第37卷 第14期 计 算 机 工 程 2011年7月
Vol.37 No.14 Computer Engineering July 2011

·图形图像处理· 文章编号：1000—3428(2011)14—0220—03 文献标识码：A 中图分类号：TP312

多核图像处理并行设计范式的研究与应用

王成良[a]，谢克家[b]，刘 昕[a]

(重庆大学 a. 软件工程学院；b. 计算机学院，重庆 400044)

摘 要：多核计算环境下采用图像处理并行算法可提高图像处理的速度，但已有的并行设计只针对边缘检测、图像投影等特定算法进行，没有形成通用的并行算法设计范式。为此，在研究图像处理算法可并行处理机制和多核架构特点的基础上，提出分析、建模、映射、调试和性能评价及测试发布等5个设计步骤的基于多核计算环境的图像处理算法并行设计范式，以图像傅里叶变换并行算法设计为例在单核、双核、四核、八核计算环境下验证了该并行范式的有效性。实验结果表明，该范式在图像处理并行设计方面可扩展图像处理的应用空间。

关键词：并行算法；并行设计范式；图像处理；多核计算环境；可扩展性

Research and Application of Multi-core Image Processing Parallel Design Scheme

WANG Cheng-liang[a], XIE Ke-jia[b], LIU Xin[a]

(a. School of Software Engineering; b. College of Computer Science, Chongqing University, Chongqing 400044, China)

【Abstract】In multi-core computing environment, the image processing parallel algorithms can greatly improve the processing speed. However, the existing parallel designs are focused on the specific algorithms such as edge detection and image projection, which can not form a universal design scheme. Thus, it is difficult to extend this application. Based on the in-depth study of the image algorithms parallel processing mechanism and the features of the multi-core architecture, this paper proposes an image processing parallel design scheme in multi-core computing environment, which has five steps, including analysis, modeling, mapping, debugging & performance evaluation and testing & release. The paper takes the algorithm design of parallel image Fourier transforms as an example to testify the effectiveness of this scheme in single-core, double-core, quad-core and eight-core computing environment. Experimental result shows that the proposed multi-core parallel design scheme has good scalability, and this scheme can extend the space of application for image processing.

【Key words】parallel algorithm; parallel design scheme; image processing; multi-core computing environment; scalability

DOI: 10.3969/j.issn.1000-3428.2011.14.074

1 概述

伴随多核处理器技术的发展，多核计算环境下并行图像处理算法的研究引起了越来越多的关注[1-4]。文献[1]设计实现一个多核计算环境下的图像分析并行计算类库，对常用的边缘检测、阈值分割、图像投影与合成算法进行了并行设计和实现；文献[2]设计实现了一个多线程的并行图像检索系统；文献[3]研究了多核处理器下实时图像处理技术；文献[4]研究了OpenMP在多核图像处理中的应用，此外还有相关文献对图像处理算法的并行处理做了一定研究。但这些文献中的并行设计方法不具通用性，难以推广使用。此外，在多核计算环境下，传统的基于并行机和并行集群算法的设计方法和编程模式不再完全适用，因此，研究多核架构下通用的并行设计范式具有重要意义。

本文利用多核架构的并行特性，提出基于多核计算环境的数字图像处理算法并行设计范式，并设计实现了多核环境下的并行图像快速傅里叶变换算法来验证范式的有效性。

2 多核架构

多核CPU是将多个CPU核集成到单个芯片中，每个CPU核作为一个独立的处理器。以Intel多核处理器为例，所有CPU共享一个统一的地址空间，有单独的L1 Cache，采用多级Cache结构，采用总线或者Crossbar作为互连结构，使用Cache一致性协议维护数据一致性，采用多线程或者多进程作为并行软件设计方法。多核处理器架构如图1所示。

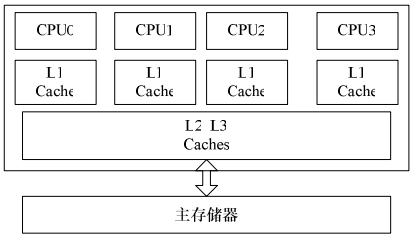

图1 多核处理器架构

多核处理器的并行分为指令级并行、线程级并行和同步多线程。其中，线程级并行是多核处理器的主流并行技术，每个多核处理器在未采用超线程技术的情况下，支持与核心数相等的线程并行执行，这为程序的并行执行提供了实现环境。且核与核之间的通信通过共享内存完成，彻底解决了并行集群存在的结点间通信延迟问题。

3 多核计算环境下图像处理算法并行设计范式

根据多核构架的特点，本文在对图像处理大量算法分析

基金项目：重庆市科技攻关计划基金资助项目(CSTC, 2009AC2060)

作者简介：王成良(1964－)，男，教授、博士，主研方向：并行分布式计算，图形与图像处理，数据库技术；谢克家，硕士研究生；刘 昕，本科生

收稿日期：2011-01-05 **E-mail：**xkj0828@163.com

的基础上，提出了多核计算环境下数字图像处理并行算法的设计范式(下称 P 范式)。该范式将数字图像算法的并行设计分为 5 个步骤：串行算法热点分析，并行建模，映射到多核架构，调试和性能评价及测试发布，如图 2 所示。

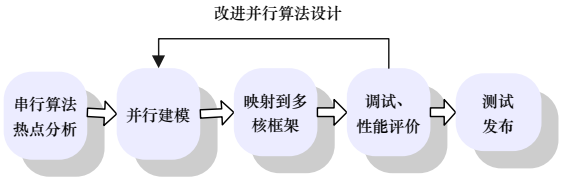

图 2　多核计算环境下数字图像处理并行算法设计范式

(1)串行算法热点分析：在进行算法并行建模之前，首先可借助于 Intel Parallel Studio 并行开发套件中的 Amplifier 性能分析器等性能分析工具确定程序的热点，即算法中耗时最多的代码。

(2)并行建模：利用并行算法设计模型，对算法的执行瓶颈部分进行并行设计。常用的并行算法设计模型有基于任务分解的并行设计模型和基于数据分解的并行设计模型[5]。数据分解的并行设计模型以并行阵列连接模式为基础，它用多个处理单元组成一个并行阵列，每一个处理单元都可独立执行任务。并行阵列连接模式如图 3 所示。

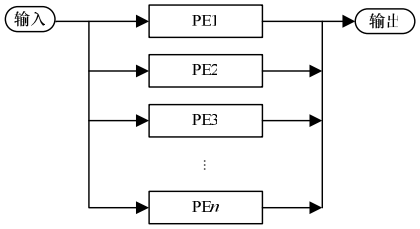

图 3　并行阵列结构

对于数字图像处理领域的算法而言，基于数据分解的并行设计模型最为直观，应用得也较为广泛。它是一种常见并有效的底层并行化图像处理方法，先把图像分成块，然后把这些块分到各个处理器上，每个处理器对自己的块进行同样的图像处理。对于底层图像处理来说，像素级及区域间的图像处理占主导地位，处理是非常局部的，因此，处理器能相对独立地处理，从而实现并行化。

(3)映射到多核架构：在完成了算法的并行建模后，需将并行算法映射到多核架构上。多核架构通过多线程并行执行来实现算法的并行执行，线程间通过共享存储器直接实现通信和数据交换，相比并行集群，解决了因节点间通信延迟带来的性能损失。

OpenMP 是基于线程的并行编程模型，是共享存储系统编程的工业标准[6]。其并行执行模型如图 4 所示。

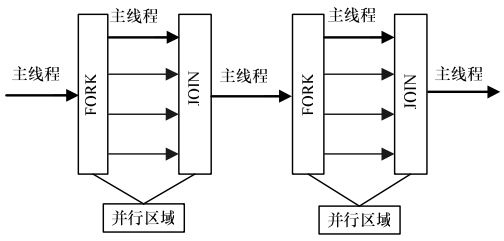

图 4　OpenMP 并行执行模型

OpenMP 采用了共享存储中标准的并行模式 fork-join，当程序开始执行时只有主线程存在，主线程执行程序的串行部分，通过派生出其他的线程来执行其他的并行部分。当重新执行程序的串行部分时，这些线程将终止。使用 OpenMP 可以方便地实现并行设计的算法到双核、四核和八核等多核计算环境的映射。因此，P 范式中采用 OpenMP 来实现并行算法到多核架构的映射。

(4)调试和性能评价：在完成并行算法到多核架构的映射之后，要对并行执行后的算法进行调试和排错以及性能分析，具体过程如果 5 所示。

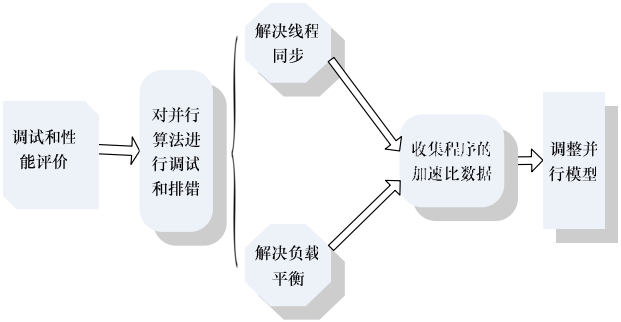

图 5　调试和性能评价实现过程

对并行算法进行调试和排错包括解决线程同步和负载平衡问题。多核架构采用共享内存机制，对共享内存的访问可能会引发数据竞争。程序结果以不确定的方式依赖于 2 个或多个线程执行的时间顺序，因此必须通过线程同步的方式解决数据竞争问题。另外，并行执行带来的负载平衡问题也是影响算法执行时间的重要因素，因此，需要通过平衡各个处理器核的计算负荷来减少算法的运行时间。

评价并行算法性能非常重要，加速比是评价并行算法性能的重要指标[7]。在性能评价阶段，通过收集程序的加速比数据，调整步骤(2)中的并行模型，使算法的总运行时间最短。

(5)测试、发布：在完成了并行算法的调试和性能评价之后，对其进行正确性测试和可扩展性测试，并最终发布应用。

4　P 范式的应用及分析

图像的快速傅里叶变换可对图像的频谱进行滤波、降噪和增强等处理。但在气象、遥感图片的处理中，由于图片尺寸较大，传统算法的运行性能受到严重影响。对图像傅里叶变换的并行设计，可使其性能得到较大提高。

4.1　图像傅里叶变换的热点分析

对图像傅里叶变换算法而言，其热点为图像数据的二维快速傅里叶变换(Fast Fourier Transforms, FFT)。可应用上文所提 P 范式，对二维 FFT 算法进行并行设计。

4.2　图像傅里叶变换的并行建模

二维 FFT 问题可转化为一维 FFT 问题来解决。对于一个 $M\times N$ 的二维 FFT$x(i,j)$而言，可先对图像每一行数据依次进行一维 FFT 运算，在完成所有行数据的一维 FFT 后，同理再对图像每一列数据依次进行一维 FFT，这种算法被称为行-列算法。此算法将每一行数据作为一个运算单元，利用基于数据分解的并行模型将运算单元分配到计算节点上。对于多核结构，一个处理器核代表一个计算节点，把运算单元分配到计算节点的过程就是将任务分配到处理器核线程上的过程。

下面以大小为 8×8 像素的图片快速傅里叶变换并行算法在四核 CPU 上的执行过程为例来说明多核环境下图像快速傅里叶的数据分解和并行执行过程，具体的任务分配情况如图 6 所示。四核 CPU 有 4 个处理器核，每个处理器上运行一个线程，编号分别为 0、1、2 和 3。根据二维傅里叶变换

的行-列算法，首先把图片每一行数据作为一个运算单元依次分配给 4 个处理器核进行 FFT 运算，所有行数据运算完成后，再把图片每一列数据作为一个运算单元分配给处理器核进行 FFT 运算，最终完成对整个图片的二维 FFT 运算。在整个运算过程中，各个处理器核并行处理分配给自己的数据以提高运算效率。

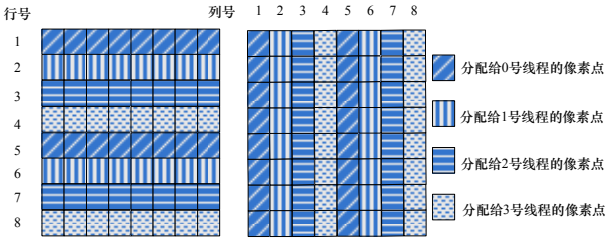

图 6 傅里叶行/列变换种任务在四核处理器上的分配

4.3 图像傅里叶变换到多核计算环境的映射

OpenMP 是用以编写可移植的多线程应用程序的 API (Application Programming Interface)库，内部使用 Windows 线程池，可很好地支持多核环境下的并行程序设计。以下代码将使二维 FFT 算法在多核架构上以多线程模式并行运行。其中，函数名为 ParallelImgFFT2D()；函数参数包括 unsigned char* imgBuf-图像数据指针、int width-图像宽、int height-图像高；傅里叶变换后的输出图像为 unsigned char *imgBufOut。核心代码如下所示：

```
void Transform_FFT::ParallelFF2D(unsigned char* imgBuf, int width, int height, unsigned char *imgBufOut)
{       //先纵向一维快速傅里叶变换
    #pragma omp parallel for private(u,v)
  for(u=0;u<width;u++)
  {             //申请 ComplexNumber 结构体数组,长度为 height
        ComplexNumber *array=newComplexNumber[height];
        for(v=0;v<height;v++)
{
              array[v].real=imgBuf[v*width+u];
              array[v].imag=0;
        }
        FFT1D(array, height);//进行一维快速傅里叶变换
        for(v=0;v<height;v++)
        {
              m_pFFTBuf[v*width+u].real=array[v].real;
              m_pFFTBuf[v*width+u].imag=array[v].imag;
        }
        delete []array;
  }//再横向一维快速傅里叶变换
  #pragma omp parallel for private(v)
  for(v=0;v<height;v++){
FFT1D(m_pFFTBuf+v*width, width);//进行一维快速傅里叶变换
}//将频谱图以图像形式存入 imgBufOut，具体代码略
}
```

4.4 并行调试与性能分析

在调试和性能分析时，采用的硬件环境为 CPU：Pentium E5300 2.8 GHz 双核处理器；主频 3.0 GHz，二级缓存 2 048 KB，内存 4 GB。采用的软件环境为：IDE：Visual studio 2008SP1，编译器：Intel C++ Compiler 11.0。

将并行的二维 FFT 算法在多核环境下和传统的单核环境下进行性能测试。所采用的实验环境分别为 DELL OPTIPLEX GX520 奔腾四单核台式机、联想启天 M690E 双核台式机、IBM System x3650 四核服务器以及 IBM System x3650 八核服务器。所采用的硬件环境的详细参数如表 1 所示。

表 1 实验环境详细参数

CPU 核数	型号	主频 /GHz	二级缓存 /KB	内存大小 /GB
1	Pentium®	2.8	1 024	1.5
2	Pentium E5300	2.8	2 048	4.0
4	Xeon E5450	3.0	2 048	4.0
8	Xeon E5450	3.0	8 192	4.0

本文在不同计算环境下开启的线程数与处理器的核心数相等。实验对不同尺寸的图像在不同环境下的处理时间进行计时，根据 10 次实验结果得到平均执行时间。实验结果如表 2 所示。

表 2 算法在不同计算环境下的运行时间

图像大小/ pixel	不同计算环境下的运行时间/ms			
	单核	双核	四核	八核
512×512	215.6	47.1	31.0	16.0
1 024×1024	1 344.2	250.7	125.0	94.5
2 048×2 048	4 813.5	1 275.2	711.1	456.0
4 096×4 096	20 656.6	6 187.5	3 406.5	2 147.6
8 192×8 192	92 484.3	24 850.9	139 353.5	8 732.8

表 2 中的数据表明，在多核环境下，二维 FFT 的时间明显缩短，不同尺寸的图片在 4 种平台上产生的加速比曲线如图 7 所示。

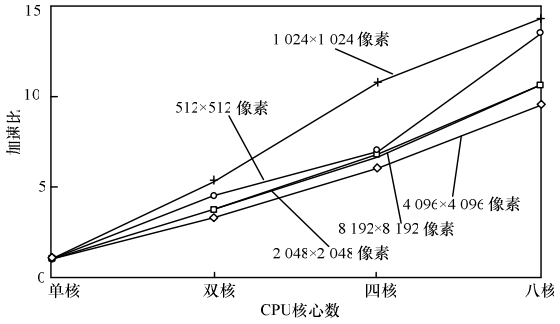

图 7 不同大小的图片在 4 种实验环境下的加速比

从图 7 中可看出，随着实验环境处理器核数的增加，每条加速比曲线呈线性递增趋势，特别是在八核环境下，加速比达到了 14.2。实验结果表明并行设计后的二维快速傅里叶变换算法达到了线性加速比，其中，1 024×1 024 像素大小的图片在多核环境下的加速效果最优。

4.5 测试发布

经调试和性能分析后的图像傅里叶变换并行算法将以 API 的形式供图像傅里叶变换的图像处理程序调用。

5 结束语

本文通过对国内外相关文献的研究，提出了多核计算环境下的数字图像处理算法并行设计范式，以充分利用多核处理器的并行计算能力。实验结果表明，该范式易于理解，便于应用，具有通用性。下一步工作主要包括解决因图片规模增大而导致并行算法性能下降问题以及将所提范式投入到实际工程应用中。

参考文献

[1] 郑 锋. 图像分析多核并行计算类库的构建与优化[D]. 厦门：厦门大学, 2008.

[2] 杨凌云. 基于多核技术的并行图像检索系统的研究[D]. 北京：北京化工大学, 2008.

(下转第 225 页)