

基于网格的散乱点曲面重构技术

冯建周, 孔令富

(燕山大学信息科学与工程学院, 秦皇岛 066004)

摘要: 提出一种利用网格技术实现散乱点曲面重构的方法, 设计适合网格并行的松耦合分层重构算法, 在此基础上研究基于 Agent 技术的智能任务分解机制和基于 Condor-G 技术的任务分配与调度策略, 并搭建网格实验平台进行测试。对重构结果的分析表明, 该技术可提高基于散乱点的曲面重构速度, 降低应用成本。

关键词: 曲面重构; 网格; 并行算法; 任务分解; 任务调度

Surface Reconstruction Technology of Dense Scattered Points Based on Grid

FENG Jian-zhou, KONG Ling-fu

(School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004)

【Abstract】 This paper proposes a new method which uses grid technology to achieve a surface reconstruction of scattered points. It designs for grid parallel loosely coupled hierarchical algorithm. Based on this, study intelligent task decomposition mechanisms based on the Agent technology and the distribution of tasks and schedule strategy based on Condor-G technology. It structures the grid experimental platform for test and analyzes the reconstruction results. Results show that the technology can improve surface reconstruction speed of dense scattered points and reduce cost application.

【Key words】 surface reconstruction; grid; parallel algorithm; task decomposition; task schedule

1 概述

逆向工程中的散乱点曲面重构就是从实体表面提取散乱点集, 在保持正确拓扑情况下, 尽可能精确地恢复成原曲面^[1]。随着激光扫描设备的发展, 包含被测物体更多细节的海量数据获取成为可能, 并成为高精度测量建模的发展方向, 但同时也存在重构速度过慢、不能适应实时性要求的问题。

网格计算^[2](grid computing)是针对复杂科学计算的一种新型计算模式, 它使用标准、开放、通用的协议和界面, 通过高速网络连接, 将地理上分布的各种异构的计算系统、软件系统, 存储系统、数字化仪器设备和控制系统等各种资源动态的整合成一体, 构成一台虚拟的超级计算机, 充分发挥和利用系统中各种资源的计算和服务能力, 实现资源的共享和协作, 并向用户提供一致的接口和高性能服务。本文将网格技术应用于曲面重构领域, 整合分散的计算资源为曲面重构服务, 通过设计适合的重构算法, 灵活的任务分解调度策略, 提高了重构速度, 从而方便用户使用。

2 曲面重构并行算法及其实现

2.1 基于轮廓线的并行算法设计

为提高基于散乱点的曲面重构速度, 研究人员相继提出一些并行算法, 如 Delaunay 并行算法^[3], 但这些算法都是基于集群的紧耦合算法, 子任务间涉及大量的数据通信, 不适合网格这样的结构松散的分分布式系统。为适应网格的特点, 必须设计子任务间更加独立的粗粒度的松散耦合并行算法。

基于轮廓线^[4]的曲面重构的基本原理是假设 2 个相邻平行切片上各有一轮廓线, 所谓轮廓线就是物体与某一特定平

面的交线, 它在相当大的程度上表述了实体在这一平面上的形状特征, 上轮廓线上的采样点列为 $P=\{P_0, P_1, \dots, P_m\}$, 下轮廓线上的采样点列为 $Q=\{Q_0, Q_1, \dots, Q_n\}$, 如果用直线把上下轮廓线上的采样点依次连接起来形成一个封闭的三角形带, 就得到一个过轮廓线的曲面的近似多边形表示。每一个直线段 P_iP_{i+1} 或 Q_jQ_{j+1} 称为轮廓线线段。连接上轮廓线上一点与下轮廓线上一点的线段称为跨距。显然, 一条轮廓线线段, 以及将该线段 2 个端点与相邻轮廓线上的 1 点相连的 2 段跨距构成 1 个三角面片, 称为基本三角面。实现 2 条轮廓线之间的曲面重构就是要用一系列相互连接起来的三角面片将上、下 2 条轮廓线连接起来。

综上所述, 由于各层数据可以独立的构造轮廓线, 因此基于分层数据集的轮廓线重构算法具有很好的并行性, 可以方便地实现松耦合的数据并行。在轮廓线重构算法的基础上, 设计并行轮廓线算法, 其基本思想如下: 采用数据并行机制, 首先将无组织数据集转换为分层数据集, 然后由网格的任务分解模块划分为若干个子集(每个子集拥有多个切片数据), 即将重建过程分为若干个子任务, 再由网格的调度模块分配计算节点, 然后采取 Master-Slave 模型, 由参与并行计算的

基金项目: 国家“863”计划基金资助项目(2006AA04Z212); 河北省自然科学基金资助项目(F2007000423)

作者简介: 冯建周(1978 -), 男, 讲师、博士研究生, 主研方向: 网格计算; 孔令富, 教授、博士、博士生导师

收稿日期: 2009-01-05 **E-mail:** fengjianzhou@tom.com

节点根据轮廓线的曲面重构思想分别处理其中的部分子集，将处理后的结果返回到 Master 节点，由 Master 节点按合并规则将 Slave 节点及本节点处理的结果进行合并，最后由 Master 节点完成绘制并在此节点上进行可视化显示。在重构过程中，采用分布存储策略，即主进程首先将完整的数据文件分发给各子进程节点，在每个参与计算的节点上都存有一份完整的数据文件，因此，主进程 Master 分发任务时，只需将任务的编号信息发给子进程，子进程根据任务编号信息获取数据的起始位置和结束位置，在本节点上读取将要处理的数据块。由于每个节点计算所得的结果即为最终显示图像的一部分，各子进程间无需在重构过程中进行过多通信，因此加快了并行算法的速度。算法流程如图 1 所示。

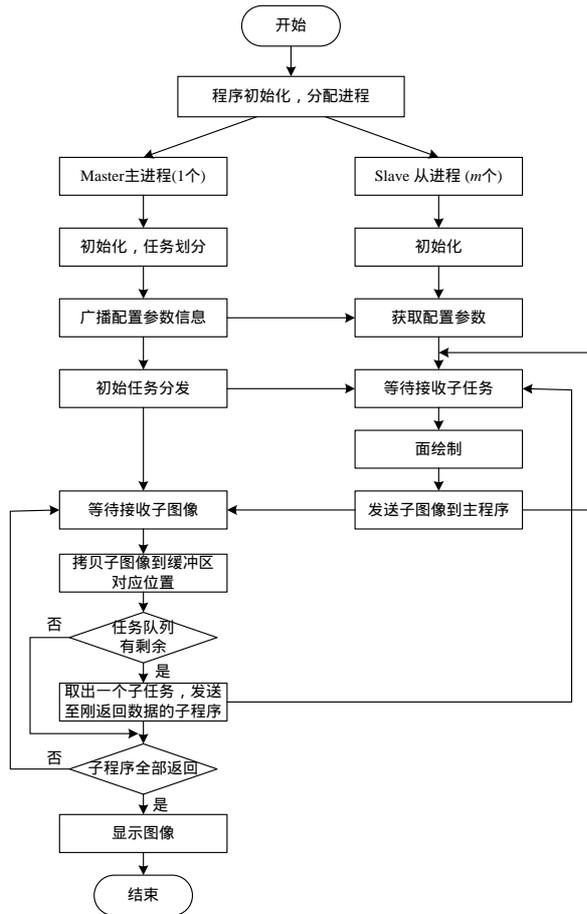


图 1 并行重构算法流程

2.2 网络环境下的并行算法实现

本文基于 Globus Toolkit4.0.1 搭建了网络实验平台，该平台由清华同方探索者 108 高性能集群(包括 8 个计算节点和一个管理节点)和 10 台 PC 资源构成，形成了 2 个网格域。基于网络的并行算法实现不同于简单的并行程序执行，由于网络资源具有异构、动态、分散等特点，所以在算法实现过程中要充分考虑这些因素，制定相应的策略。

2.2.1 任务分解

基于网络的并行任务分解有 2 种方式：一种是手动划分方式，即根据用户的经验和对数据集的了解手动划分任务集，决定并行度，这种方法操作简单，但对用户的要求较高；另一种方式是由网络系统自动划分任务集，这种方式需要网络系统为用户提供智能的任务分解机制，根据系统情况和任务情况进行分析，给出最佳的任务分解方案。

本系统采用混合型的任务分解机制，即允许用户手动选择任务并行度，也可以由系统自动选择。下文重点介绍基于 Agent 技术的自动任务分解模块，自动任务划分模块主要包括分解智能体(Agent)、关联知识库和应用接口 3 个部分。其中，分解智能体负责将用户提交的数据文件智能、高效地划分为可以调度的子任务；关联知识库主要保存网格系统的监测信息和对数据集划分的经验知识等关联信息，为任务划分提供参考；应用接口主要是与网格门户交互以获取用户参数。系统的工作过程如下：首先通过应用接口获取任务参数，包括数据文件名称，文件大小等，然后将这些信息发送到关联知识库中，关联知识库还要通过 Globus 的 MDS 获取系统相关信息，比如可用节点数、节点负载情况、各节点性能等信息，最后分解智能体通过关联知识库中的这些信息和已存储的相关知识，利用设计好的分解策略进行智能分解。任务分解模块如图 2 所示。

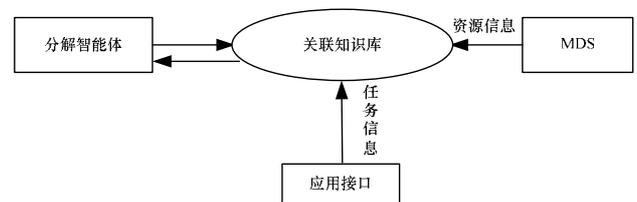


图 2 任务分解模块

在各子任务的规模上，为实现负载平衡，采用尽量等量原则，即如果数据集由 n 层数据组成，系统为数据集划分成 m 个子任务，则每个子任务的规模为 n/m ，若不能整除，则剩余部分 $r(r < m)$ 等分给前面的 r 个子节点。

2.2.2 任务分配与调度

在任务的分配与调度方面，设计了基于 2 级调度模型的优先级抢先调度策略。2 级调度模型即整个网格系统由管理整个网格的中央调度器和管理各子网格域的局部调度器组成。各子任务先提交到中央调度器，由中央调度器进行统一调度，分配到每个子网格域之后，再由局部调度器进行 2 次调度，最后分配到最合适的节点执行。优先级抢先机制是指将用户、任务和资源都设置优先级，不同用户提交的任务，根据用户优先级的不同先执行优先级高的用户的任务，同一用户的不同任务，则首先分配任务优先级高的任务，另外在任务和资源的匹配过程中，对资源也设计优先级，比如本地资源的优先级往往比远程资源的优先级高，CPU 占用率低的资源往往比 CPU 占用率高的资源优先级高等等，这样做主要是为了保证任务能与最优秀的资源进行匹配，最后形成了一个任务队列和一个资源队列，用户优先级和任务优先级高的任务首先和资源优先级高的资源匹配并进行分配。

在系统实现中本文采用了 Condor-G 技术^[5]，该技术是美国威斯康星大学研发的应用于计算密集型作业的资源管理系统 Condor 与 Globus 结合的产物，当网络环境是由多个具有不同类型的本地资源管理器管理的网格域组成时，Condor-G 便能起到作用，本系统的 PC 群和高性能集群分别采用 Condor 和 PBS 作为本地调度器，中央调度器选择 Condor-G，从而形成 2 级调度模型。另外通过设置用户优先级，任务优先级和资源优先级，充分利用 Condor 的广告(ClassAd)和匹配(MatchMaking)机制，实现了优先级抢先策略，同时 Condor-G 还支持任务迁移功能，即某一节点如果在运行过程中出现故障，则可以通过中央管理器的监听机制获悉情况，并将其任

务迁移到另一个可用节点执行。其分配调度模型如图 3 所示。

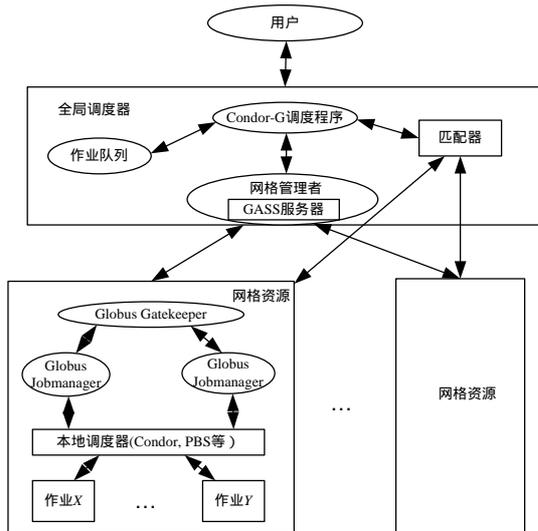


图 3 基于 Condor-G 的分配调度模型

2.2.3 并行重构与图象合并

在系统实现中，本文采用 VTK 和 Mpich-G2 结合的方法构建了并行重构环境，VTK^[6](Visualization Toolkit)是美国 Kitware 公司开发的一套 C++ 类库，是一个源代码开放、面向对象的计算机图形、可视化技术及图像处理软件系统，封装了目前许多优秀的三维数据场可视化算法，可方便地对数据集进行各种变换和操作。MPICH-G2^[7]是 MPI 和 Globus 结合的产物。它提供了与 MPI 完全一致的并行编程环境，使用 Globus 工具包提供的服务来支持在异构的网格环境中有效和透明地执行 MPI 程序。VTK 支持并行机制，将 VTK 和 Mpich-G2 结合，构建了网格并行重构环境。

在基于 VTK 的重构过程中，由于本文使用的数据都是无组织的散乱点数据，即只包含每个点的三维坐标的文本文件数据，因此须通过 VTK 中的 vtkGaussianSplatter 类和 vtkPDataSetWriter 类实现将无组织的散乱点数据转换为切片数据，然后通过 VTK 类库中的 vtkMultiProcessController 类来实现对多进程的执行控制，完成进程(线程)的安装和初始化操作，任务的分配以及远程方法调用 RMI 的注册、触发和终止；利用 vtkContourFilter 类来完成轮廓线的计算；最后通过 vtkAppendPolyData 类和 vtkPolyDataMapper 类来完成图像的合并和绘制工作。最后利用 VTK 提供的可视化类库和 VRML^[8]技术对图像进行远程可视化，包括设置显示窗口的大小、背景和位置等参数，而且还实现了可视化结果的任意旋转，从不同的角度观察重建后的图像。

3 实验结果及分析

文本采用的数据均来自 Interent，如图 4 所示为应用本文算法由 40 256 个数据点在不同并行度时重构出的兔子图象。由重构结果可见，不同并行度下的图象基本保持一致，可见本算法保证了重构精度。

另外，对于同一个数据集文件，使用不同的并行度，在一定的范围内，加速比随着并行度的增加而以不同的速率上升，但是当越过某一临界点后，加速比又呈下降趋势，如

图 5 所示，可见由于通信延迟等问题，并行度并非越高越好，而是存在一个最佳并行度，使此时的重构速度最快。

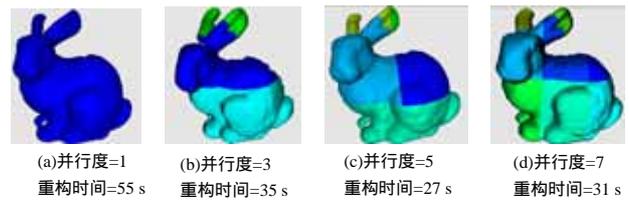


图 4 不同并行度下的重构结果

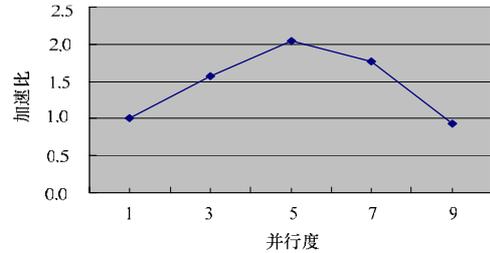


图 5 加速比分析

4 结束语

为提高散乱点曲面重构速度，本文提出基于数据分层的轮廓线并行重构算法，并基于 VTK 和 Mpich-G2 得到快速实现，同时设计了基于 Agent 技术的智能任务分解模块，基于 Condor-G 的 2 级调度模型和优先级抢先调度策略。最后通过分析不同并行度情况下重构速度的加速比，验证了并行算法对提交重构速度的有效性，由于任务间通信的影响，不同数据集在进行重构时都有一个最佳的并行度，因此下一步研究方向是进一步改进基于 Agent 技术的智能任务分解机制，保证最佳并行度。

参考文献

- [1] 周儒荣, 张丽艳, 苏旭, 等. 海量散乱点的曲面重构算法研究[J]. 软件学报, 2001, 12(2): 249-255.
- [2] 徐志伟, 冯白明, 李伟. 网格计算技术[M]. 北京: 电子工业出版社, 2004.
- [3] 易法令, 李庆华, 杨微微. Delaunay 三角剖分并行算法研究及实现[J]. 小型微型计算机系统, 2001, 22(4): 450-452.
- [4] Shi J Y, Cai W L. Visualization in Scientific Computing: Algorithm and System[M]. Beijing, China: Science Press, 1996.
- [5] Imamagic E, Radic B, Dobrenic D. An Approach Grid Scheduling by Using Condor——G Matchmaking Mechanism[C]//Proc. of ITI'06. Cavtat, Dubrovnik, Croatia: [s. n.], 2006.
- [6] Ahrens J, Brisilawn K, Martin K, et al. Large-scale Data Visualization Using Parallel Data Streaming[J]. IEEE Computer Graphics and Applications, 2001, 21(4): 34-41.
- [7] Karonis N, Foster I. MPICH-G2: A Grid-enabled Implementation of the Message Passing Interface[J]. Journal of Parallel and Distributed Computing, 2003, 63(5): 3-10.
- [8] 宋慧玲, 邓洪. 基于 VRML 的城市住宅小区交互式仿真系统[J]. 计算机工程, 2006, 32(17): 239-241.

编辑 金胡考