

文章编号:1001-2486(2011)06-0154-05

基于 BOM 的组件并行仿真引擎研究与实现*

何强, 刘晓铨, 彭春光, 郝建国

(国防科技大学 机电工程与自动化学院, 湖南 长沙 410073)

摘要:在组件化建模与仿真领域内,基于 BOM 规范的组件建模与仿真环境 KD-SmartSim 得到了广泛应用,它通过 RTI 实现了组件仿真系统的并行运算,但是其运行性能较低。讨论了 KD-SmartSim 系统存在的不足,在继承其基于 BOM 规范组件建模的基础上,提出了新的支持组件并行化计算的结构,设计了 SPMD 运行模式的并行组件仿真引擎,研究和开发了各种模型计算需要的仿真服务,实现了支持串行和保守时间推进机制的组件并行仿真引擎原型系统。测试结果表明该系统具备 BOM 组件仿真运行必须的功能,能够显著地提升组件系统的运行性能。

关键词:并行仿真;组件建模;仿真引擎;基本对象模型

中图分类号:TP391 文献标识码:A

Research and Implementation of Parallel Simulation Engine for Component Based-on BOM

HE Qiang, LIU Xiao-cheng, PENG Chun-guang, HAO Jian-guo

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

Abstract: In the field of M&S of Component, the component modeling and simulation environment KD-SmartSim based-on BOM specification has been used widely. The application composed of components in KD-SmartSim employs RTI to achieve parallelism, but its performance is inefficient. In this study, the deficiencies of KD-SmartSim were discussed, and a new parallelization structure for component which inherits the concept of BOM component modeling was presented. A parallel simulation engine for components in the form of SPMD was designed, which implements various services required by model computing. In the prototype system, the sequence and conservative time advancing were completed now. The experiment shows that the engine has all capacities which can support BOM component executing and can improve the performance of simulation system composed of components greatly.

Key words: parallel simulation; component modeling; simulation engine; base object mode

2006 年仿真互操作标准化组织 (Simulation Inter-operability Standard Organization, SISO) 提出了基于组件建模的基本对象模型 (Basic Object Model, BOM)^[1-2] 规范。在其基础上,文献[3]提出了基于 BOM 的组件建模框架,开发了支持仿真模型组件开发与运行的 KD-SmartSim。它将模型计算和仿真平台以及模型行为和数据分离,突出仿真模型的模块性和独立性,提高了模型的规范性,使模型具备了组合能力,增强了仿真模型的重用能力,简化了模型的设计和开发,能极大地促进仿真系统的开发效率,已成功应用于多个大中型复杂仿真系统。

KD-SmartSim 通过将模型组件映射到符合

HLA^[4] (High Level Architecture) 标准结构的联邦成员,使用运行支撑框架 (Run-Time Infrastructure, RTI) 来支持组件成员的并行执行。这种方式能够很好地重用已有资源,但 RTI 主要用于成员的互联互通互操作,并没有针对高速计算给予较多的优化。同时 KD-SmartSim 自身也存在一些不足,这些综合因素使得 BOM 组件式仿真应用运行效率不高。

为提高 BOM 组件的运行性能,迫切需要研究使用新的方法为其提供高性能的计算能力。本文的目的即是在继承基于 BOM 的组件化建模基础上,按照并行计算技术设计新的组件仿真运行平台,开发支持组件并行执行的仿真引擎——组件

* 收稿日期:2011-03-12

基金项目:国家自然科学基金资助项目(61074108)

作者简介:何强(1982—),男,博士生。

并行仿真引擎(Parallel Simulation Engine for Component, PaSEC),用以提高BOM组件系统的计算效率。首先简要介绍了BOM规范、BOM组件建模框架和KD-SmartSim系统并简要讨论了KD-SmartSim存在的不足,其次提出了组件并行仿真平台——PaSEC,然后详细说明了它的结构和功能,最后给出了系统的原型实现和性能测试。

1 BOM和KD-SmartSim介绍

BOM是一种模型的描述规范,其规范模板包括4个主要部分:模型标识、模型概念、模型映射和对象模型。模型标识提供了对BOM描述的元信息,主要包括:需求信息、概念模型信息、适用领域或范围信息和集成经验(如集成过程、集成产品、经验教训、使用方式等)、修订历史和其他信息(如顺序图表、应用剧情);概念模型包含相互作用模式、状态机、实体类型和事件类型;模型映射反映概念模型和HLA对象模型之间的映射关系;对象模型描述了对象类、交互类、对象类属性、交互类参数等OMT(Object Model Template,对象模型模版)信息,用于定义执行概念模型的仿真系统或联邦的接口信息。

BOM规范没有定义参考模型实现框架,在该规范以及借鉴其他一些仿真系统的基础上,文献[3]提出了基于BOM的组件建模框架,并将使用其构件的模型称为BOM组件。一个BOM组件由模型描述文件、模型数据文件和代码执行体三部分组成。模型描述文件使用XML格式的BOM规范描述模型组件的结构,模型数据文件设置模型的初始化状态和参数,代码执行体是独立的模型逻辑实现库(如在Windows上的.dll或Linux系统中的.so文件)。

KD-SmartSim是一组支持BOM组件开发、测试和运行的工具集,其核心是可扩展仿真运行框架,它使用通用的成员框架结构作为BOM组件的容器和运行载体。BOM组件被装配成不同的成员,不同的成员通过RTI连接成仿真联邦。成员内部组件之间的数据交换由可扩展仿真运行框架管理负责,成员之间使用RTI进行信息的发送与接收,整个联邦的仿真时间由RTI统一管理推进。

KD-SmartSim运行效率不高,最根本的原因是维护模型运行的开销过高,而这些开销主要是由于采用基于RTI的仿真联邦集成方式^[5]所引入。第一,HLA要求模型按照类层次设计、组织和管理,这种方式增加了事件的处理层次,降低了事件的传输效率(将事件发送给不相关的模型数

量多),并且严重束缚了实体之间的并行性;第二,KD-SmartSim将模型的运行管理分为可扩展仿真运行框架和RTI两层,虽然隔离了模型开发者对RTI的依赖,但使得几乎所有的操作都需要经历多次传递和数据转换,时间开销较大;第三,依赖RTI实现不同节点间的数据传输和时间推进,而RTI的此类操作开销较大。

2 PaSEC结构设计

PaSEC的设计思想与OpenUTF^[6]相似,采用了基于即插即用的模型集成方式^[5],在整体上将系统分为仿真模型和仿真引擎两部分,二者保持高度独立,模型能以即插即用的方式加入仿真。模型集成方式能更好地平衡通信开销与模型计算时间,从而使系统获得较高的性能。由BOM组件构成的系统可以被视为一个有向图 $G = \langle V, E \rangle$, V 是实体, E 是它们之间的数据连接关系。将系统以模型实体为单位划分成多个子集合,每个子集合部署在一个计算节点上,每个节点运行一个仿真引擎的实例,从而利用多个处理器同时计算加速系统的仿真进程。

PaSEC使用主从结构设计,采用单进程多数据(Single Program Multiple Data, SPMD)形式,系统从整体上分为组件服务器和组件仿真器两部分,其结构如图1所示。使用PaSEC对组件系统进行计算需要启动多个独立的进程,这些进程按照0,1,⋯,N的序号命名,称0进程为主节点,它充当组件服务器,其他进程为从节点,执行组件仿真器功能。

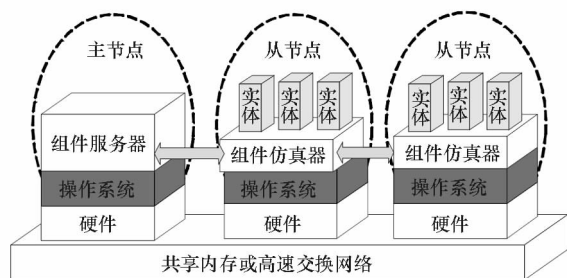


图1 PaSEC结构

Fig. 1 Architecture of PaSEC

组件服务器负责仿真任务规划、向从节点分配任务、状态监控和控制管理以及与其他仿真系统互联(如和HLA联邦)等,组件仿真器负责管理、维护和推进模型计算。在组件仿真器中,模型实体是独立自主的,它能为自己或其他实体规划新事件,也能接受仿真器的调度。模型按照事件方式建模,组件仿真器采用事件调度法管理模型的运行。组件仿真器具有组件管理功能和多种仿

真支持服务,具体将在第 3 节中详细介绍。PaSEC 运行在共享内存或高速网络相连的计算机上,所有节点之间通过消息传递的方式进行信息交换。在 PaSEC 中,各节点必须首先加载构成仿真系统的组件,然后通过任务交互在各个节点上创建出各种类型的实体。一般地,在主节点上不创建任何实体。

与 KD-SmartSim 相比,PaSEC 做了以下革新以降低系统开销。第一,基于实体层次的模型组织形式,引擎可直接调度实体而不需要经过任何中间层次,能够显著地提高事件传输和处理效率。第二,采用事件调度法推进仿真,不仅有利于提高事件调度速度,还使系统的扩展性增强。第三,将可扩展仿真运行框架的组件运行支撑功能和部分 RTI 服务融为一体,减少调用层次,如 PaSEC 核心直接支持实体层次的公布订购声明管理,允许实体定义数据的过滤区域,提供时间推进和事件传输等功能。

3 组件仿真器

图 2 示意了组件仿真器的层次结构。最底层为操作系统及相关通用工具,它对组件仿真器及模型的开发提供基本的支持;组件仿真器位于操作系统层之上,提供组件模型运行需要的所有功能;其上的建模框架层为模型开发提供支持,它是抽象模型接口定义,详细信息参考文献[7];最上面的应用层是各种类型的模型组件,遵循 BOM 组件建模规范。



图 2 组件仿真器结构

Fig. 2 Layered structure of component simulator

组件仿真器采用模块化设计方式,把仿真引擎的功能分成不同层次的服务模块,各层次具有相对独立的功能,但又相互联系,下层为上层的实现提供支持和服务,这样在保持模型接口不变的前提下,更新和添加仿真引擎的功能而不会影响已有的模型组件,从而使它们获得良好的重用性。组件仿真器按照层次结构组织,从下到上依次为

通信层、核心层和服务层,下面将依次对它们的功能进行详细介绍。

3.1 通信层

通信层位于系统的最底层,它为上层的各项功能提供支持。其基本功能包括:消息传递、同步操作和全局规约操作。消息传递用于模型间的信息交换,包括节点内部和节点之间两种类型的交互。消息传递提供阻塞和非阻塞两种传输方式的单播、组播和广播消息发送能力。同步和规约本质上是相同的,同步用于协调节点之间的运行过程,而规约用于全局虚拟时间(Global Virtual Time, GVT)的计算。

通信模块能够根据模型分布位置的不同自动透明地选择最快速的传输方式,如在同一节点内部使用指针共享,在同一台计算机上使用共享内存,而对运行在不同机器的节点上则选择 TCP/IP 或其他快速网络交换协议。

3.2 核心层

核心层功能包括组件管理、事件管理、时间管理、持久框架、数据采集和统计等。其中组件管理、事件管理和时间管理是系统最基本的功能模块,持久框架用于支持乐观时间推进,统计用于仿真引擎的性能分析。

(1) 组件管理

组件管理的功能是加载构成系统的模型组件资源,创建和管理组件对象,支持模型实体的创建、删除和初始化等操作,管理组件数据类型。

(2) 事件管理

事件管理的核心是事件的插入、排序和取出等操作,当有新事件 E 到达节点时按时戳顺序将 E 插入到实体的未来事件表(Future Event List, FEL)中,当事件被调度时从 FEL 中取出时戳最小的事件进行处理。事件队列操作是仿真计算中最频繁的过程,其效率对计算性能有非常大的影响。

(3) 时间管理

时间管理是整个系统的核心部件,其全部功能就是不断地处理 FEL 中时戳最小的事件,直至仿真结束,对不同的时间推进机制采用不同的时间管理算法。其基本开销来自两个方面:FEL 的事件删除和时间同步操作。FEL 采用了基于 AVL 方法的优先级队列,它对事件插入和删除的时间复杂度均为 $O(\log N)$ 。时间同步算法使用了 butterfly 方式,它需要 $\lceil \log N \rceil$ 步完成计算,消息传递总量为 $N \lceil \log N \rceil$ 。这两种算法的时间复杂度和可扩展性很好,有利于支持大规模的仿真。而传

统的基于步进方式的时间开销很容易随着仿真规模的扩大而增大。

(4)持久框架

持久框架是用于支持实现仿真或模型实体持久性的软件结构,它由持久库、用户接口和持久机制 3 部分组成。持久库为对象持久化提供基本支持,用于辅助模型状态的保存和恢复。用户接口只有保存和恢复两个函数,它是模型必须实现的操作,在乐观时间推进机制中由仿真引擎调用。持久机制即检查点与恢复机制,保证获得一致检查点。

(5)数据采集

数据采集模型记录模型的状态更新和交互发送,它以通用仿真数据收集算法^[8]为基础,通过采集配置信息从目标仿真系统订购感兴趣的数据,将它们存入文件或数据库中。

(6)统计

统计信息的收集对于研究分析仿真系统的性能是非常重要的。在统计模块中要收集的信息有:处理的事件数、反消息数、每个模型的处理时间、内存使用情况和等待计算 GVT 的空闲时间等。进行信息统计会带来一定的开销,可以在不需要时关闭这项功能。

3.3 服务层

PaSEC 支持 HLA 的声明管理、对象管理和数据分发管理服务,并对它们进行了适当的扩展。

(1)声明管理

声明管理的主要目的是在模型实体之间建立公布和订购关系,支持事件的分发和减少网络流量,系统构建了两层声明关系,第 1 层是节点内部模型实体之间,第 2 层是在节点之间。第 2 层和 HLA 的成员声明管理服务是相同的,它用来控制网络中的事件传输。通过建立实体间的声明关系,能够提高事件的传输效率,尤其是对单一目标类型的事件,能够完全杜绝不相干的同类别实体接收到该事件。

(2)对象管理

对象管理在声明管理的基础上,实现模型实体的注册/发现、属性值的更新/反射、交互实例的发送/接收以及模型实体的删除等功能。系统实现了 HLA 部分对象管理服务。

(3)数据分发管理

数据分发管理的目的是减少仿真运行过程中无用数据的传输和接收,从而减少网络中的数据量。此模块完全遵循 HLA 标准。

4 原型实现

4.1 原型系统结构

PaSEC 的原型系统的主要结构如图 3 所示。其中类 LogicProcess 定义了仿真运行控制的抽象接口,如仿真初始、启动、停止和结束等,组件服务器 ComponentServer 和组件仿真器 LPSimulator 从该类继承,在启动时通过多态方式实现 SPMD 的运行模式。仿真过程中,LPSimulator 管理多个仿真对象 SimObj(模型实体),SimObj 和 LPSimulator 通过桥梁模式关联起来,仿真代理接口 IFProxy 用于连接它们,它定义了模型必须的仿真服务接口,赋予模型访问仿真平台功能的能力,SimulatorProxy 是其具体实现。

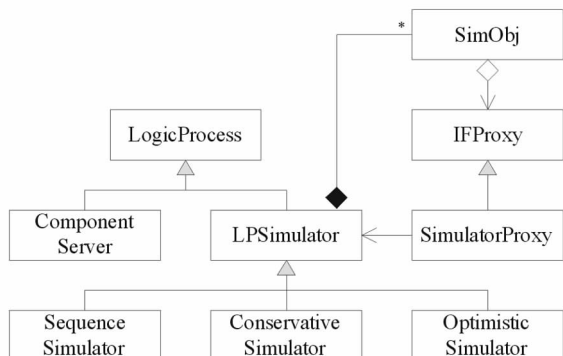


图 3 PaSEC 主要类结构图

Fig. 3 Overview classes of PaSEC

LPSimulator 是整个系统的核心,它实现了通信服务、组件管理、事件管理、时间管理、持久框架、数据采集、统计、声明管理、对象管理和数据分发管理等功能。串行仿真器 Sequence Simulator、保守仿真器 Conservative Simulator 和乐观仿真器 Optimistic Simulator 必须继承自 LPSimulator。原型系统已完成了串行和保守仿真器的开发,使用 MPI 实现进程间通信。

4.2 运行过程

PaSEC 的运行由仿真建立、仿真执行和仿真结束 3 个阶段组成,详细过程如下:

(1)仿真建立阶段

首先,对系统进行初始化,在这个过程中需要完成两项任务。①主节点选择组件服务器,其他从节点执行组件仿真器功能。②各个节点按照仿真系统配置加载组件资源,并进行相应的初始化操作。

其次,主节点读取外部任务规划信息或按照默认方法为从节点划分子任务。

第三,主节点部署规划任务,从节点接收

任务。

第四,所有节点同步以确保从节点都已完成任务的接收,并且准备好进行仿真执行。

(2) 仿真执行阶段

当仿真建立阶段结束后,主节点下发仿真开始命令,从节点收到该指令后进入事件处理阶段。依照不同的事件类型,其任务可能是创建新的模型实体、模型参数调整或者模型计算等。

(3) 仿真结束阶段

当运行到指定仿真时间或达到结束条件,主节点发布结束仿真命令,从节点收到该命令后退出仿真执行,在所有节点同步完成后,清理运行现场并退出运行。

5 性能测试

为评价 PaSEC 的性能,选用 PHOLD 模型。测试方案产生 1000 实体,平均分配到各个节点,每个实体规划 1000 个事件发送给随机目标, L 代表本地节点上的实体规划事件的概率。测试平台为两台 CPU 为 Intel Core2 Q6600 @ 2.4GHz 的四核计算机。

在性能测试中,当实体收到事件时执行约 0.1ms 的计算任务,整个方案的模型计算约为 100s。图 4 显示了 $L = 0.5$ 时 PaSEC 与 KD-SmartSim 运行该方案所用时间。从图中可以看出,在各个计算规模上,PaSEC 都比 KD-SmartSim 快 10 倍以上,这说明 PaSEC 能够降低维护组件模型运行的开销。PaSEC 的执行时间随着节点数量的增加而降低,它能有效地利用实体之间的并行性加快仿真的执行。

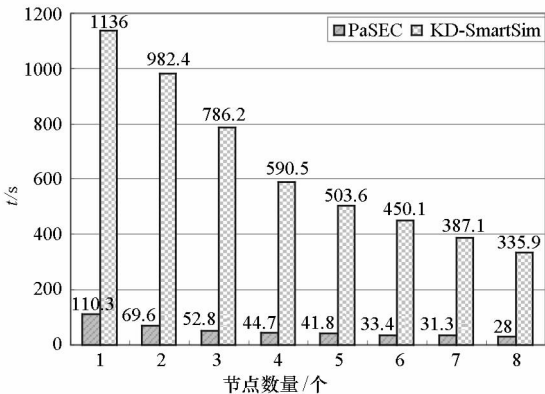


图 4 PaSEC 和 KD-SmartSim 运行时间对比
Fig. 4 Execution time of PaSEC vs. KD-SmartSim

加速比测试中,模型收到事件后不再执行负载计算。图 5 显示了 PaSEC 在 $L = 1$ 和 $L = 0.5$ 时的加速比。在 $L = 1$ 时,获得了近乎线性的加速比,而在 $L = 0.5$ 时,加速比的增长速度比 $L = 1$ 时

明显下降,在 8 节点时的加速比只达到 $L = 1$ 时的 50%。加速比增长下降原因是:随着节点数增多,节点间的事件传输数量增多,通信开销增加,从而导致加速比下降。

6 结论

通过对基于 RTI 的 KD-SmartSim 组件并行仿真系统不足的分析,在基于 BOM 的组件化建模基础上,设计了支持 BOM 组件并行离散事件仿真引擎 PaSEC,实现了原型系统。测试结果表明,PaSEC 具有组件仿真所需的各项功能,能够以很小的开销管理组件模型的运行,显著地提高仿真系统的性能。

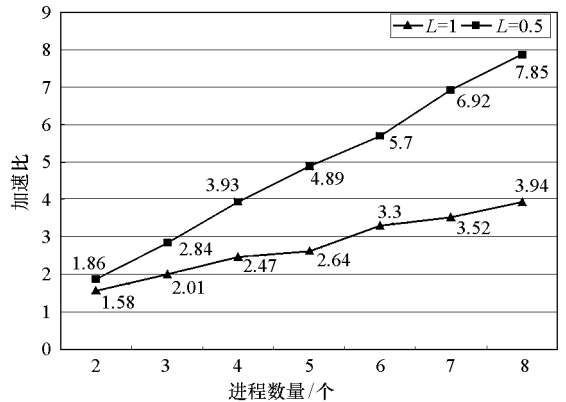


图 5 $L = 1$ 与 $L = 0.5$ PaSEC 加速比对比
Fig. 5 Speedup of PaSEC with $L = 1$ vs. $L = 0.5$

参考文献:

- [1] Guide for Base Object Model (BOM) Use and Implementation [S]. SISO-STD-003.1-2006.
- [2] Base Object Model (BOM) Template Specification [S]. SISO-STD-003-2006.
- [3] 龚建兴. 基于 BOM 的可扩展仿真系统框架研究[D]. 长沙:国防科技大学, 2007.
- [4] DMSO. High Level Architecture Run Time Infrastructure Programmer's Guide[R]. DOD, 2000.
- [5] Steinman J S, Park J, Walter B, et al. A Proposed Open System Architecture for Modeling and Simulation [G]// Simulation Interoperability Workshop, 2007.
- [6] Steinman J S, Lammers C N, Valinski M E. A Unified Technical Framework for Net-centric Systems of Systems, Test and Evaluation, Training, Modeling and Simulation and Beyond...[G]// Simulation Interoperability Workshop, 2008.
- [7] 何强, 陈彬, 钟荣华, 等. 基于 DEVs 的 BOM 组件与仿真引擎研究[J]. 系统仿真学报, 2010, 22(11):2505-2510.
- [8] 陈彬, 王全民, 龚建兴, 等. 作战模拟态势显示的关键技术研究[J]. 国防科技大学学报, 2009, 31(6):115-120.