

# 可伸缩的分布式虚拟环境

王勇军,王意洁,赵 龙,李思昆

(国防科技大学计算机学院,长沙 410073)

**摘 要:** 本文采用行为的概念来描述虚拟环境的基本仿真计算单元,提出基于行为的动态负载分配算法来支持更细粒度的负载调度,从而提高了单用户分布式虚拟环境的可伸缩性.测试结果表明,该算法能有效地调节系统负载,在满足用户实时交互能力的前提下,尽量保持了行为的真实感.

**关键词:** 分布式虚拟环境;行为建模;负载分配;可伸缩性

**中图分类号:** TP15 **文献标识码:** A **文章编号:** 0372-2112 (2000) 02-0087-03

## Scalable Distributed Virtual Environment

WANG Yong-jun, WANG Yi-jie, ZHAO Long, LI Si-kun

(School of Computer, National University of Defense Technology, Changsha 410073, China)

**Abstract:** The concept of behaviors is adopted to describe fundamental simulation computing unit of virtual environment. A dynamic load allocation algorithm based on behavior is proposed to support fine-grain load schedule and promote the scalability of single-user distributed virtual environment. The algorithm has been implemented in a development platform of distributed virtual environment. The test result shows it can tune the system load efficiently to keep the behaviors fidelity while promoting the real-time interactivity of users.

**Key words:** distributed virtual environment; behavior modeling; load distributing; scalability

## 1 引言

虚拟环境属于高密度计算,它不仅体现在生成逼真的视觉、听觉效果,还体现在不同应用领域的仿真计算的复杂性上.分布式虚拟环境系统利用分布式系统的计算资源来满足虚拟环境计算的需求,已成为虚拟现实系统实现的有效途径.由于交互性的特点,虚拟现实系统同时又是计算负载高度动态变化的系统,这要求系统本身具有动态负载分配的功能.因此,本文针对分布式虚拟环境的可伸缩性进行了重点研究. Macedonia<sup>[1]</sup>将这样的分布式虚拟环境的结构定义为具有可伸缩性:即当虚拟环境中的并发动态实体或用户数量大幅度增加时,软件结构本身不须作多大的修改就能对此进行支持.当前存在的一些较为典型的分布式虚拟环境系统<sup>[2~6]</sup>只支持或部分支持基于空间划分的实体间的并行,而本文采用行为来描述虚拟环境的基本仿真计算单元,提出基于行为的动态负载分配算法来支持更细粒度的负载调度,从而更好地支持系统的可伸缩性.

## 2 虚拟环境中的行为

### 2.1 行为

从面向对象的观点来看,虚拟环境是由若干公共属性和若干虚拟实体组成.虚拟实体则由若干行为和局部属性组成.所谓行为是实体对公共属性、局部属性或调用其它行为的操

作.行为是仿真计算的基本单元.

**定义1 实体集合** 设有一虚拟环境  $VE$ ,由  $N$  个虚拟实体组成,实体集合可记为:  $EntitySet = \{ Entity_i | i = 1, \dots, N \}$ .

**定义2 行为** 行为 behavior 可以记为三元组:  $\langle InputAttrSet, OutputAttrSet, ProcessingProc \rangle$

其中,处理过程  $ProcessingProc$  是行为计算程序;输入属性集合  $InputAttrSet$  是行为计算所需的实体属性或公共属性;输出属性集合  $OutputAttrSet$  是行为计算进行修改的实体属性或公共属性.

行为分为二类:

(1) 表现属性行为:即对实体表现属性进行操作的行为.所谓表现属性,指与仿真效果输出的有关属性,如三维几何、速度、位置等;

(2) 非表现属性行为:对实体非表现属性进行操作的行为,如涉及到智能属性的行为等.

### 2.2 虚拟环境的性能需求

虚拟环境运行的主要性能需求是,实时交互性和真实感.实时交互性是指虚拟环境计算要足够的快,使得用户的输入能实时地在输出上反映出来,如视点的变化引起的场景变化.用户执行一个动作和在屏幕上显示出来之间的时间差称为端端延迟 (EndEndLag),考虑单处理机的情况,虚拟环境的运行可以看作是由一个个仿真步组成的,实时交互性则要求每个

仿真步的  $EndEndLag < \cdot$  为特定应用所指定的时间. 而对行为而言, 虚拟环境的真实感体现在实体行为具有行为的真实感.

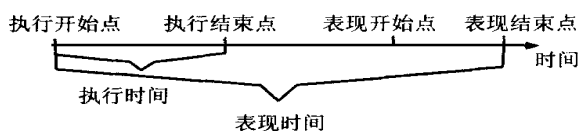


图1 行为的相关时间

如图1所示, 行为的表现时间包括执行时间, 由于多实体多行为的仿真, 其执行结果并不会马上通过虚拟环境的表现单元以图形或声音等形式表现出来, 只有当所有实体行为的表现时间都能在限定的时间内完成, 虚拟环境的行为真实感才得到保证.

### 3 基于行为的单用户分布式虚拟环境动态负载分配

相对于传统的负载分配策略, 虚拟环境的负载分配有一个更高的要求是既要保持实时交互性, 又要保证行为计算具有一定的真实感. 限于篇幅, 本文仅介绍单用户分布式虚拟环境的动态负载分配算法.

#### 3.1 基本概念

设有一个处理机集合供虚拟环境运行, 统一称集合中的处理机为结点. 结点集合可划分为两类: 用户结点集合  $UserNodeSet$ 、非用户结点集合  $NonUserNodeSet$ . 用户结点是用户参与虚拟环境的处理机, 通过它用户可以实时地与虚拟环境交互. 而非用户结点是用来支持对虚拟环境计算的分布并行处理的, 用户结点无法满足的行为计算可以迁移到非用户结点来处理. 非用户结点集合按分配到的行为计算也分为两类: 非表现属性行为结点集合  $NonPresentationBehNodeSet$  和表现属性行为结点集合  $PresentationBehNodeSet$ . 在虚拟环境运行过程中, 虚拟实体  $Entity_i$  在仿真步  $st$  有一个活跃行为表  $ActiveBehaviorTable(Entity_i, st)$ ; 活跃行为按执行计算所在的地点, 可分为局部行为集合  $LocalBehSet(Entity_i, st)$ 、远程行为集合  $RemoteBehSet(Entity_i, st)$ . 远程行为集合按等待计算结果方式分为同步行为集合  $SyncBehSet(Entity_i, st)$ 、异步行为集合  $AsyncBehSet(Entity_i, st)$ .

**定义3 行为开销** 行为开销可以记为二元组:  $\langle CompCost(behavior), CommCost(behavior) \rangle$

其中, 行为计算量  $CompCost$  可以用行为计算程序中的指令数来估算;

行为通信量  $CommCost$  可以用行为的输入、输出属性值的大小总量来估算.

**定义4 结点通信量** 结点  $P$  在仿真步  $st$  的通信量可记为:

$$CommCost(P, st) = \sum_j CommCost(behavior_j),$$

其中,  $behavior_j$  表示  $Entity_i$  的第  $j$  个行为:

$$behavior_j \in RemoteBehSet(Entity_i, st).$$

**定义5 局部行为计算时间** 结点  $P$  在仿真步  $st$  的局部

行为计算时间  $LocBehCompTime$  可记为:

$$LocBehCompTime(P, st) = \sum_j CompCost(behavior_j) / s$$

其中,  $behavior_j \in LocBehSet(Entity_i, st)$ ;

结点  $P$  处理指令的平均速度为  $s$  条指令/秒.

**定义6 行为等待时间** 行为  $behavior_{mn}$  的等待时间指从该仿真步开始到从网络上得到相应行为计算结果的时间, 可记为  $WaitTime(behavior_{mn})$ , 其中,  $behavior_{mn} \in SyncBehSet(Entity_m, st)$ .

**定义7 同步行为计算结果返回时间** 结点  $P$  在仿真步  $st$  的同步行为计算结果返回时间可记为:

$$SyncBehWaitTime(P, st) = MAX(WaitTime(behavior_{mn}))$$

**定义8 仿真计算延迟** 结点  $P$  在仿真步  $st$  的仿真计算延迟  $SimuLag$  为:

$$SimuLag(P, st) = MAX(LocBehCompTime(P, st), SyncBehWaitTime(P, st))$$

**定义9 结点开销** 结点  $P$  在仿真步  $st$  的开销记为:  $\langle SimulationTime(P, st), CommCost(P, st) \rangle$

根据行为真实感的讨论, 每个结点都有一个仿真计算延迟上限时间  $MaxSimLag$ , 如果超过了上限时间, 就会造成实时交互性能的下降; 另外, 如果网络上的通信流量过大, 必然会带来大的网络延迟和主机开销, 因此, 给每个结点界定一个通信量上限  $MaxCommCost$ , 二者构成结点的性能指标.

**定义10 结点性能指标** 记结点  $P$  的性能指标为:  $\langle MaxSimLag(P), MaxCommCost(P) \rangle$ .

设用户结点和非用户结点中的表现属性行为结点的仿真计算延迟的上限时间为  $ST$ ; 非用户结点中的非表现属性行为结点的仿真计算延迟的上限时间为  $ST$ , 存在  $ST > ST$ .

**定义11 结点的性能潜力** 结点  $P$  在仿真步  $st$  的性能潜力包括计算潜力和通信潜力, 计算潜力是指仿真计算延迟上限时间与局部行为计算时间的差额 (即  $MaxSimLag(P) - LocBehCompTime(P, st)$ ), 通信潜力是指通信量上限与结点通信量的差额 (即  $MaxCommCost(P) - CommCost(P, st)$ ).

#### 3.2 基于行为的动态负载分配算法

##### 3.2.1 算法的基本思想

**启动时机** 结点的仿真计算延迟超过上限时间  $MaxSimLag$ .

**处理方法** 如果是由于同步远程行为的计算结果返回的等待时间太长, 就选择合适的行为调入用户结点或将远程行为改为异步方式接收结果. 如果是由于局部行为的计算时间太长, 就选择合适的行为调出到合适的非用户结点上.

**行为和结点选择原则** 非表现属性行为比表现属性行为为被选择调出的优先级高; 行为的计算量与通信量的比值越大, 被选择调出的优先级越高; 而结点性能潜力越强, 被选择作为行为的接收者的优先级较高.

**调度对象** 因为非周期行为的不可预测性, 对其进行调度将会使系统极不稳定, 调度的行为都是周期性行为或长时间的行为.

##### 3.2.2 测试例子 用基于 OML<sup>[7]</sup> 开发的虚拟环境建模语言

YHVM 编写一个动画程序. 一个几何模型较为简单的智能体具有三个周期性行为, 一个是表现属性行为 MOVE 和 WAG TAIL, 一个是非表现属性行为 THINK. MOVE 是在一个三维空间移动, THINK 则是进行一个大数集合的排序, 每次排序成功, 调用 WAG TAIL 摇动尾巴.

虚拟环境运行在两台 PC (奔腾 586, 120MHz 主频, 32MB 内存, S3 64V+ 的图形加速卡) 上, 二者通过以太网 (100Mbps 共享 HUB, 10Mbps 网卡) 连接.

通过设定配置文件, 在虚拟环境中存在两个智能体, 每个都周期执行行为 MOVE 和 THINK. 当负载预分配时, 两个智能体都被分配到用户结点上运行. 测试结果如图 2 所示, 其中, RenderLag 指每一帧的图形绘制时间, SimuStepLag 即每一帧的端端延迟. 可以看到, SimuStepLag 大于 100ms 的比率约有 99%. 当启动动态负载分配算法之后, 智能体的 THINK 行为都分配到非用户结点上运行, 测试结果如图 3 所示. 可以看到, SimuStepLag 大于 100ms 的比率只有 15%, 基本上达到了实时交互的视觉效果.

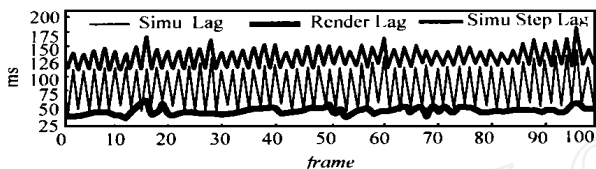


图 2 测试结果 1

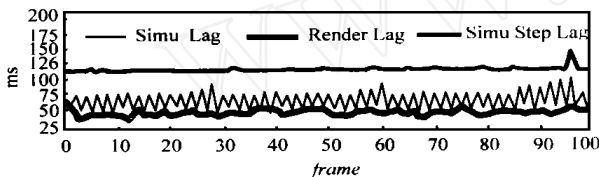


图 3 测试结果 2

#### 4 小结

本文通过采用行为的概念来统一表示虚拟环境中的仿真计算单元, 并分析了虚拟环境的性能需求. 鉴于当前分布式虚拟环境系统动态负载分配策略的不足, 本文提出的基于行为的细粒度负载分配算法使得单用户分布式虚拟环境系统具有更好的可伸缩性.

#### 参考文献

- [1] M. R. Macedonia. A Network Software Architecture for Large Scale Virtual Environments. Ph. D thesis, Naval Postgraduate School, June 95
- [2] R. Kazman. Making WAVES: On the design of architectures for low-end distributed virtual environments. IEEE Symposium on Research Frontiers in Virtual Reality, October 1993, San Jose, CA: 443 ~ 449
- [3] J. W. Barrus, R. C. Waters, and D. B. Anderson. Locals: Supporting large multiuser virtual environments. IEEE Computer Graphics and Applications, 1996, 16(6): 50 ~ 57
- [4] D. N. Snowden. AVIARY: Design issues for future large-scale virtual environments. Presence: Teleoperators and Virtual Environments, Fall 1994, 3(4): 220 ~ 241
- [5] O. Hagsand. Interactive multimedia VEs in the DIVE system. IEEE Multimedia, March 1995, 3(1): 30 ~ 39
- [6] R. Kazman. Load balancing, latency management and separation of concerns in a distributed virtual world. In Zomaya A. (ed.). Parallel Computer Paradigms and Applications, Van Nostrand: 1996, 443 ~ 454
- [7] H. Sun. A relation model for animating adaptive behavior in dynamic environments. Ph. D thesis, University of Alberta, Alberta, Canada, 1992



王勇军 1971 年生, 1992 年、1995 年和 1998 年分别获国防科技大学计算机专业学士、硕士和博士学位. 现为国防科技大学计算机学院助理研究员. 已在国内外发表论文 30 多篇, 主要研究方向为分布式虚拟环境技术、医学虚拟现实技术和分布式仿真技术.



王意洁 1971 年生, 1993 年、1998 年分别获国防科技大学计算机专业学士、博士学位. 现为国防科技大学计算机学院分布并行处理国家重点实验室助理研究员. 已在国内外发表论文 40 多篇, 主要研究方向为元计算技术、面向对象数据库技术.