

基于 HLA 的分布交互仿真数据收集系统的研究

Perspective of Data Collection System in HLA-Based Distributed Interactive Simulation

12-15

李国和 何红梅 赵沁平

(北京航空航天大学计算机科学与工程系 北京 100083)

TP391.9

TP274.2

Abstract According to the feature of data management in High Level Architecture (HLA) of the distributed interactive simulation, this paper presents three kinds of data logging structure in HLA environment and their features, and then discusses the elemental requirements of HLA-based Data Collection System, and its architecture.

Keywords Distributed interactive simulation (DIS), High level architecture (HLA)

一、引言

通过计算机网络(广域网和局域网)联结不同平台的计算环境(微机和工作站),可以构成一定地域范围内的、不断变化的分布式交互环境。在这一环境中各种仿真实体(包括真实仿真实体和虚拟仿真实体)进行与真实环境相同的活动^[1]。在仿真过程中,主要采用人在回路的驱动方式或/和以时间驱动方式推进仿真进程,具体体现在仿真实体的状态变化和仿真实体间的交互。借助数据收集系统可以全面收集仿真过程中仿真实体的状态和仿真实体间的交互。一方面,在交互仿真进行中或完全结束后,利用已收集到的状态和交互的原始数据进行驱动,以多种速度回放部分或全部的分布式交互仿真过程,即:进行仿真过程事后回顾;另一方面,全面分析已收集到的状态和交互的原始数据,可对分布式交互仿真过程中每个仿真结点、仿真应用程序和整体仿真及网络软硬件性能进行综合评价。

二、HLA 及其环境中数据管理的特点

美国国防部分布信息系统局(Distributed Information System Agency, DISA)制定了一个开放分布处理系统的技术框架,即信息管理技术系统结构框架(Technical Architecture Framework for Information Management, TAFIM)^[1,2]。在 TAFIM 的基本要求基础上确定了一种更先进的分布交互仿真技术框架——高层体系结构 HLA(High Level Architecture)。HLA 环境中的每个仿真过程和其它应用程序称为联盟成员

(Federate),所有联盟成员的集合构成了联盟^[1,6,11]。RTI(Runtime Infrastructure)是支撑 HLA 的软件平台,由 HLA 中的规则、接口规范和多种服务组成,协调和管理联盟成员的加入或退出,实现并确保联盟成员间的间接联系^[3,6]。如图 1 所示。

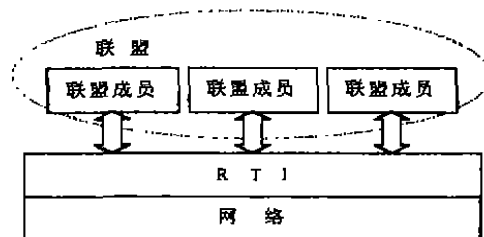


图 1

HLA 环境中数据管理的特点主要体现在:① HLA 采用面向对象技术,由用户定义 FOM(Federation of Model)为模板,创建仿真实体和仿真交互。仿真实体的属性表示实体的状态,仿真交互的参数表示仿真实体间传送的信息;②在仿真过程中,只传送更新的那一部分属性,没变化的属性不传送;③由 HLA 提供的声明管理、对象管理和数据分布管理分别实现基于类、属性和属性值的过滤;④通过 HLA 的数据发布和订购机制,以及定义区域更新(Update Region)机制实现数据的网上路由组播(Multicast)。

分布交互仿真 DIS^[1]对仿真中的实体状态和交互信息定义了固定的协议数据单元 PDU,并且以广播方

李国和 博士生,目前主要从事分布式虚拟环境数据管理和 KDD 的研究;何红梅 博士生,目前主要从事分布式交互仿真和分布式虚拟环境等的研究;赵沁平 教授,博士生导师,目前主要从事计算机软件,人工智能和虚拟现实等方面的研究。

式传递完整的 PDU。尽管采用了 DR 算法来尽可能减少 PDU 的发送,以减轻网络拥挤,但是由于即使实体仅有部分状态属性发生变化,DIS 也以广播方式发送完整的 PDU,加之仿真过程中实体的多变性,因而造成大量的 PDU 在网上流动,使得现有的网络环境成为扩大仿真规模的瓶颈。与 DIS 相比,根据 HLA 环境中数据管理的特点可以看出:HLA 除了采用面向对象技术外,更主要在于能有效减轻网络的负载,使得进一步扩大分布交互仿真的规模成为可能。

三、基于 HLA 的数据记录器的组织形式

在分布式交互仿真网上挂接的数据记录器并不主动参与交互仿真,只是被动地记录仿真过程的实体属性和交互参数。在 DIS 环境中,任何仿真结点都是以广播方式传送 PDU,因此,在交互仿真网上任何地方,只要一个数据记录器即可收集到仿真过程的所有数据(如果不是网络拥挤等原因引起数据包丢失的话)。因此,在任意网段上安置数据记录器效果都是一样的,而且不同网段上多个数据记录器记录一致性数据(备份),并不增加网络的负载。而基于 HLA 的分布交互仿真,以组播方式传递数据,挂接数据记录器的数目和位置就会影响到网络的数据流量。因此,只有选定合适的记录器数目和合理安排记录器的位置,才有可能使网络负载增加最少,保证网络的性能。根据 HLA 环境中的数据管理的特点,可以采用以下三种形式的数据记录器^[9,11~13,12]:

(1)集中式数据记录器

集中式数据记录器组织形式如图 2 所示。在交互仿真系统中只选定一个数据记录器(Logging Federation),只要数据订购充分(即不造成某一类数据的丢失),单个记录器可以挂接在交互式仿真网上的任何网段。由于数据集中记录,就可以方便地支持数据查询和分析,以及仿真的事后回顾。但在仿真过程中,任何实体更新的属性和实体交互的参数都必须路由汇集到记录器,因此,现有的网络带宽和记录器的性能可能成为数据收集的瓶颈,数据率容易达到峰值,由此可能造成

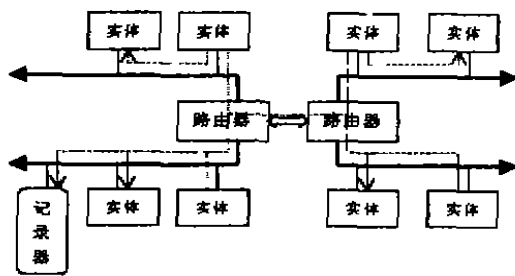


图 2

实体属性值和交互参数值的丢失。这种记录器组织方式不可能承担大规模的分布交互仿真数据的收集任务,只适合应用在较小规模、单一总线或环状结构网络的分布式交互仿真。

(2)分散式数据记录器

分散式数据记录器组织形式如图 3 所示。多个记录器可以挂接在仿真网(Federation Network)上的任何网段上,但为了提高交互仿真网络的性能,必须合理确定记录器的数目和挂接的合理位置。在具有明显分段的交互仿真网络中,考虑采用多个记录器间统一规划,“分片”订购仿真实体的属性和交互的参数,使得仿真数据传输到就近的数据记录器中,从而最大限度减少网络的负载;同时,又可以克服记录器的网络带宽的瓶颈问题,因此,这种记录器的组织形式可以支持大规模的分布交互仿真。但由于可能有不同记录器的重复订购,除了增加网络数据流量外,还造成多个记录器上对同一种类的数据重复记录;或没有按“分片就近”收集数据的原则,数据记录器过于集中在某一网段上,反而增加了网络的负载;也有可能多个记录器间订购不完备,造成订购的“空白”,而漏记某些种类数据。另一方面,由于记录数据的分布性,增加了查询和分析数据以及进行事后回顾的难度。

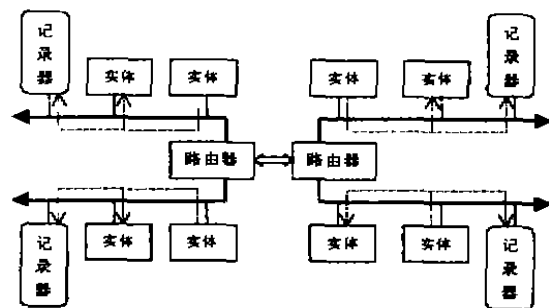


图 3

(3)完全分布式数据记录

这种形式如图 4 所示。在交互仿真网上并没有专门的记录器结点,记录数据的任务由仿真器结点自己承担。根据所有仿真实体数据(包括更新的属性和交互的参数)流入(或流出)就可以完全收集到仿真过程中的所有的数据信息,没有附加的数据流在网上流动,毫不增加网络的额外负载。这种记录方式组织起来很容易,其数据类型完全取决于仿真结点,但这种记录数据方式必然争抢仿真结点的时间和空间资源,尤其一个仿真结点仿真多个实体时,就更加降低了仿真实体的性能。由于仿真过程的所有数据完全分布于网上的所有仿真结点上,大大增加了进行及时和仿真结束后的数据查询和数据分析以及事后回顾的难度。

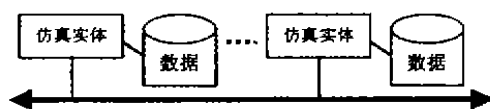


图 4

从上述分析可以看到,分散式记录器的组织形式略为增加了网络负载,但不影响仿真节点的性能,而且所收集到的数据又相对集中,因此,在大规模的分交互仿真中,大多选择这种记录数据的组织方式。目前,基于HLA环境的分布交互仿真系统—STOW(Synthetic Theater of War,战争合成演练场)的数据记录器就是以这种方式组织的。

四、分布式交互仿真数据收集系统的设计

在较大规模的分布式交互仿真网上,只有数据记录器还不够。为了最大限度方便用户,同时又能提高系统的性能和记录器资源的利用率,可以数据记录器为核心,进一步构成一个数据收集系统^[6,9]。

4.1 数据收集系统的设计基本要求

进行分布式交互仿真数据收集系统的设计时,应满足以下几点基本要求^[7]:

(1)分布性 多个记录器分布异地收集数据,尽可能降低网络负载和减轻网上数据流峰值的不良影响,确保收全网上数据;

(2)可伸缩性 增加或减少记录器数目时,不额外增加仿真系统的代价,保障适应分布交互仿真系统的数据量的变化,或分布交互仿真系统的变化;

(3)容错性 能够自动配置记录器的订购,避免记录器间的订购“空白”,此外,当数据包丢失率超过配置指定值时,启动失败处理,以及硬件异常时,能够负责数据记录器之间迁移数据等;

(4)及时查询(Runtime Query Support) 采用商用数据库管理系统提供的强壮查询机制和可靠数据存储效率和复杂查询,为仿真人员及时反馈数据。在分布仿真进行中,可以支持数据查询,而不影响数据收集和仿真正常进行;

(5)负载均衡 随着分布仿真的进行,各种类型数据的数率和经过路由空间的数据可能发生变化,应由数据收集系统在各个记录器间进行数据记录的动态调整,避免由于个别记录器太忙可能造成数据丢失,而其他记录器太闲而造成系统资源的浪费。

4.2 数据收集系统的结构

为了满足上述要求,在分布式记录器组织形式的基础上,进一步完善成为一个高度透明性、高效可行的分布交互仿真数据收集系统,还需增加其它辅助结点。

如图5所示,除了数据记录器外,还有数据收集管理工具、数据聚合库、载入管理器、数据库管理系统和图形用户界面。下面讨论各种节点的主要作用^[8,12]。

(1)数据记录器(Data Logging Federates,DLF)

通过订购机制完成分布交互仿真数据(实体属性和交互)的“分片”,这样每个数据记录器就捕获整个交互仿真过程的“一片”数据。每个数据记录器都拥有一个二进制的客户/服务器文件系统,它把收集到的二进制数据先组成文件后,再追加到数据文件服务器,最后由数据载入管理器DLM快速地把数据文件服务器中的数据载入数据聚合库。一方面,这样保证数据的收集和装载的独立性,不影响数据收集的灵活性,又便于支持数据查询和事后回顾。另一方面,数据文件服务器中的数据可支持数据安全,即使数据聚合库损坏也可由分布的数据文件重构。

(2)数据收集管理工具(Data Collection Management Tools,DCMT) 主要用于监视每个数据收集器和控制数据的订购,并提供制定可接受性能参数范围的方法。当性能参数不能满足可接受的范围,通过从盟员到盟员底层数据流的重新订购执行动态载入平衡,使得各个记录器尽可能均衡地收集数据。此外,数据收集管理工具DCMT还获取网络组播时,每组中的每个实体更新和交互参数的数目和大小、CPU的利用情况信息和网络流动的数据包以及每个记录器丢失的数据包等情况信息。

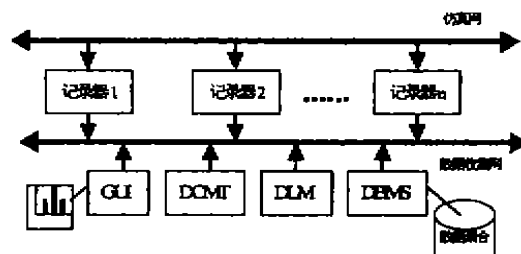


图 5

(3)数据载入管理器(Data Load Manager,DLM)

将数据记录器DLF的数据文件的数据载入数据聚合库中。数据载入管理器DLM、数据记录器DLF和数据聚合库两两相互分离,而载入管理器DLM与数据记录器DLF只通过文件系统的符号链(symbolic links)进行通信。这就保证在进行分布交互仿真的同时,可以在数据聚合库中进行数据查询、分析和事后回顾。

(4)数据库系统 在分布式交互数据采集系统中,采用关系数据库管理系统(RDBMS),在数据收集管理工具DCMT控制下,根据各个数据记录器DLF中的

各个数据文件的记录格式建立相应的数据结构,并接受数据载入管理器 DLM 载入的数据。数据记录器 DLF 和数据库模式与分布仿真的实体和交互 FOM 紧密联系。为了节省数据记录 DLF 器和数据库的空间,根据 HLA 环境中实体状态变化时只传送更新的属性,把紧密相关的属性结合在一起构成一种数据文件的记录格式和关系数据库的数据模式。这就要对仿真实体的属性进行分解。同时,为了记录仿真实体的初始状态和交互的参数,以及记录按一定时间间隔形成的仿真实体的整个状态(即所有属性),数据文件和数据库也有与仿真实体的属性和交互的参数一致的记录格式和数据模式。当从数据库中查询某时刻仿真实体的整个状态时,就没必要从仿真实体的初始状态重新逐渐构造,而是从该时刻前时间间隔时刻记录的仿真实体的完整状态重新构造即可。这样不同数据记录器的数据文件中可能有多个相同的数据记录格式,但数据库中数据模式是唯一的。数据记录器中可能有多个数据文件,数据库中就有多个数据库文件(统称数据聚合库)。

(5) 图形用户界面 GUI 用于监视记录器的状态,辅助观测数据订购和载入平衡,并提供多种可视化手段,观察系统各种性能。

结束语 HLA 在许多方面都优越于 DIS,它代表着最先进的交互式分布仿真的发展方向。基于 HLA 的分布式交互仿真采用面向对象技术和组播方式实现仿真实体间的数据交互。只有有效地规划设计分布式数据收集系统,利用 HLA 提供的数据库管理机制,才能不过分地增加网络负载,以及不影响仿真系统规模的扩大,同时,又能有效地收集到仿真过程的各种数据,达到实现高效地支持数据查询,方便仿真系统的性能分析评价和进行事后回顾。

参考文献

- 1 IEEE P1516/D1-Draft Standard Modeling and Simulation(M&S) High Level Architecture(HLA) -Federate Interface Specification
- 2 IEEE P1516/D1-Draft Standard Modeling and Simulation(M&S) High Level Architecture(HLA) -Framework and Rules
- 3 Calvin J O, Weatherly R. An introduction to High Level Architecture(HLA) runtime infrastructure(RTI). Available at <http://siso.sc.ist.ucf.edu/siw/>
- 4 Powell E T, Tarbox G H. Object-oriented framework, simulation programming paradigms, and the STOW simulation infrastructure. Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 5 Tarbox G H, Watson J F. The STOW application framework and common data infrastructure(CDI). Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 6 汪成为. 分布交互计算和分布交互仿真. 计算机研究与发展, 1998, 35(12): 1058~1063
- 7 何红梅, 王兆其, 陈小武. DVENET 应用程序框架的设计与实现. 计算机研究与发展, 1998, 35(12)
- 8 Perkinson P B, et al. How to Plan Execute Data Collection and Analysis for HLA federations. Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 9 Magee G, Shanks G. Lessons Learned an Implementation of a Fully Distributed Data Collection Tool. Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 10 Bachinsky S T, et al. Data Collection in an HLA Environment. Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 11 Calvin J O, Weatherly R. An Introduction to the High Level Architecture(HLA) Runtime Infrastructure(RTI). Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 12 Tarbox G H, et al. The STOW Application Framework and Comon Data Infrastructure(CDI). Available at: <http://siso.sc.ist.ucf.edu/siw/>
- 13 (上接第 22 页)
- 5 Kazman R, et al. SAAM: A Method for Analyzing the Properties of Software Architectures. In: Proc. of the 16th Intl. Conf. on Software Engineering. IEEE Society Press, Italy, 1994. 16~24
- 6 Kazman R, Carriere S J. Playing Detective: Reconstructing Software Architecture from Available Evidence. [Technical Report, CMU/SEI-97-TR-010]. 1997
- 7 Krikhaar R L. Software Architectuer Reconstruction. [PhD rthesis]. Unversety of Amsterdam, 1999
- 8 Kruchten P. The 4 + 1 Model View of Archirecture. IEEE Software, November 1995. 42~50
- 9 Li Bangqing, Xu Baowen, Yu Huiming. Transforming Ada Serving Tasks into Protected Objects. Proceedings of SIGAda'98, 1998. 240~245
- 10 Murphy G, Norkun D. Reengineering with Reflexion Models: A Case Study. Computer, August 1997. 29~36
- 11 Soni D, Nord R, Hofmeister C. Software Architecture in Industrial Application. In: Proc. Intl. Conf. on Software Engineering. 1995. 196~207
- 12 Storey M A D, et al. Rigi: A Visualization Environment for Reverse Engineering. In: Proc. of Intl. Conf. on Software Engineering. 1997. 606~607
- 13 Zhou Yuming, Xu Baowen. Extracting Objects of Ada Programs Using Module Features. In: Proc. of IEEE Conf. on Software Maintenance. IEEE Society Press, Oxford, 1999. 23~29
- 14 周毓明, 徐宝文. 一种利用模块内聚性的对象抽取方法. 软件学报, 2000, 11(3)