



(12)发明专利申请

(10)申请公布号 CN 111625948 A

(43)申请公布日 2020.09.04

(21)申请号 202010431751.4

(22)申请日 2020.05.20

(71)申请人 北京百度网讯科技有限公司
地址 100085 北京市海淀区上地十街10号
百度大厦2层

(72)发明人 郭正东 周杰 王家梁

(74)专利代理机构 北京品源专利代理有限公司
11332

代理人 孟金喆

(51)Int.Cl.

G06F 30/20(2020.01)

G06F 3/06(2006.01)

G06F 9/48(2006.01)

权利要求书3页 说明书14页 附图3页

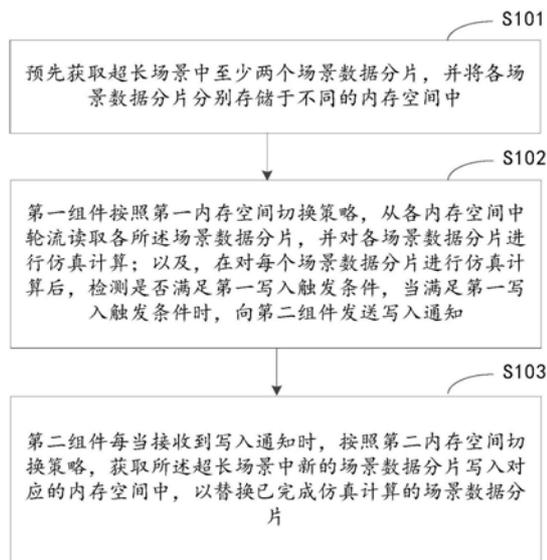
(54)发明名称

超长场景的回放型仿真方法、装置、设备和介质

(57)摘要

本申请实施例公开了一种超长场景的回放型仿真方法、装置、设备和介质,涉及自动驾驶技术,其中,该方法包括:第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。本申请实施例实现了有效降低超长场景回放仿真过程中存储设备的使用率,提高仿真服务的稳定性和准确性。

CN 111625948 A



1. 一种超长场景的回放型仿真方法,其特征在于,包括:

预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于不同的内存空间中;

第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;

第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。

2. 根据权利要求1所述的方法,其特征在于,第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算,包括:

第一组件获取第一寄存器当前值,第一寄存器当前值初始化为一个内存空间的存储编号;

第一组件获取与第一寄存器当前值匹配的第一目标内存空间后,读取第一目标内存空间中存储的场景数据分片,并对读取的场景数据分片进行仿真计算;

第一组件按照第一寄存器编号更新策略,对第一寄存器当前值进行更新后,返回执行获取与第一寄存器当前值匹配的目标内存空间的操作。

3. 根据权利要求2所述的方法,其特征在于,第一组件按照第一寄存器编号更新策略,对第一寄存器当前值进行更新,包括:

第一组件如果确定第一寄存器当前值不为末位内存空间的存储编号,则将第一寄存器当前值加1的结果,更新为新的第一寄存器当前值;

第一组件如果确定第一寄存器当前值为末位内存空间的存储编号,则将首位内存空间的存储编号,更新为新的第一寄存器当前值,其中,各所述内存空间预先进行顺序编号。

4. 根据权利要求2或3所述的方法,其特征在于,第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,包括:

第二组件每当接收到写入通知时,执行:

第二组件按照第二寄存器编号更新策略,更新第二寄存器当前值;第二寄存器当前值初始化为一个内存空间的存储编号;

第二组件根据更新前后的第二寄存器当前值确定至少一个第二目标内存空间;

第二组件按照由更新前第二寄存器当前值对应的第二目标内存空间,朝向更新后第二寄存器当前值对应的第二目标内存空间的顺序,获取所述超长场景中新的场景数据分片写入第二目标内存空间。

5. 根据权利要求4所述的方法,其特征在于,第二组件按照第二寄存器编号更新策略,更新第二寄存器当前值,包括:

第二组件将更新前第二寄存器当前值与设定增量值进行累加;

第二组件如果确定累加结果未超过末位内存空间的存储编号,则将所述累加结果更新为新的第二寄存器当前值;

第二组件如果确定累加结果超过末位内存空间的存储编号,则计算所述累加结果与所述末位内存空间的存储编号之间的差值,并将所述差值更新为新的第二寄存器当前值。

6. 根据权利要求4所述的方法,其特征在于,预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于至少两个内存空间中,包括:

第三组件预先下载超长场景中至少两个场景数据分片,并将各所述场景数据分片写入磁盘中;

第二组件获取所述磁盘中存储的至少两个场景数据分片,并将所述至少两个场景数据分片存储于至少两个内存空间中。

7. 根据权利要求6所述的方法,其特征在于,还包括:

第二组件在获取所述超长场景中新的场景数据分片写入对应的内存空间后,检测是否满足第二写入触发条件,当满足第二写入触发条件时,向第三组件发送磁盘写入通知;

第三组件每当检测到磁盘写入通知时,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中,以替换所述磁盘中当前已处理的场景数据分片;

第二组件获取所述超长场景中新的场景数据分片写入第二目标内存空间,具体包括:

第二组件从磁盘中获取所述新的场景数据分片,并将所述新的场景数据分片写入所述第二目标内存空间中。

8. 根据权利要求4或7所述的方法,其特征在于,第一写入触发条件包括:所述内存空间中当前未处理的场景数据分片的数量小于等于第一数量阈值;

第二写入触发条件包括:所述磁盘中当前未处理的场景数据分片的数量小于等于第二数量阈值。

9. 根据权利要求1所述的方法,其特征在于,第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算,包括:

第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算,直至满足结束仿真条件。

10. 根据权利要求6或7所述的方法,其特征在于,所述第一组件、第二组件以及第三组件分别由三个独立线程执行,所述方法具体包括:

通过第一线程,预先下载超长场景中至少两个场景数据分片,并将各所述场景数据分片写入磁盘中;

通过第二线程,获取所述磁盘中存储的至少两个场景数据分片,并将所述至少两个场景数据分片存储于至少两个内存空间中后,进入休眠状态;

通过第三线程,按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;在对各场景数据分片进行仿真计算的过程中,根据第一写入触发条件,唤醒第二线程;

通过所述第二线程,在被唤醒后,根据第二写入触发条件,重新触发所述第一线程,并按照第二内存空间切换策略,从磁盘中获取所述新的场景数据分片写入对应的内存空间中;在完成单次数据写入操作后,重新进入休眠状态;

通过所述第一线程,在被触发后,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中。

11. 根据权利要求1所述的方法,其特征在于,所述方法由自动驾驶仿真系统中,至少一个感知单元的计算模块执行。

12. 一种超长场景的回放型仿真装置,其特征在于,包括:

数据存储模块,用于预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于不同的内存空间中;

仿真计算模块,用于第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;

场景数据写入模块,用于第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。

13. 一种电子设备,其特征在于,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1-11中任一项所述的超长场景的回放型仿真方法。

14. 一种存储有计算机指令的非瞬时计算机可读存储介质,其特征在于,所述计算机指令用于使所述计算机执行权利要求1-11中任一项所述的超长场景的回放型仿真方法。

超长场景的回放型仿真方法、装置、设备和介质

技术领域

[0001] 本申请实施例涉及计算机技术,具体涉及自动驾驶技术,尤其涉及一种超长场景的回放型仿真方法、装置、设备和介质。

背景技术

[0002] 在自动驾驶领域,可以通过场景的回放型仿真实现无人车辆中算法模块的迭代更新。其中,需要将实际的路跑数据作为场景数据,加载至无人车辆的各个感知模块(惯性测量单元,高精雷达以及摄像机等传感器)的计算模块中进行计算处理,以回放生成真实场景。

[0003] 目前,在大规模分布式集群的仿真平台中,可以分配用于执行某个仿真任务的资源非常有限,当需要对场景数据很大的超长场景进行回放型仿真时,分配给各个计算模块使用的存储设备无法对上述场景数据的完整存储,也就无法实现对超长场景的回放仿真,这就大大降低了仿真服务的稳定性和准确性。

发明内容

[0004] 本申请实施例公开一种超长场景的回放型仿真方法、装置、设备和介质,以解决现有方法对超长场景的回放型仿真中,由于分配给各个计算模块使用的存储设备无法对上述场景数据的完整存储,导致的仿真服务的稳定性和准确性较低的问题。

[0005] 第一方面,本申请实施例公开了一种超长场景的回放型仿真方法,包括:

[0006] 预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于不同的内存空间中;

[0007] 第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;

[0008] 第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。第二方面,本申请实施例还公开了一种超长场景的回放型仿真装置,包括:

[0009] 数据存储模块,用于预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于不同的内存空间中;

[0010] 仿真计算模块,用于第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;

[0011] 场景数据写入模块,用于第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。第三方面,本申请实施例还公开了一种电子设备,包括:

[0012] 至少一个处理器;以及

[0013] 与所述至少一个处理器通信连接的存储器;其中,

[0014] 所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行如本申请实施例任一所述的超长场景的回放型仿真方法。

[0015] 第四方面,本申请实施例还公开了一种存储有计算机指令的非瞬时计算机可读存储介质,所述计算机指令用于使所述计算机执行如本申请实施例任一所述的超长场景的回放型仿真方法。

[0016] 根据本申请实施例的技术方案,通过第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真的场景数据分片,使得对场景数据分片的读取和写入可以交替进行,从而有效降低超长场景回放仿真过程中存储设备的使用率,提高仿真服务的稳定性和准确性。

[0017] 应当理解,本部分所描述的内容并非旨在标识本公开的实施例的关键或重要特征,也不用于限制本公开的范围。本公开的其它特征将通过以下的说明书而变得容易理解。

附图说明

[0018] 附图用于更好地理解本方案,不构成对本申请的限定。其中:

[0019] 图1是根据本申请实施例一公开的一种超长场景的回放型仿真方法的流程图;

[0020] 图2是根据本申请实施例二公开的另一种超长场景的回放型仿真方法的流程图;

[0021] 图3是根据本申请实施例三公开的另一种超长场景的回放型仿真方法的流程图;

[0022] 图4是根据本申请实施例四公开的一种通过三个独立线程执行第一组件、第二组件以及第三组件实现超长场景的回放型仿真方法的示意图;

[0023] 图5是根据本申请实施例五公开的一种超长场景的回放型仿真装置的结构示意图;

[0024] 图6是根据本申请实施例公开的一种电子设备的框图。

具体实施方式

[0025] 以下结合附图对本申请的示范性实施例做出说明,其中包括本申请实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本申请的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0026] 申请人在研发过程中发现,现有技术针对超长场景的回放仿真时,为了使得存储设备能够完整存储场景数据,通常会将场景数据拆分为多个小数据块,并依次加载至各计算模块进行处理,然而这种方法存在如下问题:在前一数据块计算完成后,需要首先进行后一数据块的加载,并在加载完成后,在对后一数据块进行计算,在对后一数据块的加载过程中,计算模块会停止进行计算,直至后一数据块的加载完成,上述操作会造成计算模

块间歇性的停止服务,然而,很多情况下计算模块是不能容忍短暂时间的场景数据切换造成的服务中止。例如,计算模块内部一般会维持多种时钟,如系统时钟或用户时钟等,以惯性测量单元IMU的计算模块为例,当服务开始后,由于将场景数据加载进存储设备所造成的计算模块中止时,IMU内部维持的系统时钟依然会运行,会对算法模块最终的计算误差造成较大的影响。因此,这种方法并不能根本的解决超长场景的回放仿真的稳定性和准确性较差的问题。

[0027] 实施例一

[0028] 图1是根据本申请实施例一公开的一种超长场景的回放型仿真方法的流程图,本实施例可以适用于根据采集的场景数据,通过车载的各感知模块中的计算模块进行场景回放仿真的情况。本实施例方法可以由超长场景的回放型仿真装置来执行,该装置可采用软件和/或硬件实现,并可集成在任意的具有计算能力的电子设备上,例如服务器等。

[0029] 回放型仿真的主要目的是通过模拟真实环境和构建汽车模型,找出自动驾驶过程中可能出现的问题。在控制算法、感知算法、定位算法和地图等模块的迭代升级和发版过程中,通过回放型仿真发现和复现问题,而不是真实的环境和硬件,可以极大的节约成本。此外,回放型仿真具备场景丰富、接口灵活和恢复迅速等特点。通过回放型仿真,可以把训练时间大大提高,远远超出路测时间,加快模型迭代速度。

[0030] 如图1所示,本实施例公开的超长场景的回放型仿真方法可以包括:

[0031] S101、预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于不同的内存空间中。

[0032] 在实际工作中,自动驾驶仿真系统向内置感知模块如惯性测量单元、高精雷达和摄像机等传感模块的计算模块分配的存储空间有限,通常为两个物理核和小于10G的内存空间,以及不超过50G的磁盘空间,而将超长场景的场景数据直接存储于有限的存储空间内,必将会导致场景数据的丢失。

[0033] 具体的,在本实施例中,将采集的超长场景的场景数据以场景数据分片的形式存储于服务平台中,其中场景数据分片表示计算模块进行仿真计算所需最小单位的场景数据,服务平台的本质即存储型服务器。在计算模块中的计算单元读取待处理数据之前,计算模块中的数据写入单元预先从服务平台中获取超长场景中至少两个场景数据分片,并将该至少两个场景数据分片,分别存储于与计算模块关联内存中不同的内存空间中,其中本实施例预先将内存进行空间划分,以使得内存中至少包括两个内存空间,例如将10G内存划分为两个5G的内存空间。

[0034] 本实施例通过预先获取超长场景中至少两个场景数据分片,避免了直接将超长场景的场景数据存储于内存中导致场景数据的丢失,且有效降低超长场景回放仿真过程中内存的使用率;通过将各场景数据分片分别存储于不同的内存空间中,为后续轮流读取各所述场景数据分片,以及轮流写入超长场景中新的场景数据分片,奠定了基础。

[0035] S102、第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知。

[0036] 其中,第一内存空间切换策略预先由技术人员制定并写入到自动驾驶仿真系统

中。第一内存空间切换策略用于指引第一组件从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及向第二组件发送写入通知。第一组件可选的包括自动驾驶仿真系统中各传感模块中计算模块内的计算单元。

[0037] 具体的,按照第一内存空间切换策略,第一组件获取当前读取操作关联的标识信息,并根据该标识信息确定对应的内存空间,进而访问该内存空间读取场景数据分片,最终对场景数据分片进行仿真计算;计算完成后,第一组件继续获取下次读取操作关联的标识信息,并根据该标识信息确定对应的内存空间,进而访问该内存空间读取场景数据分片,最终对场景数据分片进行仿真计算;重复上述操作,直至第一组件读取完内存空间中所有未处理的场景数据为止。其中,每次读取操作关联的标识信息都由第一标识信息更新策略预先确定,所述第一标识信息更新策略用于更新每次读取操作关联的标识信息。

[0038] 在对每个场景数据分片进行仿真计算后,第一组件会检测各内存空间中当前未处理的场景数据分片的数量,若该数量满足第一写入触发条件,则向第二组件发送写入通知。其中,第二组件用于向内存空间中写入场景数据分片,可选的包括自动驾驶仿真系统中各传感模块中计算模块内的数据写入单元。

[0039] 可选的,第一写入触发条件包括:所述内存空间中当前未处理的场景数据分片的数量小于等于第一数量阈值,其中,第一数量阈值可根据需求进行设置。

[0040] 示例性的,假设第一数量阈值为3,内存中共有6个内存空间分别存储有场景数据分片,若内存空间中当前未处理的场景数据分片的数量为3,由于内存空间中当前未处理的场景数据分片的数量“3”等于第一数量阈值“3”,则第一组件向第二组件发送写入通知。

[0041] 通过第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算,实现了对场景数据分片进行仿真计算的技术效果;通过在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知,实现了实时监测当前内存空间中是否需要写入新的场景数据分片的效果,为后续第二组件根据该写入通知在内存空间中写入新的场景数据分片,奠定了基础。

[0042] S103、第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。

[0043] 其中,第二内存空间切换策略预先由技术人员制定并写入到自动驾驶仿真系统中。第二内存空间切换策略用于指引第二组件向各内存空间中轮流写入超长场景中新的场景数据分片,以替换已完成仿真计算的场景数据分片。

[0044] 具体的,按照第二内存空间切换策略,第二组件从服务平台或者本地端获取超长场景中新的场景数据分片,进而获取当前写入操作关联的标识信息,根据该标识信息确定对应的内存空间,进而将超长场景中新的场景数据分片写入该内存空间,以替换该内存空间中已完成仿真计算的场景数据分片。写入完成后,第二组件继续获取下次写入操作关联的标识信息,并根据该标识信息确定对应的内存空间,进而将超长场景中新的场景数据分片写入该内存空间,以替换该内存空间中已完成仿真计算的场景数据分片;重复上述操作,直至第二组件将内存空间中所有已完成仿真计算的场景数据分片替换为止。其中,每次写入操作关联的标识信息都由第二标识信息更新策略预先确定,所述第二标识信息更新策略

用于更新每次写入操作关联的标识信息。

[0045] 通过第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片,实现了对内存空间中的场景数据分片进行更新的效果,保证后续第一组件对场景数据分片进行仿真计算的可靠性和准确性。

[0046] 根据本实施例的技术方案,通过第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;第二组件按照第二内存空间切换策略,向各内存空间中轮流写入超长场景中新的场景数据分片,以替换已完成仿真计算的场景数据分片,使得对场景数据分片的仿真计算和写入可以交替进行,从而有效降低超长场景回放仿真过程中存储设备的使用率,提高仿真服务的稳定性和准确性。

[0047] 在上述实施例的基础上,本实施中的超长场景的回放型仿真方法,由自动驾驶仿真系统中,至少一个感知单元的计算模块执行,其中,感知单元包括但不限于加速度计和陀螺仪。

[0048] 通过由感知单元的计算模块执行本实施中的超长场景的回放型仿真方法,使得本方法可应用于车辆或飞机等设备的自动驾驶技术上。

[0049] 在上述实施例的基础上,S101包括:

[0050] 第三组件预先下载超长场景中至少两个场景数据分片,并将各所述场景数据分片写入磁盘中;第二组件获取所述磁盘中存储的至少两个场景数据分片,并将所述至少两个场景数据分片存储于至少两个内存空间中。

[0051] 其中,第三组件用于向磁盘中写入场景数据分片,可选的包括自动驾驶仿真系统中各传感模块中计算模块内的数据下载单元。

[0052] 具体的,第三组件从服务平台中下载超长场景中至少两个场景数据分片,并将该至少两个场景数据分片写入到与计算模块对应的磁盘中。第二组件访问磁盘,并获取磁盘中存储的至少两个场景数据分片,最终将该至少两个场景数据分片存储于与内存中的至少两个内存空间。

[0053] 通过第三组件预先下载超长场景中至少两个场景数据分片,并将各场景数据分片写入磁盘中,第二组件获取磁盘中存储的至少两个场景数据分片,并将至少两个场景数据分片存储于至少两个内存空间中,实现了先在磁盘中暂存场景数据分片,当有需要时,再将场景数据分片存储于内存空间中的技术效果,从而减轻了内存空间的压力,保证所有场景数据分片都可以写入内存空间,进而保证对所有场景数据分片都能进行仿真计算,提高了仿真服务的稳定性和准确性。

[0054] 在上述实施例的基础上,所述方法还包括:

[0055] 第二组件在获取所述超长场景中新的场景数据分片写入对应的内存空间后,检测是否满足第二写入触发条件,当满足第二写入触发条件时,向第三组件发送磁盘写入通知;第三组件每当检测到磁盘写入通知时,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中,以替换所述磁盘中当前已处理的场景数据分片。

[0056] 可选的,所述第二写入触发条件包括:所述磁盘中当前未处理的场景数据分片的数量小于等于第二数量阈值,其中,第二数量阈值可根据需求进行设置。

[0057] 示例性的,假设第二数量阈值为3,磁盘中共存储有10个场景数据分片,其中,当前未处理的场景数据分片的数量为2,当前未处理的场景数据分片的数量为8,由于磁盘中当前未处理的场景数据分片的数量“2”小于第二数量阈值“3”,则第二组件向第三组件发送磁盘写入通知,第三组件检测到该磁盘写入通知,从服务平台中获取超长场景中至少两个新的场景数据分片,并将获取的场景数据分片更新写入磁盘中,以替换所述磁盘中当前已处理场景数据分片。

[0058] 通过当满足第二写入触发条件时,第二组件向第三组件发送磁盘写入通知,第三组件每当检测到磁盘写入通知时,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中,以替换所述磁盘中当前已处理的场景数据分片,使得磁盘中的场景数据分片能够时刻更新,保证后续对场景数据分片的仿真计算能够正常进行。

[0059] 在上述实施例的基础上,S102中“第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算”,包括:

[0060] 第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算,直至满足结束仿真条件。

[0061] 具体的,第一组件按照第一内存空间切换策略,向所有待读取场景数据分片的内存空间中轮流读取场景数据分片,并进行仿真计算,直至将结束仿真条件中包括的所有场景数据分片遍历进行仿真计算后停止。

[0062] 通过第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各场景数据分片,并对各场景数据分片进行仿真计算,直至满足结束仿真条件,使得所有待读取的场景数据分片都能被仿真计算,保证了仿真结果的完整性和可靠性。

[0063] 实施例二

[0064] 图2是根据本申请实施例二公开的另一种超长场景的回放型仿真方法的流程图,基于上述技术方案进一步优化与扩展,为上述实施例一的S102中“第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算”提供了一种可选的实施方式。如图2所示,该方法可以包括:

[0065] S201、第一组件获取第一寄存器当前值,第一寄存器当前值初始化为一个内存空间的存储编号。

[0066] 其中,第一寄存器当前值表示当前待读取场景数据分片所属内存空间对应的存储编号。

[0067] 示例性的,假设内存中有6个内存空间,则将这6个内存空间分别设置存储编号为1、2、3、4、5和6,若当前待读取场景数据分片所属内存空间的存储编号为3,则第一寄存器当前值为3。

[0068] S202、第一组件获取与第一寄存器当前值匹配的第一目标内存空间后,读取第一目标内存空间中存储的场景数据分片,并对读取的场景数据分片进行仿真计算。

[0069] 具体的,第一组件根据获取的第一寄存器当前值,与各内存空间对应的存储编号进行匹配,将与第一寄存器当前值相同的存储编号对应的内存空间作为第一目标内存空

间,进而访问该第一目标内存空间以读取存储的场景数据分片,读取完成后对该场景数据分片进行仿真计算,其中,仿真计算的方法包括但不限于时域有限差分法、矩量法和有限元法等。

[0070] S203、第一组件按照第一寄存器编号更新策略,对第一寄存器当前值进行更新后,返回执行获取与第一寄存器当前值匹配的目标内存空间的操作。

[0071] 其中,第一寄存器编号更新策略用于指示第一组件,按照其预设更新规则对第一寄存器当前值进行更新,以使得第一组件可以根据更新后的第一寄存器当前值不断地在未读取的内存单元中进行场景数据分片读取,并进行仿真计算,以实现超长场景中全部场景数据分片进行仿真计算的效果。

[0072] 可选的,S203中“第一组件按照第一寄存器编号更新策略,对第一寄存器当前值进行更新”,包括A和B两个步骤:

[0073] A、第一组件如果确定第一寄存器当前值不为末位内存空间的存储编号,则将第一寄存器当前值加1的结果,更新为新的第一寄存器当前值。

[0074] 示例性的,假设内存中共划分有6个内存空间,分别设置存储编号为1、2、3、4、5和6。若第一寄存器当前值为“3”,不为末位内存空间的存储编号“6”,则将“3+1”=“4”,作为更新后新的第一寄存器当前值。

[0075] B、第一组件如果确定第一寄存器当前值为末位内存空间的存储编号,则将首位内存空间的存储编号,更新为新的第一寄存器当前值,其中,各所述内存空间预先进行顺序编号。

[0076] 示例性的,假设内存中共划分有6个内存空间,分别设置存储编号为1、2、3、4、5和6。若第一寄存器当前值为“6”,为末位内存空间的存储编号,则将首位内存空间的存储编号“1”,作为更新后新的第一寄存器当前值。

[0077] 通过第一寄存器编号更新策略对第一寄存器当前值进行更新,使得第一组件可轮流的读取各内存空间中的场景数据分片,直至完成对超长场景中全部场景数据分片的仿真计算,保证所有场景数据分片都可以进行仿真计算,提高了仿真服务的稳定性和准确性。

[0078] 在本实施例中,第一组件通过获取第一寄存器当前值匹配的第一目标内存空间,进而读取第一目标内存空间中存储的场景数据分片,并对读取的场景数据分片进行仿真计算,实现了根据第一寄存器当前值对相应的场景数据分片进行仿真计算的效果;通过按照第一寄存器编号更新策略,对第一寄存器当前值进行更新后,返回执行获取与第一寄存器当前值匹配的目标内存空间的操作,使得第一组件可轮流的读取各内存空间中的场景数据分片,直至完成对超长场景中全部场景数据分片的仿真计算,保证所有场景数据分片都可以进行仿真计算,提高了仿真服务的稳定性和准确性。

[0079] 实施例三

[0080] 图3是根据本申请实施例三公开的另一种超长场景的回放型仿真方法的流程图,基于上述技术方案进一步优化与扩展,为上述实施例一的S103中“第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片”提供了一种可选的实施方式。如图3所示,该方法可以包括:

[0081] S301、第二组件按照第二寄存器编号更新策略,更新第二寄存器当前值;第二寄存

器当前值初始化为一个内存空间的存储编号。

[0082] 其中,第二寄存器当前值表示当前待写入场景数据分片所属内存空间对应的存储编号。第二寄存器编号更新策略用于指示第二组件,按照其预设更新规则对第二寄存器当前值进行更新,以使得第二组件可以根据更新前后的第二寄存器当前值,在对应至少一个内存空间中进行场景数据分片写入操作。

[0083] 示例性的,假设第一数量阈值为3,内存中共有6个内存空间分别存储有场景数据分片,若内存空间中当前未处理的场景数据分片的数量为3,由于内存空间中当前未处理的场景数据分片的数量“3”等于第一数量阈值“3”,则数据写入单元按照第二寄存器编号更新策略,更新第二寄存器当前值。

[0084] 通过将第一写入触发条件设置为内存空间中当前未处理的场景数据分片的数量小于等于第一数量阈值,使得当前未处理的场景数据分片的数量较少时,就按照第二寄存器编号更新策略,更新第二寄存器当前值,为后续在内存空间中写入新的场景数据分片奠定基础。

[0085] 可选的,S301中“第二组件按照第二寄存器编号更新策略,更新第二寄存器当前值”,包括A、B和C三个步骤:

[0086] A、第二组件将更新前第二寄存器当前值与设定增量值进行累加。

[0087] 其中,设定增量值的范围上限是根据内存空间数量减去上述第一数量阈值再减去数值“一”确定的,例如假设内存空间数量为“6”,第一数量阈值为“2”,则 $6-2-1=3$,从而设定增量的范围上限为“3”,换言之,设定增量可设置为[1,3]之间的整数。

[0088] B、第二组件如果确定累加结果未超过末位内存空间的存储编号,则将所述累加结果更新为新的第二寄存器当前值。

[0089] 示例性的,假设内存中有6个内存空间,则将这6个内存空间分别设置存储编号为1、2、3、4、5和6,更新前第二寄存器当前值为“2”,设定增量值为“3”,则累加结果为“5”,未超过末位内存空间的存储编号“6”,则将“5”更新为新的第二寄存器当前值。

[0090] C、第二组件如果确定累加结果超过末位内存空间的存储编号,则计算所述累加结果与所述末位内存空间的存储编号之间的差值,并将所述差值更新为新的第二寄存器当前值。

[0091] 示例性的,假设内存中有6个内存空间,则将这6个内存空间分别设置存储编号为1、2、3、4、5和6,更新前第二寄存器当前值为“4”,设定增量值为“3”,则累加结果为“7”,超过末位内存空间的存储编号“6”,则将“ $7-6=1$ ”更新为新的第二寄存器当前值。

[0092] 通过将更新前第二寄存器当前值与设定增量值进行累加,并根据累加结果与末位内存空间的存储编号的大小关系,确定新的第二寄存器当前值,为后续根据更新前后的第二寄存器当前值,确定至少一个第二目标内存空间,奠定了基础。

[0093] S302、第二组件根据更新前后的第二寄存器当前值确定至少一个第二目标内存空间。

[0094] 具体的,将更新前的第二寄存器当前值对应的内存空间,以及更新后的第二寄存器当前值对应的内存空间之间的内存空间,作为第二目标内存空间。

[0095] 例如,假设内存中有6个内存空间,则将这6个内存空间分别设置存储编号为1、2、3、4、5和6,更新前的第二寄存器当前值为“1”,更新后的第二寄存器当前值为“3”,则将存储

编号为“1”、“2”和“3”的内存空间作为第二目标内存空间。

[0096] S303、第二组件按照由更新前第二寄存器当前值对应的第二目标内存空间,朝向更新后第二寄存器当前值对应的第二目标内存空间的顺序,获取所述超长场景中新的场景数据分片写入第二目标内存空间。

[0097] 示例性的,假设内存中有6个内存空间,则将这6个内存空间分别设置存储编号为1、2、3、4、5和6,更新前的第二寄存器当前值为“5”,更新后的第二寄存器当前值为“2”,则按照从存储编号为“5”的内存空间,朝向存储编号为“2”的内存空间的顺序,向各第二目标内存空间写入超长场景中新的场景数据分片,即按照存储编号为“5”、“6”、“1”和“2”的先后顺序向对应内存空间内写入超长场景中新的场景数据分片。

[0098] 在本实施例中,通过根据第二组件按照第二寄存器编号更新策略,更新第二寄存器当前值,根据更新前后的第二寄存器当前值确定至少一个第二目标内存空间,按照由更新前第二寄存器当前值对应的第二目标内存空间,朝向更新后第二寄存器当前值对应的第二目标内存空间的顺序,获取所述超长场景中新的场景数据分片写入第二目标内存空间,保证第一组件从内存空间读取并仿真计算场景数据分片时,都无需先等待场景数据分片的写入,提高了场景数据分片的读取和仿真计算效率,进一步地提高仿真服务的稳定性和准确性。

[0099] 在上述实施例的基础上,S303中“第二组件获取所述超长场景中新的场景数据分片写入第二目标内存空间”,具体包括:

[0100] 第二组件从磁盘中获取所述新的场景数据分片,并将所述新的场景数据分片写入所述第二目标内存空间中。

[0101] 具体的,第二组件访问磁盘,并从磁盘中获取新的场景数据分片,进而将获取的新的场景数据分片写入所述第二目标内存空间中。

[0102] 通过从磁盘中获取新的场景数据分片,并将新的场景数据分片写入所述第二目标内存空间中,实现了先在磁盘中暂存新的场景数据分片,当有需要时,再将新的场景数据分片存储于内存空间中的技术效果,从而减轻了内存空间的压力。

[0103] 实施例四

[0104] 图4是根据本申请实施例四公开的一种通过三个独立线程执行第一组件、第二组件以及第三组件,以实现本实施例中所述的超长场景的回放型仿真方法的示意图。具体过程如下:

[0105] A、通过第一线程,预先下载超长场景中至少两个场景数据分片,并将各所述场景数据分片写入磁盘中。

[0106] 具体的,通过第一线程调用第三组件,以从服务平台中下载超长场景中至少两个场景数据分片,并将各场景数据分片写入磁盘中。

[0107] B、通过第二线程,获取所述磁盘中存储的至少两个场景数据分片,并将所述至少两个场景数据分片存储于至少两个内存空间中后,进入休眠状态。

[0108] 具体的,通过第二线程调用第二组件,以访问磁盘并获取磁盘中存储的至少两个场景数据分片,并将至少两个场景数据分片存储于至少两个内存空间中后,第二线程停止运行,进入休眠状态。

[0109] 通过第二线程将至少两个场景数据分片存储于至少两个内存空间中后,进入休眠

状态,避免了不必要的资源浪费。

[0110] C、通过第三线程,按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;在对各场景数据分片进行仿真计算的过程中,根据第一写入触发条件,唤醒第二线程。

[0111] 具体的,通过第三线程调用第一组件,按照第一内存空间切换策略,从各内存空间中轮流读取各场景数据分片,并对各场景数据分片进行仿真计算。当内存空间中当前未处理的场景数据分片的数量小于等于第一数量阈值时,第三线程向第二线程发送控制指令,以唤醒处于休眠状态的第二线程,以使得第二线程继续执行相应操作。

[0112] D、通过所述第二线程,在被唤醒后,根据第二写入触发条件,重新触发所述第一线程,并按照第二内存空间切换策略,从磁盘中获取所述新的场景数据分片写入对应的内存空间中;在完成单次数据写入操作后,重新进入休眠状态。

[0113] 具体的,第二线程被唤醒后,若磁盘中当前未处理的场景数据分片的数量小于等于第二数量阈值时,则重新触发第一线程,而第二线程按照第二内存空间切换策略,向各内存空间中轮流写入超长场景中新的场景数据分片,当完成本次的数据写入操作后,第二线程停止运行,重新进入休眠状态。

[0114] E、通过所述第一线程,在被触发后,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中。

[0115] 具体的,第一线程被触发后,从服务平台中获取超长场景中至少两个新的场景数据分片,并将获取的新的场景数据分片更新写入磁盘中。

[0116] 在本实施例中,通过第一线程、第二线程以及第三线程执行第一组件、第二组件以及第三组件,以实现本实施例中所述的超长场景的回放型仿真方法,使得场景数据分片的下载、写入、读取及仿真计算可以同步进行,避免了仿真计算出现停滞的情况,从而提高仿真服务的稳定性和准确性。

[0117] 实施例五

[0118] 图5是根据本申请实施例五公开的一种超长场景的回放型仿真装置的结构示意图,本实施例可以适用于根据采集的场景数据,通过车载的各感知单元的计算模块进行场景回放仿真的情况。本实施例装置可采用软件和/或硬件实现,并可集成在任意的具有计算能力的电子设备上,例如服务器等。

[0119] 如图5所示,本实施例公开的超长场景的回放型仿真装置50可以包括数据存储模块51、仿真计算模块52和场景数据写入模块53,其中:

[0120] 数据存储模块51,用于预先获取超长场景中至少两个场景数据分片,并将各场景数据分片分别存储于不同的内存空间中;

[0121] 仿真计算模块52,用于第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;以及,在对每个场景数据分片进行仿真计算后,检测是否满足第一写入触发条件,当满足第一写入触发条件时,向第二组件发送写入通知;

[0122] 场景数据写入模块53,用于第二组件每当接收到写入通知时,按照第二内存空间切换策略,获取所述超长场景中新的场景数据分片写入对应的内存空间中,以替换已完成仿真计算的场景数据分片。

- [0123] 可选的,所述仿真计算模块52,具体用于:
- [0124] 第一组件获取第一寄存器当前值,第一寄存器当前值初始化为一个内存空间的存储编号;
- [0125] 第一组件获取与第一寄存器当前值匹配的第一目标内存空间后,读取第一目标内存空间中存储的场景数据分片,并对读取的场景数据分片进行仿真计算;
- [0126] 第一组件按照第一寄存器编号更新策略,对第一寄存器当前值进行更新后,返回执行获取与第一寄存器当前值匹配的目标内存空间的操作。
- [0127] 可选的,所述仿真计算模块52,具体还用于:
- [0128] 第一组件如果确定第一寄存器当前值不为末位内存空间的存储编号,则将第一寄存器当前值加1的结果,更新为新的第一寄存器当前值;
- [0129] 第一组件如果确定第一寄存器当前值为末位内存空间的存储编号,则将首位内存空间的存储编号,更新为新的第一寄存器当前值,其中,各所述内存空间预先进行顺序编号。
- [0130] 可选的,所述场景数据写入模块53,具体还用于:
- [0131] 第二组件每当接收到写入通知时,执行:
- [0132] 第二组件按照第二寄存器编号更新策略,更新第二寄存器当前值;第二寄存器当前值初始化为一个内存空间的存储编号;
- [0133] 第二组件根据更新前后的第二寄存器当前值确定至少一个第二目标内存空间;
- [0134] 第二组件按照由更新前第二寄存器当前值对应的第二目标内存空间,朝向更新后第二寄存器当前值对应的第二目标内存空间的顺序,获取所述超长场景中新的场景数据分片写入第二目标内存空间。
- [0135] 可选的,所述场景数据写入模块53,具体还用于:
- [0136] 第二组件将更新前第二寄存器当前值与设定增量值进行累加;
- [0137] 第二组件如果确定累加结果未超过末位内存空间的存储编号,则将所述累加结果更新为新的第二寄存器当前值;
- [0138] 第二组件如果确定累加结果超过末位内存空间的存储编号,则计算所述累加结果与所述末位内存空间的存储编号之间的差值,并将所述差值更新为新的第二寄存器当前值。
- [0139] 可选的,所述数据存储模块51,具体用于:
- [0140] 第三组件预先下载超长场景中至少两个场景数据分片,并将各所述场景数据分片写入磁盘中;
- [0141] 第二组件获取所述磁盘中存储的至少两个场景数据分片,并将所述至少两个场景数据分片存储于至少两个内存空间中。
- [0142] 可选的,所述装置还包括磁盘写入模块,具体用于:
- [0143] 第二组件在获取所述超长场景中新的场景数据分片写入对应的内存空间后,检测是否满足第二写入触发条件,当满足第二写入触发条件时,向第三组件发送磁盘写入通知;
- [0144] 第三组件每当检测到磁盘写入通知时,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中,以替换所述磁盘中当前已处理的场景数据分片;

[0145] 第二组件从磁盘中获取所述新的场景数据分片,并将所述新的场景数据分片写入所述第二目标内存空间中。可选的,第一写入触发条件包括:所述内存空间中当前未处理的场景数据分片的数量小于等于第一数量阈值;

[0146] 第二写入触发条件包括:所述磁盘中当前未处理的场景数据分片的数量小于等于第二数量阈值。

[0147] 可选的,所述仿真计算模块52,具体还用于:

[0148] 第一组件按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算,直至满足结束仿真条件。

[0149] 可选的,所述装置设置有三个独立线程,所述装置还包括线程切换模块,具体用于:

[0150] 通过第一线程,预先下载超长场景中至少两个场景数据分片,并将各所述场景数据分片写入磁盘中;

[0151] 通过第二线程,获取所述磁盘中存储的至少两个场景数据分片,并将所述至少两个场景数据分片存储于至少两个内存空间中后,进入休眠状态;

[0152] 通过第三线程,按照第一内存空间切换策略,从各内存空间中轮流读取各所述场景数据分片,并对各场景数据分片进行仿真计算;在对各场景数据分片进行仿真计算的过程中,根据第一写入触发条件,唤醒第二线程;

[0153] 通过所述第二线程,在被唤醒后,根据第二写入触发条件,重新触发所述第一线程,并按照第二内存空间切换策略,从磁盘中获取所述新的场景数据分片写入对应的内存空间中;在完成单次数据写入操作后,重新进入休眠状态;

[0154] 通过所述第一线程,在被触发后,获取超长场景中至少两个新的场景数据分片,并将获取的所述场景数据分片更新写入磁盘中。

[0155] 可选的,所述装置配置于自动驾驶仿真系统中的至少一个感知单元的计算模块中。

[0156] 本申请实施例所公开的超长场景的回放型仿真装置50可执行本申请实施例所公开的超长场景的回放型仿真方法,具备执行方法相应的功能模块和有益效果。本实施例中未详尽描述的内容可以参考本申请任意方法实施例中的描述。

[0157] 根据本申请的实施例,本申请还提供了一种电子设备和一种可读存储介质。

[0158] 如图6所示,是根据本申请实施例的超长场景的回放型仿真方法的电子设备的框图。电子设备旨在表示各种形式的数字计算机,诸如,膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置,诸如,个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例,并且不意在限制本文中描述的和/或者要求的本申请的实现。

[0159] 如图6所示,该电子设备包括:一个或多个处理器601、存储器602,以及用于连接各部件的接口,包括高速接口和低速接口。各个部件利用不同的总线互相连接,并且可以被安装在公共主板上或者根据需要以其它方式安装。处理器可以对在电子设备内执行的指令进行处理,包括存储在存储器中或者存储器上以在外部输入/输出装置(诸如,耦合至接口的显示设备)上显示GUI的图形信息的指令。在其它实施方式中,若需要,可以将多个处理器

和/或多条总线与多个存储器和多个存储器一起使用。同样,可以连接多个电子设备,各个设备提供部分必要的操作(例如,作为服务器阵列、一组刀片式服务器、或者多处理器系统)。图6中以一个处理器601为例。

[0160] 存储器602即为本申请所提供的非瞬时计算机可读存储介质。其中,所述存储器存储有可由至少一个处理器执行的指令,以使所述至少一个处理器执行本申请所提供的超长场景的回放型仿真方法。本申请的非瞬时计算机可读存储介质存储计算机指令,该计算机指令用于使计算机执行本申请所提供的超长场景的回放型仿真方法。

[0161] 存储器602作为一种非瞬时计算机可读存储介质,可用于存储非瞬时软件程序、非瞬时计算机可执行程序以及模块,如本申请实施例中的超长场景的回放型仿真方法对应的程序指令/模块(例如,附图5所示的数据存储模块51、仿真计算模块52和场景数据写入模块53)。处理器601通过运行存储在存储器602中的非瞬时软件程序、指令以及模块,从而执行服务器的各种功能应用以及数据处理,即实现上述方法实施例中的超长场景的回放型仿真方法。

[0162] 存储器602可以包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需要的应用程序;存储数据区可存储根据超长场景的回放型仿真的电子设备的使用所创建的数据等。此外,存储器602可以包括高速随机存取存储器,还可以包括非瞬时存储器,例如至少一个磁盘存储器件、闪存器件、或其他非瞬时固态存储器件。在一些实施例中,存储器602可选包括相对于处理器601远程设置的存储器,这些远程存储器可以通过网络连接至超长场景的回放型仿真的电子设备。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。

[0163] 超长场景的回放型仿真方法的电子设备还可以包括:输入装置603和输出装置604。处理器601、存储器602、输入装置603和输出装置604可以通过总线或者其他方式连接,图6中通过总线连接为例。

[0164] 输入装置603可接收输入的数字或字符信息,以及产生与超长场景的回放型仿真的电子设备的用户设置以及功能控制有关的键信号输入,例如触摸屏、小键盘、鼠标、轨迹板、触摸板、指示杆、一个或者多个鼠标按钮、轨迹球、操纵杆等输入装置。输出装置604可以包括显示设备、辅助照明装置(例如,LED)和触觉反馈装置(例如,振动电机)等。该显示设备可以包括但不限于,液晶显示器(LCD)、发光二极管(LED)显示器和等离子体显示器。在一些实施方式中,显示设备可以是触摸屏。

[0165] 此处描述的系统和技术和各种实施方式可以在数字电子电路系统、集成电路系统、专用ASIC(专用集成电路)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括:实施在一个或者多个计算机程序中,该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释,该可编程处理器可以是专用或者通用可编程处理器,可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令,并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0166] 这些计算程序(也称作程序、软件、软件应用、或者代码)包括可编程处理器的机器指令,并且可以利用高级过程和/或面向对象的编程语言、和/或汇编/机器语言来实施这些计算程序。如本文使用的,术语“机器可读介质”和“计算机可读介质”指的是用于将机器指

令和/或数据提供给可编程处理器的任何计算机程序产品、设备、和/或装置(例如,磁盘、光盘、存储器、可编程逻辑装置(PLD)),包括,接收作为机器可读信号的机器指令的机器可读介质。术语“机器可读信号”指的是用于将机器指令和/或数据提供给可编程处理器的任何信号。

[0167] 为了提供与用户的交互,可以在计算机上实施此处描述的系统和技术,该计算机具有:用于向用户显示信息的显示装置(例如,CRT(阴极射线管)或者LCD(液晶显示器)监视器);以及键盘和指向装置(例如,鼠标或者轨迹球),用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何形式的传感反馈(例如,视觉反馈、听觉反馈、或者触觉反馈);并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0168] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如,作为数据服务器)、或者包括中间件部件的计算系统(例如,应用服务器)、或者包括前端部件的计算系统(例如,具有图形用户界面或者网络浏览器的用户计算机,用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信(例如,通信网络)来将系统的部件相互连接。通信网络的示例包括:局域网(LAN)、广域网(WAN)和互联网。

[0169] 计算机系统可以包括客户端和服务端。客户端和服务端一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务端关系的计算机程序来产生客户端和服务端的关系。

[0170] 根据本申请实施例的技术方案,实现了有效降低超长场景回放仿真过程中存储设备的使用率,提高仿真服务的稳定性和准确性。

[0171] 应该理解,可以使用上面所示的各种形式的流程,重新排序、增加或删除步骤。例如,本申请中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行,只要能够实现本申请公开的技术方案所期望的结果,本文在此不进行限制。

[0172] 上述具体实施方式,并不构成对本申请保护范围的限制。本领域技术人员应该明白的是,根据设计要求和因素,可以进行各种修改、组合、子组合和替代。任何在本申请的精神和原则之内所作的修改、等同替换和改进等,均应包含在本申请保护范围之内。

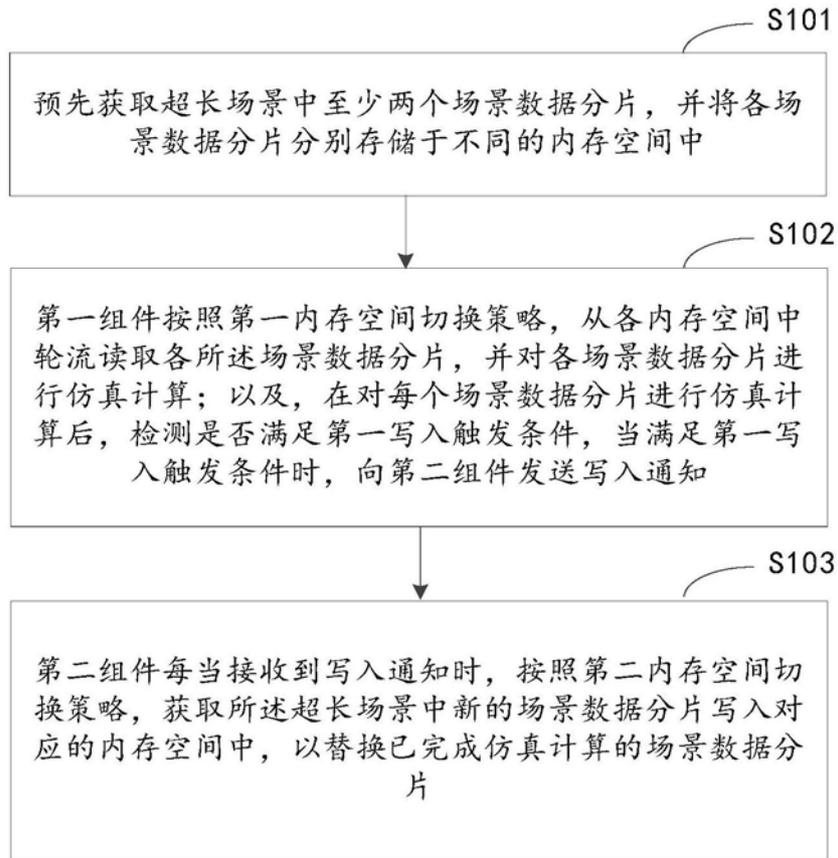


图1

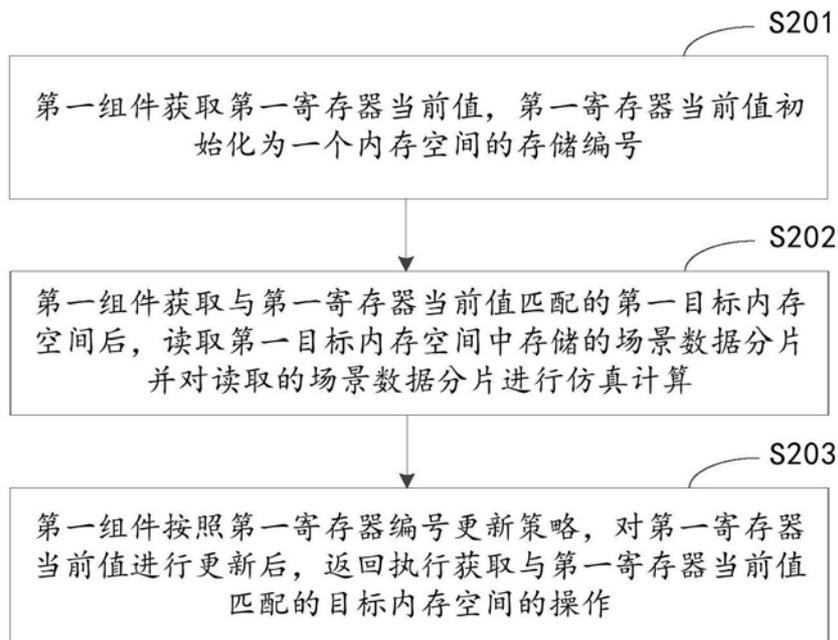


图2

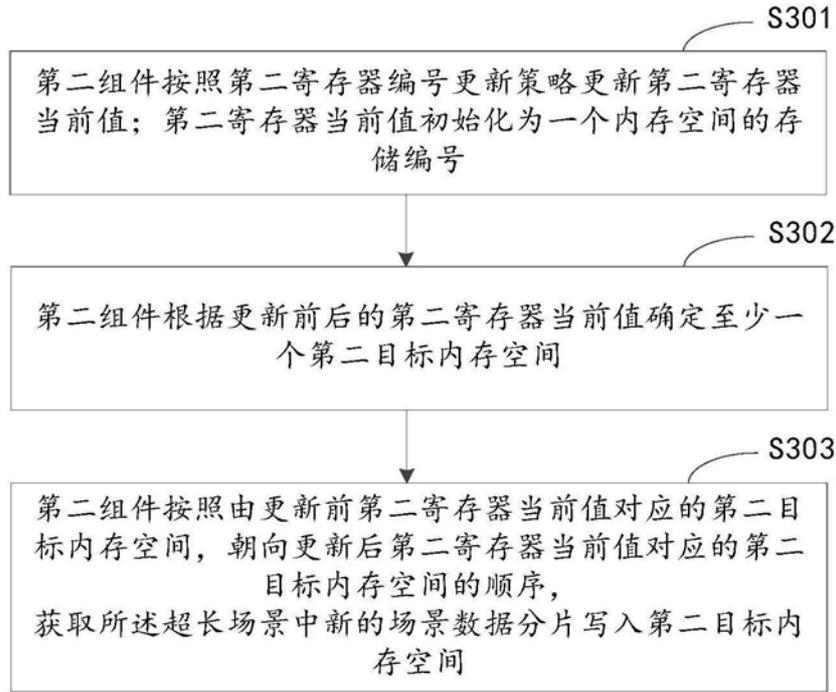


图3

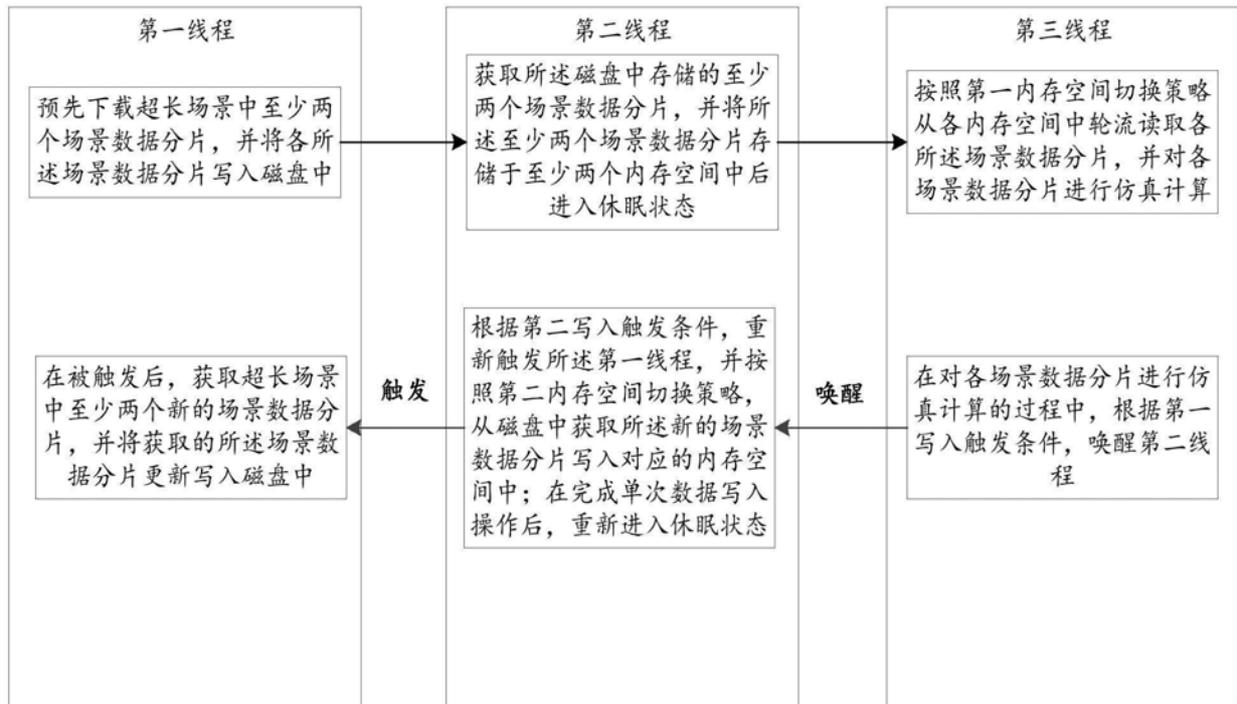


图4

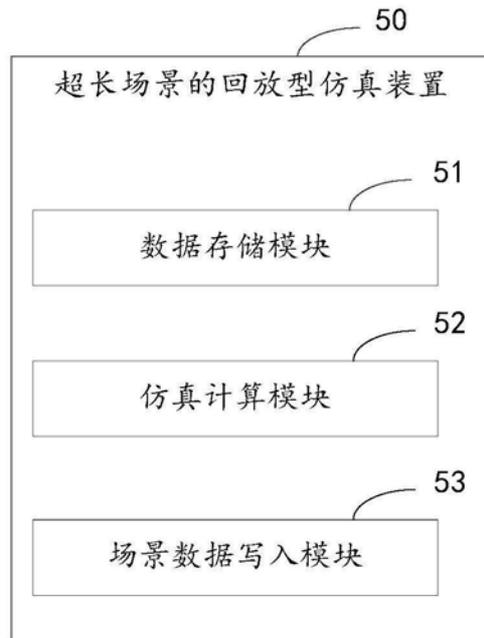


图5

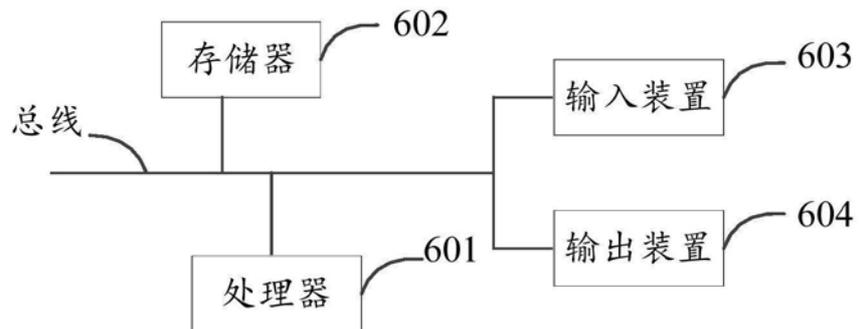


图6