第 33 卷 第 5 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.5 Computer Engineering March 2007
• 开发研究与设计技术 • 文章编号：1000—3428(2007)05—0275—03 文献标识码：A 中图分类号：TP391.41

基于保留模式并行绘制的图形对象分布策略

沈兵虎，潘瑞芳，金哲凡
（浙江传媒学院传播科技系，杭州 310018）

摘 要：保留模式并行图形系统在并行性能上有着巨大的潜力。透明、高效的图形对象分布是保留模式并行图形系统的基础。该文提出了一种完整的策略，包括通用的图形对象定义、透明的对象创建和自动一致性维护机制、状态缓存、延时更新、基于消息应答的图像同步输出。该策略实现了自动、高效的图形对象分布，并在高性能集群并行图形系统 MSPR 中得到了验证。

关键词：并行图形系统；保留模式；对象分布；延时更新

Graphics Object Distribution Strategy Based on Retained-mode Parallel Rendering

SHEN Binghu, PAN Ruifang, JIN Zhefan
(Department of Media Technology, Zhejiang University of Media and Communication, Hangzhou 310018)

【Abstract】 Retained-mode parallel graphics systems have great potential in parallel performance. A transparency, effective strategy for graphics objects distribution is the basic of retained-mode parallel graphics system. A systematic solution is presented. It includes the definition of graphics objects, transparency object creation, automatic coherence maintenance, state caching, delayed update and message-based image synchronization. The solution realizes automatic, high-performance graphics object distribution and is tested in MSPR which is high performance parallel rendering system based on cluster.

【Key words】 Parallel graphics system; Retained-mode; Object distribution; Delayed update

1 概述

计算机图形系统追求 2 个基本目标：真实感和速度。提高计算速度一直是图形系统的一个关键问题。目前微机高端显卡的三角形处理能力超过了 30M triangles/s，一些显卡带有 16 个硬件光源，4～8 条纹理通道，其性能已远远超过了早期昂贵的专用图形工作站[1,2]。在一些应用领域，如大数据集的科学计算可视化、高分辨率超大屏幕显示、顶点数在 10M以上数量级三角形的巨型几何场景交互式绘制、大纹理数据量的绘制等，孤立(stand-alone)的图形系统仍难以应付。孤立图形系统的性能在以下 4 个方面受到限制[3,4]：产生几何数据的能力、图形计算能力、几何指令发射能力和显示分辨率。并行图形技术为以上问题提供了解决办法，在各种高端图形系统中得到了应用。

图形系统分为立即模式(Immediate-Mode)和保留模式(Retained-Mode) 2 种。如图 1、图 2 所示，在立即模式里，图形数据保存在应用程序一端，每绘制一帧，应用程序将图形数据发送给图形系统，后者进行绘制计算，之后将数据丢弃。在保留模式图形系统里，应用程序定义了图形数据之后，数据就保留在图形系统一端。在图 1 的单机绘制情况下，数据保存在哪里对计算速度没有本质影响。而在分布式并行图形系统中，情况发生了变化，如图 2 所示，应用程序和绘制器分布在 2 个进程里，彼此通过高速网络连接。立即模式并行图形系统每一帧的绘制都要求所有图形数据通过网络，将造成很大的通信负担，而保留模式可以避免这种情况。对立即模式并行图形技术的研究相对比较成熟。保留模式并行图形系统中，投影墙驱动系统 Display Wall 的每个绘制结点从硬盘转载并保留场景数据，实现了一种简单、面向应用的保留模式并行绘制。分布式 3D 图形库 Repo-3D 嵌入在分布式编程语言 Repo 中，为快速构建分布式图形应用提供了一套多线程、面向对象的机制。

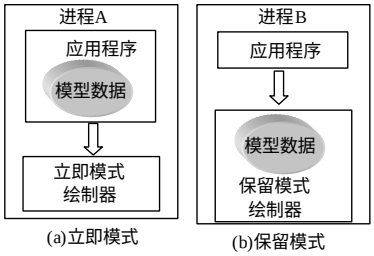

图 1 单机环境下的立即模式和保留模式

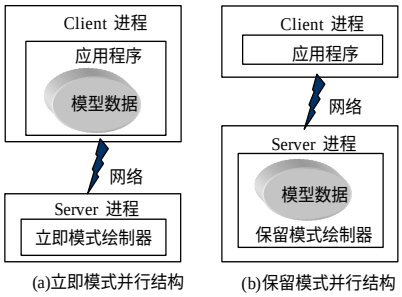

图 2 分布式并行绘制环境下的立即模式和保留模式

基金项目：浙江省自然科学基金资助项目（Y105324）；浙江省科技厅基金资助项目（2006C31065）
作者简介：沈兵虎(1954－)，男，硕士、副教授，主研方向：电子集成，图像处理；潘瑞芳，硕士、教授；金哲凡，博士、讲师
收稿日期：2006-10-24 **E-mail：**jinzf@cad.zju.edu.cn

相比立即模式图形并行绘制，保留模式的图形并行体系结构有明显的优势。由于网络通信的速度比总线速度低 3～6 个数量级，大量数据通过网络势必对系统整体性能构成冲击。如果采取保留模式并行绘制的方式，将数据保存在服务器端，则可能避免大量的数据传输，降低系统开销，提高速度。要构建高性能的保留模式并行图形系统，则必须先解决图形对象分布的问题。

2 保留模式并行的图形对象分布问题

图形对象分布是保留模式并行图形系统的基本问题，图形对象有 3 类：(1)模型对象：虚拟世界中物体，具有几何信息以及材质和纹理等属性；(2)环境对象：虚拟世界里的光照、相机参数等环境属性；(3)绘制命令对象：绘制器对场景的绘制命令，比如清除背景、交付帧缓存。

并行图形系统中的对象分布问题，可归结为以下 3 个子问题：(1)如何将模型对象高效地分布到多台服务器上；(2)如何将环境属性的改变及时传递到服务器上而不引起性能下降；(3)如何保证绘制命令在服务器端以正确顺序执行和图像的同步输出。

Princeton大学的多机大屏幕绘制系统Display Wall[5]采用了一种简单的图形对象分布策略。系统启动时，客户端和服务器从硬盘将模型对象加载到内存。在图形交互程序运行中，客户端计算当前帧的潜在可见集合(Potentially Visible Set, PVS)，得出每个服务器应分配到的部分PVS数据。之后，客户端向每个服务器发出消息，服务器进行所属PVS数据的绘制和像素的重分布，然后发送“帧结束”消息给客户端。客户端接收到所有服务器的“帧结束”消息后，命令所有服务器交付帧缓存，实现同步的图象输出。

DisplayWall 的对象分布策略存在以下问题：(1)由于图形数据格式和应用程序风格的多样性，因此从硬盘加载数据的方式削弱了系统的通用性。(2)用内定的消息分布环境对象是一种针对应用的做法，不能覆盖所有的环境参数，当新的应用出现时，常常需要定义新的消息。(3)绘制命令以一条条“命令－应答”的方式工作，执行效率比较低。

由于这些问题，Display Wall 系统整体的封装性比较差，针对特定的应用，对象分布和并行绘制部分的代码通常需要重写。成熟的保留模式分布式并行绘制系统必须建立在一个比较完备的图形对象分布策略的基础上，该图形对象分布策略应该有以下特性：(1)透明地、安全地实现对象的创建和删除。(2)自动维护对象的同步。(3)面向图形应用特点，性能上满足图形并行绘制的需要。(4)良好的兼容性。

立即模式的并行图形系统 WireGL 采用基于命令流和状态懒更新(Lazy Update)的对象分布策略，效率很高，由于体系结构上的差异，因此这套方法不能直接用于保留模式并行绘制系统。通用的分布式系统对象分布解决方案如 DCOM 和 CORBA 可以解决对象分布问题，但是图形对象的数据类型有限、绘制命令相对固定，DCOM 和 CORBA 对于图形绘制任务显得过于庞大，使用它们作为底层机制会大大降低系统的反应速度。

本文提出一种系统的保留模式并行图形系统对象分布策略，它包括完整的图形对象定义、透明的对象创建和自动的一致性维护机制、以及本地缓存和基于消息的同步。该策略吸取了 WireGL 系统策略和 DCOM/CORBA 的优点，没有 DisplayWall 系统策略的缺陷，较好地解决了保留模式图形并行系统的对象分布问题，在 MSPR(Multi-Screen Parallel Rendering)系统中得到了应用。

3 图形对象定义

根据全局场景的组织和模型几何数据的规律，可以定义保留模式并行图形系统的模型对象。图 3 显示了保留模式系统图形对象的场景/模型结构：应用程序可以拥有多个场景，每个场景内部有若干模型，而每个模型拥有顶点、向量、纹理等属性。通用图形绘制器处理的几何数据类型是有限的，本文选择了 10 种常用的几何数据，即 points、lines、line strip、line loop、polygon、quads、quad strip、triangles、triangle strip 和 triangle fan。其中 points, lines, triangles, quads 4 种类型分别固定为 1、2、3、4 个点一组，其他 6 种类型每组数据点数不固定，根据这些特性本文安排其存储办法。

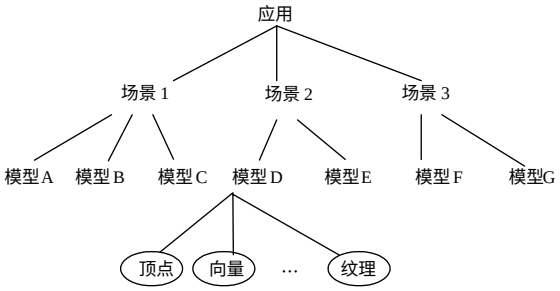

图 3 图形对象结构

4 对象创建和一致性维护

除了对象的静态结构之外，并行图形系统还必须实现对象的动态特性，系统要处理以下 2 个问题：(1)对模型对象的引用发生在本地，而实际的模型对象存在于远端，如何保证对模型对象的引用准确地映射到服务器上；(2)在什么时机建立和删除对象。

对于问题(1)，本地对模型对象的引用是通过内存地址，而该地址在远端的服务器进程中无效。为了保证模型对象在两端的正确映射，参考 DCOM 的 GUID 策略，为每一个模型对象分配一个全局唯一的标志，系统维护对象 ID 的一致性。本地通过地址对某一对象的访问在服务器端映射为对某一 ID 的对象的访问。

第(2)个问题涉及对象分布策略的一个重要特性：透明性。由于系统分布实现的事实应该对应用程序透明，应用程序不必知道绘制发生在本地还是远端，因此不能显式地出现“创建远程对象”或“删除远程对象”这样的命令，系统必须自动、安全和高效地完成这些动作，其关键是确定生成和删除对象的合适时机。

通常，这个问题可以通过统一的对象构造函数、或复杂的初始化机制来解决。本文采取了一种更加简明有效的解决方案，如图 4 显示流程，在模型对象的构造函数里，给对象分配 ID 为-1，在每一次对对象的方法调用中，先检查对象是否合法。对象不合法的情况有 2 种：

(1)ID＝-1，说明对象在远端尚未创建。碰到这种情况，系统在远端创建对象，并给本地和远端对象分配同一个合法的 ID 值。

(2)ID 合法，但是对象的本地数据缓存不为空。碰到这种情况，系统将对象缓存中数据发送到远地。

相对其他方案，这个方案的对象在本地和远地的创建不是同时发生的，这是一种异步的方法，它有以下优点：

(1)由于对对象的访问都是通过对象方法发生，而对所有引用对象的函数都作合法性检查，因此能充分保证对象的一致性和安全性。

(2)由于整数的比较在计算机内部是一种极快的操作，因此这种方案的开销很小。

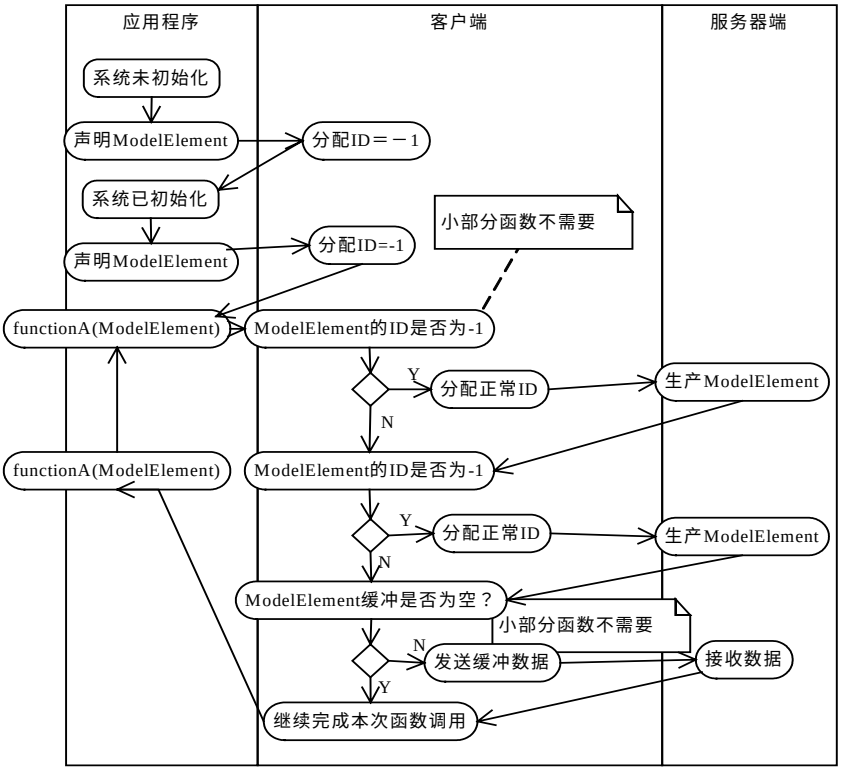

图 4 对象创建

5 缓存和同步

在交互式图形应用中，环境对象的改变很频繁，会造成大量的网络数据包在网络上传递，使系统速度下降，因此环境对象分布方法要避免过多的网络应答。由于环境对象改变的效果在帧图像交付时才可见，因此它们具有可延迟更新的特性。

利用这一点，可以对状态命令进行缓存和延迟发送，代码如下：

```
MSRP_MatrixMode(GL_MODELVIEW);
MSRP_LoadIdentity();
MSRP_Scalef(m_xScaling,m_yScaling,m_zScaling);
MSRP_Translatef(m_xTranslation,m_yTranslation,
m_zTranslation);
MSRP_Rotatef(m_xRotation,1.0f,0.0f,0.0f);
…
DrawScene();
```

对于针对变换矩阵的缩放、旋转和位移命令，系统并不将它们直接发送到服务器端，而是在本地状态缓存里生成如下片断，如图 5 所示。

MatrixMod	参数	LoadIdentity	参数	Scale	参数
Translate	参数	Rotate	参数		

图 5 生成片断

当运行到 DrawScene 命令时，系统才将累积起来的状态缓存发送到服务器端。服务器进行状态集中设置，再进行绘制。通过状态的缓存和延迟更新，有效地减少了网络消息数，提高了环境对象分布的效率。

在多服务器绘制输出时，绘制命令在多台服务器上执行的时间是不一致的，为了避免图象出现跳跃，必须对各服务器的图像输出进行同步控制。实现同步的方法有 2 类：

(1)利用专用的并行图形 API 进行同步，比如，可利用文献[6]中的 barrier 工具进行缓存交付的同步。

(2)直接用网络消息应答实现同步。

方式(2)利用客户端作为控制中心，由客户端控制绘制进程，实现同步的帧缓存交付，其流程如图 6 所示。本文在 MSPR 中采用了这种方法。

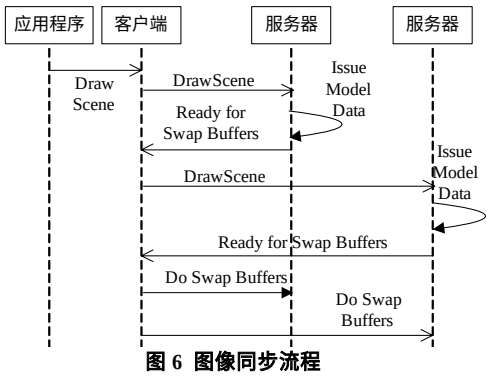

图 6 图像同步流程

6 MSPR 系统

MSPR是一个保留模式并行图形绘制系统[7]，支持 32 结点绘制集群和超高分辨率投影墙输出。本文所述图形对象分布方法在其中得到了应用。MSPR中复杂的对象分布过程对应用程序而言是透明的，对状态和数据则使用了缓存和延时更新技术以提高速度。图 7 是MSPR系统多屏拼接输出成高分辨率显示墙的几个例子。

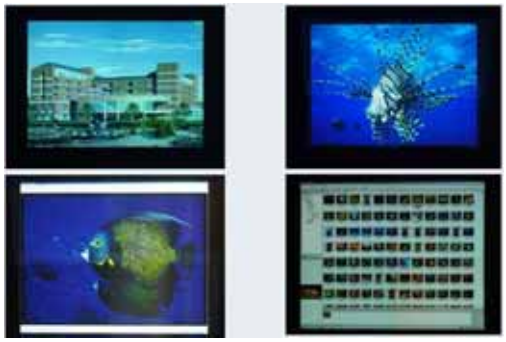

图 7 MPPR 显示墙输出

7 总结和展望

保留模式并行图形系统的关键问题之一是图形对象分布策略。对象分布方法是保留模式并行绘制系统的基础，在此之上有很多保留模式并行的问题是值得研究的，比如数据存储、归属判断、负载平衡等。

参考文献

1 Yang C, Sano B, Lebeck A R. Exploiting Instruction Level Parallelism in Geometry Processing for Three Dimensional Graphics Applications[C]//Proceedings of the 31th Annual ACM/IEEE International Symposium on Microarchitecture. 1998.

2 Accelerated Graphics Port Interface Specification[Z]. 1998. http://www.intel.com/pc-supp/platform/agfxport.

（下转第 280 页）