文章编号: 1001- 2486(2008) 05- 0108- 06

支持多剧情并发执行的仿真克隆中间件机制研究*

周 云[1], 韩守鹏[2], 刘 焱[3], 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. 海军装备研究院, 北京 100073; 3. 山东济宁电业局, 山东 济宁 272000)

摘 要: 仿真克隆可以提高分布式离散事件仿真的效率和并行性, 方便快捷地对仿真中的多种可能性进行分析、比较和评估。介绍仿真克隆的相关概念, 分析 HLA 环境下封装式仿真克隆中间件的不足, 提出并设计一种支持多剧情并发的拦截式 HLA 仿真克隆中间件, 讨论基于该拦截式仿真克隆中间件的 HLA 仿真成员体系结构, 并对联邦成员的时间延迟性能进行测试, 测试结果表明该松耦合结构使联邦成员产生的时间延迟是有限且基本稳定的, 基于拦截式仿真克隆中间件方法是可行的。

关键词: 多剧情并发; 仿真克隆; 中间件; 本地 RTI 组件
中图分类号: TP391. 9 文献标识码: B

Mechanism on Simulation Cloning Middleware of Performing Concurrent Multiple Scenarios

ZHOU Yun[1], HAN Shou-peng[2], LIU Yan[3], HUANG Ke-di[1], HU De-wen[1]
(1. College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China;
2. Naval Academy of Armament, Beijing 100073, China; 3. Jining Electric Coporation, Jining 272000, China)

Abstract: The efficiency and the parallelism of discrete event simulation can be improved through simulation cloning. Analysis, comparison and evaluation of what-if can be processed quickly and conveniently. Based on detailed introduction of simulation cloning and the analysis of the deficiency on encapsuled simulation cloning middleware, an articulated simulation cloning middleware of performing concurrent multiple scenarios is proposed and designed. The architecture of federate based on articulated simulation cloning middleware is discussed. Time delay of the federate based on articulated simulation cloning middleware is measured in experiment. The results demonstrate that the time delay caused by the decoupled federate architecture is limited and stabilized, indicating the feasibility of the method proposed.

Key words: concurrent multiple scenarios; simulation cloning; middleware; local RTI component

在传统的仿真中, 每次仿真运行的输出仅对应一个结果集, 当需要对仿真中的多种可能性(what-if)进行分析、比较和评估以做出决策时, 就需要针对不同的剧情, 使用不同的决策规则和参数, 多次重复运行仿真。在大规模分布式仿真中, 仿真模型复杂且数目众多, 反复从头重新配置和执行仿真, 资源与时间花费巨大。从效率方面来说, 这种重复运行的方式对于在线仿真就更不可取。如何进行多剧情并发仿真, 对提高仿真效率具有重要意义。仿真克隆思想的提出[1-3], 为解决该问题找到了一条可行的途径。

1 问题的提出

1.1 仿真克隆及其相关概念

仿真克隆是美国 Georgia 理工大学的 Hybinette 和 Fujimoto 最早提出的[1], 目的在于提高分布式离散事件仿真的效率和并行性, 方便快捷地对仿真中的多种可能性进行分析、比较和评估。如图 1(b) 所示, 仿真克隆的基本原理如下: 允许多个剧情在同一仿真执行中并发运行, 每个剧情对应一条特定的路径。

* 收稿日期: 2008- 03- 21
基金项目: 国家自然科学基金资助项目(60704038)
作者简介: 周云(1965—), 女, 副教授, 在职博士生。

当系统处于多个不同执行策略的决策点时,仿真克隆支持从选定的仿真剧情克隆出新的剧情副本并插入当前的仿真运行之中执行。决策点是一条仿真执行路径开始分叉的时间点,而分叉是被探索参数产生不同取值的时刻。

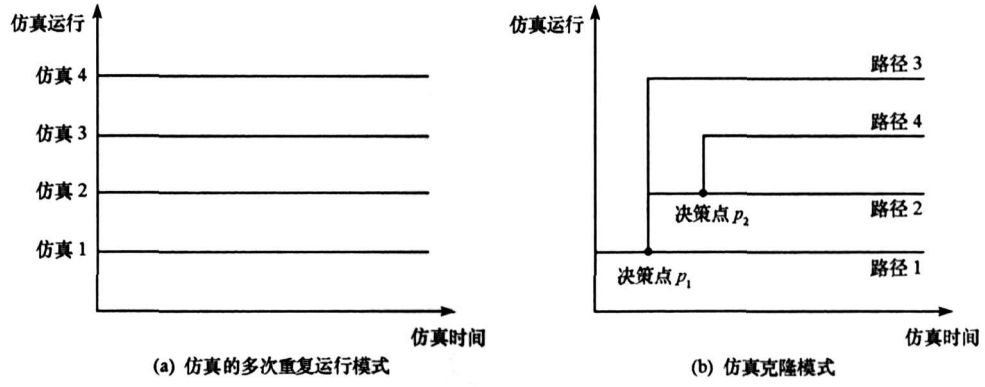

图 1　两种仿真运行方式的比较

Fig. 1　Comparison on the two executions of simulation

结合图2,下面介绍仿真克隆的几个相关概念。

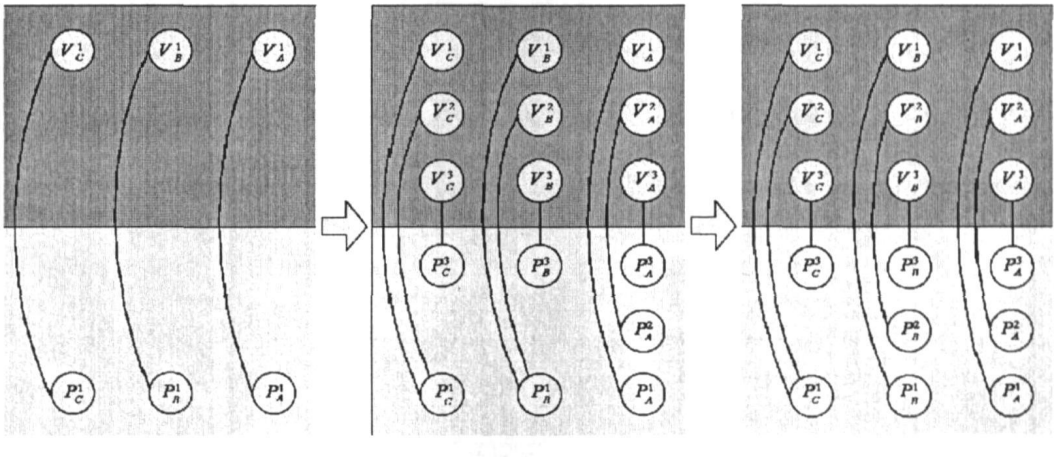

图 2　仿真克隆过程示意

Fig. 2　Process of simulation cloning

物理逻辑进程(P): 一个具有物理实现的逻辑进程。

虚拟逻辑进程(V): 以虚拟形式存在并映射到一个物理逻辑进程的逻辑进程。虚拟逻辑进程映射到物理逻辑进程记为 $V \rightarrow P$。

仿真: 由一组虚拟逻辑进程和一组物理逻辑进程组成,具有 A、B 和 C 三个逻辑进程的原始仿真记为二元组($< V_A^1, V_B^1, V_C^1 >$, $< P_A^1, P_B^1, P_C^1 >$),其中 $V_A^1 \rightarrow P_A^1$, $V_B^1 \rightarrow P_B^1$, $V_C^1 \rightarrow P_C^1$。

仿真版本: 每个被克隆的仿真称为仿真的一个版本,初始仿真为版本 1,被克隆的第一个仿真为版本2,以此类推。

主动克隆: 到达决策点而主动进行的克隆。主动克隆将创建新的仿真。

被动克隆: 随着仿真的推进,因收到已克隆逻辑进程的消息,虚拟逻辑进程被动地克隆自己而产生物理逻辑进程。被动克隆不创建新的仿真。

完全克隆: 只要到达决策点就对整个仿真剧情内的所有物理逻辑进程进行克隆。如: 当仿真到达决策点时, P_A^1 发起克隆(以“ ⇒”表示克隆),典型的完全克隆为

$$(< V_A^1,\ V_B^1,\ V_C^1 >,\ < P_A^1,\ P_B^1,\ P_C^1 >) \Rightarrow (< V_A^3,\ V_B^3,\ V_C^3 >,\ < P_A^3,\ P_B^3,\ P_C^3 >)$$

其中: $V_A^1 \rightarrow P_A^1$, $V_B^1 \rightarrow P_B^1$, $V_C^1 \rightarrow P_C^1$, $V_A^3 \rightarrow P_A^3$, $V_B^3 \rightarrow P_B^3$, $V_C^3 \rightarrow P_C^3$。

递增克隆: 利用虚拟逻辑进程的思想,仅克隆在决策点状态发生变化的物理逻辑进程,状态不受影

响的物理逻辑进程以共享方式运行于新旧剧情之中。典型的递增克隆为

$$(<V_A^1, V_B^1, V_C^1>, <P_A^1, P_B^1, P_C^1>) \Rightarrow (<V_A^2, V_B^2, V_C^2>, <P_A^2, P_B^1, P_C^1>)$$
$$\Rightarrow (<V_A^2, V_B^2, V_C^2>, <P_A^2, P_B^2, P_C^1>)$$

第一步中: $V_A^1 \rightarrow P_A^1$, $V_B^1 \rightarrow P_B^1$, $V_C^1 \rightarrow P_C^1$, $V_A^2 \rightarrow P_A^2$, $V_B^2 \rightarrow P_B^1$, $V_C^2 \rightarrow P_C^1$;

第二步中: $V_A^1 \rightarrow P_A^1$, $V_B^1 \rightarrow P_B^1$, $V_C^1 \rightarrow P_C^1$, $V_A^2 \rightarrow P_A^2$, $V_B^2 \rightarrow P_B^2$, $V_C^2 \rightarrow P_C^1$。

可见,仿真克隆从两个方面提高了仿真执行的性能: ③克隆得到的多个仿真在克隆动作发生之前共享同一执行路径; ④虚拟逻辑进程的思想避免了仿真克隆中不必要的重复计算,使多个仿真能够在决策点之后进一步共享计算。

1.2 封装式仿真克隆中间件的不足

Schulze 等将仿真克隆技术扩展到HLA 仿真范型[4],提出了单联邦克隆与多联邦克隆的体系结构。Dan Chen 等对HLA 仿真克隆进行了深入系统的研究[5-6],提出了单联邦仿真克隆中的剧情区分算法,并将标准RTI 库封装成 RTI++ 库,构建了封装式仿真克隆中间件,支持 HLA 仿真克隆。

封装式仿真克隆中间件的实现思路为:将原有的 RTI 库封装为 RTI++ 库,RTI++ 保留原有 RTI 的标准服务接口,同时在RTI++ 内部扩展仿真克隆服务,并以提供调用和回调接口的方式供用户使用,如图3 所示。

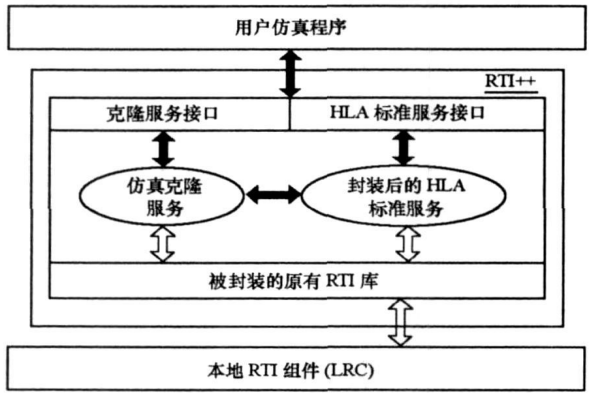

图3 封装式仿真克隆中间件

Fig. 3 Encapsulated simulation cloning middleware

封装式结构思路清晰,实现简单,但存在以下不足: ③需要修改仿真克隆的名字空间,如需要将联邦成员程序代码中的名字空间从原有的"RTI"替换为封装后的"RTI++",增加了代码的修改范围,且不利于仿真克隆对用户的透明性实现; ④封装式结构需要对原有 RTI 库的所有内容都进行封装,而原有 RTI 库的许多内容并不是HLA 仿真克隆都需要进行替换的,因此这种方式引入了许多冗余,降低了仿真克隆的效率。针对封装式的仿真克隆中间件的不足,本文提出了一种基于DDM 的拦截式的仿真克隆中间件方法。

2 拦截式仿真克隆中间件

2.1 拦截式仿真克隆中间件的设计

为使HLA 支持仿真克隆,必须增加支持仿真克隆的工具。仿真克隆工具的设计必须遵循两个重要原则:可重用性和透明性。可重用性要求最小限度地修改用户的已有代码;透明性则要求克隆动作本身的处理机制与用户仿真程序的耦合尽可能少,使用户将更多的精力关注于问题域。因此,仿真克隆工具应建立在HLA 提供的标准化服务基础之上,并由仿真应用程序调用,故应将仿真克隆工具实现为仿真应用程序与仿真支撑平台之间的一层中间件。

拦截式仿真克隆中间件是在用户仿真程序和本地 RTI 代理之间加入的一个处理仿真克隆的中间

层,该中间层可以截获RTI 本地代理和用户仿真程序之间的消息收发,并根据仿真克隆的需要对截获的信息进行处理,实现所需的仿真克隆功能。

如图 4 所示,仿真克隆中间件对仿真克隆操作进行了封装,并以服务的方式提供给用户。用户仿真程序与仿真克隆中间件之间的直接数据交互,主要用于与克隆相关的专项服务,如发起主动克隆、删除剧情、修改操作等。用户仿真程序向本地 RTI 组件(LRC)中的 RTIAmbassador 对象发出的调用消息分为两个部分,一部分被仿真克隆工具的消息拦截器所截获,经仿真克隆中间件处理之后,决定是否向 LRC 发出相应的调用,这种消息拦截仅仅截取与克隆操作相关的消息;另一部分是与克隆操作无关的调用或回调,消息拦截器不拦截这部分内容,它们仍由用户仿真程序直接传送给 LRC 进行处理,如查询当前推进时间、查询类句柄等,因此拦截机制不会带来冗余。仿真克隆中间件各组成部分的具体功能如下:

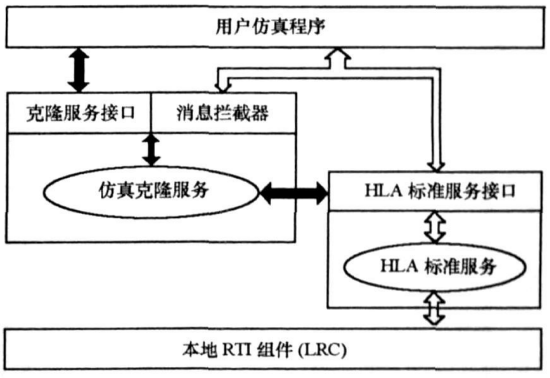

图 4　拦截式仿真克隆中间件

Fig. 4　Articulated simulation cloning middleware

(1) 消息拦截器: 负责截取与仿真克隆相关的由用户仿真程序通过 RTI Ambassador 向 LRC 发出的调用。与克隆相关的数据包括: 请求发出的更新/交互、公布/订购声明、注册的对象实例及其更新的恰当性定义设置、DDM 策略设置、时间管理策略设置等,这些数据多用于仿真管理方面的设定。因为 RTI 产品提供的开发库是以动态链接库(DLL)形式提供的,在实现中,我们采用了基于 API 钩子挂接方式的消息拦截方式[7],即将对给定的 RTI 库函数入口的调用挂接为克隆管理器的内部处理函数的入口,在完成相应处理之后再由克隆工具调用 RTI 库函数向 LRC 发送处理后的调用。

(2) 克隆服务接口: 完成两个方面的功能,一是作为仿真克隆服务的成员接口代理为用户仿真程序提供接口,用户通过该接口向仿真克隆服务发出主动克隆、剧情删除和查询等请求;二是实现回调转发功能,将仿真克隆服务接收到的来自 LRC 的回调信息发送给事件检查模块或剧情管理器进行处理,并将处理后的回调或者剧情管理器发出的其他消息以回调的方式发送给用户仿真程序中相应的 Fed Ambassador 对象。

(3) 仿真克隆服务: 具体完成消息转发、剧情管理、克隆生成、事件检查等功能。

2.2　拦截式仿真克隆中间件的实现方式

拦截式仿真克隆中间件的实现需要对原有 RTI 服务的调用进行重载,以区分不同剧情间的信息交换。仿真克隆中间层对标准 RTI 服务的重载方式如下所示:

```
// 对拦截到的消息进行克隆相关的处理
// 以拦截到某成员的 subscribeObjectClassAtrributes(theClass, attributeList, ...)为例
scenRegion = cloneManager.getFedRegion(pFed); // 获取与当前成员对应的区域
// 带区域向 LRC 发出订购
pFed->pRtiAmb->subscribeObjectClassAttributesWithRegion(theClass, scenRegion, attributeList, ...);
..// 其他处理
return;
```

2.3　拦截式仿真克隆中间件的联邦成员结构

在拦截式仿真克隆中间件的支持下,可克隆仿真中联邦成员的结构从传统的两层式结构演变为三层式结构,即加入仿真克隆工具层之后,仿真程序与 LRC 的紧耦合联系转变为一种松耦合联系,如图 5 所示。

从图中可以看出,在一般HLA 成员结构中,成员的数据流是一个单向回路,而在基于拦截式仿真克隆中间件的松耦合结构中,成员的数据流则可能构成多个回路。同时,在一般 HLA 成员结构中,一个成

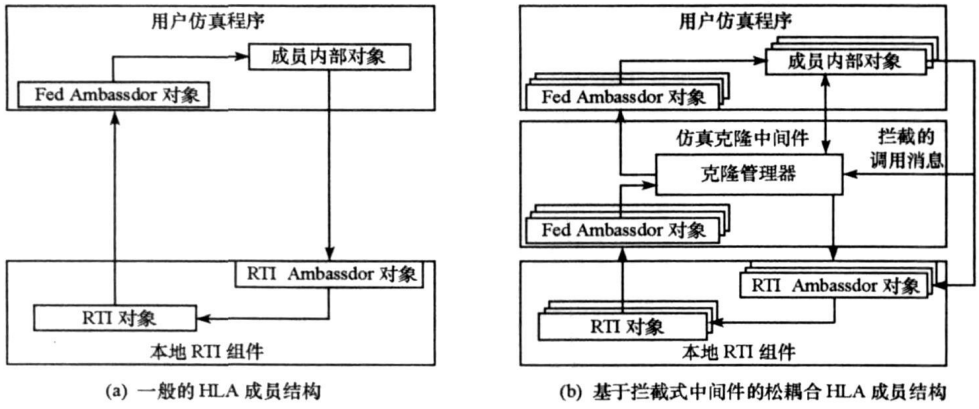

图 5 两种不同的 HLA 成员体系结构

Fig. 5 Two kinds of HLA federate architecture

员通常仅维护一个 FedAmbassdor 对象, 它与 LRC 中的一个 RTI Ambassdor 对象是一一对应的; 而在基于拦截式仿真克隆中间件的松耦合成员结构中, 用户仿真程序可以维护对应于多个成员的一组 Fed Ambassdor 对象, 同时 LRC 中也存在一组对应的 RTI Ambassdor 对象, 仿真克隆中间件负责维护它们之间的一一对应关系。

基于拦截式仿真克隆中间件的松耦合成员结构中增加了一个称为 Clone Ambassador 的成员 Ambassador 对象, 通过 CloneFedAmb 间接完成 LRC 对用户仿真程序的回调, 即 LRC 首先将回调返回给仿真克隆工具的 Clone Ambassador 对象, 然后经仿真克隆管理器处理之后, 再把处理后的回调传送给用户仿真程序中的 FedAmbassdor 对象。

3 测试与结果

3.1 结构测试

为验证基于拦截式仿真克隆中间件的松耦合成员结构的合理性, 评价仿真克隆中间件对系统时间性能的影响, 本文对该类成员结构的时间延迟性能进行了测试, 并将其与一般成员结构在同等情况下的测试结果进行了对比。测试实验的具体设置如下:

(1) 测试联邦: 由 n 对 KD _ RTI 连接的联邦成员组成, 每对成员分别表示为 A[i] 和 B[i], 其中 $1 \leqslant i \leqslant n$, 且 A 类和 B 类成员分别位于两个不同的物理节点上;

(2) 测试方法与指标: A[i] 向 B[i] 周期性地发送实例属性更新事件, 并等待从 B[i] 返回的响应。测量从发出事件到返回响应的时间间隔, 以足够多次(实验采用了 1000 次) 测量结果的平均值作为测量时间延迟的指标。其中对松耦合成员的测试引入了事件拦截与重发机制, 并给每对成员分配不同的 DDM 区域, 拦截之后的事件重发采用 DDM 区域发送的方式进行;

(3) 测试参数: 事件的大小 L, 成员对数目 n。

在 L= 0.1KB 和 L= 1KB 时, 实验的测试结果如图 6 所示, 从该结果可以看出, 松耦合成员和普通成员的时间延迟差值基本保持稳定, 且该差值随事件大小的增加而增大的幅度并不明显。这说明, 松耦合成员截获原有事件并带 DDM 区域重新发送这一附加操作, 所带来的时间延迟是有限且基本稳定的, 因此对于仿真克隆来说, 松耦合成员结构的性能是可以保证的。

3.2 系统运行测试

为检验本文研究的仿真克隆方法的效率, 测试子剧情数目和决策点时间对克隆效率的影响, 我们将该方法运用于 HLA 联邦的具体应用, 并进行了测试。具体测试如下:

(1) 测试联邦: 包括红蓝双方的空地攻防对抗联邦, 红方为 XX 飞机编队, 蓝方为 XX 高炮营。

(2) 具体方案:

1 克隆启动条件: 在仿真开始之后的 t= 200s, 400s, 600s, 800s 时刻, 分别启动 8 组不重复的完全

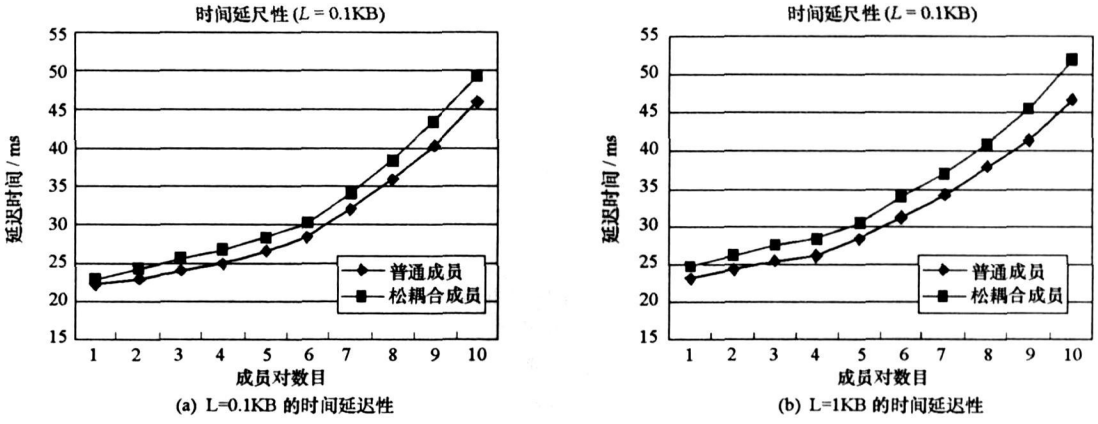

图 6　松耦合成员的时间延迟性能

Fig. 6　Performance of time delay on decoupled federates

克隆实验,每组实验中又包括 8 个子实验,它们分别对应于产生的子剧情个数为 $k(1 \leqslant k \leqslant 8)$ 的情形;

④仿真终止条件: 系统推进到给定的仿真时间 t = 1000s 时,每个子实验停止仿真运行。

㈣测试指标: 设仿真终止时子实验的物理时间开销为 Δt,传统仿真执行方式下一次运行的平均物理时间开销为 Δt_0,则测试指标取为仿真执行的加速比: $\alpha = (k \cdot \Delta t_0 / \Delta t)$。

(3) 预期结果: 克隆动作发生的时间越迟,仿真克隆执行中各个子路径的共享计算就越多,因而加速比越大;同样,克隆出的子剧情数目越多,共享计算量就越多,因而加速比越大。

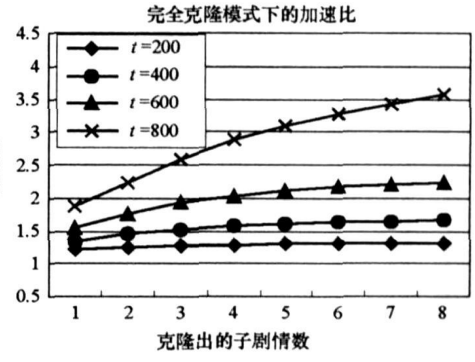

图 7　系统运行效率测试结果

Fig. 7　Efficiency of the system

(4) 测试结果如图 7 所示。

从图中可以看出: ¹ 给定克隆的子剧情数目时,克隆动作发生的时间越迟,仿真执行的加速比越大; ④给定克隆的动作发生时间时,克隆出的子剧情数目越多,仿真执行的加速比越大。实验与预期一致。

4　结 论

根据仿真克隆思想,针对封装式仿真克隆中间件的不足,提出了拦截式仿真克隆中间件的概念,并进行了设计与实现。与封装式中间件相比,虽然这种方式在仿真克隆中间件的设计与开发上较为复杂,但是,由于只是为原有程序添加了仿真克隆这类额外的服务,对用户代码的修改较少,因此,灵活性更好,透明性更强。从松耦合成员结构下的时间延迟性能测试结果可看出,该松耦合结构使联邦成员产生的时间延迟有限且基本稳定,可以保证联邦成员的时间性能,基于拦截式仿真克隆中间件方法是可行的。当然,要实现复杂大系统的实时在线多剧情并发仿真,还需要进一步研究克隆的相关机制。

参 考 文 献:

[1] Hybinette M, Fujimoto M. Cloning: A Novel Method for Interactive Parallel Simulation[C]// Proceedings of the Winter Simulation Conference. Atlanta, Georgia, USA, 1997: 444- 451.

[2] Hybinette M, Fujimoto M. Cloning Parallel Simulations[J]. ACM Transactions Modeling and Computer Simulation (TOMACS). 2001, 11(1): 378- 407.

[3] Hybinette M. Just-in-time Cloning[C]// Proceedings of the Eighteenth Workshop on Parallel and Distributed Simulation. Kufstein, Austria, 2004, 45- 51.

[4] Schulze T, Straβburger S, Klein U. Online-data Processing in Simulation Models: New Approaches and Possibilities Through HLA[C]// Proceedings of the 1999 Winter Simulation Conference. Washington DC, USA, 1999, 1602- 1609.

[5] Chen D, Turner S J, Cai W, et al. A Decoupled Federate Architecture for Distributed Simulation Cloning[C]// Proceedings of the 15th European Simulation Symposium, 2003, 131- 140.

[6] Chen D. Cloning Mechanisms for HLA-based Distributed Simulations[D]. Ph.D Thesis, School of Computer Engineering, Nanyang Technological University, 2005.

[7] 王建华,译. Windows 核心编程[M]. 北京: 机械工业出版社, 2000.