# Journal of System Simulation

Volume 27 | Issue 8

Article 22

8-3-2020

# Group-based Image Compositing in Heterogeneous Environments

Liu Ning

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;2. University of Chinese Academy of Sciences, Beijing 100049, China;

Dengming Zhu 1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;

Zhaoqi Wang 1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;

Wei Yi

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; ;

Follow this and additional works at: https://dc-china-simulation.researchcommons.org/journal

Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

# Group-based Image Compositing in Heterogeneous Environments

## Abstract

Abstract: Image compositing requires dense inter-node communications and could easily become the bottleneck of a parallel visualization system. Conventional image compositing algorithms assume the environment is homogeneous and take no heterogeneity into consideration. In a heterogeneous environment, this assumption is likely to cause unreasonable load distribution among the nodes. *To address this problem, we present a group-based image compositing algorithm in heterogeneous environments. It is able to distribute the workloads among the nodes according to computing and networking characteristics of the heterogeneous environments. We conduct a comprehensive evaluation on the ns-3 network simulator. Experiment results shows that when the environment has great heterogeneity, our algorithm gains significant speedup over conventional methods.* 

## Keywords

image compositing, heterogeneous environment, parallel visualization, ns-3 simulator

# **Recommended Citation**

Liu Ning, Zhu Dengming, Wang Zhaoqi, Wei Yi. Group-based Image Compositing in Heterogeneous Environments[J]. Journal of System Simulation, 2015, 27(8): 1801-1808.

第27卷第8	期
2015年8月	

# 异构环境下基于分组的图像合成算法

刘宁<sup>1,2</sup>,朱登明<sup>1</sup>,王兆其<sup>1</sup>,魏毅<sup>1</sup>

(1.中国科学院计算技术研究所,北京 100190; 2.中国科学院大学,北京 100049)

**摘要:**图像合成由于需要密集的节点间通信,因此很容易成为整个并行可视化系统的瓶颈部分。传统的图像合成算法均假定环境是同构的,这种假定在异构环境下可能会造成严重的负载不均衡。*针对这个问题,提出了一个异构环境下基于分组的图像合成算法。它能够根据环境的计算与网络特性更合理地进行任务分配。*在 ns-3 网络模拟器上进行了详尽的实验评估。实验结果表明当环境存在 比较大的异构性时,该方法相对于传统图像合成算法能够取得显著的加速比。

关键词:图像合成;异构环境;并行可视化; ns-3 网络模拟器

中图分类号: TP391 文献标识码: A 文章编号: 1004-731X (2015) 08-1801-09

#### Group-based Image Compositing in Heterogeneous Environments

Liu Ning<sup>1,2</sup>, Zhu Dengming<sup>1</sup>, Wang Zhaoqi<sup>1</sup>, Wei Yi<sup>1</sup>

(1.Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; 2.University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Image compositing requires dense inter-node communications and could easily become the bottleneck of a parallel visualization system. Conventional image compositing algorithms assume the environment is homogeneous and take no heterogeneity into consideration. In a heterogeneous environment, this assumption is likely to cause unreasonable load distribution among the nodes. *To address this problem, we present a group-based image compositing algorithm in heterogeneous environments. It is able to distribute the workloads among the nodes according to computing and networking characteristics of the heterogeneous environments. We conduct a comprehensive evaluation on the ns-3 network simulator. Experiment results shows that when the environment has great heterogeneity, our algorithm gains significant speedup over conventional methods.* 

Keywords: image compositing; heterogeneous environment; parallel visualization; ns-3 simulator

# 引言

并行可视化<sup>[1-3]</sup>是分析由科学模拟计算产生的 大规模数据的一种有效方法。Monar<sup>[4]</sup>将并行绘制 算法分为 sort-first、sort-middle 和 sort-last 三种, 其中 sort-last 方法由于简单易行、良好的负载均衡



收稿日期: 2015-06-04 修回日期: 2015-07-02; 基金项目: 国家自然科学基金(61173067, 61379085): 作者简介: 刘宁(1987-), 男,博士生,研究方向为科 学数据可视化:朱登明(1973-),男,博士,副研究员, 研究方向为自然现象模拟和数据可视化; 王兆其 (1966-), 男,博士,研究员,研究方向为虚拟现实和 智能人机交互。 特性和良好的可扩展性而得到了广泛的应用。

在 sort-last 类型的并行可视化系统中,数据首 先被分发到各个节点上去,然后各节点执行局部 渲染得到一个部分结果。这些部分结果会在图像 合成阶段被合成为最终结果。对于图像合成来 说,每个节点都需要直接或者间接与其它所有节 点进行通信,因此它很容易成为整个系统中的瓶 颈部分。

在同构环境下,已经对图像合成问题进行了 大量的研究,其中比较有代表性的算法包括:

http://www.china-simulation.com

第 27 卷第 8 期	系统仿真学报	Vol. 27 No. 8
2015年8月	Journal of System Simulation	Aug., 2015

direct send<sup>[5]</sup>, binary swap<sup>[6]</sup>, 2-3 swap<sup>[7]</sup>, radix-k<sup>[8]</sup> 和 pipelining image compositing<sup>[9]</sup>。这些算法均假 定环境是同构的,即节点计算能力、链接带宽与 链接延迟都是相同的。这个假定会导致各节点分 配到的任务量是相同的。这适用于传统的计算环 境,如集群和超级计算机,但对于一个异构环境 来说,它很可能会导致严重的负载不均衡。例如 一个快速的链接会承载与一个慢速链接相同的流 量,一个快速的节点会处理与一个慢速节点相同 的数据。这显然是不合理的。

在本文中我们提出了一个异构环境下基于分 组的图像合成算法。分配到各节点组的任务量是 由环境的计算特性与网络特性所决定的。首先我 们形式化地定义了这个问题,并且提出了相应的 耗费函数与目标函数。为了计算给定环境下的最 优合成方案,我们实质上需要确定一系列介于 0 与 1 之间的实值分割点。我们提出了一个最优化 方法,它可以在多项式时间内计算出给定环境下 的最优合成方案。经过我们方案优化之后,快速 的链接更有可能会传输更多的流量,而快速的节 点会更有可能处理更多的数据。因此,我们的算 法能够有效提高异构资源的使用效率。

为了验证本文方法的有效性,我们在 ns-3<sup>[10]</sup> 网络模拟器上进行了详尽的实验。我们从不同方 面对我们的方法与经典的合成算法 binary swap 进 行了比较,如节点计算能力、链接带宽、链接延 迟、图像空间大小和参与节点的数目等。当环境 存在较大异构性时,我们的方法相对于 binary swap 能够取得显著的加速比。

在本文的后续章节中,我们将在第 1 部分介 绍相关工作,在第 2 部分形式化地定义异构环境 下基于分组的图像合成问题,并在第 3 部分提出 能够在多项式时间内计算出给定环境下最优合成 方案的优化算法。我们在 4 部分给出并详细分析 与 binary swap 比较的实验结果。最后,第5部分总 结了我们的工作并指出了未来工作的研究方向。

# 1 相关工作

图像合成算法可以分为两类:基于分组的和 基于流水线的。基于分组的合成算法主要包括: direct send<sup>[5]</sup>, binary swap<sup>[6]</sup>, 2-3swap<sup>[7]</sup>和 radix-k<sup>[8]</sup>。 参与节点被划分为若干个节点组,只有属于同一 个节点组内的节点相互才能通信。分组方案一般 会随着合成过程的推进而演化。这类算法通常是 利用图像合成操作符的结合律来减少节点间通信 的次数。

Direct send<sup>[5]</sup>可以算是最简单的合成算法。它 将图像空间均匀地划分,然后每个节点负责其中 一部分。它只包含一个合成阶段,每个节点会将 它持有的其它部分的数据直接发送到对应的节点 中去。这个算法主要的缺点是当参与节点数目比 较多时会造成网络阻塞,因为它需要执行一个 all-to-all 的通信过程。

Binary swap<sup>[6]</sup>是一个受到广泛应用的合成算 法。最开始节点被划分为两组,其中一组负责图 像空间的上半部分,另一组负责图像空间的下半 部分。然后这两组会被递归地划分为更小的节点 组。在任一个合成阶段,每个节点都会且只会与 一个节点通信,因此不会有网络阻塞的问题。当 节点数目比较多时,该方法能够比 direct send 更 有效地利用网络资源。由于数据交换模式的特 点,该算法将节点数目限制为2的幂次方。

2-3 swap<sup>[7]</sup>对 binary swap 进行了扩展,使得它可以处理任意数目的节点。在每个阶段,一个节点最多可以与两个节点通信,而不是像 binary swap 所规定的一个。除此之外,2-3 swap 的合成过程与 binary swap 十分相似。

Radix-k<sup>[8]</sup>是一个可以配置的合成算法,它统 一了 direct send 与 binary swap 算法。在每个阶 段,它允许一个节点与多个节点进行通信,因此 能够重叠通信与计算来提高合成效率。同时,每 个节点所通信的节点的数目通常远小于 direct send,因此又能避免网络阻塞问题。因此它通常

第 27 卷第 8 期		Vol. 27 No. 8
2015 年 8 月	刘宁, 等: 异构环境下基于分组的图像合成算法	Aug., 2015

能取得比 direct send 和 binary swap 更好的性能。

基于管线的合成算法主要包括 pipelining image compositing<sup>[9]</sup>。该算法将节点组织为一个管线,部分渲染结果在这个管线中流动并逐渐融合为最终结果。它支持合成结果的渐进式显示。相对于最终结果显示时间来说,它的性能要优于 binary swap。因此它更适用于远程可视化的场景。

上述所有的算法均假定环境是同构的。因此 每个节点会承担相同的任务量。在一个异构环境 下,这种假定很可能会造成比较严重的负载不均 衡。与之不同,我们的算法会根据环境计算与网 络特性调整各节点的任务量,使得快速的节点更 可能处理更多的数据,快速的链接更可能传输更 多的流量。因此我们的方法能够更有效地利用异 构环境的计算与网络资源。

## 2 问题定义

一共有 N 个节点,每个节点会被分配一个 rank 号 *i*,其中1≤*i*≤N。如果*i*<*j*,那么认为 节点*i*在节点*j*前面。局部渲染阶段之后,每个节 点持有一个部分结果。在图像合成阶段,这些部 分结果经由图像合成操作符融合为最终结果。

#### 2.1 图像合成概述

假设一共有N个节点,其中节点i持有一个部 分结果 x<sub>i</sub>。图像合成的目的是将这些部分结果融 合以得到一个最终结果 x<sub>F</sub>,即

$$x_F = x_1 \otimes x_2 \otimes \dots \otimes x_N \tag{1}$$

在这里⊗表示一个图像合成操作符。图像合成操 作符通常符合结合律而不符合交换律。在图像合 成领域最常用的一个合成操作符是 over 操作 符。最终合成结果 x<sub>F</sub> 可以驻留在一个或者多个 节点上。

#### 2.2 异构环境下基于分组的图像合成

基于分组的图像合成是将那些部分结果融合 为最终结果的一种特定方式。它利用合成操作符 的结合律来减少节点之间数据交换的次数。参与 节点被分为若干组,只有属于同一组内的节点可 以交换数据。最开始的时候,所有的节点均属于 同一个组。随着合成过程的推进,这些节点组被 递归地分割为一系列更小的组。

一个 节点组 $S = \{i_1, i_2, \dots, i_m\}$ 是由m个节点组成的节点集合,其中 $s, t \in [1, m]$ ,并且如果s < t则 $i_s < i_t$ 。异构环境下基于分组的图像合成可以表示为一系列节点组的分割。节点组S上的一个分割P定义为

$$P(S) = (S_U, S_L, \lambda) \tag{2}$$

在这里  $S_U = \{i_1, i_3, \dots, i_{m-1}\}$  是节点组 S 中负责图像空间上半部分的节点组成的集合,  $S_L = \{i_2, i_4, \dots, i_m\}$  是由节点组 S 中负责图像空间下半部分的节点组成的集合,  $\lambda$  是一个介于 0 和 1 之间的实数,它代表  $S_U$  所负责的图像空间大小与 S 所负责图像空间 大小的比例。

在节点组 *S* 中,两个相邻的节点会组成一个 节点对并相互交换数据。之后,其中一个节点会 被分配到 *S*<sub>U</sub> 中,另外一个节点会被分配到 *S*<sub>L</sub> 中。自此之后,*S*<sub>U</sub> 中的节点与 *S*<sub>L</sub> 中的节点不会 再有数据交换。图1是异构环境下8个节点基于分 组的图像合成的示意图。



图 1 异构环境下基于分组的图像合成示意图

# 3 我们的方法

#### 3.1 符号

首先我们在表 1 列举了本文中使用的符号。 链接带宽 B、链接延迟 L 和节点计算能力 C 都是给 定的全局常量,可以在任意位置使用它们,而且 它们通常会从函数的参数列表中省略。

第 27 卷第 8 期	系统仿真学报	Vol. 27 No. 8
2015 年 8 月	Journal of System Simulation	Aug., 2015

表 1 本文中使用的符号		
符号	含义	
B(p,q)	节点 p 与节点 q 之间的链接带宽	
L(p,q)	节点 p 与节点 q 之间的链接延迟	
C(p)	节点 p 的计算能力	
S	一个节点集合	
Ι	图像空间大小	
Р	一个节点集合的划分	
$T_p(S,I,P)$	S针对I和P的最优合成时间	
$T_{opt}(S,I)$	S针对 I的最优合成时间	
$t_p(S,I,P)$	S针对I和P的分割时间	

#### 3.2 图像合成耗费函数

给定图像空间大小 *I*, 节点组 *S* 的最优合成方案是其所有可能分割中合成时间最小的, 即,

$$T_{opt}(S,I) = \min T_P(S,I,P)$$
(3)

在这里*T<sub>p</sub>(S,I,P*)是节点组 *S*关于图像空间大小 *I*与分割 *P*的最优合成时间,它定义为

$$T_{P}(S, I, P) = t_{p}(S, I, P) + \max(T_{opt}(S_{U}, \lambda I), T_{opt}(S_{L}, (1 - \lambda)I))$$
(4)

在这里 *t<sub>p</sub>*(*S*,*I*,*P*) 是节点组 *S* 关于图像空间大小 *I* 与分割 *P* 的分割时间,它定义为

$$t_{p}(S, I, P) = \max_{i_{s} \in S_{U}} (\max(L(i_{s+1}, i_{s}) + \frac{\lambda I}{B(i_{s+1}, i_{s})} + \frac{\lambda I}{C(i_{s})}),$$
  
$$L(i_{s}, i_{s+1}) + \frac{(1 - \lambda)I}{B(i_{s}, i_{s+1})} + \frac{(1 - \lambda)I}{C(i_{s+1})}))$$
(5)

在同一个节点组内,相邻的节点会进行配对并交换 数据。两个配对的节点会被分配到不同的节点组, 因此在后续合成阶段中不会再有数据交换。

#### 3.3 最优化方法

首先*t<sub>p</sub>(S,I,P*)可以表示为下面两部分中较大的一个:

$$\max_{i_{s}\in S_{U}}(L(i_{s+1},i_{s})+\frac{\lambda I}{B(i_{s+1},i_{s})}+\frac{\lambda I}{C(i_{s})})$$
(6)

$$\Re \max_{i_s \in S_U} (L(i_s, i_{s+1}) + \frac{(1-\lambda)I}{B(i_s, i_{s+1})} + \frac{(1-\lambda)I}{C(i_{s+1})})$$
(7)

相对于 $\lambda$ ,公式(6)是非递减的,而公式(7)是非递 增的。因此, $t_p(S,I,P)$ 的整体形状如图 2 所示。 相对于 $\lambda$ ,它存在一个全局最小值。



图 2  $t_p(S, I, P)$  示意图

其次,对于 $\lambda$ 来说, $T_{opt}(S_U,\lambda I)$ 是非递减的,而 $T_{opt}(S_L,(1-\lambda)I)$ 是非递增的。因此,这两者的最大值 max( $T_{opt}(S_U,\lambda I), T_{opt}(S_L,(1-\lambda)I)$ )的形状与图 2 近似。

最后,*T<sub>P</sub>(S,I,P*)是上述两个函数之和,它的 形状如图 3 所示。由于函数的单调性,最优的 *λ* 一 定是 *λ*<sub>1</sub> 或者 *λ*<sub>2</sub>。具体的 *λ*<sub>1</sub> 与 *λ*<sub>2</sub> 可由基于二分查找 的算法进行求解。





#### 3.4 时间复杂度分析

令 *F*(*n*) 表示计算 *T<sub>opt</sub>*(*S*,*I*) 的时间复杂度,令 *f*(*n*) 表示计算 *T<sub>p</sub>*(*S*,*I*,*P*) 的时间复杂度。根据公式
(3)和公式(4),我们有

$$F(n) = Mf(n)$$
  

$$f(n) = O(n) + 2F(n/2)$$
(8)

这里 *M* 是根据相应的误差阈值在求解相应最优 *λ* 时的最大迭代次数。因此,

$$F(n) = O(n) + 2MF(n/2)$$
 (9)

根据主定理,

$$F(n) = \Theta(n^{\log_2 2M}) \tag{10}$$

http://www.china-simulation.com

第 27 卷第 8 期 2015 年 8 月

## 4 结果

我们在 ns-3 网络模拟器上进行了详尽的实验 验证。ns-3 是一个被广泛使用的离散事件网络模拟 器,主要用于教育和科研目的。实验中参与节点的 数目从 8 到 64 个不等。

#### 4.1 环境搭建

所有节点通过 Point-to-Point 链接直接相连。 链接的带宽从带宽范围 BR 中随机选取,链接的延 迟从延迟范围 LR 中随机选取,节点的计算能力从 计算能力范围 CR 中随机选取。实验中用到的主要 控制参数如表 2 所示。

	表 2	实验中主要的控制参数
符号		含义
Nodes		参与节点的数目
Ι		图像空间大小(MB)
BR		链接带宽范围(Mbps)
LR		链接延迟范围(s)
CR		节点计算能力范围(MBps)

我们使用 ns-3 模拟器提供的 OnOffApplication 来产生网络流量。OnOffApplication 在 OnTime 时 间段内持续产生 UDP 数据包,而在 OffTime 时间 段内不产生数据流量。在实验过程中,我们根据图 像空间的大小来设定 OnTime,同时将 OffTime 设 为 0。OnOffApplication 产生数据的速度由相应网 络链接的带宽来决定。

在实验过程中,我们的算法与经典的图像合成 算法 binary swap 进行比较。Binary swap 算法的合 成方案是确定的。我们在确定环境的计算与网络特 性之后,计算出该环境下最优的合成方案。然后我 们将 binary swap 的合成方案与我们方法的合成方 案分别在 ns-3 模拟器上运行,并将实验结果记录 下来,以进行后续的分析。

对于每一组控制参数,我们运行 R 次 (R 通常 取 50)来减少随机性带来的影响。对于第 i 轮模拟 实验,其加速比 s<sub>i</sub>定义为

$$s_i = \frac{t_{bs}^i}{t_{our}^i} \tag{11}$$

其中, *t*<sup>*i*</sup><sub>*bs*</sub> 是第 *i* 轮中 binary swap 的合成时间, *t*<sup>*i*</sup><sub>*our*</sub> 是第 *i* 轮中我们方法的合成时间。每一轮实验由相应的*实验 ID* 标识。相应地,这 *R* 轮实验的平均加速比定义为

$$\frac{1}{R}\sum_{i=1}^{R}s_i \tag{12}$$

#### 4.2 图像空间大小

不同图像空间大小的结果如图 4 所示。*I* 取值 分别为 1MB、4MB 和 16MB,其它控制参数分别 设为 Nodes=32,*CR*=[1,1000],*BR*=[0.01,100], *LR*=[e-5,0.1]。随着图像空间大小逐渐增大,我们 方法相对于 binary swap 的平均加速比也在增加。 这是因为当图像空间增大时,更多的数据需要被传 输和处理。因此,我们的方法有更大的优化空间。

#### 4.3 节点数目

不同节点数目的结果如图 5 所示。节点数目分 别为 8,16 和 32,其它控制参数分别为 *I*=4MB, *CR*=[1,1000],*BR*=[0.01,100],*LR*=[e-5,0.1]。当 节点数目增加时,在一个合成阶段将会包含更多的 节点组。每个合成阶段的合成时间由这个阶段中最 慢的一个节点组决定,因此 binary swap 在这种情 况下更容易受到慢速节点与慢速链接的影响。我们 的方法能够为这些慢的节点组分配更少的任务量, 因此能够有效减少这些慢速节点组的影响,从而提 高图像合成效率。

# 4.4 链接延迟

网络链接的结果如图 6 所示。链接延迟范围 LR 被置为[0.01, 1],其它控制参数分别设为 Nodes=64,I=4MB,CR=[1000,1000],BR=[1000, 1000]。节点计算能力和链接带宽被设为一个比较 大的值,使得它们对最终合成时间的影响可以忽略 不计。我们的方法只会调整分配到各节点的任务 量,但不会改变节点间的通信模式。对于给定的节 点间通信模式,链接延迟再来的耗费是恒定且不可 避免的。因此在这种情况下,我们的方法与 binary swap 算法的性能基本是一致的。



系统仿真学报 Journal of System Simulation

Vol. 27 No. 8 Aug., 2015



http://www.china-simulation.com





#### 图 6 链接延迟的结果

#### 4.5 链接带宽

使用不同类型的链接带宽的结果如图 7 所示。 BR 分别设为[0.01, 1], [0.01, 10], [0.01, 100], 其 它 控 制 参 数 分 别 设 为 Nodes=32, *I*=4MB, *CR*=[1000, 1000], *LR*=[0.001, 0.001]。当链接带宽 范围增大时,各链接所分配到的链接带宽的变化范 围也倾向于增加,此时环境会变得更加异构。由于 我们的方法能够根据环境计算出相应的最优的合 成方案,因此环境异构性变大时,我们的方法能够 取得更大的加速比。



图 7 不同链接带宽的结果

#### 4.6 链接带宽与链接延迟

链接带宽与链接延迟共同影响的结果如图 8 所示。链接带宽范围 BR 被置为[0.01, 100],链接 延迟范围 LR 分别被置为[0.001, 0.001], [0.5, 0.5], [1, 1],其它控制参数分别被置为 Nodes=32, I=4MB, CR=[1 000, 1 000]。节点计算能力被设为 一个比较大的值,使得它对最终合成时间的影响可 以忽略不计。链接带宽带来的耗费与所传输的数据 量成正比,而链接延迟带来的耗费与传输的数据量 无关,是固有且不可避免的。我们的方法可以调整 分配到各节点的任务量,但不会改变节点间的通信 模式,因此不会对延迟带来的耗费产生影响。在每

第 27 卷第 8 期	系统仿真学报	Vol. 27 No. 8
2015 年 8 月	Journal of System Simulation	Aug., 2015

组模拟实验中,链接延迟均被置为一个常量。对于 第一组模拟实验,延迟被置为一个比较小的值 (1ms),此时带宽对整个合成时间起决定性的作用。 由于 *BR* 范围较大,此时我们方法相对于 binary swap 取得的加速比较大。当延迟越来越大并逐渐 对图像合成耗费起决定性作用时,由于我们的方 法不能改变延迟带来的耗费,因此相应的加速比 会越来越小。



图 8 链接带宽与链接延迟共同作用的结果

# 5 结论

在本文中我们提出了一个异构环境下基于分 组的图像合成算法。首先我们对于这个问题进行了 形式化的定义并提出了相应的耗费模型。然后我们 给出相应的最优化算法,它能够在多项式时间内计 算出相应环境的最优合成方案。经过我们方法的优 化,各节点与链接所分配到的任务量会根据环境相 应的计算与网络特性进行调整,而不是像传统合成 算法那样进行平均分配,因此我们的方法能够更有 效地利用异构环境的计算资源和网络资源。我们在 ns-3 模拟器上进行了详尽的验证,并从图像空间大 小、节点数目、链接带宽和链接延迟等多方面对实 验结果进行了分析。结果表明当环境呈现出比较大 的异构性时,我们的方法相对于经典图像合成算法 binary swap 能够取得显著的加速比。 现在我们将节点的数目限制为2的幂次方。将 来我们准备研究异构环境下针对任意节点数目的 图像合成算法。

# 参考文献:

- Bentes C., Labronici B. B., Drummond L. M., Farias R. Towards an efficient parallel raycasting of unstructured volumetric data on distributed environments [J]. Cluster Computing (S1386-7857), 2014, 17(2): 423-439.
- [2] 单桂华,田东,谢茂金,等.千万亿次科学计算的原位 可视化[J].计算机辅助设计与图形学学报,2013, 25(3):286-293.
- [3] 丁治宇,陈海东,吴斐然,等.多变量空间数据场可视化综述[J]. 计算机辅助设计与图形学学报,2013,25(11):1597-1605.
- [4] Molnar S., Cox M., Ellsworth D., Fuchs H. A sorting classification of parallel rendering [J]. IEEE Computer Graphics and Applications (S0272-1716),1994, 14(4): 23-32.

(下转第1814页)

http://www.china-simulation.com