

9-17-2021

Research and Implementation of a High Performance Distributed Object-oriented Simulation Engine

Xiaodong Huang

Service College, Naval Aviation University, Yantai 264001, China;

Kongshu Xie

Service College, Naval Aviation University, Yantai 264001, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Research and Implementation of a High Performance Distributed Object-oriented Simulation Engine

Abstract

Abstract: Simulation engine is the core of simulation platform and the operating system of simulation field. *On the basis of the reflection-based object-oriented simulation modeling framework and programing specification, aiming at the developing integration and operation requirements for the large-scale multi-granularity computation intensive simulation system, a High-performance Distributed Object-oriented Simulation Engine (HDOSE) is developed.* Combined with the brief review of HDOSE development, on the basis of expounding the HDOSE function services and architecture, *the key technology, large-scale multi-granularity parallel computation, programming model and interface, parameterized componentized modeling and so on are mainly introduced.* The HDOSE typical application cases and performance test cases are provided. The application results show that HDOSE has good performance and extendibility, and can rapidly help to construct the computation and interaction intensive complex application systems.

Keywords

simulation engine, object oriented, high extendibility, distributed parallel integration

Recommended Citation

Huang Xiaodong, Xie Kongshu. Research and Implementation of a High Performance Distributed Object-oriented Simulation Engine[J]. Journal of System Simulation, 2021, 33(9): 2215-2226.

高性能分布式面向对象仿真引擎研究与实现

黄晓冬, 谢孔树

(海军航空大学 战勤学院, 山东 烟台 264001)

摘要: 仿真引擎是仿真平台的核心, 是仿真领域的操作系统。在反射式面向对象仿真建模框架和编程规范基础上, 针对大规模多粒度计算密集型仿真系统开发集成及运行的需求, 开发了高性能分布式面向对象仿真引擎(High-performance Distributed Object-oriented Simulation Engine, HDOSE)。简要回顾了 HDOSE 的发展历程, 在阐述 HDOSE 的功能服务和体系结构基础上, 重点介绍了大规模多粒度并行计算、编程模型与接口、参数化组件化建模等关键技术。给出了 HDOSE 的典型应用案例和性能测试案例。结果表明: HDOSE 具有性能好、可扩展性强等优点, 可帮助用户快速构建计算和交互均密集的复杂应用系统。

关键词: 仿真引擎; 面向对象; 高扩展性; 分布并行一体化

中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X (2021) 09-2215-12

DOI: 10.16182/j.issn1004731x.joss.21-0228

Research and Implementation of a High Performance Distributed Object-oriented Simulation Engine

Huang Xiaodong, Xie Kongshu

(Service College, Naval Aviation University, Yantai 264001, China)

Abstract: Simulation engine is the core of simulation platform and the operating system of simulation field. On the basis of the reflection-based object-oriented simulation modeling framework and programming specification, aiming at the developing integration and operation requirements for the large-scale multi-granularity computation intensive simulation system, a High-performance Distributed Object-oriented Simulation Engine (HDOSE) is developed. Combined with the brief review of HDOSE development, on the basis of expounding the HDOSE function services and architecture, the key technology, large-scale multi-granularity parallel computation, programming model and interface, parameterized componentized modeling and so on are mainly introduced. The HDOSE typical application cases and performance test cases are provided. The application results show that HDOSE has good performance and extendibility, and can rapidly help to construct the computation and interaction intensive complex application systems.

Keywords: simulation engine; object oriented; high extendibility; distributed parallel integration

引言

建模仿真技术具有安全性、复用性、经济性等特点, 已成为装备论证、型号研制、试验鉴定、效能评估、模拟训练、编配优化、作战运用、技术保

障等国防应用领域的关键支撑技术。仿真平台是支撑仿真应用系统的开发集成及运行应用的规范、工具、模型、数据的集合, 也是连接仿真应用与信息技术的桥梁, 对提高仿真应用系统的开发效率和质量具有重大的促进作用^[1-2]。

收稿日期: 2021-03-19 修回日期: 2021-04-14

基金项目: 装备预研项目(41401020401, 41401050102)

第一作者: 黄晓冬(1975-), 男, 博士, 教授, 研究方向为计算机软件、系统建模与仿真、人工智能应用。E-mail: 3065351527@qq.com

仿真引擎是仿真平台的核心，是仿真领域的“操作系统”。仿真引擎既是仿真系统的“发动机”，也是其“神经中枢”和“骨架”。仿真引擎在定义一套开放可扩展的模型开发和运行规范基础上，对仿真对象、事件、时间进行精确管理并对底层资源(计算、存储、通信等)动态分配，使仿真系统能有条不紊的按要求运行。仿真引擎与模型框架是既分离又密切相关的，模型框架主要定义仿真领域内主要对象的功能、属性、交互、参数及对象之间的关系；仿真引擎为模型框架的实现提供基础服务，使框架关注领域内本身的问题。因此，狭义的仿真引擎，指仿真系统集成和调度的内核，而广义的仿真引擎，则包括仿真系统内核和模型框架。本文介绍的仿真系统内核与模型框架是分离的，因此，在无特别说明的情况下，仿真引擎指的是仿真系统内核。

1 相关工作

20 世纪 90 年代初，欧洲航天局为解决航天领域仿真标准化规范化的问题，设计了 Model-API 并发展为仿真模型可移植规范(Simulation Model Portability Specification, SMP)^[3]。SMP 既是仿真模型的标准规范，更是仿真引擎开发与实现的标准规范，较好解决了仿真引擎的组件化、参数化、仿真运行流程及调度等问题，现已发展到 2.0 版。SMP 的主要问题是规范过于繁琐和笨重，严格按它的要求去实现在性能和易用性方面会大打折扣，因此并没有得到很广泛的应用。

美军在开发大型仿真系统的同时，也同步开发了仿真引擎。如在联合战区仿真系统开发中，形成并应用了离散事件仿真引擎来支持大规模兵力对象的推演仿真^[4]。联合仿真系统 JSIMS(Joint Simulation System)中，仿真引擎 CCSE(Common Component Simulation Engine)以公共组件形式存在，其借鉴了组件化建模思想，实现了实体和组件 2 种基类，在实体级对组件进行组装，然后统一集成到 CCSE 仿真框架中，但其发布订购机制与模型的解耦度不高，导致开发运行效率不高^[4-5]。

EADSim (Extended Air Defense Simulation)，CMANO (Command: Modern Air Naval Operations) 等海空作战仿真系统的仿真引擎支持事件和实体的并行调度，大幅提高了运行性能，但这 2 个系统并没有单独把仿真引擎单独解耦出来，因此难以支持二次开发。

在国内，北京仿真中心较早研究并开发了多个版本的仿真引擎，并基于它们形成了面向虚拟样机的平台 COSIM(Collaborative Simulation Platform)和面向体系仿真的平台 CISE(Component-Based Integrated Modeling and Simulation Environment)^[2,6-7]。乔海泉等提出了并行离散事件仿真引擎框架，开发实现了支持并行与分布计算的 KD-PADSE (KD Parallel and Distributed Simulation Engine)仿真引擎^[8]，并将其应用于联合建模与仿真环境 JMASE (Joint Modeling and Simulation Environment)^[9]。苏年乐等^[10-11]基于 SMP2 开发了支持并行计算的仿真引擎。何强等基于 DEVS(Discrete Event System Specification)实现了以事件为中心并支持 BOM (Base Object Model)组件集成的仿真引擎^[12]。华如科技的可扩展仿真平台(XSIM)和中船 716 所的体系建模与仿真平台(MARS)也将仿真引擎作为核心内容进行了开发和实现，二者均设计了事件管理器为每个执行线程建立单独的事件列表，大幅提高仿真事件调度与执行效率。

计算和调度能力是仿真引擎最重要的指标之一，美国推出了以 SPEEDES (Synchronous Parallel Environment for Emulation and Discrete-Event Simulation)为代表的支持高性能计算的仿真引擎^[13]。随着并行计算的规范化，目前大多数仿真引擎均采用 MPI(Message Passing Interface)或 OpenMP (Open Multi-Processing)实现并行计算。

国内外仿真引擎普遍存在的问题：①与应用捆绑过紧，通用性和可移植性较差；②功能与接口设计不合理，模型开发难度大；③与底层通信、数据库等基础设施绑定，适应性差；④并行计算时需要模型配合，对编程人员要求太高；⑤集成方式少，

构建方式不灵活, 应用范围窄; ⑥对大规模多粒度模型调度和解算能力不足, 容量小, 性能差。

针对上述问题, 对仿真引擎进行了深入研究和持续 20 多年的改进及应用, 在系统构建及集成方式、实体对象组织、多粒度模型调度、资源优化分配、开发应用模式等方面取得了较大的进展, 研制了高性能分布式面向对象仿真引擎 (High Performance Distributed Object-Oriented Simulation Engine, HDOSE), 并取得了良好的应用效果^[14-20]。

2 HDOSE 发展历程

HDOSE 是分布并行一体化的高性能仿真引擎, 它起源于某大型飞机模拟器项目^[14], 其最初目标是针对分布实时紧耦合仿真系统集成需求, 封装底层中间件并为上层应用提供简单易用的集成框架和接口。经过 2001—2004 年的开发, 形成 1.0 版本, 成功保障了该型飞机模拟器的研制。HDOSE 1.0 的主要特征是具有反射式对象化的编程模型及便捷的开发接口。

HDOSE 2.0 开发于 2005—2009 年, 主要新增特征是将底层通信协议/中间件与核心服务分离, 使两者可以独立演化; 同时使模型以组件方式开发和集成, 并引入仿真参考标记语言 (Simulation Reference Makeup Language, SRML) 作为模型描述规范。

HDOSE 3.0 开发于 2009—2013 年, 目标是提高系统的性能。新增特征包括: 支持多核并行调度和多级调度; 通过报文合并、按需推送等方法优化了通信性能; 编程模型上新增对流对象的支持; 实现了 Linux 和 Windows 64 位环境下多个版本。

HDOSE 4.0 开发于 2013—2016 年, 目标是为异类异构紧耦合系统集成, 而不仅仅局限于为仿真系统服务。主要新增特征是对桥接器的接口进行了标准化设计和重构, 开发了基于 TCP/IP、共享内存 SM (Shared Memory)、反射内存 RM (Reflected Memory) 等多种类型桥接器。

HDOSE 5.0 开发于 2016—2019 年, 针对大规模多粒度仿真计算的需求, 通过优化多核环境下的

资源分配方案, 进一步大幅提升了计算效率和系统容量。

3 HDOSE 体系结构与功能服务

在体系结构设计上, HDOSE 分为应用层、服务层和连接层, 如图 1 所示。应用层包括基于 HDOSE 的模型、工具、GUI 应用等, 是由引擎用户定制开发而成 (若应用层由非 C++ 语言开发, 则需进行包裹); 服务层以反射式对象系统为内核^[14], 以对象化的 API 接口提供系统管理、通信管理、组件管理、对象管理、事件管理、时间管理、场景管理、调试支持功能服务, 实现了集成组件化、编程对象化、配置参数化、调度自动化等特征。连接层封装了 RTI (Run Time Infrastructure), DDS (Data Distribution Service), TENA (Test and Training Enabling Architecture), ACE (Adaptive Communication Environment) 等中间件, 以及共享内存 SM、反射内存 RM, TCP 等更底层的通信协议, 同时将本地的事件或对象状态适时与远程节点交互和同步, 为分布式系统提供发布订阅模式的通信服务并协调仿真逻辑时间, 实现了分布并行一体化的计算, 体现了协议多元化和通信透明化的特点。

从功能设计来看, HDOSE 系统管理主要包括: 引擎服务进程和服务对象的创建、初始化、配置及状态维护; 仿真进程控制 (如开始、暂停、恢复、重演、结束等); 系统结构及组件、模型、实体、对象、参数、事件的运行状态监控及日志记录。

通信管理主要包括: 以配置的方式描述对象、事件的订阅和发布关系, 并在运行时实现分布环境下仿真对象间透明地交互; 封装底层中间件 (如 RTI, DDS) 的声明管理和对象管理等功能并为之兼容; 为非订阅发布模式的通信方式 (如 TCP/IP、共享内存) 提供订阅发布机制; 进行报文的封装、合并、分发、接收、解析、推送。

组件管理主要包括: 加载和初始化组件, 实现组件化集成; 检查和匹配组件, 对不符合规范的组件进行告警; 提供组件查询、序列化等服务等。

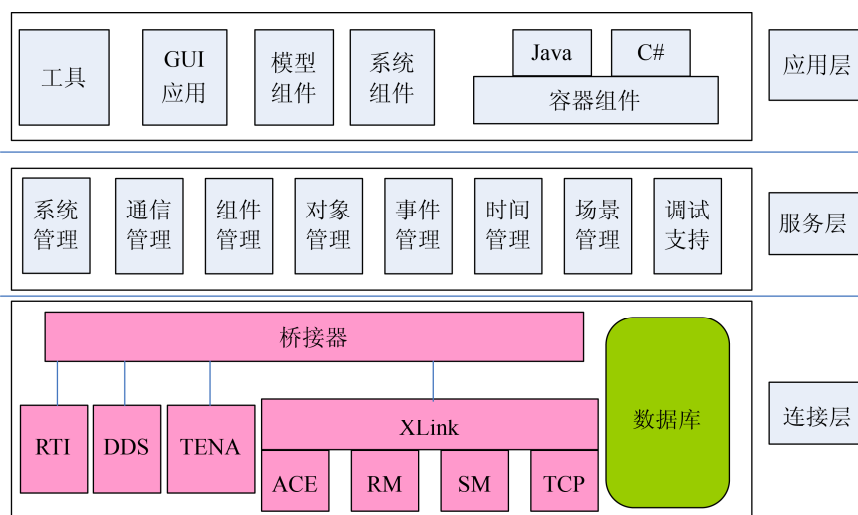


图 1 HDOSE 体系结构与功能
Fig. 1 HDOSE architecture and function

对象管理主要包括：为联邦范围内的所有对象实例提供反射式对象树结构的组织方式；提供高级的对象创建、命名、ID 号管理、查找、删除等服务和回调通知；提供对象的聚合和组装机理，支持将仿真实体分解为多个仿真对象；与通信管理相结合，按设定的策略自动完成各个节点对象状态的更新和映射；监视和管理为对象分配的内存块，防止内存泄露；将批量对象封装为流对象进行管理传输等。

事件管理主要包括：为联邦、成员、实体 3 个层级的隐式调用提供统一的方式和接口；实现以静态方式和动态方式对事件进行创建；根据事件传播范围和事件接收描述信息传送事件；按给定的同步或异步策略进行事件调度等。

时间管理主要包括：为联邦内的所有仿真对象和事件提供统一的逻辑时钟和同步机制，确保因果关系正确；与底层中间件(如 RTI)时间管理服务(TAR, TAG 等)的时间管理策略(TC/TR/NTC/NTR)集成；提供动态改变仿真逻辑步长和物理触发间隔的服务；提供毫秒级的触发时钟服务等。

场景管理主要包括：解析基于 SRML 的场景描述文件，根据描述的内容按时间序列构建仿真对象、仿真实体、仿真事件、行为计划；在运行时执行场景指令(如打开或关闭场景、创建或删除对象、创建事件等)；调度场景中的仿真对象；由实验方

案动态生成实验样本文件；根据需要记录与回放场景等。

调试支持主要包括：输出引擎关键状态或事件日志、检查内存泄露、运行异常处理等等。

4 HDOSE 关键技术

4.1 大规模多粒度模型并行计算技术

仿真引擎的计算和调度能力不仅影响系统运行的速度，还在很大程度上决定了系统的规模和精度。当前，仿真系统规模越来越大，模型种类和数量越来越多，比如，在军事体系对抗仿真领域，需要对整个战场的兵力实体进行仿真计算，实体数量通常要达到 5 000 个以上，实体类别涉及车辆、舰船、导弹、飞机、卫星等^[17]；模型的种类涉及运动、感知、通信、控制、决策等，模型的物理机理涉及专业包括力、声、热、电、光等^[17-18]。这对仿真引擎的计算和调度能力提出了更高的挑战，而且这种挑战不是简单地通过提升硬件能力就能解决的。

受到摩尔极限的制约，目前计算机的计算能力主要通过增加运算核心数量来实现^[21-22]，因此，仿真引擎软件如何充分利用多核并行计算的潜能，并针对仿真的特点设计合理的模型调度和资源分配策略显得尤为关键。

国内一些学者把并行仿真分为作业级、任务级、模型级、线程级^[2]。考虑到作业级与任务级本质上都是松耦合的并行, 本文将并行仿真计算分为样本级、模型级和算法级。样本级并行比较容易实现^[23], 而算法级并行则根据领域特点进行具体设计和开发, 因此, 对于仿真引擎来说, 核心问题是支持模型级的并行。

与样本级并行相比, 模型级并行主要特点是仿真模型间的耦合非常紧密, 通常在毫秒或微秒级, 因此, 调度和管理仿真时间的精度远远高于样本级。如果时间精度过低, 将导致采样间隔大, 最终使得模型运算结果严重失真, 这是高精度实时仿真所不允许的, 特别是高速飞行器和电子装备的仿真尤为突出^[17]。模型的调度间隔比操作系统调度线程的间隔更小, 通常一个线程被执行的时间片段内, 会有若干个仿真模型被执行^[24-27]。

HDOSE 提高计算性能的主要方案是通过精准控制和管理每个模型对象及仿真时间, 确保时空一致情况下将计算、存储、通信等资源适时分配到每个对象。其中, 计算资源的分配最为关键。如图 2 所示, HDOSE 采用了基于线程池的分配方案, 采用了基于 OpenMP 提供的 Fork-Join 并行计算模式, 即在给定的同步点主线程分叉(Fork)产生若干子线程, 每个子线程分配到类型相同的仿真对象, 然后在所有子线程完成计算后汇聚(Join)到主线程, 以实现模型对象级的并行。



图 2 并行计算资源分配示意图

Fig. 2 Parallel computing resource allocation diagram

本文方法的核心是将计算资源按需和按时精准地分配到仿真模型上。其中, 按需分配意味着模型自身对计算资源的需求要进行自描述, 由于不同粒度模型的需求是不一样的, 从而防止了不必要的分配所形成的浪费; 按时分配意味着在定义调度接口规范基础上, 模型的回调函数按规定的策略在线程上执行, 在保证并行调度的同时确保在规定的时间内进行同步。具体调度主要采用了以下策略:

策略 1: 将模型对象按模型类别进行分类, 同类别模型对象调度完成后, 再调度下一类别模型对象。由于同一类模型的算法是相同的, 计算量也相同, 因此, 将这些模型对象同时分配给多核线程进行计算时, 可以有效保证负载均衡, 使模型对象在各个核上同时完成并达到同步点, 大幅提高系统的并行性。比如, 假设飞机实体中有飞机运动模型和机载雷达探测模型, 如果应用本方法调度 1 000 架飞机实体, 应该是先调度 1 000 架飞机的运动模型, 再调度 1 000 架飞机的雷达探测模型。否则, 若将 1 000 架飞机的运动模型和相应的雷达探测模型放在一起进行调度, 由于每架飞机的状态不同(比如有的飞机没开雷达), 每次调度计算量可能并不相同, 最终导致负载不平衡。

策略 2: 仿真模型的每个仿真步分解为不同功能的回调函数, 调度系统按规定的时序逻辑扫描这些回调函数, 并将符合调度条件的函数分配到线程池中执行。由于每个回调函数具有特定的语义, 这样可以保证调度程序按语义顺序进行调度, 从而避免了数据互斥造成的相互等待和死锁。本文定义的主要回调函数如下:

- (1) void OnInit();
- (2) void Tick();
- (3) void Simulation();
- (4) void Output();
- (5) void OnClose()。

上述 5 个函数定义了模型对象的 3 个阶段。首先是初始化阶段, 即 OnInit 函数, 该函数在对象的

生命周期中只被调用一次,用以执行一些必要的初始化操作。其次是运行阶段,有 3 个函数,其中 Tick 函数用以执行模型自身功能的解算, Simulation 函数用于接收到其他对象数据后进行反应性解算, Output 函数用以执行将模型数据推送出去的操作。运行阶段的函数在每次仿真推进时都可能被调用,这是被调用的主体。最后是退出阶段,其中的 OnClose 函数用于执行对象将被删除时的一些操作,该函数只被调用一次。

模型的调用主要发生在运行阶段。结合策略 1 所描述的按同类合并进行调度的策略,在运行期的调度方案如图 3 所示。调度算法的核心在于保持并行调度的同时,在预定的同步点上等待,以防止多

线程间因数据互斥而出现死锁或等待^[28-29]。在每一个调度块内部,比如 Tick 调度块,按策略 1 先调度某一类模型对象的所有 Tick 函数,再调度另一类模型对象的所有 Tick 函数,其他调度块类似。从图 3 可以看出,当实体对象规模越大,分配负载越容易达到均衡,系统的并行度越高。

4.2 编程模型与 API 接口设计

编程模型和接口是仿真引擎设计的重要内容,极大地影响着模型开发效率。HDOSE 的接口服务分仿真对象服务和仿真支撑服务 2 部分^[16],规范明确定义了它们的职责和接口形式,实现两者的高效集成和交互,如图 4 所示。

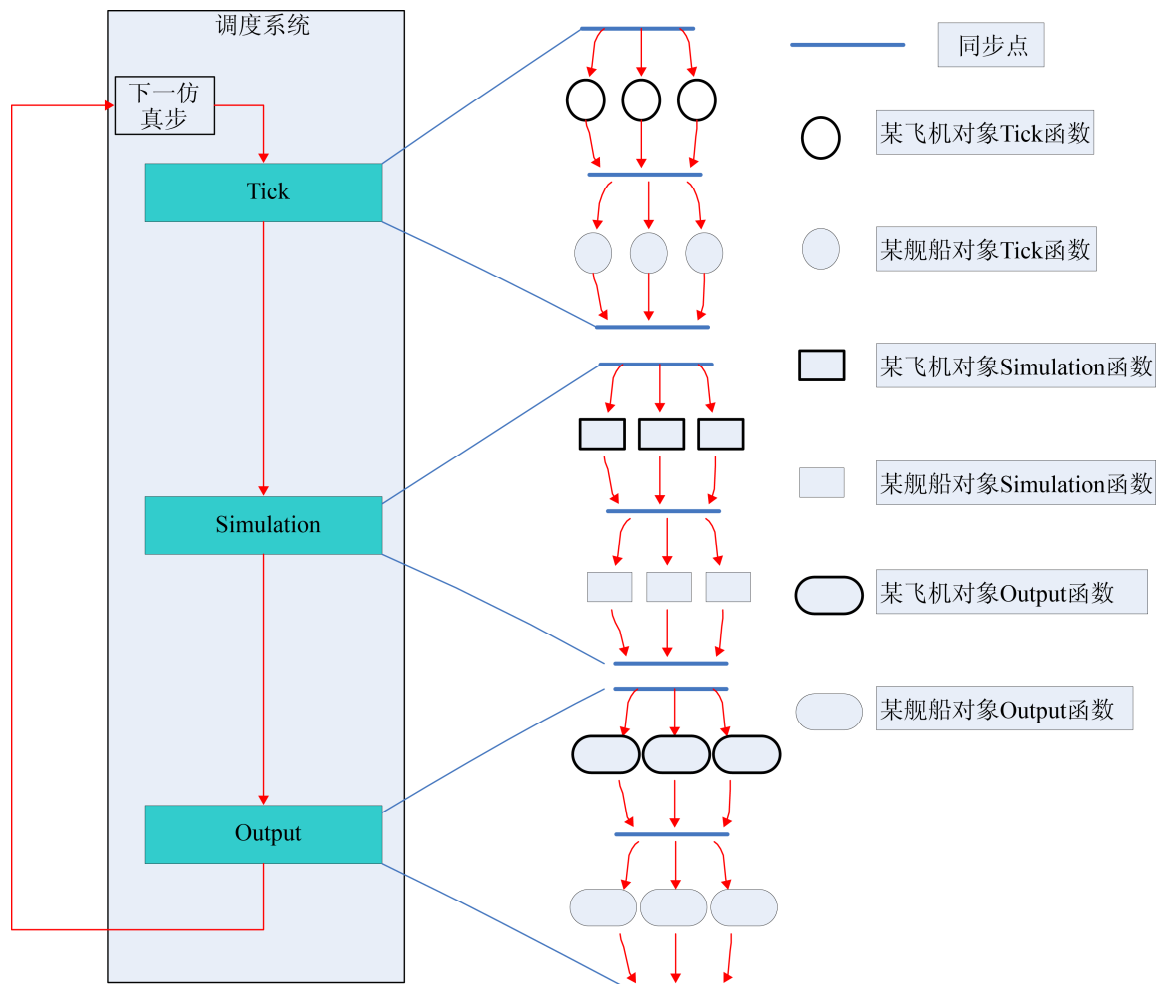


图 3 并行调度策略

Fig. 3 Parallel scheduling strategy

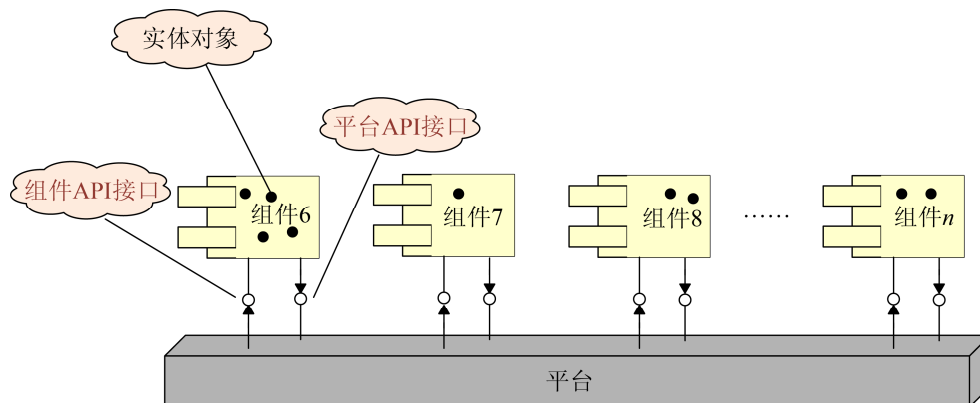


图 4 仿真支撑服务平台与仿真对象

Fig. 4 Simulation support service platform and simulation object

针对分布环境开发的需求, HDOSE 对象化的编程模型隐藏了通信与同步的细节, 实现了通信的透明化。其核心设计是采用了在分布环境下的反射式对象系统, 以树结构对各类仿真对象进行组织管理^[6]。对象化的编程模型和通信透明化相结合, 使得仿真对象可以透明访问远程对象。如图 5 所示, 成员 1 订购了成员 2 的 B 对象, 因此在成员 1 形成一个 B1 对象作为 B 对象的代理, 成员 1 的 C 对象需要访问成员 2 的 B 对象时, 只需访问成员 1 的 B1 对象即可。

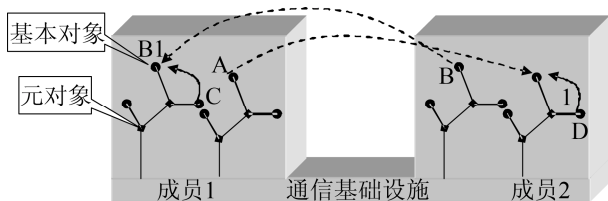


图 5 HDOSE 对象化编程模型

Fig. 5 HDOSE objectification programming model

4.3 参数化建模技术

参数化是 HDOSE 的一个重要设计理念, 其中模型参数化是重点。模型参数化是指模型的计算部分与数据部分相分离, 通过模型解算与数据集的组合, 可以使模型具有不同的表示行为和能力^[17-18]。从复用的角度看, 模型参数化主要实现模型计算部分的复用, 而参数部分则在运行初始时加载。结合具体装备型号来看, 模型参数化意味着“装备型号模型=通用模型+性能参数的定制”, 使得模型具有

较好的通用性、可扩展性和快速修改能力, 适应动态需求下的体系作战效能评估^[14,17-20], 如图 6 所示。

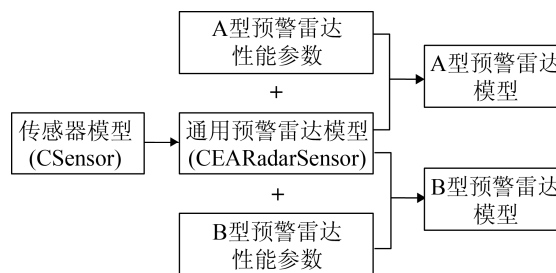


图 6 参数化建模方法

Fig. 6 Parameterized modeling method

除了支持模型的参数化, HDOSE 定义一套的模型和数据描述语言, 实现了模型、系统、场景的一体化描述。该语言在 SRML 基础上, 通过增加对组件的描述和对复杂结构、交互和行为的表达能力, 使其成为一个可扩展的演化式仿真系统描述语言 (eXtensible Evolution Simulation Language, XESL)^[19]。XESL 为对象类、实体类型、对象静态参数、对象初始属性 4 级参数提供统一的表示规范, 极大地方便了仿真系统开发和演化, 也为各类基于 HDOSE 的工具提供了集成规范。

4.4 组件化建模技术

HDOSE 严格区分了组件化建模和组合化建模的概念。组件化建模是指模型以相对独立的物理模块开发、管理、部署和集成。组件是模型的容器, 模型是组件的内容^[18]。组合化建模是指模型在逻辑

上可以组合装配形成具有一定行为和功能的实体。组件化与组合化均是支持复用的手段,但组件化是为了灵活控制物理模块的粒度,而组合化是为了让模型的逻辑功能可以灵活地组合。

HDOSE 的组件被设计为一个容器载体,它装载的内容是若干仿真模型对象类。组件被定义为一个 3 元组,即:

COM: =<name, handle, ClsList>

其中, name 为组件的名称; handle 为组件在系统中的句柄; ClsList 为组件所装载的类模型链表。HDOSE 通过定义一套平台无关的组件规范,利用操作系统的动态库技术,实现仿真系统基于组件的开发和集成。

4.5 组合化建模技术

组合化建模方面, HDOSE 针对体系仿真的需求,在经典 EATI (即实体、动作、任务、交互) 实体建模方法^[17-18]基础上,引入了状态池的概念,并支持实体的多级复用迭代。

如图 7 所示, HDOSE 定义的实体由标识、状态池(兵力实体的静态属性集)、功能模型(传感器、控制器、执行器)、交互模型等模型对象复合装配而成,不同功能的模型组合将赋予实体不同的能力和特征。实体可分为聚合实体(如舰艇编队、航空兵编队等)和平台实体(如飞机、水面舰、导弹等)。

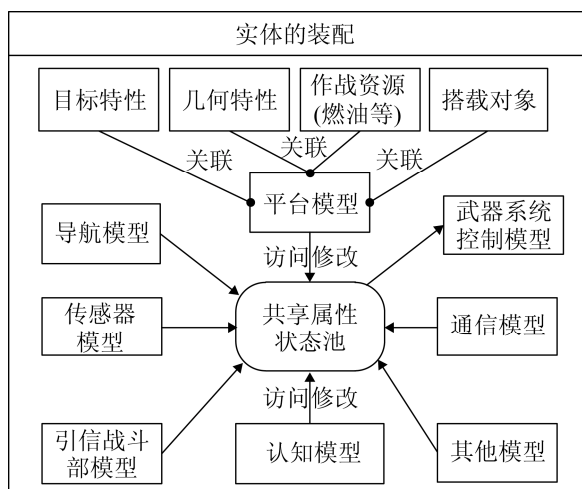


图 7 实体组合与装配

Fig. 7 Entity combination and assembly

状态池是描述实体状态的属性集(如地理坐标、姿态、毁伤程度等),它可被实体内部的仿真模型访问并修改,并通过底层通信服务进行发布以被其他实体所感知。状态池类型在不同的抽象程度上覆盖了控制对象、实体、聚合实体、单个实体、单兵、水面舰艇、车辆、固定翼飞机、旋转翼飞机、水下平台、导弹等类型。

模型是对实体感知、决策、动作等行为进行仿真的功能模块;对于平台实体,大部分模型在实现中映射成为平台所装配的装备;平台的模型体系提供传感器、控制器、执行器、数据链 4 类模型供应用层扩展,分别实现实体的感知模型、决策模型、物理模型和基于信道的通信模型。

5 基于 HDOSE 的应用与测试

5.1 HDOSE 的典型应用

经过多年的实践应用, HDOSE 已成功在诸多仿真应用领域得到应用,比如某型飞机模拟训练系统、某编队指挥模拟训练系统、某空防作战效能评估系统、某型飞机战技指标论证系统、武器装备体系对抗支持平台等^[14,17,23,30-31]。

HDOSE 的典型应用模式有 3 种: ①作为高级的分布交互仿真集成框架,利用其支持流对象、远程对象、对象透明交互等特性实现异构系统互联互通互操作; ②作为计算密集型仿真运行框架,通过构建计算服务器模式的系统结构开展大规模多粒度仿真计算; ③混合前两者的特点,构建分布并行一体化的仿真系统,每个节点均为高性能计算节点并同时进行高速信息交换。

(1) 在某型飞机模拟训练系统^[30]中,以 HDOSE 作为公共总线实现分布的仿真节点集成,每个节点功能相对独立、部署的模型也不相同,实现各节点之间信息互联互通,系统实现方案如图 8 所示。

(2) 在某型飞机装备战技指标论证仿真系统^[17]中,需要对多个战技指标参数进行反复设计、实验、分析与优化,在此过程中要进行大量多次复杂的仿真

计算, 对计算效率要求非常高。基于 HDOSE 的高性能计算服务、友好的编程模型、组件化参数化建模能力等功能, 可以很好地实现上述仿真需求。如图 9 所示, 以 HDOSE 的一个或多个节点作为模型计算服务器(或模型计算服务器集群), 其余节点作为导调节点、控制节点或者可视化节点。在这种模式下, 服务器运行可选取超实时、实时的运行方式。

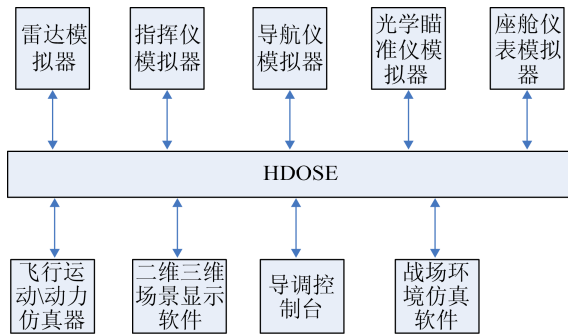


图 8 某型飞机模拟训练系统应用示例

Fig. 8 Application example of a certain type of aircraft simulation training system

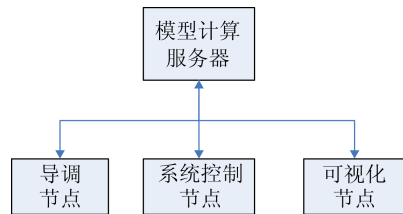


图 9 某装备效能评估仿真系统应用示例

Fig. 9 Application example of an equipment effectiveness evaluation simulation system

5.2 HDOSE 性能测试

本测试案例以固定翼飞机六自由度飞行行为测试算例, 单个飞机对象模型的单步计算复杂度相当于执行 500 次双精度浮点乘法运算。在测试想定中配置 35 架飞机, 即每个仿真步需要同时对这 35 架飞机的飞行模型执行运算, 且每架飞机每个仿真步向网络中产生长度为 140 byte 的数据报文。测试环境分为笔记本和 workstation, 仿真步长为 30 ms, 成员数量分为 1 个仿真成员和 2 个仿真成员, 运行模式分别使用服务器模式、共享内存模式、TCP 模式、pRTI 模式等 4 种模式。详细测试条件与测试结果见表 1, 测试程序运行状态见图 10。

测试结果表明, 在上述测试条件下, 当仿真步长为 30 ms 时, HDOSE 计算的更新频率最高可达 58 000 次/s, 此时加速倍率为 1 740 倍、CPU 利用率为 60% 以上(如果再适当增加飞机数量, CPU 利用率可达到 100%)。同样的测试条件下, 如果将仿真步长设置为 1 ms, 则其加速倍率可达 50 倍左右, 依然具有很强的超实时计算能力, 尤其适合于航空航天领域高精度高采样率的实时/超实时仿真计算。

表 1 HDOSE 性能测试结果

Tab. 1 HDOSE performance test results

运行模式	成员数量	双核笔记本计算机: CPU T730 2.0G		双 CPU 8 核 DELL 工作站: CPU 至强 E5 2.4G	
		更新频率/(次/s)	加速倍率	更新频率/(次/s)	加速倍率
服务器模式	1	34 000	1 020.0	58 000	1 740.0
	2	12 500	375.0	32 500	975.0
共享内存	1	10 500	315.0	28 500	855.0
	2	7 950	238.5	12 950	388.5
TCP 协议	1	6 050	181.5	10 050	301.5
	2	5 700	171.0	6 200	186.0
pRTI	1	750	22.5	790	23.7
	2				

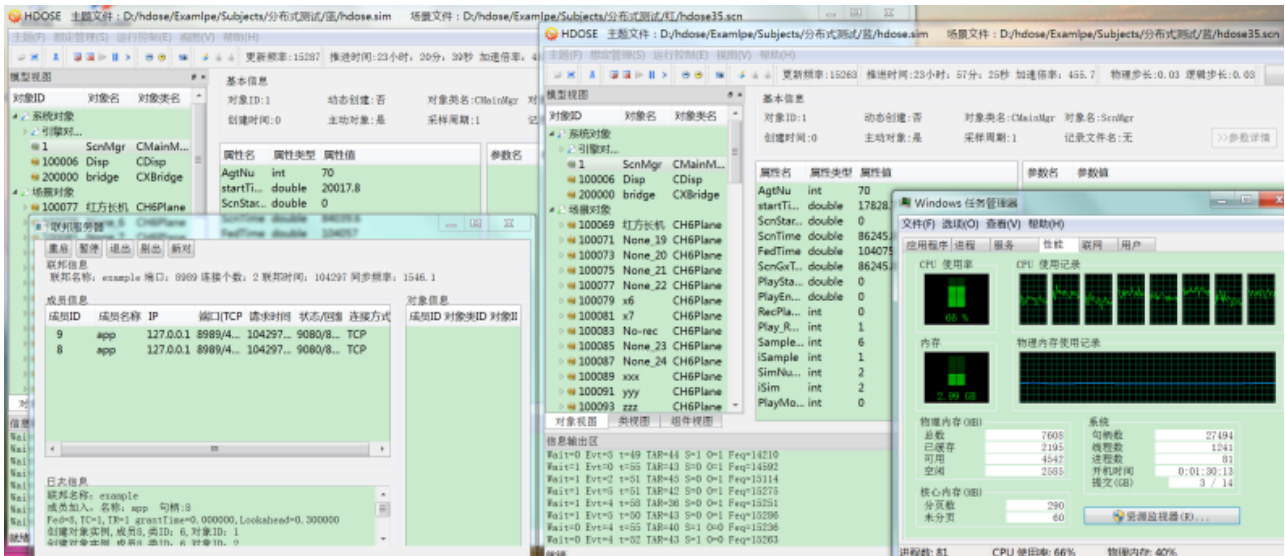


图 10 HDOSE 性能测试程序
Fig. 10 HDOSE performance test program

6 结论

HDOSE 定义了一套分布并行一体化的仿真建模框架和编程规范,为复杂的分布式仿真系统或计算密集型仿真系统提供常用的支撑服务,实现通信透明化、集成组件化、协议多元化、编程对象化、配置参数化、调度自动化、应用定制化,以简化仿真系统开发和集成的难度。HDOSE 友好的编程模型和方便的复用机制帮助用户快速构造仿真系统,并能够基于高效的调度策略和充分利用硬件计算潜能提供高性能计算服务。HDOSE 适用于模型计算和信息交互均密集的复杂应用系统研制,其灵活的架构和强扩展性鼓励二次开发,以便为专用领域定制平台或产品,这些特点可以极大地拓展 HDOSE 的应用范围。HDOSE 可推广用于其他实时紧耦合信息系统研制,作为计算密集的命令系统^[31]、情报传输系统、航空管制系统等公共的基础平台。

参考文献:

- [1] 李伯虎,柴旭东,朱文海,等. 现代建模与仿真技术发展中的几个焦点[J]. 系统仿真学报, 2004, 16(9): 1871-1878.
Li Bohu, Chai Xudong, Zhu Wenhai, et al. Some Focusing Points in Development of Modern Modeling

and Simulation Technology[J]. Journal of System Simulation, 2004, 16(9): 1871-1878.

- [2] 李伯虎,柴旭东,张霖,等. 面向新型人工智能系统的建模与仿真技术初步研究 [J]. 系统仿真学报, 2018, 30(2): 349-362.

Li Bohu, Chai Xudong, Zhanglin, et al. Preliminary Study of Modeling and Simulation Technology Oriented to Neo-type Artificial Intelligent Systems[J]. Journal of System Simulation, 2018, 30(2): 349-362.

- [3] European Space Agency Directorate of Operations and Infrastructure. SMP 2.0 Handbook EGOS-SIM-GEN-TN-0099 Issue 1 Revision 2[Z]. European Space Operations Centre, 2005.

- [4] 李进,钱大豪. 美军大型仿真系统[J]. 国防科技, 2009, 30(1): 87-91.

Li Jin, Qian Dahao. Large-scale Simulation System for the Us Military[J]. National Defense Science & Technology, 2009, 30(1): 87-91.

- [5] Joint Simulation System (JSIMS) Common Component Simulation Engine (CCSE) Software User Manual (SUM) Version 0.5.1 JSIMS-CCSE-SUM-C010 Preliminary [EB/OL] [2020-12-03]. <http://www.jsims.mil/>.

- [6] 李伯虎,柴旭东,李潭. 复杂系统高效能仿真技术研究 [J]. 中国电子科学研究院学报, 2012(3): 221-228, 245.

Li Bohu, Chai Xudong, Li Tan. Research on High-Efficiency Simulation Technology of Complex System[J]. Journal of CAEIT, 2012(3): 221-228, 245.

- [7] 贺东京,宋晓,张霖,等. 基于扩充 Modelica 的复杂系统多层次并行仿真[J]. 北京航空航天大学学报, 2012, 38(9): 1240-1244.

- He Dongjing, Song Xiao, Zhang Lin, et al. Multi-layer Parallel Simulation for Complex System Based on Extended Modelica[J]. Journal of Beijing University of Aeronautics and Astronautics, 2012, 38(9): 1240-1244.
- [8] 乔海泉, 王学慧, 李革, 等. 基于模板的并行离散事件仿真持久框架[J]. 系统仿真学报, 2007, 19(3): 563-566, 574.
- Qiao Haiquan, Wang Xuehui, Li Ge, et al. Template-based Persistence Framework for Parallel Discrete Event Simulations[J]. Journal of System Simulation, 2007, 19(3): 563-566, 574.
- [9] 乔海泉, 张耀程, 李革, 等. JMASE 分布式仿真功能的设计与实现[J]. 计算机仿真, 2006, 23(10): 113-118.
- Qiao Haiquan, Zhang Yaoheng, Li Ge, et al. Design and Implimentation of JMASE Distributed Simulation Capability[J]. Computer Simulation, 2006, 23(10): 113-118.
- [10] 苏年乐, 周鸿伟, 李群, 等. SMP2 仿真引擎的多核并行化[J]. 宇航学报, 2010, 31(7): 1883-1891.
- Su Nianle, Zhou Hongwei, Li Qun, et al. Parallelization of SMP2 Simulation Engine on Multi2Core Platform[J]. Journal of Astronautics, 2010, 31(7): 1883-1891.
- [11] 李群, 王超, 王维平, 等. SMP2.0 仿真引擎的设计与实现[J]. 系统仿真学报, 2008, 20(24): 6622-6626.
- Li Qun, Wang Chao, Wang Weiping, et al. Design and Implementation on Simulation Engine Compliant with SMP2.0[J]. Journal of System Simulation, 2008, 20(24): 6622-6626.
- [12] 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.
- He Qiang, Chen Bin, Zhong Ronghua, et al. Research of BOM Component and Simulation Engine Based on DEVS[J]. Journal of System Simulation, 2010, 22(11): 2505-2510.
- [13] The Synchronous Parallel Environment for Emulation and Discrete-Event Simulation. User's Guide[EB/OL]. (2003-04-30)[2021-02-19]. <http://www.speedes.com/documentation.html>.
- [14] 黄晓冬. 基于反射的适应性软件开发方法及其在分布交互仿真中的应用[D]. 烟台: 海军航空工程学院, 2005.
- Huang Xiaodong. Reflective Software Development Approaches for Adaptive Software with Applications in Distributed Interactive Simulation[D]. Yantai: Naval Aeronautical Engineering Institute, 2005.
- [15] 黄晓冬, 李伯虎, 柴旭东, 等. 基于反射的分布交互仿真软件框架[J]. 北京航空航天大学学报, 2007, 27(4): 994-999.
- Huang Xiaodong, Li Bohu, Chai Xudong, et al. Reflection Based Distributed Interactive Simulation Framework[J]. Journal of Beijing University of Aeronautics and Astronautics, 2007, 27(4): 994-999.
- [16] 黄晓冬, 何友, 徐俊艳, 等. 一种基于 HLA 的弹性软件框架及其应用[J]. 系统仿真学报, 2005, 17(1): 95-99.
- Huang Xiaodong, He You, Xu Junyan, et al. An HLA-based Flexible Framework with Applications[J]. Journal of System Simulation, 2005, 17(1): 95-99.
- [17] 凌绪强. 多领域协同建模仿真支撑平台研究[D]. 烟台: 海军航空工程学院, 2012.
- Ling Xuqiang. Research on Collaborative Modeling and Simulation Platform for Multiple Application Fields[D]. Yantai: Naval Aeronautical Engineering Institute, 2012.
- [18] 温玮, 何友, 李牧. 基于参数数据库的 CGF 实体框架结构研究[J]. 系统仿真学报, 2007, 19(6): 1231-1237.
- Wen Wei, He You, Li Mu. Research of CGF Entity Framework Based on Parameters Database[J]. Journal of System Simulation, 2007, 19(6): 1231-1237.
- [19] 黄晓冬, 凌绪强, 温玮, 等. 基于 SRML 的仿真语言研究及应用[J]. 计算机仿真, 2013, 30(3): 285-289.
- Huang Xiaodong, Ling Xuqiang, Wen Wei, et al. Research on SRML-Based Simulation Language and Its Application[J]. Computer Simulation, 2013, 30(3): 285-289.
- [20] 温玮, 黄晓冬, 杨瑞平, 等. 计算机生成兵力仿真引擎 HYCGF 的研究[J]. 火力与指挥控制, 2014, 39(7): 128-132.
- Wen Wei, Huang Xiaodong, Yang Ruiping, et al. Research on Simulation Engine for Computer Generated Forces HYCGF[J]. Fire Control & Command Control, 2014, 39(7): 128-132.
- [21] Fujimoto R. M. Parallel and Distributed Simulation Systems[M]. New York: John Wiley & Sons, Inc, 2000.
- [22] Pedretti K, Kelly S, Levenhagen M. Summary of Multi-Core Hardware and Programming Model Investigations [R]//Technical Report No: SAND2008-3205. Albuquerque, New Mexico, USA: Sandia National Laboratories, 2008: 15-30.
- [23] 柳林, 李涛, 荆涛. 基于多节点多核的样本级并行仿真技术研究[J]. 系统仿真学报, 2010, 22(11): 2597-2599.
- Liu Lin, Li Tao, Jing Tao. Research on Sample-level Parallel Simulation Based on Multi-node Multi-core Environment[J]. Journal of System Simulation, 2010, 22(11): 2597-2599.

- [24] Xu J W, Chen M Y, Zheng G, et al. SimK: A Large-scale Parallel Simulation Engine[J]. Journal of Computer Science and Technology(S1000-9000), 2009, 24(6): 1048-1060.
- [25] 苏年乐, 吕建军, 王维平, 等. 插件式多核并行仿真引擎[J]. 系统仿真学报, 2011, 23(12): 2597-2602, 2622.
Su Nianle, Lü Jianjun, Wang Weiping, et al. Plug-in Architecture for Parallel Simulation Engine on Multi-core Platform[J]. Journal of System Simulation, 2011, 23(12): 2597-2602, 2622.
- [26] Sun Microsystems, Inc. OpenMP API User Guide Book[Z/OL][2021-02-19]. www.sun.com.
- [27] 钱立兵, 陈波, 晏涛, 等. 多线程并发访问无锁队列的算法研究[J]. 先进技术研究通报, 2009, 3(8): 50-55.
Qian Libing, Chen Bo, Yan Tao, et al. Research on Algorithm of Multi-thread Concurrent Access to Lock-free Queue[J]. Bulletin of Advanced Technology Research, 2009, 3(8): 50-55.
- [28] 陈莉丽, 姚益平, 蔡潞. 多核环境下负载均衡的并行离散事件全局调度机制[J]. 国防科技大学学报, 2012, 34(4): 108-113.
Chen Lili, Yao Yiping, Cai Lu. A Global Schedule Mechanism for PDES on Multi-core Environments[J]. Journal of National University of Defense Technology, 2012, 34(4): 108-113.
- [29] 王学慧, 张磊. 并行与分布式仿真时间推进同步机制综述[J]. 计算机仿真, 2010, 27(2): 126-129.
Wang Xuehui, Zhang Lei. The Summary of the Time Synchronization Mechanism Synchronization in Parallel and Distributed Simulation[J]. Computer Simulation, 2010, 27(2): 126-129.
- [30] 黄晓冬, 何友, 欧阳文, 等. 一种基于 HLA 的飞机训练模拟器的研制[J]. 系统仿真学报, 2005, 17(4): 897-900.
Huang Xiaodong, He You, OuYang wen, et al. Research and Implementation of HLA Based Training Aircraft Simulator[J]. Journal of System Simulation, 2005, 17(4): 897-900.
- [31] 李皓, 倪忠德, 李小军, 等. 基于仿真的编队作战指挥系统设计方法及环境研究[J]. 系统仿真学报, 2011, 23(增 1): 228-231.
Li Hao, Ni Zhongde, Li Xiaojun, et al. Research on Designing Method and Supporting Environment of Formation Combat Command System Based on Simulation[J]. Journal of System Simulation, 2011, 23(S1): 228-231.