文章编号:1001-2486(2011)04-0163-05

社会计算实验动态仿真引擎的研究与实现*

周 云[1],乔海泉[2],邱晓刚[1],黄柯棣[1],胡德文[1]
(1. 国防科技大学 机电工程与自动化学院,湖南 长沙 410073; 2. 北京清河大楼子6,北京 100085)

摘 要:应对新形势下非常规突发事件的严峻挑战是各国政府亟待解决的问题,综合人工社会-计算实验-平行执行的ACP方法,是解决非常规突发事件应急管理问题的有效方法。本文引入ACP方法,阐述构建高性能社会计算实验动态仿真引擎的作用;按照层次化模块化思想设计动态仿真引擎D-PARSE的体系结构,详细描述各组成部分的具体功能,并基于并行仿真引擎 μsik 实现所设计的动态仿真引擎D-PARSE;测试了不同事件发送策略对仿真系统性能的影响。

关键词:动态仿真引擎;通信库;微内核;微进程
中图分类号:TP39119 **文献标识码**:A

Research and Implement on Dynamic Simulation Engine for Society Computing Experiments

ZHOU Yun[1], *QIAO Hai-quan*[2], *QIU Xiao-gang*[1], *HUANG Ke-di*[1], *HU De-wen*[1]
(1. College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China;
2. Beijing Qinghe-Building Zi 6, Beijing 100085, China)

Abstract: Responding to the serious challenge of unconventional emergencies under the new situation is the urgent problem of governments. Integrating Artificial Societies—Computational Experiments—Parallel Execution, ACP is an effective method to solve the unconventional emergency management. Firstly, the ACP method was introduced, which was used as the dynamic simulation engine for constructing high performance society computing experiments. Then, the architecture of the dynamic simulation engine D-PARSE was designed following the concept of hierarchy and modularity. The specific function of each component was described in detail. D-PARSE was implemented based on the (sik parellel simulation engine. Finally, the influence of different event-send-strategy on simulation system performance was tested.

Key words: dynamic simulation engine; communication library; micro kernel; micro process

当今社会,人口与财富向城市及特定区域高度集中,发达的交通网络使世界变成了地球村,万维社会媒体的出现及其广泛应用急剧放大了人为因素影响的不确定性。这些变化和趋势正在使突发事件逐渐向非常规化、复杂化、网络化、社会化方向发展,各国政府面临着应对新形势下非常规突发事件的严峻挑战。从2001年美国"9·11"事件开始,以美国为首的西方国家开始对非常规突发事件应急管理研究进行大规模倾斜性资助,从2003年我国"SARS流行"事件开始,我国全面展开了对危机事件应急管理的研究[1]。

随着网络信息化的跨越式发展,以万维社会媒体为代表的新兴媒体正在发挥越来越强的主导作用,非常规突发事件的社会化倾向日趋明显[2-3]。传统的应急管理模式难以应对非常规突发事件的分析、管理和控制等方面的挑战,更无法满足基于非常规突发事件情景的应急演练、心理培训、预案评估、自适应情景模拟等更高层次的需求[4]。利用计算机进行模拟仿真的方法是解决非常规突发事件应急管理问题的最重要途径。

1 ACP方法的引入

非常规突发事件发生的时间和地点难以预测,传播和转换迅速,涉及面广,社会化倾向程度加剧。这些问题对建模与仿真理论和方法的研究以及相应的建模与仿真平台的开发提出了新的挑战,需要引入针对复杂巨系统研究的人工社会(Artificial Societies)-计算实验(Computational

* 收稿日期:2010-12-31
基金项目:国家自然科学基金资助项目(61074108,91024030/G30)
作者简介:周云(1965—),女,副教授,博士。

Experiments) – 平行执行(Parallel Execution)方法,即 ACP 方法的综合解决方案。

人工社会系统最先由美国兰德公司在 20 世纪 90 年代初提出,其核心思想是在计算机上构建一个与现实社会并行的封闭式的人工社会,以此来研究信息技术对社会政治文化的影响[5-6]。ACP 方法中的人工社会系指基于建模与仿真建立的、与真实系统对应的虚拟世界。根据这一方案,利用人工社会的思想和复杂系统理论,构建与实际社会系统平行运行的人工社会,通过人工社会与实际社会系统的平行执行,采用计算实验的方法在人工社会中进行实验,从而提出定性和定量的决策依据,更好地对实际系统进行管理与控制[1]。

人工社会系统方法主要基于对底层个体的规定,利用计算机和智能体技术,实现自下而上的主动培育。它不以逼近现实社会系统为唯一目的,追求的是逻辑自洽,更多地关注实验过程中的涌现情景。

人工社会中参与的智能体数量多,相互之间的关系复杂,实验运行的计算量与数据通信量巨大,同时需要多次运行才能满足评估决策的要求。为了能够超实时获得结果,仿真系统对计算速度提出了很高的要求,需要研究分布与并行相结合的运行支撑技术,提供多种运行支撑软件,充分发挥分布技术和并行技术各自的优点。为此我们研究并实现了提高需多次重复运行的多智能体仿真整体执行效率的动态仿真引擎。

2 动态仿真引擎的体系结构设计

我们按照层次化模块化思想设计动态仿真引擎 D-PARSE,如图 1 所示,自下而上依次为通信层、核心层、RTI 服务层和功能扩展层[7]。

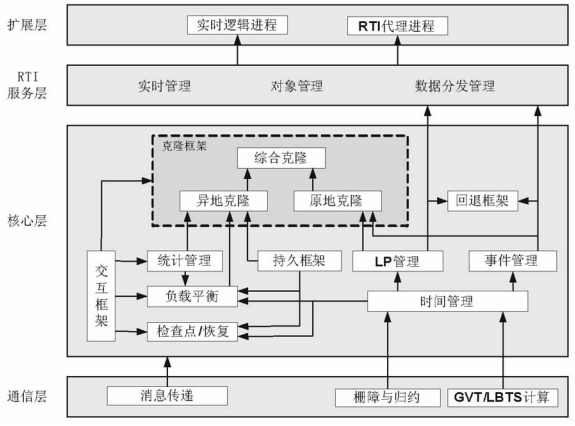

图 1 动态仿真引擎的逻辑结构

Fig.1 Logical architecture of the dynamic simulation engine

2.1 通信层

处理器间的通信是并行仿真的关键之一,通信性能直接影响仿真系统的性能。通信层主要包括基本消息传递、栅障与归约、GVT/LBTS 计算功能。D-PARSE 能够根据节点间互联设备的不同自动选择通信手段,在不同通信协议之上提供用户透明的消息传递接口。

2.2 核心层

核心层包括逻辑进程(Logical Process,LP)管理、事件管理、时间管理、回退框架、统计管理、检查点/恢复、负载平衡、持久框架、克隆框架、交互框架等功能。

• LP 管理:LP 是并行仿真引擎中最重要的数据结构之一,主要由状态变量、事件队列、本地时钟变量三部分组成。LP 管理主要包括创建 LP 和 LP 队列的排序、插入、删除等操作。

• 事件管理:事件管理主要包括产生、传递、缓冲、处理、暂存和提交事件。

• 时间管理:时间管理功能负责在事件处理过程中不断更新本地时间和全局仿真时间。

• 回退框架:发生因果关系错误时,利用回退机制加以纠正。

• 统计管理:统计模块收集的信息包括处理的事件数、反消息数、每个事件的处理时间、内存使用情况等。

• 检查点/恢复:系统发生故障后,将相关进程恢复至发生故障前的最近检查点,从该检查点处继续执行,避免重新运行整个任务。

• 负载平衡:通过提高处理器利用率或减小通信量达到提高系统总体性能的目的。

• 持久框架:持久化在仿真中用于检查点与恢复操作、负载平衡和仿真克隆中的仿真对象迁移等。

• 克隆框架:通过共享相同计算的方式,仿真克隆技术能够提高需多次重复执行的仿真的执行效率。

• 交互框架:提供用户与系统交流的途径。

2.3 RTI 服务层

在 D-PARSE 中增加 RTI 服务层,其接口与 HLA/RTI 接口兼容,使 D-PARSE 对外既提供并行仿真接口,又提供 RTI 接口,增强并行仿真中的模型重用与互操作性,可以满足各种应用的需要。RTI 服务层提供的服务有声明管理、对象管理和数据分发管理。

2.4　功能扩展层

功能扩展层通过实时逻辑进程为用户提供实时仿真功能。实时逻辑进程既可以处理实时事件,也可以处理非实时事件,有效地支持了动态仿真系统中仿真的多模式运行。功能扩展层通过 RTI 代理进程提供与外部 HLA 联邦的互连。

3　动态仿真引擎的原型实现

μsik(micro simulation kernel) 并行仿真引擎是 Georgia 大学 Perumalla 开发的开源并行与分布仿真微内核仿真引擎,它的设计借鉴了操作系统中的微内核思想,将并行仿真引擎中的功能划分成不同层次的服务,最基本的服务放在核心,其他的服务放在外围,外层可以调用内层的服务。Perumalla 基于微内核思想设计了仿真内核 μsik,实现了基本的时间管理和进程调度等功能。但他没有论述并行仿真引擎中应该有哪些服务以及各个服务之间的关系[8-9]。供下载的 μsik 版本实现的功能较少,我们在 μsik 的基础上增加了交互框架、多种仿真克隆方法、持久框架、在分布节点上仿真、多种事件发送策略、回退框架、实时仿真、与 HLA 联邦互连、RTI 服务接口等功能,实现了所设计的动态仿真引擎 D-PARSE。

3.1　通信功能

通信是仿真引擎中最难实现的部分之一,μsik 利用 libsynk 通信库支持多种通信设备以及混合平台仿真[10]。libsynk 通信库采用了模块化的设计思想,主要功能模块包括:时间管理模块 TM、消息传递模块 FM、归约模块 RM、栅障同步模块 RM BAR。D-PARSE 采用 μsik 的 libsynk 通信库实现系统的通信功能。

3.2　基本数据结构

D-PARSE 中,微内核管理着多个仿真进程,而仿真进程又管理着多个事件。D-PARSE 的数据结构可分为三个基本部分:微内核类、微进程类和事件类,其层次结构如图 2 所示。

(1)微内核

D-PARSE 中,微内核只提供三种最基本的服务:命名服务、路由服务和调度服务。微内核利用命名服务的仿真进程标识符映射表统一定位和引用仿真进程;微内核以统一的方式为仿真进程提供路由服务,确保事件透明转发给接收进程。微内核利用调度服务以最有利于仿真进程推进的方式在多个仿真进程间分配 CPU 时间,确保不出现死锁。

(2)微进程

如图 2 所示,D-PARSE 中,进程分为两类:内核进程和仿真进程。内核进程分为两类,一类是远程通信代理进程 RemoteFederateProcess,实现远程事件的发送与接收;另一类是反射进程 ReflectorProcess,实现仿真进程间的间接通信和组播通信以及 RTI 服务层的声明管理与对象管理。仿真进程 SimProcess 就是通常所说的逻辑进程。类 Phisycal _ lp _ struct 是对类 MicroProcess 的包装,通过类 physical _ lp _ struct 中的 parent/child/left/right 指针,指明逻辑进程类 MicroProcess 之间的克隆关系。仿真克隆类 SimClone 是实现递增克隆的主要场所。类 CloneRelation 记录每个仿真之间的克隆关系。

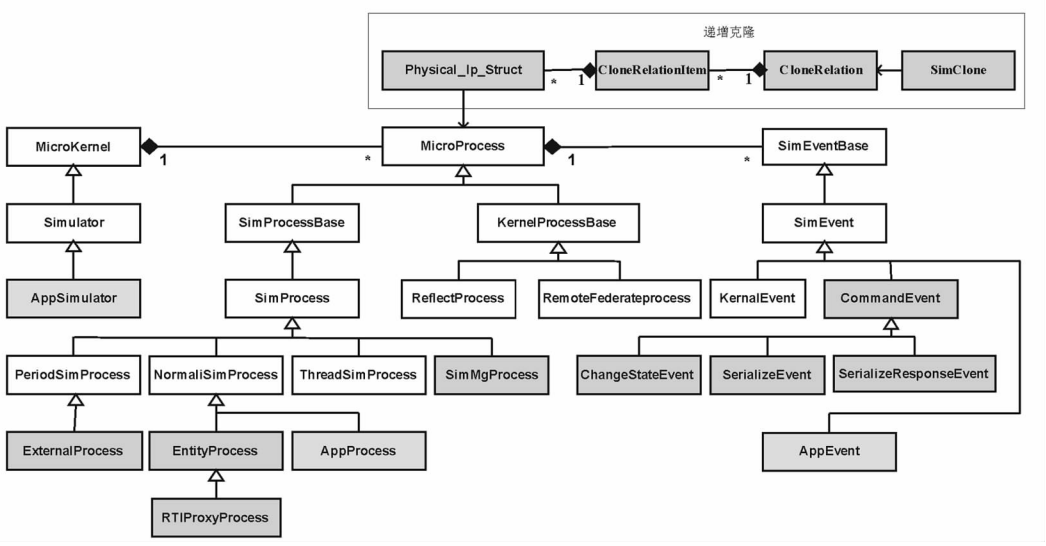

图 2　D-PARSE 的数据结构

Fig.2　Data structure of D-PARSE

(3)事件

每个仿真进程保存自己的事件。普通仿真进程中有两个事件列表:待处理事件队列(Future Event List,FEL)和已处理事件队列(Processed Event List,PEL)。FEL 中按时戳顺序保存了当前尚未处理的事件,PEL 中按时戳顺序保存了已处理但还没有提交的事件。

3.3 进程调度

D-PARSE 实现了进程的保守调度与乐观调度的混合调度。从进程调度的角度,D-PARSE 将仿真进程事件的生命周期划分为四个阶段,使得仿真中任一时刻,所有事件都可纳入以下四个不同阶段:

- 可发送阶段:由仿真进程处理当前可提交事件和当前可处理事件时可能调度给其他仿真进程的事件组成。
- 可处理阶段:已被仿真进程接收且正等待处理。
- 可提交阶段:已经处理但正在等待提交。
- 已提交阶段:已经处理和提交且已经释放内存。

每个阶段都有一个重要的时间量,它们分别是事件的最早可发送时间 EETS(earliest emittable timestamp)、事件的最早可处理时间 EPTS(earliest processable timestamp)、事件的最早可提交时间 ECTS(earliest committable timestamp)和事件的最晚提交时间 LCTS(latest committed timestamp)。EETS 是可发送事件队列中事件的最小时戳。EETS 表明一个仿真进程向其他仿真进程发送事件时间的下限(发给自身的不计算在内),用于计算 LBTS。EPTS 用于调度一个乐观的仿真进程。对于乐观进程,ECTS 是 PEL 中事件的最小时戳;对于保守进程,每次处理完一个事件都可以立即提交,ECTS 等于 FEL 中事件的最小时戳。LCTS 是提交的最后一个事件的时间。D-PARSE 采用如图 3 所示算法确定仿真进程事件各阶段的重要时间量。

进程调度过程如图 4 所示,首先按照 ECTS 值由小到大的顺序调度拥有可提交事件的仿真进程,即 ECTS 值小于 LBTS 的进程,采用保守方式推进时间。没有 ECTS 值小于 LBTS 的进程时,说明所有仿真进程中的事件都是不安全的,这时开始异步计算 LBTS。LBTS 计算非常耗时,保守进程在这段时间内必须等待,微内核就利用这段时间对乐观进程进行乐观时间推进。采用次小 EPTS 值限制乐观程度[11]。

```
输入:仿真进程 i,FEL_i,PEL_i,Lookahead(前瞻量);
输出:ECTS_i,EPTS_i,EETS_i;
算法:
    if FEL_i = ∅  then  FEL_i^top ← ∞
    else FEL_i^top ← Min(FEL_i 中的事件时戳)
    if 仿真进程 i 为保守仿真进程  then PEL_i^top ← ∞
    else
        if PEL_i = ∅  then  PEL_i^top ← ∞
        else PEL_i^top ← Min(PEL_i 中的事件时戳)
    ECTS_i = Min(FEL_i^top, PEL^top i)
    if 仿真进程 i 为保守仿真进程  then  EPTS_i ← ∞
    else EPTS_i = FEL_i^top
    EETS_i = Min(FEL_i^top + Lookahead, PEL_i^top)
```

图 3 仿真进程事件时间的确定算法

Fig.3 Determining algorithm on event time of simulation process

```
取 ECTS_i 最小的进程 cpb
if(cpb〉ECTS() < LBTS){
        //如果 cpb 的最早提交时间小于 LBTS
    advance _ process(cpb,LBTS);
        //cpb 的仿真时间推进到 LBTS
}
else{
    接收事件,计算 LBTS
    取 EPTS 最小的进程 ppb
    advance _ opt(ppb,ECTS _ min2);
        //ppb 乐观推进到时间 ECTS _ min2)
        //ECTS _ min2 是 EPTS 次小的进程的 EPTS 值
}
```

图 4 仿真进程的调度算法

Fig.4 Shedule algorithm of simulation process

4 测试与结果

μsik 仅实现了乐观发送的事件发送策略,即:不管事件的目标在本地,还是在异地,都立即发送。由于取消异地事件的代价很大,适当控制发送到异地的事件,对仿真性能有一定的提高。通过修改代理进程,我们实现了如下两种异地事件的发送策略:

- 保守发送:本地事件可立即发送,只有安全时才发送异地事件。
- 受限乐观发送:不存在安全事件时,可以发送不大于 N 个的不安全事件。

下面以 PHOLD 为例,测试不同事件发送策略对仿真系统性能的影响。

(1)测试方法:PHOLD 中每个联邦成员设置 NP 个仿真进程,初始化时,每个仿真进程向随机选择的目标发送 M 个事件,每个事件的接收时间也是随机产生的。在运行过程中,仿真进程每收到一个事件,就再随机选择一个目标进程发送一个随机时间的事件。整个仿真运行过程中,事件总数不变,为 NP × M 个。测试中设置 NP = 10,M = 100,仿真进程使用乐观时间推进机制,仿真时间为 100s。测试中使用的两个联邦成员参数设置相同。

(2)硬件环境:100M 以太网互联的局域网内的两台单处理器个人电脑,CPU 为 AMD Athlon XP 3000 + 2.17GHz,内存 256M。

(3)测试指标:联邦成员发送的反消息的个数;事件吞吐率。

测试每个项目时,程序独立运行 10 次,每次的反消息个数如图 5 所示。保守发送时没有反消息,因此,图中只有 5 条曲线,分别是无限制和 N = 1,2,4,10 时的反消息个数。事件吞吐率取 10 次的平均值,如图 6 所示。

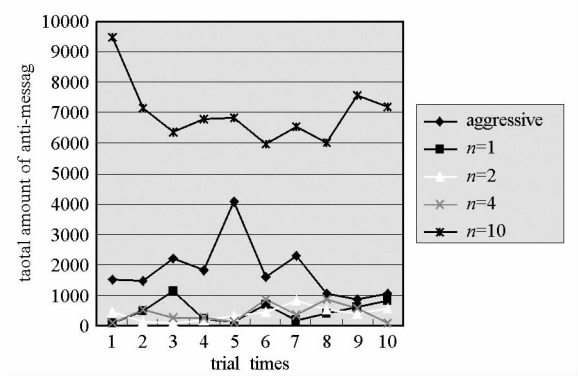

图 5　反消息数目

Fig.5　Total amount of anti-message

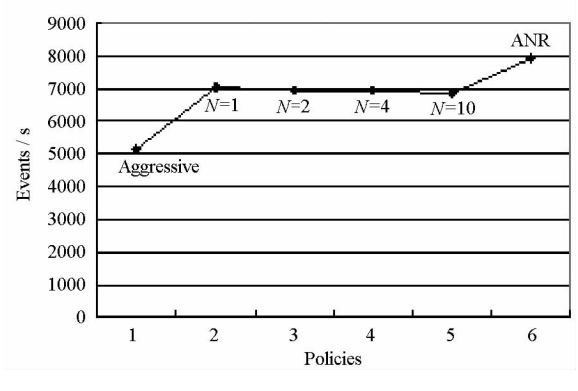

图 6　事件吞吐率

Fig.6　Throughput of events

从图 5 看出,限制因子为 1、2 和 4 时的反消息个数比乐观发送时有明显减少,但 N 取 1、2 和 4 对限制反消息个数的影响不很明显。

图 6 是远程事件发送策略分别采用乐观、限制乐观 N = 1,2,4,10、保守方法时,事件的吞吐率。可以看出当采用乐观策略时性能最低,保守策略时性能最高,而采用限制乐观的策略时性能在二者之间,且随限制因子的增大,性能有下降的趋势,这也是合乎逻辑的。

从这两组数据可以看出,系统性能并不决定于反消息个数,而是受多种因素的影响,也许对于其他仿真应用和其他运行环境,性能表现会大不相同。但这都不重要,重要的是增加了事件发送策略,使用户化系统性能有多种可供选择的方案。

5　结 论

信息化使得突发事件的转化与演变加速、不确定性加强和社会化程度加剧,因此,要求平台具有很强的在线决策支持能力,能够适应万维社会媒体给非常规突发事件应急管理带来的挑战。本文设计并实现的具备交互功能并支持仿真克隆技术的动态仿真引擎 D-PARSE 能够有效提高需多次重复执行的多方案分析仿真的运行速度,支持社会计算实验中的超实时仿真。下一步的工作主要是将 D-PARSE 运用于具体应用。

参 考 文 献:

[1] 邱晓刚. 基于平行应急管理的非常规突发事件动态仿真与计算实验集成升华平台[R]. 长沙:国防科技大学,2010.

[2] 王飞跃. 万维社会媒体在防灾应急中的作用[J]. 科技导报, 2008, 26(10):30 – 31.

[3] 王飞跃. 加强信息技术在应急管理中的作用[N]. 科学时报, 2008.

[4] 范维澄. 国家突发公共事件应急管理中科学问题的思考和建议. 中国科学基金[J], 2007, 21(2):71 – 76.

[5] Epstein R A. Growing Artificial Societies: Social Science from the Bottom Up[Z]. Mit Press, 1996.

[6] Epstein R A. Artificial Societies and Generative Social Science[R]. Artificial Life and Robotics, 1997, 1(1): 33 – 34.

[7] 周云. 面向实时作战决策支持的动态数据驱动仿真理论和方法研究[D].长沙:国防科技大学, 2010.

[8] Perumalla K S. μsik-A Micro-kernel for Parallel/Distributed Simulation Systems[C]//Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation, 2005.

[9] Perumalla K M. μsik-A Micro-kernel for Parallel/Distributed Simulation[R]. Georgia Tech Technical Report GIT-CERCS-TR-04-20, http://www.cercs.gatech.edu/tech-reports/tr2004/git-cercs-04-20.pdf./2010 – 09.

[10] Perumalla K S. Libsynk Website[EB].http://www-static.cc.gatech.edu/people/home/kalyan/libsynk.htm./2006 – 04.

[11] 乔海泉, 鞠儒生, 张锰,等. 并行/分布式仿真微内核 μsik 的研究与改进[J]. 系统仿真学报, 2006,8(18):2:214 – 217.