于图像复杂、占用内存多的情况，图像渲染耗时长，工作效率低的问题。

[0005] 为了实现上述目的，本发明的实施方式提供如下技术方案：

[0006] 在本发明的实施方式的第一方面中，提供了一种智能分配渲染任务的方法，该方法由智能分配渲染任务的服务器执行，包括：

[0007] 获取用户通过用户上传的待渲染任务，及任务上传时间；

[0008] 根据任务上传时间，将待渲染任务加入到任务队列；

[0009] 按照预获取的渲染节点性能，对渲染节点进行排序；

[0010] 从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点，以便处于当前执行顺序的渲染节点按照预设规则对待渲染任务进行渲染，预设规则至少包括渲染执行启动时间和用户渲染需求；

[0011] 接收当前执行顺序的渲染节点反馈的渲染结果，将渲染结果反馈至用户的用户端，渲染节点为已授权的任一终端设备。

[0012] 在本发明的一个实施例中，渲染节点性能根据渲染节点的硬件配置以及当前渲染节点的空闲资源综合确定。

[0013] 在本发明的另一实施例中，当用户通过用户上传的待渲染任务包括至少两个子任务时，将每一个子任务分别分配给一个渲染节点执行渲染。

[0014] 在本发明的又一个实施例中，从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点之前，方法还包括：

[0015] 接收用户为待渲染任务选择的第一执行区域的操作指令；

[0016] 当根据操作指令，确定第一执行区域内的第一渲染节点符合执行启动时间和用户

渲染需求时,将待渲染任务分配至第一渲染节点;

[0017] 或者,当根据操作指令,确定第一执行区域内没有任何渲染节点符合执行启动时间,和/或用户渲染需求时,自动为用户分配至符合执行启动时间和用户渲染需求的第二执行区域,并规定执行渲染任务的第二渲染节点,其中执行区域包括至少两个,每一个执行区域均包含至少一个渲染节点。

[0018] 在本发明的再一个实施例中,将待渲染任务分配至第一渲染节点,或者第二渲染节点之后,方法还包括:为待渲染任务确定最佳传输路径,以便以最短时间将待渲染任务传输至少第一渲染节点或第二渲染节点。

[0019] 在本发明的再一个实施例中,每一个执行区域均分别包含至少一个中转存储节点,待渲染任务及任务上传时间存储至预设中转存储节点中,预设中转存储节点为待渲染任务执行区域中的一个中转存储节点,且中转存储节点和用户的用户端建立通信连接后,传输数据时间最短;

[0020] 为待渲染任务确定最佳传输路径,具体包括:

[0021] 根据预设中转存储节点、待渲染任务执行区域中的其他中转存储节点以及执行搜索渲染任务的渲染节点之间的通信关系,确定最佳传输路径。

[0022] 在本发明的实施方式的第二方面中,提供了一种智能分配渲染任务的服务器,接收单元,用于获取用户通过用户上传的待渲染任务,及任务上传时间;

[0023] 处理单元,用于根据任务上传时间,将待渲染任务加入到任务队列;

[0024] 按照预获取的渲染节点性能,对渲染节点进行排序;

[0025] 从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点,以便处于当前执行顺序的渲染节点按照预设规则对待渲染任务进行渲染,预设规则至少包括渲染执行启动时间和用户渲染需求;

[0026] 接收单元,还用于接收当前执行顺序的渲染节点反馈的渲染结果;发送单元,用于将渲染结果反馈至用户的用户端,渲染节点为已授权的任一终端设备。

[0027] 在本发明的实施方式的第三方面中,提供了一种智能分配渲染任务的系统,包括:用户端、智能分配渲染任务的服务器和渲染节点;

[0028] 用户端用于根据用户操作指令,将待渲染任务上传至智能分配渲染任务的服务器;

[0029] 智能分配渲染任务的服务器用于执行如权利要求1-7任一项的方法;

[0030] 渲染节点用于按照预设规则,对待渲染任务进行渲染;

[0031] 将渲染结果反馈至智能分配渲染任务的服务器。

[0032] 在本发明的实施方式的第四方面中,提供了一种计算机存储介质,计算机存储介质中包含一个或多个程序指令,一个或多个程序指令用于被智能分配渲染任务的服务器执行如上一种智能分配渲染任务的方法中的任一方法步骤。

[0033] 根据本发明的实施方式,具有如下优点:在获取到用户上传的待渲染任务后,按照上传时间进行排序,然后为每一个待渲染任务优先分配渲染性能较高的渲染节点进行渲染。渲染节点将会按照预设规则进行渲染,多个渲染节点并行执行渲染工作。每一个渲染节点实际就是网络上连接的终端设备,在已授权的情况下就可以和服务器建立通信连接,接收渲染任务。而规定执行启动渲染时间,是为了避免渲染工作和渲染节点其他的正常工作

冲突,在渲染节点处于空闲状态或者有部分资源可以执行渲染工作时,接收渲染任务并执行渲染工作。从而实现保证其他工作正常进行的前提下,降低图像渲染的工作时长,而且网络上任一个终端设备都有可能在经过授权的情况下,成为渲染节点,从而实现类似渲染节点集群的形式对渲染任务进行渲染,大大提升图像渲染工作效率。

附图说明

[0034] 为了更清楚地说明本发明的实施方式或现有技术中的技术方案,下面将对实施方式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示例性的,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,还可以根据提供的附图引伸获得其它的实施附图。

[0035] 图1为本发明的一实施例提供的一种智能分配渲染任务的方法流程示意图;

[0036] 图2为本发明的另一实施例提供的一种智能分配渲染任务的服务器结构示意图;

[0037] 图3为本发明的另一实施例提供的一种智能分配渲染任务的系统结构示意图。

具体实施方式

[0038] 以下由特定的具体实施例说明本发明的实施方式,熟悉此技术的人士可由本说明书所揭露的内容轻易地了解本发明的其他优点及功效,显然,所描述的实施例是本发明一部分实施例,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,都属于本发明保护的范围。

[0039] 本发明实施例1提供了一种智能分配渲染任务的方法,具体如图1所示,图1为本发明实施例提供的一种智能分配渲染任务的方法流程示意图。该方法由智能分配渲染任务的服务器执行,可以包括:

[0040] 步骤110,获取用户通过用户上传的待渲染任务,及任务上传时间。

[0041] 具体的,用户通过自身的用户上传待渲染的任务。智能分配渲染任务的服务器(以下称服务器)在接收到用户上传的待渲染任务的同时,还包括记录任务上传时间。这里的用户端可以是以APP形式存在,也可以是以网页形式存在,具体不做限定。

[0042] 步骤120,根据任务上传时间,将待渲染任务加入到任务队列。

[0043] 为了保证渲染任务执行的公正性,则按照任务上传时间,依次将待渲染任务加入到任务队列中。便于后续按照任务队列中的渲染任务排序依次安排渲染节点执行渲染任务。

[0044] 步骤130,按照预获取的渲染节点性能,对渲染节点进行排序。

[0045] 具体的,待渲染节点性能计算可以按照预设规则综合评定。例如,根据待渲染节点的硬件配置以及当前所剩余的空闲资源等综合确定。这里的渲染节点实际是任一网络用户对应的终端设备,终端设备经过授权后可以与服务器建立通信连接,从而方便服务器获取终端设备自身对应的性能参数,例如硬件配置和当前所剩余的空闲资源等,然后根据这些性能参数对该终端设备进行性能排序。比如,将该终端设备的GPU、CPU等型号与预建立的数据库中配置硬件型号相匹配,数据库中包含不同型号硬件和对应的分数。类似的,还可以获取该终端设备的所剩余的内存资源,剩余资源越多,其得分越高。进而获取该终端设备的综合得分,根据综合得分,确定该终端设备的性能排序,也即是渲染节点的性能排序。

[0046] 步骤140,从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点。

[0047] 步骤150,接收当前执行顺序的渲染节点反馈的渲染结果,将渲染结果反馈至用户的用户端。

[0048] 具体的,上述操作是为了便处于当前执行顺序的渲染节点按照预设规则对待渲染任务进行渲染,预设规则至少包括渲染执行启动时间和用户渲染需求。

[0049] 而且,在分配过程中,可以优先为性能高的渲染节点安排任务,如此一来,渲染任务的工作效率必然提升。而且,网络中所有的终端设备都可能作为渲染节点,只要持有终端设备的用户出于自愿行为,自行在智能分配渲染系统上进行注册。经过授权后,终端设备就可以作为渲染节点。而且,还可以自行设定执行工作的时间,也即是渲染执行启动时间,比如该渲染节点希望在晚上19点至第二天7点之间执行渲染任务,而其他时间则需要完成自己的正常工作。那么,系统可以分配任务时,考虑其可以执行任务的时间,并非随时为其分配任务,而是在其可以执行任务的时间分配渲染任务。又或者,当当前可以执行任务的渲染节点均已被分配任务,系统也可以筛选就近的某一个时间段内可以启动执行任务的渲染节点,为其分配渲染任务,用以渲染节点在渲染执行启动时间开始,执行渲染任务。或者,如果用户不自定义渲染节点的执行渲染任务的时间,则默认渲染节点当前有空闲资源可以使用时就可以执行渲染任务。服务器可以将任务先分配给该渲染节点,让渲染节点自行安排渲染时间。另外,渲染任务中包含了用户的渲染需求,因此,需要渲染节点按照用户的渲染需求进行渲染。

[0050] 通常而言,每一帧图片作为一个渲染任务,如果用户的渲染任务中包含多帧图片时,则将每一帧图片作为一个子任务。每一个子任务可以分别分配给一个渲染节点,由不同的渲染节点并行执行渲染。在渲染完成后,反馈至服务器,由服务器进行汇总。在分配任务之前,可以为每一个子任务进行具有标识性的命名,用以在服务器获取渲染结果后,对子任务根据具有标识性的名称进行汇总,构成总的渲染任务结果,反馈给上传渲染任务的用户。当然,如果上传渲染任务时,用户已经命名好的,则无需重新命名,根据其自定义的命名进行汇总即可。而且,汇总后的渲染结果还可以进行备份,分别备份至不同的中转存储节点中,以便于数据传输过程中丢失或者出线其他意外的情况下,还可以及时进行补救。

[0051] 可选的,为了进一步提高渲染效率,可以将渲染任务按照执行区域进行分配。例如,国内的渲染节点分布在各个城市,不同的城市又属于不同点区域,例如以省为准。那么将每一个省作为一个执行区域的话,那么服务器在执行步骤140之前,还可以包括:接收用户为待渲染任务选择的第一执行区域的操作指令。

[0052] 也即是说,用户可以自主选择进行渲染任务执行的执行区域。其主要目的是为了加快渲染任务上传时间。每个执行区域包含一个子服务器,而上文说的服务器可以理解为总服务器,用以进行宏观控制。而子服务器则负责控制该执行区域内的渲染任务的分配,以及相应指令的控制。

[0053] 一般建议用户选择距离自己所在位置最近的执行区域,例如辽宁省内的用户选择辽宁省对应的执行区域。或者,如果辽宁省内包含多个执行区域时,可以根据地理位置选择距离用户最近的执行区域。具体实现可以根据实际情况设定。

[0054] 总服务器根据操作指令确定第一执行区域内的第一渲染节点是否符合执行启动

时间和用户渲染需求,或者总服务器将用户需求反馈至子服务器,由子服务器确定自己对应的执行区域内的第一渲染节点是否符合上述要求。如果符合,则将待渲染任务分配至第一渲染节点。

[0055] 否则,若确定第一执行区域内没有任何渲染节点符合执行启动时间,和/或用户渲染需求时,则由总服务器自动为用户分配至符合执行启动时间和用户渲染需求的第二执行区域,并规定执行渲染任务的第二渲染节点,其中执行区域包括至少两个,每一个执行区域均包含至少一个渲染节点。

[0056] 进一步的,在将待渲染任务分配至第一渲染节点,或者第二渲染节点之后,方法还包括:为待渲染任务确定最佳传输路径,以便以最短时间将待渲染任务传输至少第一渲染节点或第二渲染节点。

[0057] 这里的传输路径依然是为了加快文件传输速率,也就是加快待渲染任务能够尽快传输至目标渲染节点(总服务器已经为其分配的渲染节点)。

[0058] 需要说明的是,每一个执行区域中均可以包含至少一个中转存储节点。待渲染任务及任务上传时间实际上是存储至预设的中转存储节点中。该中转存储节点和用户的用户端建立通信连接后,传输数据时间最短。具体确定传输数据时间可以再正式传输渲染任务之前,用户端向子服务器发送通信连接请求,子服务器根据通信连接请求反馈该执行区域中所有中转存储节点的相关信息(例如中转存储节点名称和IP地址等能够建立通信连接的信息)至用户端。用户端可以分别和每一个中转存储节点建立通信连接,依次向每一个中转存储节点发送一个测试数据包,用以验证传输时间,从而确定传输数据时间最短的中转存储节点作为预设中转存储节点。并将待渲染任务传输至该中转存储节点。在通过该中转存储节点经过其他传输路径传输至渲染节点。

[0059] 具体的,任一网络用户端和渲染节点之间的传输路径中,包含了至少一个中转存储节点。理想状态中,就是用户的用户端将待渲染任务传输至预设中转存储节点后,直接通过预设中转存储节点传输至需要执行该渲染任务的目标渲染节点。这个传输过程时间最短,那么则确定该传输路径为最佳传输路径。

[0060] 而如果期间还需要通过其他通信节点传输,那么首选通过多个中转存储节点之间进行通信传输,最终通过中转存储节点传输至需要执行该渲染任务的目标节点。其仅通过中转存储节点之间进行中转的速率往往高于渲染节点之间的传输速率,中转存储节点之间的传输可以理解为内网传输,因为效率往往更高。

[0061] 还需要说明的是,网络中与服务器(包括上文所说的总服务器和子服务器)建立通信连接的所有终端设备都可以加载用户端,用以传输待渲染的任务,又都可以作为渲染节点,在服务器中进行注册后,执行渲染任务。当然,渲染任务完成后并非是义务完成,而是可以获取一定的报酬。用户提交渲染任务也需要支付一定的报酬。这个报酬可以以积分的形式,或者以数字货币的形式体现。具体可以根据实际情况设定,这里不做过多限定。

[0062] 本发明实施例提供的一种智能分配渲染任务的方法,在获取到用户上传的待渲染任务后,按照上传时间进行排序,然后为每一个待渲染任务优先分配渲染性能较高的渲染节点进行渲染。渲染节点将会按照预设规则进行渲染,多个渲染节点并行执行渲染工作。每一个渲染节点实际就是网络上连接的终端设备,在已授权的情况下就可以和服务器建立通信连接,接收渲染任务。而规定执行启动渲染时间,是为了避免渲染工作和渲染节点其他的

正常工作冲突,在渲染节点处于空闲状态或者有部分资源可以执行渲染工作时,接收渲染任务并执行渲染工作。从而实现保证其他工作正常进行的前提下,降低图像渲染的工作时长,而且网络上任一个终端设备都有可能在经过授权的情况下,成为渲染节点,从而实现类似渲染节点集群的形式对渲染任务进行渲染,大大提升图像渲染工作效率。

[0063] 与上述实施例1对应的,本发明实施例2还提供了一种智能分配渲染任务的服务器,具体如图2所示,该服务器包括:接收单元201、处理单元202和发送单元203。

[0064] 接收单元201,用于获取用户通过用户上传的待渲染任务,及任务上传时间。

[0065] 处理单元202,用于根据任务上传时间,将待渲染任务加入到任务队列。按照预获取的渲染节点性能,对渲染节点进行排序,从任务队列中按序调取待渲染任务分配给处于当前执行顺序的渲染节点,以便处于当前执行顺序的渲染节点按照预设规则对待渲染任务进行渲染,预设规则至少包括渲染执行启动时间和用户渲染需求。

[0066] 接收单元201,还用于接收当前执行顺序的渲染节点反馈的渲染结果;

[0067] 发送单元203,用于将渲染结果反馈至用户的用户端,渲染节点为已授权的任一终端设备。

[0068] 可选的,渲染节点性能根据渲染节点的硬件配置以及当前渲染节点的空闲资源综合确定。

[0069] 进一步可选的,当用户通过用户上传的待渲染任务包括至少两个子任务时,处理单元202具体用于,将每一个子任务分别分配给一个渲染节点执行渲染。

[0070] 可选的,接收单元201还用于,接收用户为待渲染任务选择的第一执行区域的操作指令;

[0071] 处理单元202还用于,当根据操作指令,确定第一执行区域内的第一渲染节点符合执行启动时间和用户渲染需求时,将待渲染任务分配至第一渲染节点;

[0072] 或者,当根据操作指令,确定第一执行区域内没有任何渲染节点符合执行启动时间,和/或用户渲染需求时,自动为用户分配至符合执行启动时间和用户渲染需求的第二执行区域,并规定执行渲染任务的第二渲染节点,其中执行区域包括至少两个,每一个执行区域均包含至少一个渲染节点。

[0073] 可选的,处理单元202还用于,为待渲染任务确定最佳传输路径,以便以最短时间将待渲染任务传输至少第一渲染节点或第二渲染节点。

[0074] 可选的,每一个执行区域均分别包含至少一个中转存储节点,待渲染任务及任务上传时间存储至预设中转存储节点中,预设中转存储节点为待渲染任务执行区域中的一个中转存储节点,且中转存储节点和用户的用户端建立通信连接后,传输数据时间最短;

[0075] 处理单元202具体用于,根据预设中转存储节点、待渲染任务执行区域中的其他中转存储节点以及执行搜索渲染任务的渲染节点之间的通信关系,确定最佳传输路径。

[0076] 本发明实施例提供的一种智能分配渲染任务的服务器中各部件所执行的功能均在上述实施例中做了详细介绍,因此这里不做过多赘述。

[0077] 本发明实施例提供的一种智能分配渲染任务的服务器,在获取到用户上传的待渲染任务后,按照上传时间进行排序,然后为每一个待渲染任务优先分配渲染性能较高的渲染节点进行渲染。渲染节点将会按照预设规则进行渲染,多个渲染节点并行执行渲染工作。每一个渲染节点实际就是网络上连接的终端设备,在已授权的情况下就可以和服务器建立

通信连接,接收渲染任务。而规定执行启动渲染时间,是为了避免渲染工作和渲染节点其他的正常工作冲突,在渲染节点处于空闲状态或者有部分资源可以执行渲染工作时,接收渲染任务并执行渲染工作。从而实现保证其他工作正常进行的前提下,降低图像渲染的工作时长,而且网络上任一个终端设备都有可能在经过授权的情况下,成为渲染节点,从而实现类似渲染节点集群的形式对渲染任务进行渲染,大大提升图像渲染工作效率。

[0078] 与上述实施例相对应的,本发明实施例3还提供了一种智能分配渲染任务的系统,具体如图3所示,该系统包括:用户端301、智能分配渲染任务的服务器302和渲染节点303。

[0079] 用户端301用于根据用户操作指令,将待渲染任务上传至智能分配渲染任务的服务器302;

[0080] 智能分配渲染任务的服务器302用于执行上述实施例提供的一种智能分配渲染任务的方法中任一项方法步骤;

[0081] 渲染节点303用于按照预设规则,对待渲染任务进行渲染;

[0082] 将渲染结果反馈至智能分配渲染任务的服务器302。

[0083] 虽然图3中仅是简单示意出一个用户端301、一个智能分配渲染任务的服务器302以及一个渲染节点303,但是图3仅是为了示意性说明智能分配渲染任务的系统中三个部件之间的通信连接关系,并不代表具体数量。具体数量根据实际情况设定,这里不做过多限定。

[0084] 本发明实施例提供的一种智能分配渲染任务的系统,在获取到用户上传的待渲染任务后,按照上传时间进行排序,然后为每一个待渲染任务优先分配渲染性能较高的渲染节点进行渲染。渲染节点将会按照预设规则进行渲染,多个渲染节点并行执行渲染工作。每一个渲染节点实际就是网络上连接的终端设备,在已授权的情况下就可以和服务器建立通信连接,接收渲染任务。而规定执行启动渲染时间,是为了避免渲染工作和渲染节点其他的正常工作冲突,在渲染节点处于空闲状态或者有部分资源可以执行渲染工作时,接收渲染任务并执行渲染工作。从而实现保证其他工作正常进行的前提下,降低图像渲染的工作时长,而且网络上任一个终端设备都有可能在经过授权的情况下,成为渲染节点,从而实现类似渲染节点集群的形式对渲染任务进行渲染,大大提升图像渲染工作效率。

[0085] 与上述实施例相对应的,本发明实施例还提供了一种计算机存储介质,该计算机存储介质中包含一个或多个程序指令。其中,一个或多个程序指令用于被一种智能分配渲染任务的服务器执行如上所介绍的一种智能分配渲染任务的方法。

[0086] 虽然,上文中已经用一般性说明及具体实施例对本发明作了详尽的描述,但在本发明基础上,可以对之作一些修改或改进,这对本领域技术人员而言是显而易见的。因此,在不偏离本发明精神的基础上所做的这些修改或改进,均属于本发明要求保护的范围。

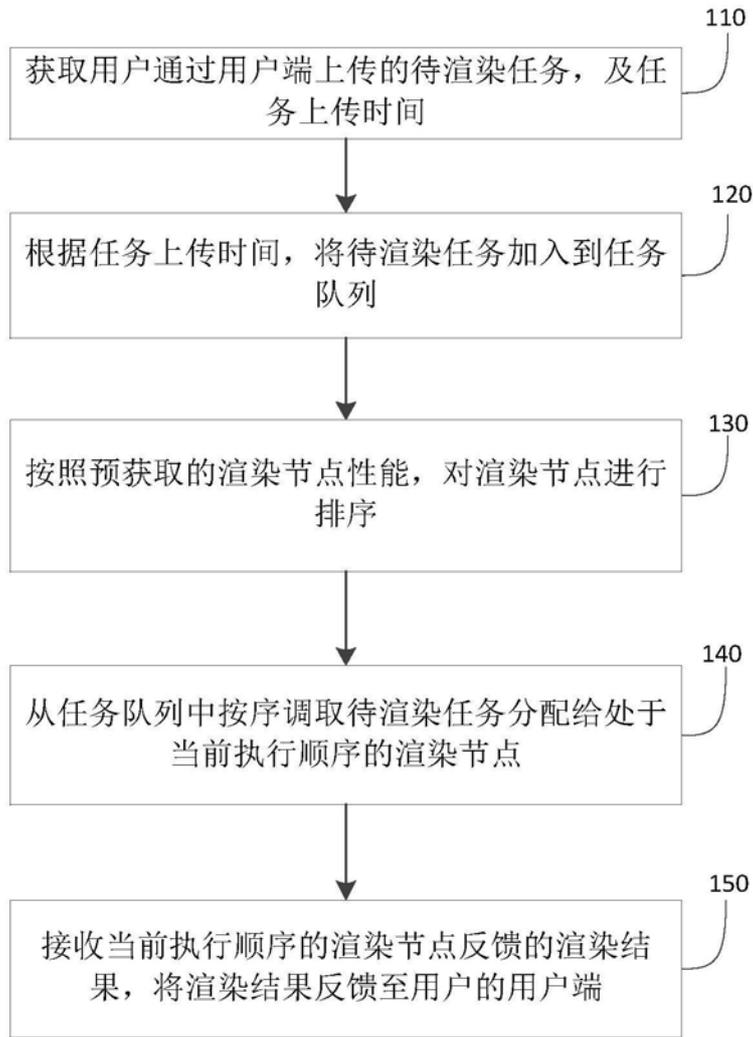


图1



图2

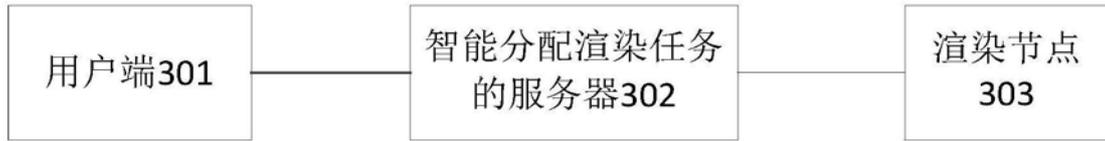


图3