

一种面向大规模 HLA 仿真的并行区域匹配算法*

梁洪波^{1,2}, 朱卫国¹, 姚益平², 刘步权², 慕晓冬³, 杨勇²

(1. 92117 部队, 北京 100072;

2. 国防科技大学 计算机学院, 湖南 长沙 410073;

3. 第二炮兵工程大学, 陕西 西安 710025)

摘要:HLA 仿真中,数据分发管理实现基于值的过滤,可以有效减少盟员接收冗余数据的可能性和网络中的数据流量。大规模 HLA 仿真系统在仿真推进中需要大量的区域匹配计算以维护数据分发管理的正确性。现有的区域匹配算法大多需要对所有区域进行匹配计算,造成了大量计算资源的浪费;同时,主要基于串行匹配思想,难以充分发挥多核平台的并行计算优势。针对现有区域匹配算法的局限性,提出了一种面向大规模 HLA 仿真的并行区域匹配算法,该算法能够实现对一次仿真推进中多个改变区域的并行匹配计算,同时在匹配计算中采用基于移动相交的基本思想,利用区域范围移动前后的历史信息,将匹配限定在移动区间之内,减少了大量的无关计算。理论分析与实验结果表明该算法尤其适合基于多核计算平台构建大规模分布式仿真的应用需求。

关键词:分布式交互仿真; 运行支撑环境(RTI); 数据分发管理(DDM); 区域匹配; 并行计算

中图分类号:TP391.9 **文献标志码:**A **文章编号:**1001-2486(2013)03-0084-08

A parallel region matching algorithm for large scale HLA simulation

LIANG Hongbo^{1,2}, ZHU Weiguo¹, YAO Yiping², Liu Buquan², MU Xiaodong³, YANG Yong²

(1. Unit 92117, Beijing 100072, China;

2. College of Computer, National University of Defense Technology, Changsha 410073, China;

3. Second Artillery Engineering University, Xi'an 710025, China)

Abstract:In HLA simulation, data distribution management (DDM) service provides data filtering capability, which can effectively reduce the transmission and reception of redundant data. During the advancing of large-scale HLA simulation system, much region matching calculation is needed. However, in most of the existing regional matching algorithms, all regions are calculated in every matching calculation process, which results in a waste of computing resources. Meanwhile, the current matching algorithms are mostly based on serial region matching, and the advantage of multi-core cannot be fully taken. Aiming at the limitation of existing algorithms, a parallel region matching algorithm for large-scale HLA simulation is proposed. By this algorithm, the matching process of multiple regions which changes in the same simulation advancing can be calculated in parallel, the advantage of parallel computing environment can be taken. During the process of matching calculation, a method which is based on intersecting information was used. By the history of moving regions, matching calculation only happened within a limited range, and the redundant calculation could be avoided. Theoretical analysis and experimental results showed that the algorithm is particularly suitable for large-scale distributed simulation applications which execute on multi-core computing platform.

Key words:distributed interactive simulation; runtime infrastructure; data distribution management; region matching; parallel computing

计算机仿真技术是缩短研发和训练周期、提高研发和训练质量、节省研发和训练经费的有效手段,在国民经济和国防建设领域有着十分广泛的应用。为了实现分布于不同地理位置的各仿真组件之间的互操作,提升仿真组件的可重用性,美国国防部建模与仿真办公室(Defense Modeling and Simulation Office, DMSO)于1996年9月正式提出了高层体系结构 HLA(High Level Architecture),

并在2000年9月被定为国际分布仿真通用标准 IEEE 1516^[1]。HLA 定义了一个技术框架,它是一个灵活的、可伸缩的、可重用的软件体系结构,基于它可创建基于组件的分布式仿真,构成系统的各类模块或各类仿真体均可直接接入该框架,并能容易地实现相互间的互操作及仿真组件的可重用^[2]。HLA 规范主要由三部分组成:HLA 规则^[3]、HLA 接口规范^[1]和 HLA 对象模型模板^[4]。在 HLA 仿真

* 收稿日期:2012-10-20

基金项目:国家自然科学基金资助项目(61170047)

作者简介:梁洪波,(1983—),男,河北邢台人,工程师,博士,E-mail:lianghongbo@nudt.edu.cn;

姚益平(通信作者),男,研究员,博士,博士生导师,E-mail:ypiao@nudt.edu.cn

中,一个分布式仿真应用被称为一个联盟(Federation),该仿真应用中的每一个仿真组件称为一个盟员(Federate)^[5]。作为联盟执行的基础设施,运行支撑平台 RTI(Run Time Infrastructure)是 HLA 接口规范的具体实现,它提供了支持联盟成员之间互操作的标准服务实现,这些服务包括联盟管理服务(Federation Management, FM)、声明管理服务(Declaration Management, DM)、对象管理服务(Object Management, OM)、时间管理服务(Time Management, TM)、所有权管理服务(Ownership Management, OWM)和数据分发管理服务(Data Distribution Management, DDM)等^[6]。

基于 HLA/RTI 的大规模分布式仿真应用通常包含分布于不同盟员的数目庞大的仿真实体,这些实体在仿真运行过程中将进行大量的通信,如军事作战仿真中的雷达实体在每一步仿真过程中都需要接收导弹、飞机等众多其他实体的位置信息,而当战场中雷达、导弹、飞机及舰船等作战实体的规模增大时,系统的通信量往往成级数增长,保证这类仿真效率的重要途径是数据分发管理 DDM(Data Distribution Management)的使用。它是在声明管理实现基于类的数据过滤的基础上实现基于值的过滤,其目的是减少仿真运行过程中盟员间无用数据的传输和接收,从而进一步减少仿真中盟员接收冗余数据的可能性和网络中的数据流量。DDM 实现的关键是更新区域和订购区域的匹配(区域交叠)计算,数据发送盟员通过计算更新区域与订购区域是否交叠来决定是否将更新数据发送到订购盟员。随着仿真实体数的增加,在大规模 HLA 仿真的每一次仿真推进中,都会有大量的区域发生移动,为了保证数据分发管理的正确性,需要对区域间的交叠关系重新进行匹配计算,匹配计算的复杂度随着实体数的增加而不断增大,因此区域匹配算法的设计决定着 DDM 的效率,也直接影响到仿真系统的运行效率和可扩展性。研究高效的区域匹配算法对于提高 HLA 仿真的运行性能和可扩展性等具有重要的理论和现实意义。

1 区域匹配相关概念

在 HLA 1516 接口规范中,数据分发管理以区域为基础,数据生产者(发送盟员)通过对更新区域的指定来表明所生产数据的范围,数据消费者(接收盟员)通过对订购区域的指定来表明所消费数据的范围,当且仅当公布区域与订购区域存在交叠时,发送盟员才能将数据直接发送到接收盟员。DDM 通过判断公布区域和订购区域是否交叠来决

定是否将属性更新数据发送给接收盟员,如图 1 所示,更新区域 U_1 与订购区域 S_1 相交叠, U_1 与订购区域 S_2 不交叠,因此 U_1 所属发送盟员会将更新数据发送到 S_1 所属的接收盟员,而不会向 S_2 所属的接收盟员发送。DDM 相关定义如下:

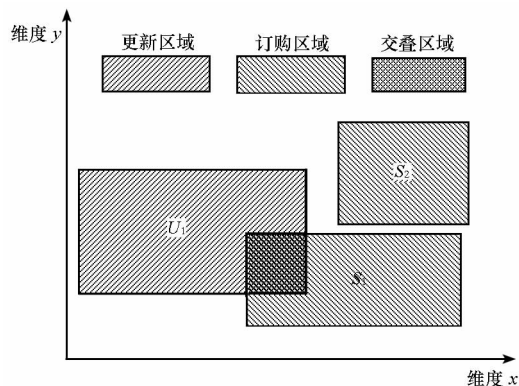


图 1 二维兴趣空间的区域交叠描述

Fig. 1 Region overlapped description in 2-dimension space

定义 1 维度(Dimension):维度是指一个命名的非负整数区间,由一个有序整数对定义,其下界为 0,上界随维度不同而不同,其值在联盟对象模型文档数据(Federation Object Model Document Data, FDD)维度表中规定。

定义 2 范围(Range):范围是一个连续整数的半开区间,是维度的子集,由一个有序整数对定义,整数对的第一个数称为范围下界,第二个数称为范围上界,上界严格大于下界,其差至少为 1。

定义 3 区域描述(Region Specification):区域描述是一个范围的集合,一个区域描述中的维度就是其所包含范围所属的维度,在一个区域描述的每个维度上最多只有一个范围。

定义 4 区域实现(Region Realization):是关联到实例属性更新、交互发送或对象类属性和交互类的订购的区域描述。

定义 5 区域(Region):区域描述及区域实现的通称。

定义 6 更新区域(Publish Region):用于表示发送交互或更新实例属性的区域实现,盟员利用它来表明发送数据的限制条件。

定义 7 订购区域(Subscribe Region):用于类属性订购或交互类订购的区域实现的集合,盟员利用它来表明接收数据的限制条件。

定义 8 范围交叠(Range Overlap):同一维度上两个范围 $U = [U_{lower}, U_{upper}]$ 、 $S = [S_{lower}, S_{upper}]$ 交叠当且仅当 $U_{lower} = S_{lower}$ 或 $(U_{lower} < S_{upper}$ 且 $S_{lower} < U_{upper})$ 。

定义 9 区域交叠(Region Overlap):两区域交

叠且仅当这两个区域的所有公共维度上的范围交叠,如果两个区域没有公共维度,则不交叠。

2 几种典型的区域匹配算法及局限性

区域匹配算法的主要任务是判断基于区域描述的个体兴趣域是否重叠,其匹配速度和精度决定着兴趣匹配的整体效率。现有的典型区域匹配算法概括起来主要有直接匹配法 (Brute-Force Approach)、网格法 (Grid-Based Approach)、混合匹配法 (Hybrid Approach)、基于分类的匹配算法 (Sort-Based Approach) 和基于移动相交的匹配法 (Intersecting-Based Approach) 等。

直接匹配法通过两个单独的数据结构分别保存更新区域和订购区域,当查询任意一个更新或订购区域的匹配情况时,该算法会依次查询当前所有的订购或更新区域的每一维上的范围,然后根据区域交叠条件进行判断。该算法实现逻辑简单,不需要存储额外的信息,且匹配精确;但该算法计算开销非常大,假设共有 N 个 d 维订购区域需要进行匹配,则所需的查询时间为 $O(d * N)^{[7]}$,因而对于大规模仿真中的大量区域匹配的需求难以适应。

基于网格的匹配算法将仿真系统中 d 维全域预先划分为规则的 d 维网格,每一个网格逻辑上都对应一个数据通道;对区域边界进行计算即可确定区域覆盖的网格,进而确定数据传输通道,一个更新区域和订购区域交叠的充要条件是它们覆盖了至少一个相同的网格^[8-10]。网格法仅需 $O(d * K)$ 的时间去计算动态 d 维区域相交 (K 是被覆盖的网格的数量),但网格法会产生虚假连接和冗余连接,需要接收盟员进行进一步的数据过滤。从整体上看,基于网格的匹配算法在匹配速度和匹配精度上存在着矛盾,整体性能受多维空间中网格大小影响较大。

针对直接匹配法和基于网格的匹配法所存在的问题,部分研究者提出了混合匹配算法^[11-12],该算法首先利用基于网格的匹配算法将所有区域映射到不同的网格上,然后利用直接匹配法对每个网格中的区域精确匹配;该算法能降低直接匹配算法的时间复杂度,并提供精确匹配结果,但与基于网格的匹配算法一样,混合匹配算法的整体性能仍然受制于网格大小。

基于分类的区域匹配算法^[13-16]的基本思想是将系统中所有区域在每个维度上进行投影排序,若两个或多个范围在投影上有重叠,则表明这些范围相交。基于分类的匹配算法的计算复杂度为 $O(d * (M + N) * \log(M + N))$,其中 M, N 分别

表示更新区域和订购区域的数目, d 为维数。基于分类的算法一次可完成对所有区域的交叠判断,虽然这种判断可通过采用位操作而提高匹配效率,但由于每次匹配都得对所有范围节点进行排序和相交检测,而实际上在每次仿真推进中,只有部分区域发生改变,因而存在大量的无用计算,导致其难以获得很高的匹配效率。

基于移动相交信息的动态区域匹配算法^[17-18]的基本思想是:针对大规模分布式仿真系统的一次仿真推进中往往只有部分区域发生改变的实际特点,利用区域范围移动前后的信息,将匹配计算限定在移动区间之内,从而有效地减少了需要进行匹配的候选范围,进而减少了匹配计算量。该算法匹配操作对 d 维空间的时间复杂度不超过 $O(d * K)$, K 表示该区域各维度上投影范围在移动过程中经过的平均区域数,因此其算法复杂度仅与变化区域所经过的区域个数相关,与系统区域总数无关,因此该算法具有较高的匹配效率。

随着多核逐渐成为 CPU 发展的主流方向,多核计算平台被越来越多地应用于大规模分布式交互仿真中,人们希望通过多核平台提供的高性能并行计算环境来提高仿真性能,增强对复杂系统仿真的支持能力。以上算法通过不同的区域匹配计算方法,虽然在一定程度上提高了 DDM 服务性能,然而对大规模仿真中数据分发管理的支持能力仍然有限,同时目前的 DDM 匹配算法都属于串行算法,将其应用于多核计算平台时,受算法本身的制约,并不能发挥多核体系结构的并行计算优势,从而很难满足基于多核平台构建大规模分布式仿真的应用需求。面向多核计算平台的高效并行区域匹配计算正成为目前大规模 HLA 仿真亟须解决的问题。

3 基于移动相交信息的并行区域匹配算法

针对目前主要区域匹配算法的局限性,本文提出了一种面向多核计算平台的并行区域匹配算法。该算法结合移动相交区域匹配思想^[17]与并行计算方法,将一次仿真推进中的多个移动区域的匹配计算任务划分到不同的 CPU 核上,通过多线程的形式实现区域匹配的并行计算,以提高区域匹配计算性能;同时在每一个计算线程内,针对每一个移动区域采用基于移动相交的区域匹配方法,有效减少冗余匹配计算,提高匹配效率。

3.1 算法思想

该算法基于如下原理:①当单个区域移动时,其他区域与该区域交叠的变化必定与该区域刚才的移

动相关,因此区域在更新时不用与所有区域进行匹配,从而无需考虑在区域移动区间之外的区域,因为与这些区域的交叠情况是不可能变化的;②当一次仿真推进中有多个区域移动时,属于同一种类型(公布区域或订购区域)的区域范围改变存在无关性,从而可以将同一类型的多个区域交叠变化交由多个计算线程并行计算其交叠信息。如图2所示。

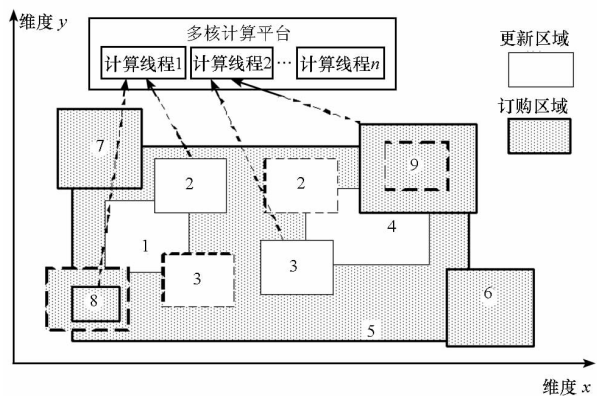


图2 区域移动相交变化示意图

Fig. 2 Changing of overlap relationship between moving regions

图2是一次仿真推进中的二维区域移动交叠变化情况,订购区域8发生移动时,途中经过更新区域1的边界(与更新区域1的交叠情况发生了变化),而没有经过其他更新区域的边界,因此订购区域8的交叠变化只可能与移动所经过的更新区域,即更新区域1有关,而与其他更新区域无关,因此只需要重新计算与更新区域1的交叠关系是否发生改变,而不必考虑其他更新区域。同时,更新区域2与更新区域3都发生移动时,各自的交叠变化情况只与本区域移动所经过的订购区域有关,相互之间的计算过程彼此独立,因此更新区域2与更新区域3的区域交叠变化计算过程可以同时交由计算线程1与计算线程2并行计算,同理订购区域8与订购区域9的区域交叠变化计算过程也可以分别交由不同计算线程并行计算,从而可以实现区域交叠过程的并行计算,有效发挥多核并行计算性能。

3.2 数据结构

当一个区域发生改变时,由该区域改变引起的匹配计算的核心是如何根据移动前后的位置信息快速查询该区域移动经过的区域。这里首先通过索引有序表来存放区域的每一维数据,即多维空间中的每一维的数据由两组索引有序表来存储,一组索引有序表用来存放所有更新区域投影在该维度上的范围界点值,另一组索引有序表用来存放所有订购区域投影在该维度上的范围界点

值;然后每一组由两个索引有序表组成,其中一个索引有序表存储范围的下界闭界点,另一个索引有序表存储范围的上界开界点。定义节点(Node)数据结构来存储范围界点(下界闭界点或上界开界点),在查询时可以直接根据下界闭界点或上界开界点移动前后的位置信息在对应的下界索引有序表或上界索引有序表中查找相关的节点即可,节点的数据结构定义如下:

```
struct Node {
    int id; //范围 id
    Struct Node * Next; //节点指针
}
```

每一个索引有序表通过一个指针数组进行索引,指针数组的大小设为维度的上界,数组元素的下标值对应界点值,即指针数组元素所指向的节点所表示的范围界点值与该数组元素的下标值相等。如一个含四个订购范围 $S_1 = [0, 1)$ 、 $S_2 = [0, 4)$ 、 $S_3 = [4, 6)$ 、 $S_4 = [5, 8)$ 的索引表表示如图3(a)所示。

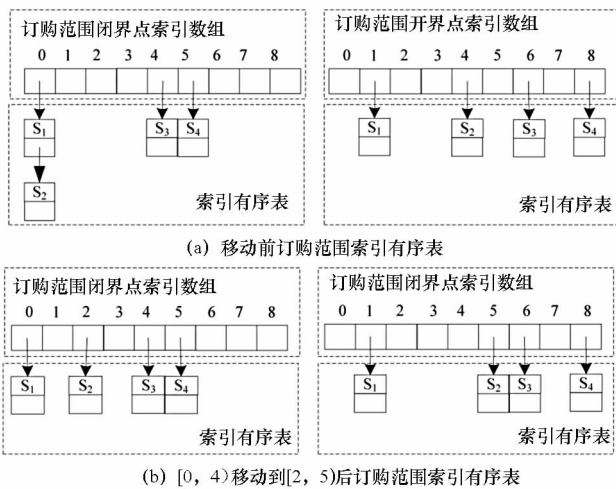


图3 基于索引有序表的范围表示

Fig. 3 Description of ranges based on index-ordered table

若范围 S_2 从 $[0, 4)$ 移动到 $[2, 5)$, 则相应的索引有序表变化如图3(b)所示。此时假设一个更新范围 U 从 $[0, 3)$ 移动到 $[4, 5)$, 则我们首先遍历订购范围闭界点索引数组找到索引值在 $[3, 5)$ 区间的订购范围,这里有一个范围 S_3 , 范围 U 与范围 S_3 由不相交变为相交;再遍历订购范围开界点索引数组找到索引值在 $(0, 4]$ 区间的订购范围,这里有一个范围 S_1 , 范围 U 与范围 S_1 由相交变为不相交。由此可以看出,对于不在移动范围之内的节点是不需要对它进行匹配的,从而避免了与订购范围 S_2 及订购范围 S_4 的冗余匹配操作,有效节省了计算资源,同时实现了范围之间的精确匹配,不

存在虚假连接问题。

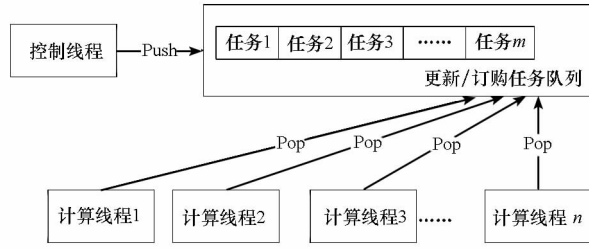


图 4 并行区域匹配方法框架

Fig. 4 Infrastructure of parallel region matching

3.3 算法描述

如图 4 所示,本文基于多线程实现区域的并行交叠计算,共包括一个控制线程与多个计算线程,同时通过任务队列描述相关的计算任务,控制线程在区域发生移动后负责生成任务节点并将其放入任务队列,然后控制计算线程完成任务计算。任务队列由更新区域任务队列与订购区域任务队列组成,分别存储由更新区域移动引起的区域匹配计算任务以及由订购区域引起的区域匹配计算任务,在一次仿真推进中,各计算线程首先处理更新区域任务队列,在处理完毕后,再处理订购区域任务队列,直到所有任务处理完毕为止,线程间采用信号量机制进行同步。控制线程采用的算法如表 1 所示。

表 1 控制线程算法描述

Tab. 1 Control thread algorithm

控制线程算法描述
for (每一次仿真推进过程)
1. 对于每一个公布范围 a_m , 如果 a_m 发生移动, 则将 a_m 插入公布任务队列 pq ;
2. 对于每一个订购范围 b_n , 如果 b_n 发生移动, 则将 b_n 插入订购任务队列 sq ;
3. 如果 pq 不为空, 对于每一个计算线程 i , 释放信号量 $m_publish_i$, 用于通知计算线程开始处理更新任务队列;
4. 等待每一个计算线程 i 的完成信号量 $m_publish_finish_i$, 直到所有更新任务处理完成;
5. 如果 sq 不为空, 对于每一个计算线程 i , 释放信号量 $m_subscribe_i$, 用于通知计算线程开始处理订购任务队列;
6. 等待每一个计算线程 i 的完成信号量 $m_subscribe_finish_i$, 直到所有订购交叠计算完成。
end for

每一个计算线程基于以上算法思想,对每一个移动范围进行相关匹配计算,相关算法描述如表 2 所示。

表 2 计算线程算法描述

Tab. 2 Calculate thread algorithm

计算线程算法描述
for (每个计算线程 i)
1. for (更新任务队列 pq_i , 中每个发生移动的范围 R)
2. 设范围 R 修改之前的区间为 $[a, c)$ (对于新创建范围, $a=c=0$), 修改之后的区间(新范围)为 $[b, d)$, 该范围所属维上的公布范围下界闭界点索引表为 P_L , 公布范围上界开界点索引表为 P_U , 订购范围下界闭界点索引表为 S_L , 订购范围上界开界点索引表为 S_U , 假设与 R 交叠的范围集合为 $R_{overlap}$;
3. 定义索引节点指针 Mod_L 与 Mod_U 分别指向移动范围所在的下界闭界点索引表与上界开界点索引表, 同时定义 Mat_L 与 Mat_U 分别指向需要与 R 进行匹配计算的集合的下界闭界点索引表与上界开界点索引表。若范围 R 为公布范围, 则 $Mod_L = P_L, Mod_U = P_U, Mat_L = S_L, Mat_U = S_U$; 若 R 为订购范围, 则 $Mod_L = S_L, Mod_U = S_U, Mat_L = P_L, Mat_U = P_U$ 。
4. 若 R 为新创建的范围, 则为 R 分别创建个闭节点与开节点, 将闭节点链入 Mod_L 所指链表中, 将开节点链入 Mod_U 所指链表中; 否则, 将 $Mod_L[a]$ 所指链表中 R 闭节点移入 $Mod_L[b]$ 所指链表中, 将 $Mod_U[c]$ 所指链表中 R 开节点移入 $Mod_U[d]$ 所指链表中。
5. if($c < d$) then 将 $Mat_U[c] \sim Mat_U[d-1]$ 所指链表中的所有节点对应的范围加入 $R_{overlap}$ 中;
6. if($a < b$) then 将 $Mat_L[a+1] \sim Mat_L[b]$ 所指链表中的所有节点对应的范围从 $R_{overlap}$ 中删除;
7. if($a > b$) then 将 $Mat_L[b+1] \sim Mat_L[a]$ 所指链表中的所有节点对应的范围加入 $R_{overlap}$ 中;
8. if($c > d$) then 将 $Mat_U[d] \sim Mat_U[c-1]$ 所指链表中的所有闭节点对应的范围从 $R_{overlap}$ 中删除;
9. $R_{overlap}$ 中即为与 R 相交叠的所有范围。
end for
10. 释放信号量 $m_publish_finish_i$;
11. for(订购任务队列 sq_i 中每个发生移动的范围 R)
12. 执行 2 ~ 9 的匹配计算过程;
end for
13. 释放信号量 $m_subscribe_finish_i$;
end for

设全域多维空间中包含 d 个维度,对于 m 个变化区域,设 K 表示该区域各维度上投影范围在变化过程中经过的平均界点数,假设在 x 核计算平台上有 $n(n \leq x)$ 个计算线程,则理想情况下上述算法

匹配操作的时间复杂度为 $O(m * d * K/n)$, 由此可以看出,该算法复杂度与变化区域所经过的界点个数相关,而与系统中的总区域个数无关;同时,通过多核并行计算进一步降低了算法复杂度,提高了计算效率。

4 实验结果

为了比较该并行算法与其他匹配算法的性能,本文设计了如下测试程序:首先根据用户输入的维度上限 m 生成一个二维空间,并在该二维空间上随机生成 n 个公布区域和 n 个 ($n > 500$) 订购区域,为模拟实际复杂系统应用中个体的多样性,区域的大小均随机生成;然后随机选取 0 至 $n - 1$ 中的 500 个区域,将选择区域的每一个范围随机移动 x 距离,分别采用直接匹配、分类匹配、网格匹配(网格数为 100)、网格匹配(网格数为 500)、并行匹配等方法进行区域的交叠计算,并记录匹配完成时间 t ,重复该过程 10 次,计算平均时间 t_{avg} ,再计算每一个区域移动的平均完成时间 $t'(t' = t_{avg}/500)$,在不同的维度上限与区域个数情况下,通过不同算法下的 t' 大小来分析相关算法性能。同时为了分析并行算法的加速性能,针对不同的移动步长及计算线程,设计了加速比测试对并行算法进行了充分测试。实验环境为 Intel 四核 Q8400 处理器,2.66GHz,3.5GB RAM,操作系统为 Windows XP SP3。

4.1 不同区域个数时的算法性能测试

图 5 显示了维度上限分别为 2000、4000、8000 的情况下,各匹配算法匹配耗时随区域数变化的情况。从测试结果可以看出,直接匹配算法性能随着区域数的增加呈指数级增长,在维度上限为 4000 和 8000 时,由于其耗时太长,在对比时未进行考虑;其他几个算法耗时都随着区域数的增加而呈现线性增长的趋势,但并行算法的耗时增长最慢。在维度上限为 2000 和 4000 时,并行匹配算法性能最好,维度上限为 8000 时,网格数为 100 的网格匹配算法性能略优于并行匹配,但由于其存在大量虚假连接,因而数据过滤效果很差。

4.2 不同维度上限时的算法性能测试

图 6 显示了区域个数分别为 2000、4000、8000 的情况下,各匹配算法匹配耗时随维度上限变化的情况。从测试结果可以看出,网格匹配法性能波动很大,这主要是因为当网格数确定时,随着维度上限的增大,网格粒度增大,区域在网格内移动的概率增加,由于网格内移动不会改变区域的交叠

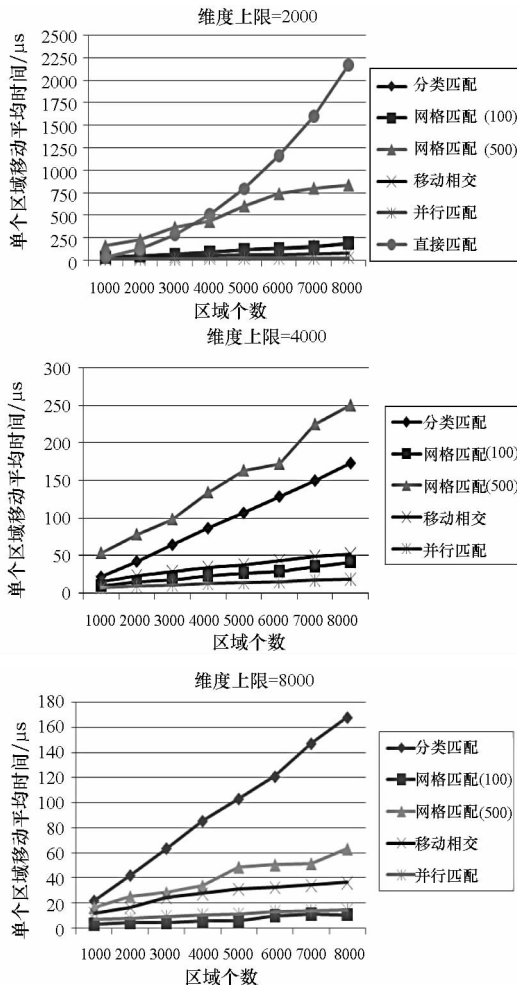


图 5 不同区域个数的算法性能测试结果

Fig. 5 Performance test with different number of region

关系,因此其算法耗时随着维度上限的增加而减小,但过滤效果变差。而分类匹配、移动相交以及并行匹配算法性能与维度上限改变无关,因而更适合于大规模仿真。同时,并行匹配算法性能明显优于移动相交以及分类匹配算法,在维度上限大于 5000 时,算法性能与网格数为 100 的网格匹配性能相当,并且不存在网格匹配中的虚假连接现象。

4.3 算法加速比测试

并行匹配算法性能与计算线程数存在很大相关性,多个计算线程更有利于充分发挥多核计算性能;同时并行匹配算法与移动经过的范围数密切相关。图 7 显示了在不同维度上限与区域个数的情况下,针对不同的移动距离进行的加速比测试。由图中可以看出,对于测试采用的四核计算平台来说,计算线程数为 4 时,算法性能最好,线程数大于 4 时,由于线程间的同步及切换开销增加,加速性能将随着线程数的增加而逐渐降低。同时,由图中可以看出,在移动距离较大时,算法加速性能明显优于移动距离小的情况,当移动距离为

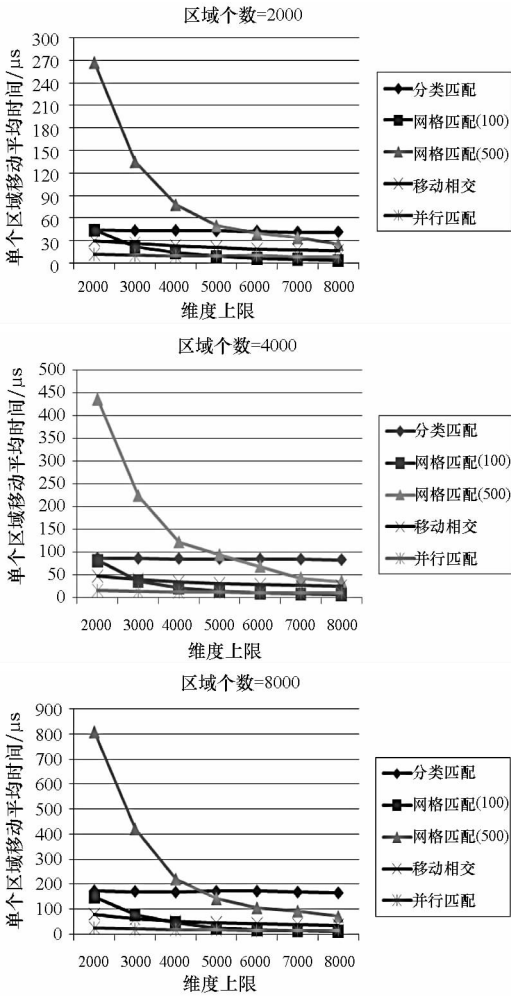


图 6 不同维度上限时的算法性能测试结果

Fig.6 Performance test with different number of region

1200 时,四线程加速比约为 3.2,移动距离为 200 时,四线程加速比约为 2,这主要是因为移动距离大时经过的区域较多,区域匹配的计算开销占总开销的比例增加。同时也表明,该并行区域匹配算法虽然会导致额外的同步开销,但与串行算法相比,仍能获得较好的加速性能。

由以上测试可以看出,本文提出的并行区域匹配算法性能优于目前主要的匹配算法性能,与文献[17]中移动相交区域匹配算法相比,在上述测试环境中,单次移动平均时间仅为后者的 1/3 ~ 1/2;该算法不仅不会随区域数的增加而出现明显的性能下降,而且与维度上限的改变无关,具有高效、无虚假连接等优点;同时通过多核并行计算获得了较好的加速比,尤其是对于大范围的区域改变加速性能更好。然而,由实验结果也可以看出,由于线程间需要一定的同步开销,目前的算法加速比并不能达到线性加速比,因此如何减少同步开销将是下一步的研究方向。总体来说,该算法非常适合于基于多核计算平台构建的大规模复杂

系统仿真。

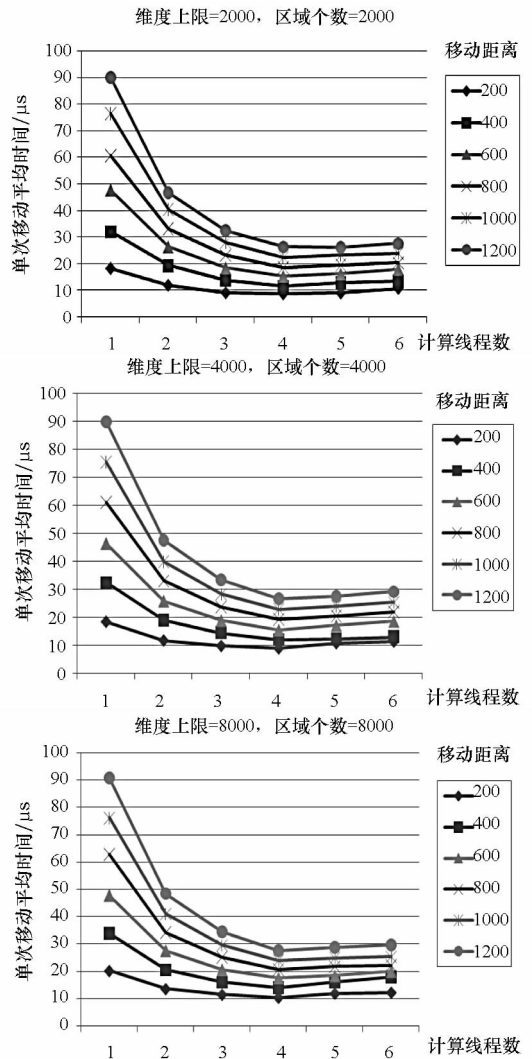


图 7 算法加速性能测试结果

Fig.7 Result of acceleration performance test

5 结 论

论文针对基于多核平台构建大规模 HLA 仿真的应用需求,分析了目前几种主要的区域匹配算法的优缺点,提出了一种面向多核计算平台的并行快速区域匹配算法。该算法的创新在于基于多核平台通过多线程实现了区域匹配的并行计算,同时在区域匹配计算中,基于移动相交的基本原理,利用区域范围更新前后的历史信息,将匹配限定在移动区间之内,因而可大大减少冗余的匹配计算。算法性能测试表明,该算法具有很高的匹配效率,不会引起虚假连接,同时具有较好的加速性能,可以充分发挥多核计算平台的计算性能,满足大规模分布式仿真数据分发管理需求。本文的下一步工作是如何减少线程间的同步开销,以进一步提高加速性能。

参考文献 (References)

- [1] IEEE Standard for Modeling and Simulation. High Level Architecture [HLA]-Federate Interface Specification [S]. IEEE Std 1516.1-2000, 2001.
- [2] 姚益平. 高性能分布式交互仿真运行支撑平台关键技术研究[D]. 长沙:国防科技大学, 2003.
YAO Yiping. Research on technologies of high performance distributed simulation runtime infrastructure [D]. Changsha: Nation University of Defense Technology, 2003. (in Chinese)
- [3] IEEE Standard for Modeling and Simulation. High Level Architecture [HLA]-Federate Interface Specification [S]. IEEE Std 1516-2000, 2001.
- [4] IEEE Standard for Modeling and Simulation. High Level Architecture [HLA]-Federate Interface Specification [S]. IEEE Std 1516.2-2000, 2001.
- [5] Fujimoto. Parallel and distributed simulation systems [M]. Wiley Interscience, 2000.
- [6] 周彦,戴剑伟, HLA 仿真程序设计[M]. 北京:电子工业出版社, 2002.
ZHOU Yan, DAI Jianwei. The design of HLA simulation program [M]. Beijing: Publishing house of electronics industry. (in Chinese)
- [7] Petty M D, Morse K L. Computational complexity of HLA data distribution management [C]//Proceedings of 2000 Fall Simulation Interoperability Workshop. Orlando, FL, 2000.
- [8] Tan G, Ayani R, Zhang Y S, et al. Grid-based data management in distributed simulation [C]//Proceedings of Simulation Symposium, 33rd annual, Washington, DC, 2000: 7-13.
- [9] Ayani R, Moradi F, Tan G. Optimizing cell-size in grid-based DDM [C]//Proceedings of Parallel and Distributed Simulation, Bologne, 2000: 93-100.
- [10] 任文, 傅忠谦. 一种改进的基于网格的数据过滤方法 [J]. 中国科学技术大学学报, 2010, 40(11).
REN Wen, FU Zhongqian. An improved data filtering algorithm based on grid [J]. University of Science and Technology of China, 2010, 40(11). (in Chinese)
- [11] Prabhakar S. Query indexing and velocity constrained indexing: scalable techniques for continuous queries on moving objects [J]. Computers, IEEE Transactions on, 2002, 51(10): 1124-1140.
- [12] Boukerche A, McGraw N J, Araujo R B. A grid-filtered region-based approach to support synchronization in large-scale distributed interactive virtual environments [C]//Parallel Processing, International Conference Workshops, 2005, 525-530.
- [13] Pan K, Turner S J, Cai W T, et al. An efficient sort-based DDM matching algorithm for HLA applications with a large spatial environment [C]//Principles of Advanced and Distributed Simulation, San Diego, 2007: 70-82.
- [14] Pan K, Turner S J, Cai W T, et al. A dynamic sort-based DDM matching algorithm for HLA applications [J]. ACM Transactions on Modeling and Computer Simulation (TOMACS), 2011, 21(3): 17.
- [15] Raczky C, Tan G, Yu J. A sort-based DDM matching algorithm for HLA [J]. ACM Transactions on Modeling and Computer Simulation (TOMACS), 2005, 15(1): 14-38.
- [16] 王磊. 一种改进排序匹配算法在 DDM 中的应用与实现 [J]. 计算机工程与应用, 2007, 43(33): 4.
WANG Lei. Research and implementation of improved sort-based matching algorithm in data distribution management [J]. Computer Engineering and Applications, 2007, 43(33): 4. (in Chinese)
- [17] 姚益平. 基于索引排序的快速动态区域匹配算法 [J]. 计算机学报, 2009, 32(7): 7.
YAO Yiping. Dynamic region matching algorithm based on index-order [J]. Chinese Journal of Computers, 2009, 32(7): 7. (in Chinese).
- [18] 张颖星. 面向复杂系统应用的并行离散事件仿真性能优化技术研究 [D]. 长沙:国防科技大学, 2011.
ZHANG Yingxing. Research on performance optimization of parallel discrete event simulation for complex system application [D]. Changsha: Nation University of Defense technology, 2011. (in Chinese)