

doi: 103969/j.issn.0490-6756.2016.11.011

基于 InfiniBand 集群并行全景图拼接系统设计

陆应通, 李文强, 吴志红

(四川大学计算机学院视觉合成图形图像国防重点学科实验室, 成都 610065)

摘要: 全景图拼接是通过对序列图像进行拼接, 从而实现对场景全方位环视的技术, 是计算机视觉和图像处理的研究热点. 针对大规模高分辨率的序列图像拼接, 单计算机的拼接效率已经难以满足实际应用需求的问题, 提出了一种基于 InfiniBand 高速网络通信的多机并行架构实现全景图并行拼接的解决方案. 首先, 对全景图拼接的关键步骤进行并行性分析, 将全景图拼接尽可能划分成细粒度和高并行度的子任务; 然后, 通过并行架构中的控制节点和计算节点进行任务的分配和执行; 最后, 针对随着图像数据量增大, 采用传统的 TCP/IP 网络传输, 计算节点间的数据传输速率低下的问题, 设计了基于 InfiniBand 与 TCP/IP 结合的高速网络传输模块, 有效地提高节点间的传输速率. 实验结果表明, 在此并行架构上实现的分布式计算, 能够有效地提升全景图拼接效率, 有明显的实际应用价值.

关键词: 全景图拼接; 并行架构; 并行性分析; InfiniBand

中图分类号: TP391 **文献标识码:** A **文章编号:** 0490-6756(2016)06-1247-08

Design of parallel panoramic stitching system based on InfiniBand cluster

LU Ying-Tong, LI Wen-Qiang, WU Zhi-Hong

(National Key Laboratory of Fundamental Science on Synthetic Vision,
College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: Panoramic stitching is a technology to achieve all-round look of the scene by stitching image sequence. It is a research hotspot of Computer Vision and Image Processing. For large-scale and high-resolution image sequence stitching, stitching efficiency of single computer is hard to meet the needs of the practical application. A new solution for panorama parallel stitching based on a multi-machine parallel architecture with high-speed network InfiniBand was proposed. At first, do the parallelism analysis for the key steps of panoramic stitching and divide into fine-grained and high degree of parallelism sub-tasks; Secondly assign and execute the tasks through the control node and the compute nodes in the parallel architecture; Finally, in order to solve the problem that the rate of the data transmission between the computer nodes using the traditional TCP/IP network is low with the amount of image data increasing, the authors designed a high-speed network transmission module based on InfiniBand and TCP/IP which can effectively improve the transmission rate. The experimental results proved that the distributed computing on this parallel architecture can effectively improve the panorama stitching efficiency with a significant practical value.

Keywords: Panoramic stitching; Parallel architecture; Parallelism analysis; InfiniBand

收稿日期: 2016-02-29

基金项目: 国家高技术研究发展计划(2015AA016405); 国家自然科学基金(61472261)

作者简介: 陆应通(1991-), 男, 广东佛山人, 硕士, 研究方向为计算机图形图像处理、计算机图形学以及虚拟现实. E-mail: 370528668@qq.com

通讯作者: 吴志红. E-mail: wuzhihong@scu.edu.cn

1 引言

全自动高分辨率的全景图拼接是计算机视觉、图像处理和计算机图形领域中的研究热点^[1]。近年来,全景图拼接技术也接近成熟。随着现代计算机硬件技术飞速发展,计算速度有了很大的提升,但传统的串行全景图拼接方法显然是满足不了大规模高分辨率全景图拼接应用对拼接效率的需求。本文通过对传统的拼接算法进行并行性分析,将拼接任务划分成细粒度、高并行性的子任务,并结合具有高计算能力的计算机集群并行架构,设计并实现了并行全景图拼接系统。针对并行架构中网络传输能力不足的问题,本文提出了基于 InfiniBand 高速网络的并行架构。InfiniBand 是具有高带宽、低延迟的网络通信技术^[2-4],非常适合应用于计算机集群的通信场景。在 InfiniBand 的基础上,设计与传统 TCP/IP 结合的网络通信模块,提升了网络传输速率,解决了计算节点间的数据传输问题。并行系统整合了计算机集群的计算能力,实现了全景图的并行计算,有效地提升了拼接效率,有很好的实际应用价值。

2 相关工作

在全景图拼接领域中,传统的全景图拼接技术是使用几何特征点,计算并优化两幅连续图像间的投影变换,最后基于特征点的关系来构造全景图。1999年,Shum等人提出了一种基于全局、局部对准的全景图拼接技术^[1],这种技术就是给每张输入图像关联一个变换矩阵,然后采用全局对准方法减少累积的配准误差和局部对准方法补偿由相机平移引起的少量运动视差,通过结合全局和局部对准,提高了图像拼接质量,也使通过手持照相机拍摄得到的图像来构造全景图成为现实。移动物体是影响构造全景图质量的关键因素。2003年,Hsieh提出了一种基于边缘的拼接技术^[5]来检测移动物体和构造全景图,主要原理是采用边缘对准和基于相关性方法得到一个较好的相机初始化参数,然后采用网格划分方法来优化相机参数,这种技术能够更好地解决图像间出现移动物体和光照差异。同年,为了提升全景图拼接效率,Kim提出了一种基于重叠区域种子点的全景图拼接技术^[6],这个技术首先采用相位相关法提取重叠区域的种子点,然后通过加权重的块匹配算法来最小化相机旋转引起的图像畸变,能够快速地建立一幅高质量

的全景图。近年来,针对实时航拍图像拼接的应用需求,Kekec在2014年提出了一种基于“分离轴理论”检测图像交集的航拍图像拼接技术^[7],这种技术首先采用用于计算机图形碰撞检测的“分离轴理论”来检测图像交集,然后利用一种新型仿射细化过程方法来增加全局连续性,最后采用离焦措施曝光补偿和多频率融合方法来弥补光照差异和由错误配准引起的缝隙。2016年,Pandey提出了一种基于优化的图割方法^[8]用于全景图拼接。这种技术主要使用优化的图割算法来融合重叠区域,能够有效解决由相机运动,传感器噪声和光照差异等因素导致全景图拼接质量降低的问题。

3 系统概述

本文设计的全景图并行系统将全景图拼接过程拆分成高并行度的子任务,并将这些子任务按照一定依赖顺序分配至计算机集群中执行,这样凭借多台具有高计算能力的计算机,并行执行高分辨率序列图像拼接子任务,能够有效地提升全景图拼接效率。因此,该系统的输入是大规模序列图像数据(可包含多组全景图),输出则是多组全景图拼接结果。完整的系统设计流程如图1所示。

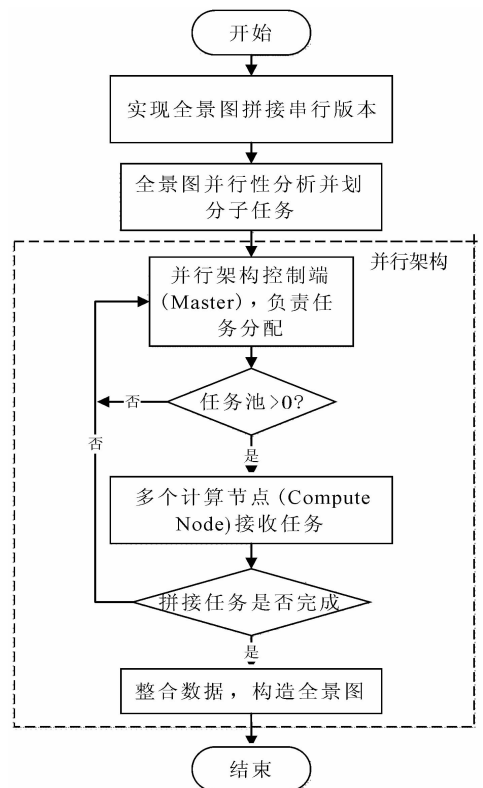


图1 系统设计流程图

Fig. 1 System design flow chart

该全景图并行拼接系统的主要设计流程为:首先以高内聚,低耦合的设计原则,实现全景图拼接串行版本,完成全景图拼接流程中的各个功能模块,为后面的并行设计奠定基础;然后对全景图拼接各个功能模块间进行并行性分析,对原有串行版本中的功能模块进行拆分或重新组合,划分出细粒度、高并行度的子任务并根据子任务间的数据依赖关系得到拓扑排序图;最后将产生的所有子任务输入到并行架构控制端(Master),Master会根据子任务的数据依赖图实时生成可执行任务,当任务池中的任务数和空闲计算节点数都大于零的条件满足时,就会进行任务分配,多个计算节点并行执行任务而且能够动态地加入和退出并行架构,计算机集群中的消息传输采用传统的TCP/IP协议,数据传输采用 InfiniBand 技术,降低了网络通讯消耗,提高了系统的性能.

4 系统设计

本文中提出的并行全景图拼接系统的核心设计思想是,结合计算机集群具有很高的数据处理能力优势,将全景图拼接拆分成可并行的小任务,让计算集群中的计算节点并行处理这些小任务,尽可能将所有计算机资源利用起来,从而达到加速的效果.系统设计主要包含全景图拼接技术实现,全景图拼接并行性分析和并行架构设计三部分.

4.1 全景图拼接技术

在进行全景图并行拼接前,首先需要实现全景图拼接的串行版本,完成全景图拼接过程中的各个功能模块.本文采用了传统的全景图拼接方法,这种方法是基于几何特征点和优化连续图像间的投影变换来构造全景图.使用该方法进行全景图拼接,对序列图像拍摄要求不高,只需要用手持照相机以一定重叠区域定点环绕拍摄即可.完整的全景图拼接流程如图2所示.

全景图拼接主要的功能模块如下

(1) 特征点检测:采用 sift 算法^[9],sift 特征点具有对旋转、缩放、亮度变化不变性.该算法应用于全景图拼接更有效地对准图像及缓解图像间的缝隙问题.

(2) 特征点匹配:采用了欧式距离匹配算法^[10]对 sift 算法检测出的特征点特征向量描述子进行匹配,并利用 KD Tree 来优化搜索,加速匹配.

(3) 相机标定^[11]:首先通过特征点匹配对信息来初始化连续图像间的投影变换和相机参数,然

后采用最小二乘法最小化重投影误差,调整相机参数.

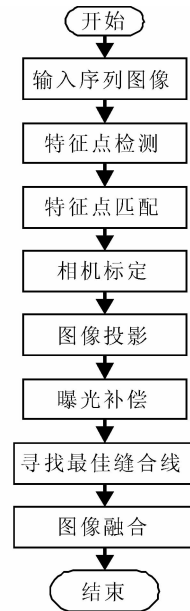


图2 全景图拼接流程图

Fig. 2 Panoramic mosaicing flow chart

(4) 图像投影:首先通过图像对应的相机参数求出像素对应三维点 $P(x, y, z)$,然后将三维点 P 投影到柱面 (θ, v) 或球面坐标 (θ, ϕ) ,其投影公式如式(1)、式(2)所示.

$$\begin{cases} \theta = \tan^{-1}(x/z) \\ v = y/\sqrt{x^2 + z^2} \end{cases} \quad (1)$$

$$\begin{cases} \theta = \tan^{-1}\left(\frac{x}{z}\right) \\ \phi = \tan^{-1}\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \end{cases} \quad (2)$$

(5) 曝光补偿^[12,13]:全景图中缝隙产生的主要原因之一是图像间存在光照差异,本文采用了增益补偿法来平滑光照.增益补偿法是基于优化问题的算法,通过最小化图像间重叠区域的光照差异来得到所有图像的光照增益值.

(6) 寻找最佳缝合线:当拍摄场景中存在运动目标或拍摄造成了配准误差时,将会导致鬼影的出现.本文采用基于图割的算法^[14],找到重叠区域最佳的缝合线,缝合线两边像素分别取自不同图像,达到消除鬼影的目的.

(7) 图像融合:本文采用了基于拉普拉斯图像金字塔融合算法^[15],构造一组带通图像,并使用不同的加权函数进行颜色融合,该方法能够使得图像间平滑地过渡,同时在有配准误差的位置保留细节.

4.2 全景图拼接并行性分析

在实现了全景图拼接串行版本的各个功能模块后,需要分析功能模块之间的数据依赖,然后根据各个模块间的数据依赖,便可得到全景图拼接流程中各个子过程的并行度. 三张序列图像的情况下,各功能模块间的数据依赖划分得到子任务的拓扑排序图,如图 3 所示.

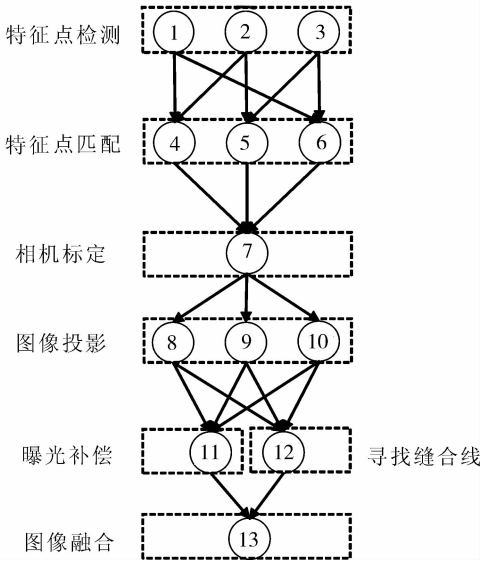


图 3 子任务拓扑排序图

Fig. 3 Subtasks topological-sort chart

拓扑排序图描述了子任务间的数据依赖关系以及它们的执行先后关系. 结合各功能模块与任务拓扑排序图,可以得到各功能模块的并行性和子任务的分解过程如下.

(1) 系统首要执行的是特征点检测任务,每张图像都会产生一个特征点检测任务,即图 3 中的 1、2、3 号任务,这三个任务是系统功能的开始,并且不依赖于其他任务,因此可以并行执行.

(2) 当有两组特征点检测结果生成时,即可生成一个可执行的特征点匹配任务. 根据两图像间需要进行一次特征点匹配原则,可产生如图 3 中 4、5、6 号匹配任务. 系统会实时监测每个任务的前驱依赖任务是否完成,若完成,则把该号任务置于可执行队列.

(3) 相机标定任务通过输入所有图像间的特征点匹配信息,求得所有图像对应的相机参数. 因此,图 3 中 7 号相机标定任务需要等待所有匹配任务完成后,才能开始执行.

(4) 恢复出图像的相机参数后,即可进行图像投影,每个图像生成一个图像投影任务,均依赖于

7 号相机标定任务结果,投影任务间没有数据依赖,可并行执行.

(5) 系统在执行完所有的图像投影任务后,将会执行图 3 中 11 号曝光补偿任务和 12 号寻找缝合线任务,这两个任务都是输入所有图像投影结果,之间不存在数据依赖,可并行执行.

(6) 系统最后执行的是 13 号图像融合任务,该任务依赖于曝光补偿和寻找缝合线的结果,图像融合任务输出全景图拼接结果.

4.3 并行架构

并行全景图拼接系统主要由三个模块组成:任务控制与计算模块(一个控制节点 Master 和多个计算节点 Compute Node);负载均衡模块;基于 InfiniBand 与 TCP/IP 的高速网络通信模块(如图 4).

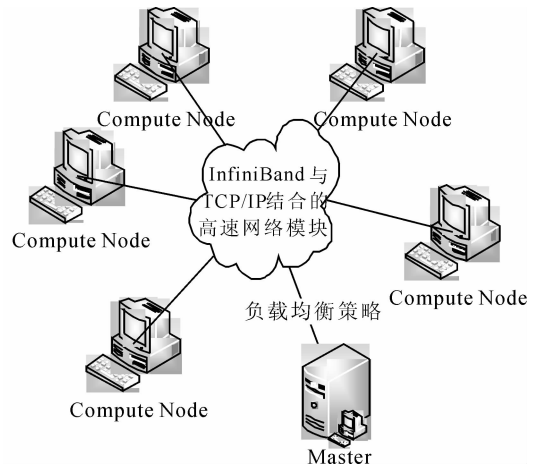


图 4 全景图拼接并行架构

Fig. 4 Panoramic mosaicing parallel architecture

(1) 任务控制与计算模块.

控制节点 Master 是并行全景图拼接系统的控制中心,保证全景图拼接子任务有条不紊地执行. Master 程序设计主要完成以下功能:(a) 根据设计的任务拓扑排序规则,对输入的序列图像生成一系列的子任务对象,并保存在一个数组中,任务对象主要包含有任务索引 Index,任务类型 Task Type,任务状态 (READY, DOING, FINISH, EXCEPTION),前驱依赖任务索引集,执行该任务计算节点的 IP 地址等信息;(b) 监听计算节点,计算节点能够动态注册进入或退出并行架构,而不影响并行架构的正常运行,Master 将计算节点信息保存在一个计算节点对象数组中,计算节点对象主要包含有 IP 地址,计算机状态(可用 CPU,可用物理内存)等信息;(c) Master 实时遍历任务数组和计算节点数组,采用负载均衡策略进行任务分配,使得

各个计算节点的资源尽可能使用起来.

计算节点 Compute Node 是并行全景图拼接系统中的任务执行节点. Compute Node 程序设计主要完成以下功能:(1)任务监听,能够接收从 Master 分配过来的任务;(2)Compute Node 实时向 Master 发送 CPU 可用率、物理内存可用率更新信息,以便 Master 进行实时任务分配;(3)当任务的输入数据不在本机时,当前结点能够向目标节点发起数据请求信息,并且目标节点能够处理数据请求信息,往回发送请求的数据;(4)任务完成后,把任务结果保存到一块内存区,给内存区设置一个阈值大小,当超过阈值时,将最久未使用的数据内存序列化到文件,最后通知 Master 该任务状态.

(2) 负载均衡策略设计.

负载均衡策略是影响并行全景图拼接系统性能的关键因素. 一个不合理的负载均衡策略将会导致某个计算节点任务超载或空闲,浪费计算资源,使得整体全景图拼接效率下降. 本文首先根据每个计算节点的可用 CPU、可用物理内存计算该计算节点在总的计算集群资源中所占的权重. 权重计算公式如下所示.

$$w_i = \frac{c_i + m_i}{\sum_{i=1}^n (c_i + m_i)}, (i = 1, 2, \dots, n) \quad (3)$$

其中, n 为计算节点个数; w_i 为第 i 个计算节点占总资源的权重; c_i 为第 i 个计算节点的 CPU 可用率; m_i 为第 i 个计算节点的物理内存可用率.

Master 再根据任务数组中任务对象的前驱依赖任务索引集的属性来生成 Ready 任务,假设当前有 k 个可执行任务,则每个计算节点能够得到任务个数的计算公式如下所示.

$$t_i = k * w_i, (i = 1, 2, \dots, n) \quad (4)$$

其中, t_i 是第 i 个计算节点得到的任务个数.

(3) 基于 InfiniBand 网络通信模块.

一个高速网络通信模块是提高并行全景图拼接系统性能的重要因素. 本文设计了基于 InfiniBand 与传统的 TCP/IP 协议结合的高速网络通信模块,降低了网络通讯耗时. InfiniBand 是结合了高传输速率硬件、专门的协议和远程数据内存访问(RDMA)的技术. 在计算机集群中采用 InfiniBand 技术进行网络互联,能够有效地提高 CPU 利用率,降低通讯延迟和简化了数据中心的管理问题.

InfiniBand 在节点间进行大规模数据发送时,主要采用程序控制流的方法来保证数据的正常发

送与接收,InfiniBand 数据收发时序图如图 5 所示. 当一个计算节点收到数据发送指令时,该节点首先计算数据的大小,并采用 UDP 给数据接收节点发送一个数据空间准备包,为接收数据开辟空间,然后接收节点同样采用 UDP 给数据发送节点发送一个准备确认包,最后发送节点使用 InfiniBand 发送大规模数据.

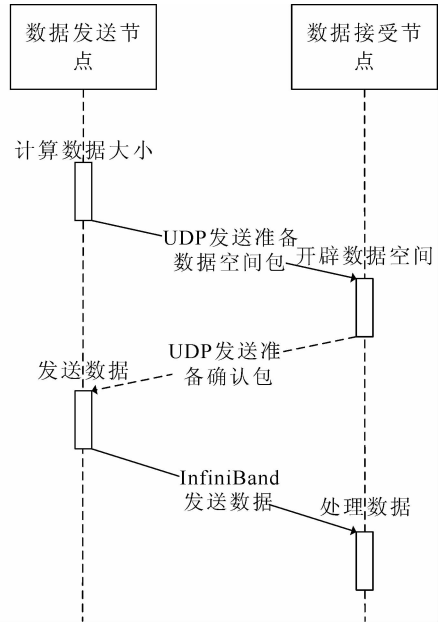


图 5 InfiniBand 数据收发时序图 Fig. 5 InfiniBand sequence chart

在并行全景图拼接系统网络中,主要发送两种类型的数据,一种是消息数据,如任务消息,状态消息等,具有数据量小的特点,另一种是全景图拼接过程中的中间数据,随着序列图像的增加,中间数据规模也会越来越大. 为此,本文设计并实现了 InfiniBand 与 TCP/IP 结合的高速网络通信模块,首先底层实现了 InfiniBand、TCP 和 UDP 这三种基本的网络通信功能,然后设计了一个自动选择网络的核心功能,自动选择网络的主要设计思想是根据数据包的大小来进行网络选择,在本系统中主要用到两种协议,消息数据由于数据量小,采用 UDP 发送,拼接过程中的中间数据规模大,则采用 InfiniBand 发送,最终只需要给用户提供一个网络发送接口,便可根据数据包的信息自动选择最合适网络进行数据发送.

5 结果分析

5.1 实验平台

针对本文的并行全景图拼接系统,搭建了一个

具有 5 台计算机的多机并行系统, 每台计算机具有相同的硬件配置. 配置如下:

处理器: Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30 GHz 四核; 内存: 16.0 GB; 显卡: NVIDIA GeForce GTX 980; 显示器分辨率: 1920x1080; 操作系统: Windows 7(64 位).

多机并行系统采用基于 InfiniBand 硬件技术进行网络互联. 配置如下:

适配器卡: Mellanox MHQH19B-XTR InfiniBand; 交换机: Mellanox IS5022 8 端口非阻塞交换机, 速率可达到 40 Gb/s.

5.2 实验结果

为了证明系统的有效性, 我们分别进行了在单机串行版本、基于传统 TCP/IP 协议的并行拼接系统和基于 InfiniBand 高速网络并行拼接系统下的拼接效率测试. 表 1~表 3 分别表示在单机串行版本、基于 TCP/IP 协议并行系统和基于 InfiniBand 高速网络并行系统对于不同序列图像数所需的拼接时间.

表 1 单机串行版本下的拼接时间(s)

Tab.1 Stitching time on serial version

图像数(张)	单机串行时间(s)
36	374.58
72	998.76
108	1985.32
144	2652.26
180	3359.07
216	4030.89
288	6315.44

表 2 TCP/IP 并行系统的图像拼接时间(s)

Tab.2 Stitching time on parallel system(TCP/IP)

图像数(张)	2 节点(s)	3 节点(s)	4 节点(s)
36	518.67	492.34	408.32
72	1173.04	1022.43	822.943
108	1811.56	1393.45	1228.97
144	2178.81	1920.32	1544.49
180	2868.12	2499.00	1960.54
216	3596.08	2925.91	2363.14
288	4617.10	3959.52	3220.59

表 3 InfiniBand 并行系统的图像拼接时间(s)

Tab.3 Stitching time on parallel system(InfiniBand)

图像数(张)	2 节点(s)	3 节点(s)	4 节点(s)
36	244.50	199.23	163.81
72	639.65	496.62	390.28
108	957.29	674.62	536.35
144	1318.74	942.89	725.67
180	1697.35	1151.71	891.64
216	1992.36	1445.83	1202.8
288	2962.42	2063.59	1654.02

表 1 和表 2 分别列出了单机串行和基于 TCP/IP 协议多机并行全景图拼接系统对于不同序列图像数的拼接效率, 其折线图如图 6 所示. 在本文实验计算集群局域网中, 采用 TCP/IP 协议最大传输速率大概是 10 MB/s, 而一张 4384×2466 分辨率的图像读取到内存中大概是 30 MB. 由于节点间涉及大量图像数据与中间数据传输, 从表 2 可以看出, 当并行系统中只输入 36 张序列图像时, 采用 TCP/IP 协议进行数据传输的通讯延时影响大于 2 台到 4 台计算节点同时执行任务的效率影响, 所以只有 4 台计算节点的并行全景图拼接系统对于输入 36 张序列图像的拼接效率低于单机串行版本. 当并行系统输入 72 张序列图像且有 4 台计算节点时, 能够生成足够的任务使得并行全景图拼接系统克服了采用 TCP/IP 网络数据传输延时影响, 拼接效率超越了单机串行版本. 当输入 288 张序列图像, 4 台计算节点的并行系统相对于单机串行版本效率提升将近两倍, 并随着图像数和计算节点数的增加, 并行全景图拼接系统有更好的性能.

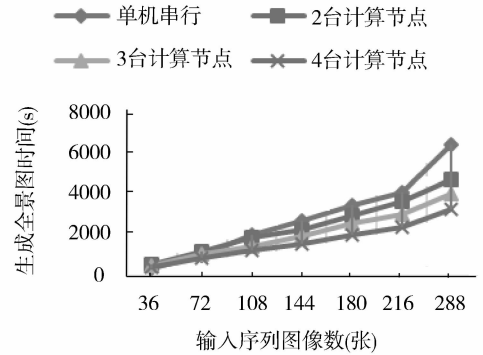


图 6 基于传统 TCP/IP 网络并行全景图系统效率

Fig.6 Efficiency line chart of the parallel panorama system based on TCP/IP

表 3 列出了基于 InfiniBand 高速网络并行全景图拼接系统对于不同序列图像数的拼接效率, 其折线图如图 7 所示. 由于采用了 InfiniBand 高速网络进行数据传输, 最高速率可达到 40 Gb/s, 解决了网络通讯延时问题. 由图 7 可知, 具有 2 台计算节点的并行全景图拼接系统对于各组的输入序列图像数的拼接效率都高于单机串行版本, 而且随着计算节点数的增加, 并行全景图拼接系统的拼接效率都有一定的提升. 从表 3 可以看出, 4 台计算节点的并行全景图拼接系统相对于单机串行版本效率提升达到 3 倍多.

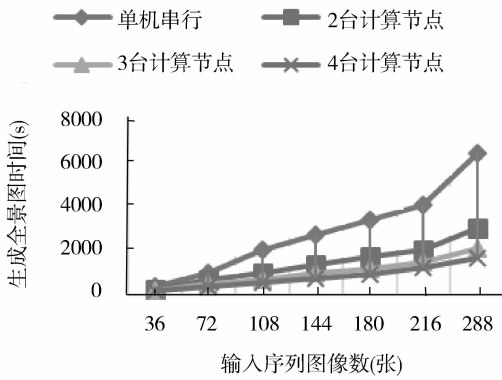


图 7 基于 InfiniBand 高速网络并行全景图系统效率
Fig. 7 Efficiency line chart of the parallel panorama system based on InfiniBand

图 8 分别显示了单机串行全景图拼接版本(a)和基于 InfiniBand 并行全景图拼接系统(b)对于



(a)



(b)

图 8 单机串行版本(a)与基于 InfiniBand 并行全景图拼接系统(b)全景序列图像拼接结果对比
Fig. 8 The compared figure of panoramic image sequence about single serial version(a) and the parallel panorama system based on InfiniBand (b)

该系统的局限性在于:全景图拼接流程某些功能模块未能拆分出子任务,如相机标定模块,需要等到前面所有数据才能执行;控制节点 Master 在进行任务分配时没有考虑不同的任务占用的 CPU 与物理内存不一致,可能导致某些计算节点超载或

96 张全景序列图像的拼接结果. 综上对比,基于 InfiniBand 高速网络并行拼接系统在保证拼接结果一致的前提下,有效地降低计算节点间的通讯延迟,提升了全景图拼接效率,符合大规模全景图拼接应用要求.

6 结 论

本文设计实现了一个并行全景图拼接系统. 该系统能够同时对多组全景图序列图像进行处理,提高了系统的吞吐量,然后将这些全景图拼接任务划分出细粒度,高并行性的子任务,在计算机集群中并行执行这些子任务,有效地提升了全景图拼接效率. 同时,本文采用了基于 InfiniBand 的高速网络进行计算节点间的数据传输,大大降低了网络通讯延时,也极大地提高了并行全景图拼接系统的性能.

空闲;过多的数据传输,影响了该系统在仅有 TCP/IP 网络情况下的性能.

因此,根据以上局限性,本文未来工作将致力于拆分一些瓶颈功能模块,改善负载均衡策略,以及完善任务调度策略,减少节点间的数据传输,提

升系统在 TCP/IP 网络下的性能,进一步优化并行全景图拼接系统.

参考文献:

- [1] Shum H Y, Szeliski R. Construction of panoramic image mosaics with global and local alignment[C]//Panoramic Vision. New York: Springer, 2001.
- [2] Pentakalos O I. An introduction to the infiniband architecture[C]//High Performance Mass Storage and Parallel I/O: Technologies and Applications. Wiley: IEEE Press, 2002.
- [3] Grun P. Introduction to InfiniBand for end users[M]. Beaverton: InfiniBand Trade Association, 2010.
- [4] Vivek D, Deshmukh. InfiniBand: A new Era in networkint[C]//Proceedings of National Conference on Innovative Paradigms in Engineering & Technology. New York, USA: Foundation of Computer Science, 2012.
- [5] Hsieh J W. Fast stitching algorithm for moving object detection and mosaic construction[C]//Proceedings of 2012 IEEE International Conference on Multimedia and Expo. Melbourne, Australia: IEEE, 2003.
- [6] Kim D H, Yoon Y I, Choi J S. An efficient method to build panoramic image mosaics[J]. PRL, 2003, 24(14): 2421.
- [7] Kekec T, Yildirim A, Unel M. A new approach to real-time mosaicing of aerial images [J]. RAS, 2014, 62(12): 1755.
- [8] Pandey A, Pati U C. Panoramic image mosaicing: an optimized graph-Cut approach[C]//Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics. Bhubaneswar, India: Springer, 2016.
- [9] Lowe D G. Object recognition from local scale-invariant features[C]//The Proceedings of the Seventh IEEE International Conference on Computer Vision. Corfu, Greece: IEEE, 1999.
- [10] Hess R. An open-source SIFTLibrary[C]//Proceedings of the international conference on Multimedia. Firenze, Italy: ACM, 2010.
- [11] Stein G. Accurate internal camera calibration using rotation, with analysis of sources of error[C]//Proceedings of Fifth International Conference on Computer Vision. Boston, USA: IEEE Computer Society Press, 1995.
- [12] Xu W, Mulligan J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching[C]//Proceedings of IEEE Conference on Computer Vision & Pattern Recognition. San Francisco, USA: IEEE, 2010.
- [13] Uyttendaele M, Eden A, Szeliski R. Eliminating ghosting and exposure artifacts in image mosaics. [C]//Proceedings of IEEE Conference on Computer Vision & Pattern Recognition. Hawaii, USA: IEEE Computer Society Press, 2001.
- [14] Kwatra V. Graphcut textures: image and video synthesis using graph cuts[J]. ACM TOG, 2003, 22(3): 277-286.
- [15] Xiong Y, Turkowski K. Registration, calibration and blending in creating high quality panoramas [C]//Proceedings of Fourth IEEE Workshop on Applications of Computer Vision, Princeton, New Jersey, USA: IEEE Computer Society, 1998.