

文章编号: 1001- 2486(2008) 02- 0118- 05

并行离散事件仿真中的 DDM 机制实现*

张耀程, 李 革, 黄柯棣

(国防科技大学 机电工程与自动化学院, 湖南 长沙 410073)

摘要: 在并行离散事件仿真系统中实现仿真模型间的互操作机制需要引入类似 HLA 中定义的声明管理(DM)和数据分发管理(DDM)服务。我们开发的 KD-PADSE 高性能并行离散事件仿真环境采用谓词断言和对象代理方法在面向对象的并行离散事件仿真系统中实现了基于平行结构和事件调度机制的声明管理(DM)和数据分发管理(DDM)服务。在以千兆以太网互联的 10 节点 Linux 集群计算机上,采用 DDM 机制后仿真模型每秒(物理时间)可以进行 1 万次属性更新,是没有采用 DDM 机制时的 20 倍,在功能性和性能上验证了该 DDM 机制在并行离散事件仿真系统中的适用性。

关键词: 兴趣管理; 声明管理; 数据分发管理; 并行离散事件仿真

中图分类号: TP319.9 文献标识码: A

DDM Mechanism Implementation in Parallel Discrete Event Simulation

ZHANG Yao-cheng, LI Ge, HUANG Ke-di

(College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Parallel Discrete Event Simulation (PDES) Systems demand a Declaration Management (DM) service and Data Distribution Management (DDM) service similar to those defined in HLA to realize model interoperability. In KD-PADSE, a high-performance PDES environment was implemented, and those services in a parallel structure based on event-scheduling using predication and object proxy techniques were realized. In an experiment carried out on a 10-node Linux cluster machine connected via 1Gbps switch Ethernet, 10 thousand attribute updates per second (wall clock time) was achieved in a synthetic benchmark test, which is 20 times larger than that of the same experiment without DDM services. The experiment proves the functionality and high performance of the implemented DDM service in a PDES environment.

Key words: interest management; declaration management; data distribution management; parallel discrete event simulation

以 HLA 标准为代表的分布式仿真系统所提供的声明管理(DM)和数据分发管理(DDM)服务能够减小仿真模型之间的无关通讯流量,极大提高系统执行效率和可扩展性。并行离散事件仿真技术(PDES, Parallel Discrete Event Simulation)将离散事件仿真技术和并行计算技术相结合,通过将离散事件仿真模型的计算分解到多个计算节点上并行执行来提高计算能力。PDES 技术的发展已有近 30 年的历史,已开发出多种建模和计算环境^[1],但是这些成果普遍在模型的互操作性以及 DDM 方面考虑甚少,因为 PDES 系统本质上是一种 Single-Program-Multi-Data 结构,考虑的是将一个单一的仿真模型分解到多个节点上进行并行解算,以提高计算性能,仿真模型中各对象之间的通讯拓扑关系是事先已知的,与分布式仿真中的将多个独立的联邦成员集成到一个一致的虚拟时空中进行交互仿真的思想不同。

随着软件模型复用思想的普及, PDES 领域也在探索如何将各种已开发的组件模型快速组织成新的并行仿真系统模型。PDES 中没有邦员的概念,其模型的粒度更细,一般以实体(以对象为表现形式的仿真模型)作为模型的基本单位。为了实现实体模型重用和快速组合的目标,面向对象的 PDES 系统也需要实现类似于 HLA 中的 DM 和 DDM 服务,以提供标准的模型互操作能力。

* 收稿日期: 2007- 05- 30

基金项目: 国家自然科学基金资助项目(60574057)

作者简介: 张耀程(1977-),男,博士生。

本文在比较了分布式仿真与并行仿真中 DDM 技术的异同的基础上, 结合我室开发的高性能并行离散事件仿真软件 KD-PADSE^[10-13] 中的实现方法, 介绍了适用于并行离散事件仿真系统的、支持实体级模型重用技术的 DDM 技术, 最后通过负载实验验证了其有效性和优越性。

1 分布与并行仿真中的 DDM 机制比较

DDM 技术最初源自分布式仿真应用的实际需要。在早期的分布式仿真技术(DIS)中, 网络上的各仿真程序都必须接收所有的交互信息(如仿真实体的属性更新和交互操作), 大量的无关数据挤占了有限的网络带宽, 消耗了 CPU 资源。为此, 在第二代分布式仿真技术标准 HLA 中, 详细定义了数据分发管理服务(DDM)和声明管理服务(DM), 这两种服务提供了仿真中基于类属性和实例属性的双重过滤机制, 其实现方法在国内外文献中已有大量研究^[3-6], 主要的匹配算法包括区域法和网格法, 前者计算量较大, 可扩展性差, 而后的数据过滤精度与网格大小有直接的关系, 同时还易引起重复匹配及冗余数据问题。从软件系统结构角度考虑, 无论何种 DDM 算法, 都必须将公布信息与订购信息进行匹配, 这种匹配可以在数据发送端进行(最大限度杜绝无效信息发送的方法), 或在数据接收端, 或中介端(如 RTI 服务器)进行。

分布式仿真系统的组成结构和运行方式与 PDES 系统有显著区别, 所以 HLA 标准中的 DDM 技术并不能直接适用于 PDES 系统中。首先, 基于 HLA 技术的分布式系统是一种集中分布相结合的结构, 联邦运行时必须有 RTI 服务器进行运行管理和兴趣匹配计算; 而 PDES 中没有这种集中控制的存在, 各计算节点上运行的是同一个程序的不同实例, 各程序之间是平行关系。其次, PDES 系统是由事件驱动执行的, 不仅仿真模型的状态修改、公布订购及交互发送都以带有时间戳的离散事件表示和处理, 在系统层内部对模型属性的公布、传递、接收、反射, 以及属性过滤器的创建、传递、删除等系统行为也以普通事件的方式在事件处理系统中进行传递和处理, 与仿真模型平等地竞争 CPU 周期等计算资源。

因此, 要在面向对象的 PDES 系统中实现 DDM 机制, 要解决如下三个问题:

(1) 基于实体和事件的设计——在 PDES 中, 仿真引擎以驱动事件执行的方式运行, 实体与事件是系统的基本组成单位。控制兴趣管理和数据分发服务的系统部件也应以对象的形式存在, 以调度事件的方式运行;

(2) 在平行结构中实现公布订购管理——这就要求在没有中央控制的情况下由各节点共同协调进行兴趣管理;

(3) 基于实体类型和属性的兴趣管理——HLA 中的 DDM 服务建立在区域(Region)、限域(Extent)和维(Dimension)这些由联邦或联邦成员定义的概念上^[9]; PDES 中没有联邦成员的概念, 每个实体都可以独立表达对其他类型实体的订购兴趣, 也可以进一步限定实体中某属性的取值范围。系统应能根据实体类型和实体属性值的限定范围进行数据分发。

2 面向对象 PDES 系统中的 DDM 机制实现

我们采用实体级建模(Entity Level Modeling)和对象代理(Object Proxy)技术, 在面向对象的 PDES 环境 KD-PADSE 中实现了基于实体模型和事件驱动的 DDM 服务。本节将介绍 KD-PADSE 的系统模型基础, 及其 DDM 机制的设计实现。

2.1 KD-PADSE 系统模型

KD-PADSE 仿真系统包括用户模型(模型层)及为在并行机上进行模型层计算所需的软件支撑环境(系统层)两部分。软件支撑环境由提供 C++ 接口的库实现, 为模型层的计算和通讯提供仿真时间管理、事件驱动机制、回退计算、网络通信服务、持续性管理、运行控制、统计跟踪、及公共算法库等服务, 同时为用户建模提供了基本的建模模块, 包括实体、属性、事件、过程、消息、仿真时间表示等。仿真开发人员利用这些建模模块, 使用面向对象的方法描述和建立用户模型, 并由 KD-PADSE 驱动模型的运行。用户模型由独立的实体构成, 实体间通过调度事件进行交互, 实体也是公布订购的基本单位, 各实体通过表达公布和订购兴趣来互相感知对方的状态。KD-PADSE 仿真环境同时支持共享内存通信、网络通信、

及两者混合的通信环境,可以运行在单机、SMP、SMP 集群、局域网工作站等典型串行或并行计算环境中。

在KD-PARSE中,每个计算节点上驻留一个KD-PADSE程序实例,各个程序实例之间是平行的关系,没有中央控制结构,每个实例都包括用户模型的一部分和系统层支持环境,把它们连接起来的是底层的共享内存或TCP/IP网络。针对仿真模型中的每种实体,KD-PARSE提供两种实体集分解策略供用户选择,分别为轮序法(round robin),即将实体轮流分配到每个处理器上;或块分解法(block),即将实体连续地分配到每个处理器上。

2.2 实体描述及实体代理机制

KD-PADSE系统中的实体模型以“Entity”表示,而每个Entity的状态由多个属性(PO_Attribute)来描述。PO_Attribute是多种类型类型的基类抽象,包括PO_int,PO_double,PO_string,PO_position等基本数据类型和复杂数据类型。它们由系统提供,供用户用来描述实体状态。用户使用各种PO_Attribute类来定义实体的可公布属性变量,当模型修改这些属性值时,系统就可以自动探测到该变化并将其属性值分发到所有感兴趣的订购实体上。实现这种自动数据分发机制的方法是使用一种称为“代理对象”的技术。称为PO(Proxy Object)的代理对象是一个实体的属性全集中可公布属性子集在远程节点上的映像。每种PO代表一种实体约定的可公布属性集,它包含若干PO_Attribute类的成员,对应其代理的实体类中的各PO_Attribute,一个PO对象代表一个公布了自身属性的实体,它们具有一致的类型。一个实体可以利用系统服务生成与自己类型对应的PO代理,并将自己的可公布属性与该PO中的对应变量相关联,再通过系统服务公布(publish)该PO对象。如果某个节点上的实体根据其表达的订购兴趣能够发现该对象,系统就会将该PO对象信息打包发送到该订购节点上,并在订购节点上重构该PO对象的一个副本,订购节点上的订购实体就可以通过该PO副本读取公布实体的可公布属性值了,如图1所示。在图1中,节点1上的公布实体A生成一个与自己类型对应的PO代理,并把描述其可公布属性的各PO_Attribute属性映射到该PO代理中,然后公布(Publish)该PO代理。如果节点2上有至少一个订购实体(Subscribing Entity)满足订购条件,系统将在节点2生成该PO的一个副本,并通知节点2上所有满足订购条件的订购实体。订购实体在接到通知后,可以从该PO副本中读取实体A的状态信息,或执行其他相应的逻辑。当实体A的可公布属性发生变化时,这种变化会在同时反应到其所有PO代理副本上,所有的订购实体也会得到通知。每个节点只会重构一个PO副本,该节点上的所有订购实体都共享这一个副本,以提高数据分发效率。

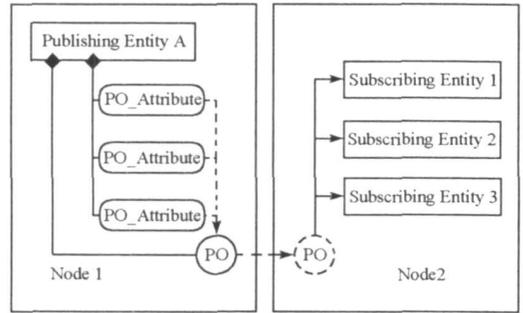


图1 代理对象机制示意图

Fig. 1 Proxy object mechanism

2.3 兴趣管理对象(Interest Manager)和本地PO管理对象(Local PO Manager)

为了实现上面的机制,还需要两个系统部件。一个进行订购兴趣管理和匹配计算,为每个公布实体找到其匹配的订购实体集;另一个负责每个节点上PO副本的管理和维护。KD-PADSE系统中提供两种系统对象——Interest Manager和Local PO Manager——来完成上述功能。

在组建仿真模型时,仿真开发人员必须将所有可能的PO类型和每个PO的详细结构记录在一个声明文件上(类似于HLA中的FOM表)。仿真运行初始化阶段,系统读取该声明文件并针对每种PO类型生成一个Interest Manager对象,该对象由系统定义,与用户模型中的实体对象一样是一个自治对象,通过处理事件和调度新事件来运行,不过它的任务是在整个系统内负责其对应PO类对象的订购兴趣管理和数据匹配计算。它一方面记录整个系统内所有订购实体声明的订购条件,另一方面维护所有公布的PO信息,并在公布实体进行公布、公布撤销、属性更新等操作前根据全局的订购兴趣为公布实体生成新的订购实体列表并返回给公布实体。公布实体按照新的订购列表将PO副本发送到每个远程订购

节点上。如果一个仿真模型中共定义了 N 种 PO 类, 则系统会生成 N 个 Interest Manager 对象, 每个对应一种 PO 类, 这些 Interest Manager 对象被按照轮序方法分配到各个处理节点上并负责协调整个系统中其对应类型的 PO 对象的公布和分发。通过将各 Interest Manger 分散到各个处理节点上的方法可以获得一定的并行处理能力。

Local PO Manage 对象用来实现 PO 副本的重构和管理。该对象负责接收所有到达本节点的 PO 副本并维护每个 PO 副本在本地的定购实体列表。一个公布实体在公布其 PO 对象时经过与其对应的 Interest Manager 对象协商后得到该 PO 对象的定购实体列表, 然后将所有定购实体按照节点分组, 为每个目的节点上的 Local PO Manager 调度一个事件并把该 PO 对象打包发送到该节点, 并把该 PO 副本在该节点上的定购实体列表也传送到该 Local PO Manager。Local PO Manager 收到消息后生成该 PO 的副本, 并将其与本地定购实体列表相关联, 然后逐一通知这些定购实体新 PO 对象的到来。这样所有的定购实体都可以感知到一个新 PO 对象的存在, 并可以获取其属性值。

当公布实体更新属性时, 系统将捕获 PO _Attribute 的变化, 产生相应事件通知每个节点上的 Local PO Manager, 并把变化的属性值集打包传递给它们, 再由 Local PO Manager 通知每个定购实体(都以调度事件的形式进行)。这样, 一个属性更新事件只需在两个节点间传送一次, 而不用分别传送给每个远程定购实体, 大大节约了开销。当实体定购某个 PO 类时, 它将首先通知该 PO 类对应的 Interest Manager, 然后 Interest Manager 对象将新的定购实体告知公布实体, 公布实体可以将该定购实体添加到自己的定购实体列表中, 然后通知该定购实体所在的节点上的 Local PO Manager 加入新的定购实体。图 2 中以公布实体飞机和定购实体雷达为例说明了各对象之间的拓扑关系。

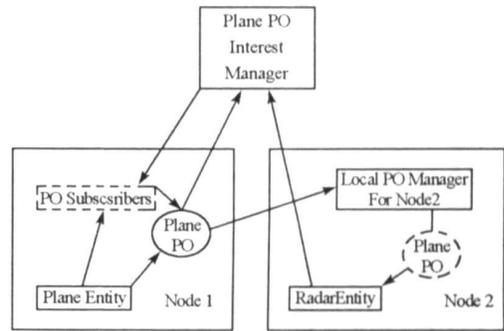


图 2 Entity, Interest Manager 与 Local PO Distributor 关系图
Fig. 2 Relationship between entity, interest manager and local PO distributor

2.4 兴趣表达式及 PO 过滤器

在 DDM 技术中通常有两种策略来进行数据过滤: 基于实体属性和谓词的方法(如 Joint Precision Strike Demonstration 系统^[7]), 或基于区域和网格的方法(如 ModSAF^[8]及多数 HLA 系统)。根据 KD-PARSE 面向对象的建模特点, KD-PADSE 中采用基于谓词的方法来进行数据过滤, 以实体每个属性的具体值来进行兴趣匹配计算, 能够达到完全匹配, 不会出现重复匹配或数据冗余问题。

KD-PADSE 采用兴趣表达式和 PO 过滤器来进行精确的数据过滤。兴趣表达式是对某个 PO 属性的一个谓词断言, 并提供方法来判断该属性值是否满足谓词断言, 定购实体可以通过声明各种兴趣表达式来对公布实体的各 PO _Attribute 进行进一步的限定。

PO 过滤器是对某个 PO 对象多个谓词断言的集合表达。例如, 一个 C4I 指控中心实体对作战平台实体的定购兴趣的限定可包括: 部队标志属性为红方 (PO _String ForceFlag= "Red"), 健康属性为生存 (PO _Logical Alive= 1), 范围属性为 100km 内 ($PO_double|ForcePosition-MyPosition| < 100,000$) 等。定购实体针对每个判断条件定义一个兴趣表达式, 同时生成一个 PO 过滤器来存储所有这些兴趣表达式。定购实体将 PO 过滤器发送到作战平台类的 Interest Manager 对象上, 该 Interest Manager 对象将每个过滤器及生成该过滤器的定购实体标识作为一个完整的定购信息单位进行保存, 作为为每个公布实体确定定购实体范围的依据。并在每次有作战平台类 PO 公布/取消公布或作战平台实体修改其 PO 对象的属性时将该 PO 对象与各个过滤器对比来确定新的定购实体列表。在这里, 生成针对某类 PO 对象的过滤器并发送到其对应 Interest Manager 对象上的行为等价于声明了基于实体类型的定购兴趣(DM), 而 PO 过滤器中的各个兴趣表达式则进一步根据实体属性值限定了定购范围(DDM)。定购实体也可以随时修改自己的 PO 过滤器, 这些修改会立即反映到 Interest Manager 对象中。

Interest Manager 对象将定购实体声明的 PO 过滤器及公布实体公布的 PO 对象实例进行兴趣匹配,

可以为每个公布实体确定有感兴趣的定购实体集。公布实体与定购实体所在节点的 Local PO Manager 对象建立数据传输路径(通过共享内存或 TCP/IP 网络)来进行数据的选择性传输。

3 功能与性能验证实验

设计了一个负载实验来验证上述 DDM 机制的功能性及计算性能。实验硬件环境为 20 节点 Linux 高性能计算集群,通过千兆以太网互联,每个计算节点拥有双 Itanium2 CPU (1.3GHz),通过共享内存通信。取其中的 10 个节点参与计算,共 20 个 CPU。在本实验中,生成 1000 个 PublishingEntity 对象和 20 个 SubscribingEntity 对象,都按照轮序方式平均分配到 20 个 CPU 上。

仿真模型利用 DDM 服务在运行时进行逻辑分组,将每 50 个公布实体与 1 个定购实体结成一组并建立数据传递路径,共划分为 20 个逻辑通信组。

为模拟最坏情况,每组的公布实体与定购实体都在不同的节点上,数据传递必须跨节点。每个 PublishingEntity 将一个 PO_int 类的属性 Attribute 从 1 更新到 1000,每次增加 1。SubscribingEntity 必须能够反射自己所在通讯组内的公布实体的这些更新情况。实验共进行 6 次,其中三次是公布实体分别以 1、5、10Hz 的频率进行属性更新,并利用 DDM 机制进行数据过滤;另外三次为对照实验,分别以同样的频率更新属性,但是不使用 DDM 服务,每个 SubscribingEntity 都会接收所有 1000 个 PublishingEntity 的属性更新。我们记录仿真计算的 STAR (Simulation Time Advance Rate) 平均值作为性能评估依据。该值是每个 GVT 周期中仿真时间与墙上时间的推进速率比值,它反映了仿真时间相对于物理时间的推进速度。例如,STAR 平均值为 12 说明观察者的时间每前进 1s,模型的仿真时间可前进 12s。因为 KD-PADSE 采用 TCP 协议进行跨节点消息传递,所以每个属性更新消息都可以保证到达,因此应以仿真计算的 STAR 值作为 DDM 机制性能的主要评价指标。实验结果如图 3 所示。

由实验结果可以看出,因为该仿真模型所进行的唯一操作就是更新自身属性(极低粒度事件)及发送、接收该更新信息,因此该实验中的计算开销主要为 DDM 计算的开销。从实验结果可以看出,增加 DDM 服务后模型计算速度增加约 20 倍,符合模型分组情况,从而验证了其在功能上可以满足数据过滤、降低无关开销的需求。

参考文献:

- [1] Fujimoto R M. Parallel and Distributed Simulation Systems [M]. New York: John Wiley & Sons, Inc., 2000.
- [2] Steinman J S. The Distributed Simulation Management Services Layer in SPEEDES [P]. 04S- SIW- 098.
- [3] 张亚崇,孙国基,严海蓉,等. 分布式交互仿真中一种新的数据分发管理算法的研究[J]. 系统仿真学报,2005,17(1): 91- 99.
- [4] 曲庆军. 高层体系结构(HLA)中兴趣管理的研究和实现[D]. 长沙:国防科技大学,2003.
- [5] 曲庆军,杨建池,黄柯棣. 关于 DDM1.516 与 DDM1.3 的研究[J]. 计算机应用,2002,22(6):35- 37.
- [6] Tacic I. Efficient Synchronized Data Distribution Management in Distributed Simulations[D]. Georgia Institute of Technology, Doctoral Thesis, Feb. 2005.
- [7] Powell E T, Mellon L, Watson J F, et al. Joint Precision Strike Demonstration(JPSD) Simulation Architecture[P]. 14th Workshop on Standards for the Interoperability of Distributed Simulations. Orlando, Florida, 1996: 807- 810.
- [8] Russo K L, Shuette L C, Smith J E, et al. Effectiveness of Various New Bandwidth Reduction Techniques in ModSAT[C]//Proceedings of the 13th Workshop on Standards for the Interoperability of Distributed Simulations, 1995: 587- 591.
- [9] High Level Architecture Interface Specification[S]. IEEE1516, 2000.
- [10] 张耀程,乔海泉,李革,等. 并行离散事件仿真中的回退和持续机制研究[J]. 系统仿真学报,2007,19(1): 67- 71.
- [11] 张耀程,李革,黄柯棣. 并行离散事件仿真软件 KID-PARSE 中的事件对象及事件队列建模技术[J]. 系统仿真学报,2007,19(9): 1949- 1953.
- [12] Zhang Y C, Li G, Huang K D. Optimizing Model Interoperability in Parallel Discrete Event Simulation for Cluster Environment[C]//Proceedings of Summer Computer Simulation Conference (SCSC'07), San Diego, California, USA, July 15- 18, 2007.
- [13] 乔海泉. 并行仿真引擎及相关技术研究[D]. 长沙:国防科技大学,2006.

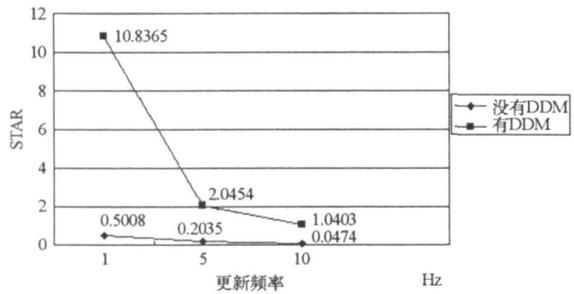


图 3 DDM 实验结果对比

Fig. 3 Experiment results comparison of DDM