

文章编号:1001-9081(2010)02-0547-04

## 基于PC集群的三维图形并行渲染性能分析

彭 帅,李冬梅,李朝晖

(中国传媒大学 信息工程学院,北京 100024)

(ldm@cuc.edu.cn)

**摘 要:**研究基于PC集群的三维图形并行渲染性能问题,从网络性能、算法复杂度、并行分配机制等几方面分析了影响并行渲染性能的关键因素。在千兆以太网PC集群上进行了基于通用MPI和OpenGL的三维图形并行渲染仿真测试,给出了数据及分析结果,给出了合理构建并行三维图形渲染系统的建议,通过平衡图形算法复杂度和网络性能以达到最佳并行性能。

**关键词:**并行渲染;OpenGL;MPI;并行效率

**中图分类号:**TP338.6 **文献标志码:**A

## Parallel rendering efficiency analysis of 3D graphics on PC cluster

PENG Shuai, LI Dong-mei, LI Zhao-hui

(Information Engineering School, Communication University of China, Beijing 100024, China)

**Abstract:** In this paper, the efficiency of parallel 3D graphics rendering on general-purpose PC cluster was investigated. Three key factors affecting the efficiency of parallel rendering were analyzed, including network performance, algorithm complexity and parallel rendering mechanism. Parallel rendering simulations based on MPI and OpenGL were carried out on gigabit Ethernet PC cluster. The experimental results were analyzed and the advices for designing efficient parallel rendering system on PC cluster were given.

**Key words:** parallel rendering; OpenGL; MPI; parallel efficiency

### 0 引言

三维图形渲染是数字媒体制作的重要技术,涉及大量计算。高清晰高复杂度的三维图形渲染非常耗时,成为三维图形应用的瓶颈。有效解决方法之一是使用昂贵的专业图形工作站。随着PC机性能的提高和价格的降低,利用多个PC构建集群进行并行处理来解决高复杂度大运算量的三维图形渲染不失为一种值得尝试的有价值的解决方法。针对不同的应用需求,国内外研究机构在这方面进行了相关研究和应用尝试,例如根据场景独立性,通过子任务分配进行并行绘制<sup>[1]</sup>;基于PC集群和Linux系统进行地形场景的绘制和多通道屏幕输出实现实时漫游<sup>[2]</sup>。本文重点研究PC集群并行渲染的效率问题。与单机三维图形渲染相比,并行渲染从理论上讲可以大大缩短三维图形渲染时间,显著提高工作效率。然而并行三维图形渲染在实现过程中却会不可避免地遇到几个问题影响并行性能。从仿真测试结果具有通用性的角度,本文在千兆网PC集群<sup>[3]</sup>上进行了基于OpenGL图形库<sup>[4]</sup>和MPI消息传递接口<sup>[5-7]</sup>的并行三维图形渲染仿真实验,从三个方面研究和分析了影响并行渲染性能的主要因素,分别是集群系统硬件及网络性能、三维图形渲染算法复杂度及并行分配机制。综合考虑这些因素后,可以找到合理构建并行平台、平衡图形算法复杂度和网络性能以到达最佳并行性能的方法。

### 1 并行渲染仿真

测试三维图形并行渲染性能,最好能够使用实际的商业

应用软件。由于商业软件各有特性,且不公开代码,无法移植和修改底层程序,所以本课题所使用的程序是在开源OpenGL代码基础上修改设计的,结合MPI进行并行移植、设计并行机制,并增加了细致的算法复杂度调节控制,以尽可能真实地模拟实际应用中的复杂度需求。本文的并行渲染测试在实验室集群平台上进行,共5台PC,每个PC内部有4个节点(核),所用集群系统的实物如图1所示,集群平台配置如表1所示,网络系统为千兆局域网。仿真测试渲染几个三维对象构成的场景,包含一个三维容器、两个圆环和四个球体。连续渲染360帧,整个场景做一次完整旋转。并行渲染流程如图2所示。并行渲染过程中,参与计算的节点(node 1, ..., n)计算生产各图像帧,然后把渲染结果传回主进程节点(node 0, process 0)。以3进程并行渲染为例,进程1顺序计算第1, 4, 7, 10, ..., 258帧;进程2顺序计算第2, 5, 8, 11, ..., 259帧;进程3顺序计算第3, 6, 9, 12, ..., 360帧。子进程计算完一帧后立即读取数据并向主进程(node 0, process 0)发回数据,主进程顺序接收数据。主进程控制各个渲染进程的参数设置及数据收发顺序。

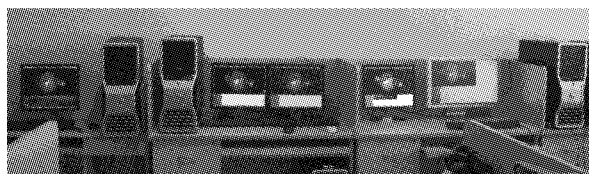


图1 PC集群系统实物图

收稿日期:2009-08-30。 基金项目:中国传媒大学211重点学科建设项目(21103040115)。

作者简介:彭帅(1983-),男,江西抚州人,硕士研究生,主要研究方向:视频信号并行处理与编码;李冬梅(1968-),女,新疆石河子人,副教授,博士,主要研究方向:数字视频处理、图像编码;李朝晖(1969-),女,河北涿州人,副教授,博士,主要研究方向:数字视频处理、图像编码。

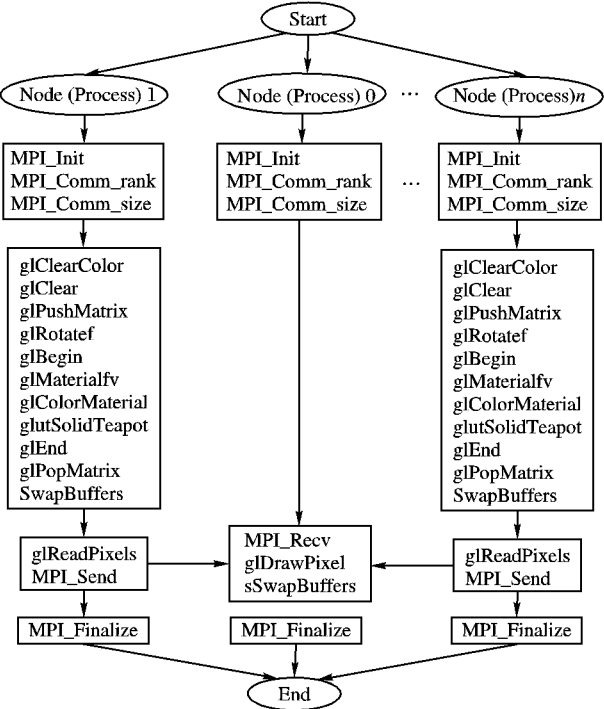


图2 并行渲染流程

表 1 PC 集群系统配置

设备	具体配置			
	CPU	内存	网卡	显卡
DELL AW690n	2 Intel Xeon	4GB	Intel 9400PT	NVIDIA
	5130	DDRII ECC	千兆	Quadro NVS
	2.0GHz	SDRAM	以太网卡	285

表 3 CPU 不同负荷下千兆局域网传输速率测试

Mbps									
CPU 负荷	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	平均值
26%	399.63	453.54	345.05	433.73	354.68	419.01	403.92	410.84	402.55
51%	326.78	410.84	400.74	414.99	314.41	341.10	390.07	325.06	354.50
76%	351.48	414.64	309.57	296.61	340.34	412.97	435.23	424.86	373.21

3 渲染算法复杂度分析

只有当渲染算法足够复杂的时候,并行渲染才有意义,才能有效地发挥并行效率。如果单机就能进行实时渲染(对 720×576 以上的图像),改成并行渲染后不会大幅提高渲染速度,因为网络拥挤的问题会明显起来,阻碍并行性能的提高。那么渲染算法的最低复杂度由什么因素界定呢?主要取决于网络速率的限制,同时也与所使用图形渲染软件平台有关。本文的渲染测试程序是基于 OpenGL 的,要利用显示卡的显存,渲染算法在处理数据时需要反复读写显存,所以显卡的性能将影响渲染速度。如果 CPU 性能好,而显卡性能差,渲染速度将会受到很大影响。这种情况下,并行渲染不能发挥优势。但是只要渲染算法足够复杂,读写显卡对渲染速度的影响就成为次要的了。渲染算法越复杂,显卡对并行性能的影响越小。

表 4 是对两种不同机型的测试,测试算法复杂度、硬件性能和渲染时间长度之间的关系。CPU 性能不同、显卡性能不同,显卡对渲染速度的影响也不相同。关于算法复杂度,其影响因素很多,取决于具体应用的具体渲染算法,没有统一的标

2 网络性能分析

不同的 PC 集群,其网络性能是不同的。本文在两组型号的集群上做了网络传输性能比较测试,如表 2 所示。为了排除读写硬盘及 CPU 工作负荷不同带来的差异,测试时仅作纯传输测试。

表 2 千兆局域网传输速度测试

表 2    千兆局域网传输速度测试						Mbps
机型	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	平均值
A	444.22	483.22	454.26	432.00	424.15	447.57
B	294.88	268.67	293.28	291.40	295.38	288.72

由表 2 可以看出,千兆局域网的网络速度不是恒定的,不同配置的集群网络速度差异很大。表 2 中机型 A 构成的千兆局域网平均速率为 447.57 Mbps,而机型 B 构成的千兆局域网平均速率仅为 288.72 Mbps,即使是同一个网络,其传输速率也有相当大的波动。在进行集群选型的时候,需要特别考虑这一点。

另外,影响网络传输速率的因素还有 CPU 工作负荷,也就是繁忙程度,表 2 所测试的是纯传输测试,CPU 并不做其他工作。如果在进行网络传输时,CPU 同时做复杂计算,那么传输速率将会会进一步降低并且波动,参见表 3。

网络速度不仅受网卡影响,也受 CPU 性能影响,同时也跟 CPU 使用率(繁忙程度)相关,所以在构建集群平台时要做充分的考虑,对网络速率要留有合理的裕量。鉴于这个考虑,对于本课题所用的集群平台(机型 A),能利用的网络传输速率应该预期在 350 Mbps 以下,加上频繁零碎的控制信息传输,真正能用于图形数据传输的网络速率应该更低。

准表示方法。为了描述和测试方便。表 4 中仅用三维图形对象的几何分割线数来表示。几何形状越复杂越细致,渲染也越细致越耗时。机型 A 的 CPU 性能高,显卡性能也相对较高。通常,算法复杂度增加,渲染时间也该相应增加,但表中数据显示,机型 A 的算法复杂度在一段较大的范围内从低到高增加(C1:10~C13:130),渲染所用时间并没有明显增加,这是因为 CPU 性能高,运算足够快,而反复读写显存制约了渲染速度。图 3 中机型 A 的曲线也清楚地反映了这个特点,当复杂度增加到一定程度(C14:140)后,曲线斜率明显增大,这时 CPU 的性能成为影响渲染速率的最主要因素,而显卡的影响将逐渐变为次要,以至无需考虑。

表 4 机型 C 属于 CPU 性能相对较低,显卡性能也相对较低的情况。观察表 4 和图 3 的机型 C 的曲线,可以看出,随着计算复杂度增加,渲染时间也成比例相应增加,CPU 的性能始终是影响渲染速度的最主要因素,而读写显存的影响是相对次要的。

为了能够合理仿真测试并行渲染性能,同时充分利用网络性能,应该合理地配置并行渲染应用系统,包括集群硬件平

台和渲染程序算法复杂度。设计复杂度适当的渲染测试程序,对不同尺寸的图像,要有一个最低限度的算法复杂度门限,这也非常符合图形渲染应用的实际需求。鉴于本文所用集群平台的具体性能(网络性能、CPU 处理能力、显卡性能等),最后用于并行测试的渲染程序算法复杂度应该按表 5 来定位。表 5 中列出了三种常用的影视数字媒体图像尺寸,图像越大,实时传输的数据量也越大,需要的网络速率也越大,但是对于一个具体网络,其数据传输能力是一定的。表 5 中假设网络稳定持续的有效数据传输能力为 200 Mbps,这个假设是合理的,是在前面网络分析基础上,保留了充分裕量之后提出的。表 5 根据网络承受力计算出合理的渲染复杂度门限(最低值),就能保证在网络承载力允许的范围内,充分利用并行系统,有效提高渲染效率。由于算法复杂度最终体现在渲染时间上,所以可以用单机单节点渲染 360 帧所需的时间来表示,表 5 及后面表中的复杂度都是用这种方式来表示的。另外假定集群网络中参与渲染运算的节点为 5 个。从表 5 可看出,图像越大,能发挥并行优势的最低算法复杂度也

相应增加。

表 4 渲染 360 帧所用的时间(图像尺寸 1 280 × 720)

算法 复杂度	分割 线数	时间/s	
		机型 A	机型 C
C1	10	4.782	1.218
C2	20	4.766	1.453
C3	30	4.782	1.797
C4	40	4.766	2.281
C5	50	4.781	2.781
C6	60	4.782	3.781
C7	70	4.781	5.015
C8	80	4.782	6.594
C9	90	4.782	8.094
C10	100	4.766	9.375
C11	110	4.782	10.875
C12	120	4.812	12.969
C13	130	4.969	15.078
C14	140	9.547	18.672

表 5 网络性能、图像尺寸和渲染算法复杂度之间的关系

图像尺寸	25 fps 实时图 像速率/Mbps	网络平均 速率/Mbps	网络允许的 最大帧率/fps	每节点分配 的帧率/fps	每节点渲染 1 帧 所需时间/s	每节点渲染 360 帧所需时间/s
720 × 576	248.832	200	20.094	4.019	0.249	89.580
1 280 × 720	552.960	200	9.042	1.808	0.553	199.066
1 920 × 1 080	1 244.160	200	4.019	0.804	1.244	447.898

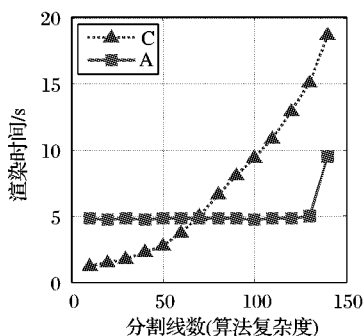


图 3 渲染时间与算法复杂度的关系

## 4 并行渲染性能分析

仿真测试时可以通过三维对象的参数来控制算法复杂度,并行性能可用并行效率或并行加速比表示。并行后总的渲染时间会缩短,通过跟单机渲染所需时间对比,可以方便地计算出并行效率。并行加速比可用单机渲染时间除以并行渲染时间来表示<sup>[6]</sup>。

### 4.1 单节点 PC 构成的 PC 集群

这部分测试的条件是,集群中共有 5 台 PC 机,每个 PC 只取一个(核)节点参与渲染运算,所以集群最多有 5 个节点并行运算。对 360 帧图像进行渲染,从单机单节点渲染逐步到 5 节点并行渲染,测试结果如表 6 所示。

分别对三种标准尺寸的图像进行测试,720 × 576,1 280 × 720,1 920 × 1 080,每种尺寸的图像有两级复杂度,表 6 中列出了不同情况下的并行加速比。表 6 中,D 表示显示,R 表示渲染,L 表示低复杂度,H 表示高复杂度。复杂度的差别可以从单机单节点渲染 360 帧图像所需的时间体现出来。从表 6 中第 4 列可以看出,对于 720 × 576 尺寸的图像,PC0 单机单节点既渲染又显示 360 帧图像所用的时间,低复杂度时为 90。

187 s,高复杂度时为 180.031 s,相差一倍。表 6 第 5 列为双节点并行工作的情况,从 PC0 中取一个节点用于显示,从 PC1 中取一个节点用于渲染。第 6 列为 3 节点并行工作的情况,从 PC0 取一个节点用于显示,从 PC1 和 PC2 中各取一个节点用于渲染。第 9 列为 6 节点并行工作的情况,从 PC0 中取一个节点用于显示,再从 PC0 中取一个节点用于渲染,从 PC1,PC2,PC3,PC4 中各取一个节点用于渲染,即共有 5 个节点并行作渲染运算。

从表 6 的结果可以看出,由单节点 PC 构成的 PC 集群并行渲染能够很好地、高效率地并行工作,非常有效地缩短了渲染总时间。如 1 920 × 1 080 高复杂度下 5 进程并行渲染时,并行加速比为 4.736,效率为 94.72%。另外从图 4 可以看出,不论图像大小和复杂度高低,只要算法复杂度满足网络传输承受能力的最低门限,并行渲染都可以保持高效率,并且稳定一致。从图 4 可以看出复杂度增大时,更有利于提高并行性能。

### 4.2 多节点 PC 构成的 PC 集群

多节点 PC 构成的集群并行渲染测试时,集群中共 5 台 PC 机,每台 PC 中可以取一个或多个的节点参与渲染运算,一个 PC 中实际上有 4 个核(节点),所以集群中最多可有 20 个节点并行。为了便于和前面的结果进行比较,测试中所取的总渲染节点数最多是 5 个。对 360 帧图像进行渲染,从单机单节点渲染逐步到 5 节点并行渲染,测试结果如表 7 所示。分别对两种标准尺寸的图像进行测试,720 × 576,1 280 × 720,每种尺寸的图像有两级复杂度,表中列出了不同情况下的并行加速比。表中 D 表示显示,R 表示渲染,L 表示低复杂度,H 表示高复杂度。

从表 7 和图 5 的结果可以看出,与单节点 PC 构成的集群相比,多节点 PC 构成的集群的并行渲染性能明显降低,并行

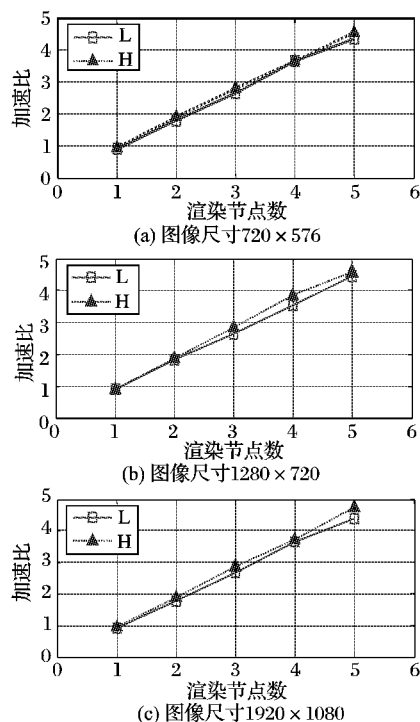


图4 单节点 PC 构成的 PC 集群的并行加速比

表6 单节点 PC 构成的集群并行渲染测试

图像尺寸	算法	PC0	R, D	D	D	D	D	D + R
	复杂度	PC1		R	R	R	R	R
		PC2			R	R	R	R
		PC3				R	R	R
		PC4					R	R
720 × 576	L	时间/s	90.187	100.826	49.943	33.940	24.728	20.665
		加速比		0.894	1.806	2.657	3.647	4.364
	H	时间/s	180.031	193.117	94.074	64.470	48.864	39.394
		加速比		0.932	1.924	2.792	3.677	4.570
1280 × 720	L	时间/s	200.531	219.612	109.625	75.690	56.602	45.202
		加速比		0.913	1.829	2.649	3.543	4.436
	H	时间/s	407.953	439.608	218.692	142.520	106.430	89.073
		加速比		0.927	1.865	2.862	3.833	4.580
1920 × 1080	L	时间/s	448.703	492.774	250.646	166.750	122.888	101.720
		加速比		0.911	1.790	2.690	3.651	4.411
	H	时间/s	904.500	947.702	475.756	314.530	214.264	190.993
		加速比		0.954	1.901	2.876	3.749	4.736

表7 多节点 PC 构成的集群并行渲染测试

图像尺寸	算法	PC0	R, D	D	D	D + R	D	D + R
	复杂度	PC1		R	R + R	R + R	R + R	R + R
		PC2					R + R	R + R
		PC3						
		PC4						
720 × 576	L	时间/s	90.187	100.826	93.062	60.910	46.251	38.693
		加速比		0.894	0.969	1.481	1.950	2.331
	H	时间/s	180.031	193.117	187.449	125.910	95.289	76.561
		加速比		0.932	0.960	1.430	1.998	2.351
1280 × 720	L	时间/s	200.531	219.612	225.187	156.970	113.863	92.475
		加速比		0.913	0.786	1.275	1.761	2.168
	H	时间/s	407.953	439.608	464.892	309.080	232.168	189.617
		加速比		0.927	0.876	1.319	1.757	2.151

加速比约下降了一半。最为明显的性能下降发生在2节点渲染时,与1节点渲染相比,性能几乎没有提高。这很可能是因

为两个进程共用显存、内存以及网卡,造成数据等待<sup>[6]</sup>。如1920×1080高复杂度下5进程并行渲染时,并行加速比为2.151,效率为43.02%。多节点PC构成的集群进行并行渲染不利于发挥最大并行效率,所以用单CPU PC机来构建并行渲染系统更为经济实用,即使网络性能不高的情况下,也能有效地提高渲染性能。

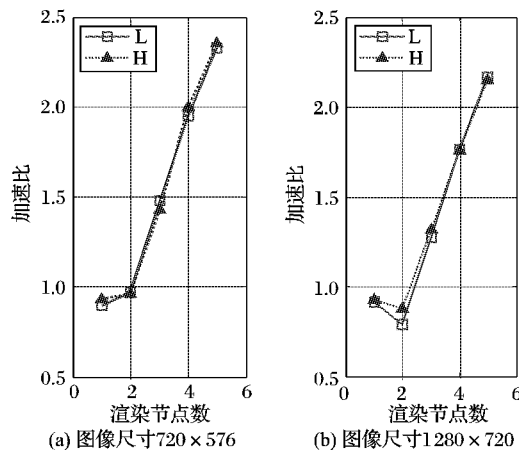


图5 多节点 PC 构成的 PC 集群的并行加速比

另外从图4和5看,不论图像大小和复杂度高低,只要算法复杂度满足网络传输承受能力的最低门限,并行渲染性能都能稳定一致。

## 5 结语

本文对基于OpenGL和MPI的并行渲染系统进行了较为全面的测试和分析,包括网络性能、渲染复杂度及并行方式之间的关系和影响,数据结果和分析报告对构建经济实用的PC集群并行渲染系统提供了有价值的实用参考。另外也验证了基于PC集群的并行渲染系统在合理规划下能够有效地提高渲染效率,并行性能稳定一致。

### 参考文献:

- [1] 杨菲,张小龙. 基于OpenGL的三维地形场景绘制及并行方法探讨[J]. 现代计算机, 2005(10): 105-108.
- [2] 徐永志,李利军. 地形场景的并行绘制及多通道图形输出[J]. 计算机工程, 2005, 31(8): 180-181.
- [3] YUETAL W. Parallel finite difference time domain method [M]. Norwood: Artech House, 2006.
- [4] 白建军,朱亚平,梁辉,等. OpenGL三维图像设计与制作[M]. 北京:人民邮电出版社,1999.
- [5] 都志辉. 高性能计算并行编码技术—MPI并行程序设计[M]. 北京:清华大学出版社,2001.
- [6] QUINN M J. MPI与OpenMP并行程序设计[M]. 陈文光,武永卫,译. 北京:清华大学出版社,2004.
- [7] PACHECO P. Parallel programming with MPI[M]. San Francisco: Morgan Kaufmann, 1997.