



(12) 发明专利申请

(10) 申请公布号 CN 104778299 A

(43) 申请公布日 2015. 07. 15

(21) 申请号 201510051451. 2

(22) 申请日 2015. 02. 01

(71) 申请人 浙江成功软件开发有限公司

地址 310012 浙江省杭州市文三路 583 号康
新商务大厦 c 座 2001-2004 室

(72) 发明人 张子健 陆宇飞 陈奇 江洁羽

(74) 专利代理机构 杭州求是专利事务所有限公
司 33200

代理人 邱启旺

(51) Int. Cl.

G06F 17/50(2006. 01)

权利要求书1页 说明书3页 附图2页

(54) 发明名称

并行微观交通仿真中非阻塞式的车辆移动方
法

(57) 摘要

本发明公开了一种并行微观交通仿真中非阻塞式的车辆移动方法,该方法首先为每个路段中的车道创建一个车辆管理装置,用于管理车道上的车辆,并将组成路网的所有路段划分成多个组,每个组分配一个线程负责仿真车辆的状态。当车辆跨路段行驶时,则会带来并发与线程竞争问题,为解决这个问题,本方法利用一条硬件级的原子指令 CAS 指令,实现基于冲突检测的乐观并发策略,也就是非阻塞式的同步方案,避免了锁的使用,降低了并发仿真环境下线程间交互的性能损耗,使得整个微观交通仿真系统的性能有了明显提升,满足了城市规模道路交通系统的实时仿真需要。

1. 一种并行微观交通仿真中非阻塞式的车辆移动方法,其特征在于,包括以下步骤:

(1) 仿真初始化阶段,包括以下子步骤:

(1.1) 为每一个路段中的每一条车道创建一个车辆管理装置(LVC);

(1.2) 将每一个路段映射到一个线程;

(2) 仿真进行阶段,若车辆仅在路段内部移动,则不涉及线程间的交互,否则,若车辆从上游路段的车道进入下游路段中的车道,执行以下操作:

(2.1) 上游路段所处的线程将车辆从当前车道的LVC中除去,并将车辆标记为游离状态,表示车辆不再属于任何路段;

(2.2) 上游路段所处的线程将车辆提交到下游车道的缓冲区中;

(2.3) 下游路段的线程从缓冲区中提取车辆并把车辆加入到下游车道的LVC中,从而在并发环境下高性能地实现车辆的跨路段移动。

2. 根据权利要求1所述的并行微观交通仿真中非阻塞式的车辆移动方法,其特征在于,所述步骤1.1中,每个LVC与每个路段中的每条车道是一一对应关系,用于管理车道上的车辆,LVC由两部分组成:一个缓冲区,最多容纳一个车辆对象的指针,用于保存刚刚从车道上游新进入的车辆;一个车辆容器,用于保存车道上的车辆,车辆按位置的先后顺序排序。

3. 根据权利要求1所述的并行微观交通仿真中非阻塞式的车辆移动方法,其特征在于,所述步骤1.2具体为:首先确定线程数量,用户可通过参数指定线程数,若未指定,则默认线程数量即为CPU核心数量,假设线程数量为n,则创建n个线程,存放在线程池之中;接下来把整个路网包含的所有路段分为n组,每组固定分配给一个线程,组内所有路段中的车辆状态更新由该线程驱动;路段分组方法需保证每个组的路段数尽可能相等,以使每个线程负载均衡。

4. 根据权利要求1所述的并行微观交通仿真中非阻塞式的车辆移动方法,其特征在于,所述步骤2.2具体为:利用compare-and-swap(CAS)指令,上游线程将车辆与下游车道缓冲区中的值进行一次原子交换,若交换得到空值,则表示原缓冲区为空,表示车辆已经成功放入缓冲区中;否则上游线程交换得到另一辆车,说明缓冲区中已有其它车辆,表示车辆发生碰撞,流程终止。

5. 根据权利要求1所述的并行微观交通仿真中非阻塞式的车辆移动方法,其特征在于,所述步骤2.3具体为:利用CAS指令,下游线程将空值与缓存区中的值进行一次原子交换,交换后下游线程得到缓冲区中的值(车辆或空值),而缓冲区被换入空值;交换得到两种结果:a)若下游线程交换得到空值,说明缓冲区中原本就没有车辆,此时终止流程;b)若下游线程交换得到的值非空,则该值即为从上游车道驶入的车辆,此时将车辆加入到下游车道的LVC中,将车辆状态更新为属于下游路段。

并行微观交通仿真中非阻塞式的车辆移动方法

技术领域

[0001] 本发明涉及系统仿真领域、微观交通仿真领域,尤其涉及一种并行环境下的微观交通仿真中非阻塞式的车辆移动方法。

背景技术

[0002] 交通问题一直是一个严重的社会问题,制约着城市的发展。由于道路交通系统是一个复杂的巨系统,单纯的理论研究或实验方法都不能很好的胜任,而基于计算机仿真技术的微观交通仿真系统则是分析和解决交通问题的有效手段。由于微观交通仿真系统需要模拟每一辆车的行为,因此对计算机的性能要求较高,传统的微观交通仿真系统不能仿真大范围的交通系统。

[0003] 目前已有的并行微观交通仿真方法分两类,一类是创建多个线程,每个线程完成特定的工作,但线程是非对称的,也就是每个线程负责不同的任务,因此难以平衡线程之间的负载,也不能使线程数量根据 CPU 核心数灵活增减;另一类并行仿真方法线程是对等的,但是在控制线程相互作用时使用了锁,即阻塞式的同步方案。无论哪种方案,都还没有达到硬件性能的极限,影响微观仿真系统性能的进一步提升。

发明内容

[0004] 本发明的目的在于针对现有技术的不足,提供一种并行微观交通仿真中非阻塞式的车辆移动方法。

[0005] 本发明的目的是通过以下技术方案来实现的:一种并行微观交通仿真中非阻塞式的车辆移动方法,包括以下步骤:

[0006] (1) 仿真初始化阶段,包括以下子步骤:

[0007] (1.1) 为每一个路段中的每一条车道创建一个车辆管理装置(以下简称 LVC);

[0008] (1.2) 将每一个路段映射到一个线程;

[0009] (2) 仿真进行阶段,若车辆仅在路段内部移动,则不涉及线程间的交互,否则,若车辆从上游路段的车道进入下游路段中的车道,由于不同路段分属于不同的线程,因此会产生并发和竞争,此时需要执行以下操作:

[0010] (2.1) 上游路段所处的线程将车辆从当前车道的 LVC 中除去,并将车辆标记为游离状态,表示车辆不再属于任何路段;

[0011] (2.2) 上游路段所处的线程将车辆提交到下游车道的缓冲区中;

[0012] (2.3) 下游路段的线程从缓冲区中提取车辆并把车辆加入到下游车道的 LVC 中,从而在并发环境下高性能地实现车辆的跨路段移动。

[0013] 进一步地,所述步骤 1.1 中,每个 LVC 与每个路段中的每条车道是一一对应关系,用于管理车道上的车辆,LVC 由两部分组成:一个缓冲区,最多容纳一个车辆对象的指针,用于保存刚刚从车道上游新进入的车辆;一个车辆容器,用于保存车道上的车辆,车辆按位置的先后顺序排序。

[0014] 进一步地,所述步骤 1.2 具体为:首先确定线程数量,用户可通过参数指定线程数,若未指定,则默认线程数量即为 CPU 核心数量,假设线程数量为 n,则创建 n 个线程,存放在线程池之中;接下来把整个路网包含的所有路段分为 n 组,每组固定分配给一个线程,组内所有路段中的车辆状态更新由该线程驱动;路段分组方法没有特殊要求,只需要保证每个组的路段数尽可能相等,以使每个线程负载均衡。

[0015] 进一步地,所述步骤 2.2 中,所述下游路段车道对应的缓冲区属于临界资源,上游线程将车辆提交到该缓冲区使用的是一种基于冲突检测的乐观并发策略,也就是非阻塞式同步 (Non-Blocking Synchronization) 方案,不使用锁,具体为:利用 CAS 指令,上游线程将车辆与缓冲区中的值进行一次原子交换,若交换得到空值,则表示原缓冲区为空,表示车辆已经成功置入缓冲区中;否则上游线程交换得到另一辆车,说明缓冲区中已有其它车辆,表示车辆发生碰撞,流程终止;所述 CAS (compare-and-swap) 指令为硬件级的原子指令,CAS 指令包含 3 个操作数:内存地址 V,旧的预期值 A,新值 B,当且仅当 V 中原有的值和 A 值相等时,处理器用 B 值更新 V 中存储的值,否则就不执行更新,指令的返回值视硬件的不同或者返回一个布尔值表示是否更新成功,或者返回 V 中的旧值(不是新值 B),上述整个操作是原子性的。

[0016] 进一步地,所述步骤 2.3 中,下游路段所处的线程从缓冲区中获取车辆同样使用 CAS 指令,具体为:下游线程将空值与缓存区中的值进行一次原子交换,交换后下游线程得到原缓冲区中的值(车辆),而缓冲区被置空;若下游线程交换后仍为空值,说明缓存区中原本就没有车,此时终止流程;否则,下游线程得到从上游线程驶入的车辆,此时应将车辆加入到下游车道的 LVC 的车辆容器中,并更新车辆的位置状态为隶属于下游车道,同时进行安全性检查;安全性检查指的是判断车辆位置是否与前面车辆重叠,若重叠表示车辆发生碰撞,此时终止流程。

[0017] 与现有技术相比,本发明的有益效果是:1) 线程之间是对称的,平等的,可以任意增加或减少线程数量,可以完全利用不同核心数量的 CPU;2) 显著提高了车辆移动的性能,从而提高了微观交通仿真的整体性能,利用硬件级的原子指令,完全避免了锁的使用,消除了线程之间的同步开销,在普通 4 核心电脑上支持超过 100 万台车辆的仿真,满足城市规模交通的实时仿真。

附图说明

[0018] 图 1 是车道上车辆管理装置 (LVC) 的结构示意图;

[0019] 图 2 是上游车道的线程将车辆提交给下游车道的流程图;

[0020] 图 3 是下游车道的线程接收车辆的流程图。

具体实施方式

[0021] 下面结合附图对本发明作进一步详细说明。

[0022] 本发明一种并行微观交通仿真中非阻塞式的车辆移动方法,包含以下步骤:

[0023] (1) 仿真初始化阶段,包括以下子步骤:

[0024] (1.1) 为每一个路段中的每一条车道创建一个车辆管理装置 (LVC)。

[0025] 车辆管理装置(以下简称 LVC)用于管理车道上的车辆,其结构如图 1 所示,由两

部分组成：一个缓冲区，最多容纳一个车辆对象的指针，用于保存刚刚从车道上游新进入的车辆；一个车辆容器，用于保存车道上的车辆，车辆按位置的先后顺序排序。仿真初始化阶段需要为每个路段中的每条车道创建一个 LVC，LVC 与车道是一一对应关系。

[0026] (1.2) 将每一个路段映射到一个线程。

[0027] 该步骤首先要确定线程数量，用户可通过参数指定线程数，若未指定，则默认线程数量即为 CPU 核心数量，假设线程数量为 n ，则创建 n 个线程，存放在线程池之中，以供后续使用。接下来把整个路网包含的所有路段分为 n 组，每组固定分配给一个线程，组内所有路段中的车辆状态更新由该线程驱动。路段分组方法没有特殊要求，只需要保证每个组的路段数尽可能相等，以使每个线程负载均衡。

[0028] (2) 仿真行进阶段，当车辆在路段之内移动时，不产生进程间的交互，故不需要额外的同步措施。但是当车辆从一个路段进入另一个路段时，由于不同路段分属于不同的线程，因此会产生并发和竞争，此时需要执行以下步骤，即完成非阻塞式的车辆移动：

[0029] (2.1) 上游路段所处的线程将车辆从当前车道的 LVC 中除去，并将车辆标记为游离状态，表示车辆不再属于任何路段。

[0030] 上游线程将车辆从上游车道的 LVC 中删除，并将车辆位置更新为游离状态，表示不再属于任何路段。车辆在跨路段移动过程中是不属于任何路段的，只有在移动完成后才会属于某个特定的路段。因此在移动开始时需要先把车辆的位置更新为游离状态。

[0031] (2.2) 上游路段所处的线程将车辆提交到下游车道的缓冲区中。

[0032] 下游车道 LVC 的缓冲区属于临界资源，上游线程将车辆提交到该缓冲区时本发明使用的是一种基于冲突检测的乐观并发策略，也就是非阻塞式同步 (Non-Blocking Synchronization) 方案，不使用锁。具体为：

[0033] i) 利用一条硬件级的原子指令：compare-and-swap (CAS) 指令，CAS 指令包含 3 个操作数：内存地址 V ，旧的预期值 A ，新值 B ，当且仅当 V 中原有的值和 A 值相等时，处理器用 B 值更新 V 中存储的值，否则就不执行更新，指令的返回值视硬件的不同或者返回一个布尔值表示是否更新成功，或者返回 V 中的旧值（不是新值 B ），最后上述整个操作是原子性的。目前大部分微处理器都已经提供了 CAS 指令，如 x86，IA64 中的 CMPXCHG 指令。

[0034] ii) 利用 CAS 指令，上游线程将车辆与缓冲区中的值进行一次原子交换。若交换得到空值，则表示原缓冲区为空，表示车辆已经成功置入缓冲区中；否则上游线程交换得到另一辆车，说明缓冲区中已有其它车辆，表示车辆发生碰撞，流程终止。

[0035] 步骤 (2.1)、(2.2) 的过程如图 2 所示。

[0036] (2.3) 下游路段所处的线程从缓冲区中获取车辆并把车辆加入到 LVC 中。

[0037] 下游线程从缓冲区提取车辆也是使用 CAS 指令，下游线程将空值与缓存区中的值进行一次原子交换，交换后下游线程得到原缓冲区中的值（车辆），而缓冲区被置空。另一方面，若下游线程交换后仍为空值，说明缓存区中原本就没有车，此时应该终止流程。否则，下游线程得到从上游线程驶入的车辆，此时应将车辆加入到下游车道的 LVC 的车辆容器中，并更新车辆状态，同时进行安全性检查。其中更新车辆状态指的是将车辆位置更新为属于下游路段，安全性检查指的是判断车辆位置是否与前面车辆重叠，若重叠表示车辆发生碰撞，此时应终止流程。整个过程如图 3 所示。



图 1

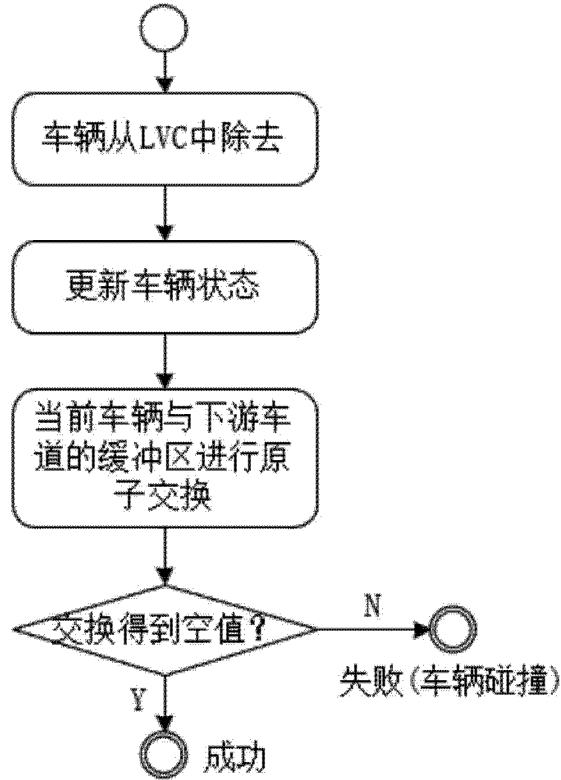


图 2

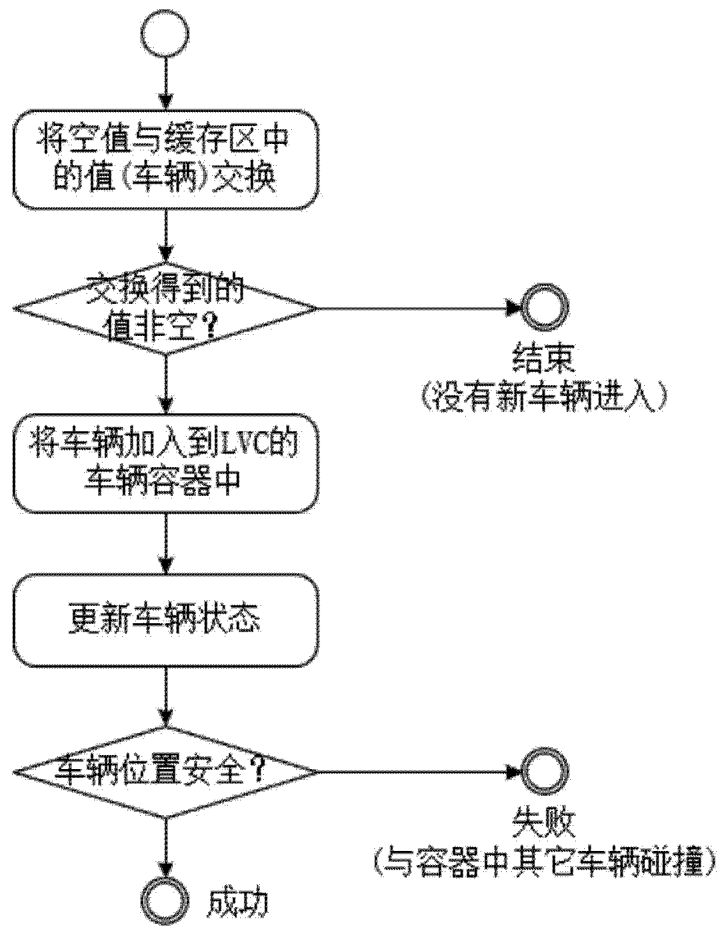


图 3