

多核与众核上 MNF 并行算法与性能优化*

方民权^{1,2}, 张卫民^{1,2}, 高畅¹, 方建滨¹

¹(国防科学技术大学 计算机学院, 湖南 长沙 410073)

²(国防科学技术大学 海洋科学与工程研究院, 湖南 长沙 410073)

通讯作者: 方民权, E-mail: fmq@hpc6.com

摘要: 高光谱遥感影像降维最大噪声分数变换(maximum noise fraction rotation, 简称 MNF rotation)方法运算量大, 耗时长。基于多核 CPU 与众核 MIC(many integrated cores)平台, 研究 MNF 算法的并行方案和性能优化。通过热点分析, 针对滤波、协方差矩阵运算和 MNF 变换等热点, 提出相应并行方案和多种优化策略, 量化分析优化效果, 设计 MKL(math kernel library)库函数实现方案并测评其性能; 设计并实现基于多核 CPU 的 C-MNF 和基于 CPU/MIC 的 M-MNF 并行算法。实验结果显示, C-MNF 算法在多核 CPU 取得的加速比为 58.9~106.4, 而基于 CPU/MIC 异构系统的 M-MNF 算法性能最好, 加速比最高可达 137 倍。

关键词: 集成众核; 多核并行; 高光谱影像降维; 最大噪声分数变换; MKL 性能分析

中文引用格式: 方民权, 张卫民, 高畅, 方建滨. 多核与众核上 MNF 并行算法与性能优化. 软件学报, 2015, 26(Suppl. (2)): 247-256. <http://www.jos.org.cn/1000-9825/15035.htm>

英文引用格式: Fang MQ, Zhang WM, Gao C, Fang JB. Parallelizing and optimizing maximum noise fraction rotation on multi-cores and many-cores. Ruan Jian Xue Bao/Journal of Software, 2015, 26(Suppl. (2)): 247-256 (in Chinese). <http://www.jos.org.cn/1000-9825/15035.htm>

Parallelizing and Optimizing Maximum Noise Fraction Rotation on Multi-Cores and Many-Cores

FANG Min-Quan^{1,2}, ZHANG Wei-Min^{1,2}, GAO Chang¹, FANG Jian-Bin¹

¹(School of Computer, National University of Defense Technology, Changsha 410073, China)

²(Academy of Ocean Science and Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: Maximum noise fraction (MNF) rotation is a classical method of hyperspectral image dimensionality reduction, and it needs a large amount of calculation and thus is time-consuming. This paper investigates the code transplantation and performance optimization for the maximum noise fraction algorithm on multi-core CPU and many integrated core (MIC) architecture. By analyzing hotspots of the MNF algorithm, parallel schemes are first designed for filtering, covariance matrix calculating and MNF transforming. Then, a series of optimization methods are presented and validated for various parallel schemes of different hotspots, including using math kernel library (MKL) functions. Finally, a C-MNF algorithm on multi-cores CPUs and an M-MNF algorithm on the CPU/MIC heterogeneous system are constructed. Experiments show that the C-MNF algorithm achieves impressive speedups (ranging from 58.9 to 106.4), and the M-MNF parallel algorithm runs the fastest, reaching a maximum speed-up of 137X.

Key words: many integrated cores; multi-cores parallel; hyperspectral images dimensionality reduction; maximum noise fraction rotation; performance analysis for MKL

高光谱遥感影像数据具有波段连续、光谱分辨率高的特点, 能从其光谱空间中获取丰富的地物特征信息, 因其图谱合一的优势, 在农业、林业、军事、环境科学、地质等领域有广泛应用^[1]。在数据处理过程中, 连续波

* 基金项目: 国家自然科学基金(61272146, 41375113)

收稿时间: 2015-08-07; 定稿时间: 2015-10-12

段成像导致了高光谱影像数据量庞大,且连续波段之间数据相关性强,信息冗余大,存储处理困难,直接处理将导致维数灾难、Hughes 现象等严重的计算问题^[2-4].因此,数据降维应运而生.

怎样将高维空间数据映射到低维子空间,同时保持信息量基本不变是高光谱数据降维遵循的基本原则.高光谱影像降维过程涉及大量矩阵操作,如滤波、矩阵乘法、协方差计算等,是典型的计算密集型和访存密集型应用.传统的降维过程一般采用串行方式进行,计算复杂度高,耗时长,无法满足各应用领域对高光谱数据及时处理的需求.如何提高降维速度已成为重要研究热点,王和勇等人^[5]研究基于聚类和改进距离的局部线性嵌入方法可缩短 5.5 倍降维时间;而并行降维是加快降维过程的另一个重要研究方向.

2004 年以来,由于能耗限制,通过提高时钟频率提升 CPU 性能的方案彻底破产,多核 CPU 成为 CPU 发展主导方向^[6].2007 年 Nvidia 公司发布基于 NVIDIA GPU(graphic processing unit)的统一计算设备架构(compute unified device architecture,简称 CUDA),引领科学计算步入众核时代,基于 GPU 架构的天河 1A、泰坦等超级计算机相继成为 TOP500 榜首^[7].自 Intel 公司 2012 年推出集成众核产品 Xeon Phi^[8],集成众核架构成为目前高性能计算领域的研究热点,搭载 MIC 的天河 2 号超级计算机已连续 5 届蝉联 TOP500 榜首^[7].

高光谱影像并行处理在传统并行系统已有成熟的应用:Valencia 等人^[9]基于异构 MPI(message passing interface)在网络机群系统研究了高光谱影像处理的技术;Plaza 等人^[10]提出基于神经网络的高光谱影像并行分类算法.基于 CPU/GPU 异构系统,高光谱影像并行处理也有部分研究:Sánchez 等人^[11]对高光谱解混进行了 GPU 移植;Platoš 等人^[12]和 Ramalho 等人^[13]分别用 GPU 加速了非负矩阵分解和独立成分分析等降维方法.但基于 CPU+MIC 这一新型高性能体系结构研究相对较少.在文献[14,15]中,我们分别在 GPU 和 MIC 上探讨了高光谱影像主成分分析(principal component analysis,简称 PCA)并行降维,利用 2 个 GPU 获得 128 倍加速比,在 3 个 MIC 上加速 133 倍.本文将面向多核与众核架构继续研究加速高光谱影像降维.

最大噪声分数变换^[16]分离原始数据中的噪声信息,进而提取影像数据的主要特征,表征主要信息.该算法由两层主成分变换构成,在主成分分析的基础上考虑了噪声对图像的影响,具有更好的效果,是目前主流的线性降维算法.

本文以最大噪声分数变换算法为对象,面向多核 CPU 与众核 MIC 架构展开并行算法与性能优化研究,本文贡献包括:1) 分析算法加速热点并设计相应的并行方案;2) 针对各热点并行方案提出并验证了多种建设性的优化策略;3) 设计并实现了基于 MKL 库的并行方案,量化测评 MKL 库函数性能瓶颈;4) 提出并实现了基于多核 CPU 的 C-MNF 和基于 CPU/MIC 的 M-MNF 并行降维算法.

1 集成众核体系结构与优化策略

1.1 集成众核体系结构

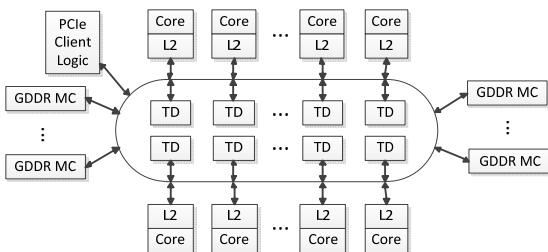


图 1 MIC 架构

英特尔集成众核(MIC)架构集成了超过 50 个 x86 核心,由片上高速环形互联总线联接(如图 1 所示).MIC 共有 8 个基于 GDDR5 的存储控制器,16 通道,峰值带宽可达 350GB/s^[8].图中 TD(tag directory)为标签目录,其为每个核的 L2 Cache 建立标识目录副本,从而全局监听所有核的 L2 Cache.MIC 的每个计算核心拥有 1 个 512b 的向量处理单元(vector processing unit,简称 VPU)、1 个 x86 架构标量处理单元、32KB 的 L1 cache(指令和数据各 1 个)和 512KB 的 L2 cache.MIC 支持硬件多线程技术,每个核心有 4 个硬件线程,每 2 个时钟发射 1 条指令^[8].

1.2 集成众核优化策略

MIC 是众核协处理器,要获得最优性能,可以采取的优化策略包括最大化并行度、合理利用内存空间、最小化数据通信、优化 Cache 访问、向量化、均衡负载等.对于 CPU/MIC 异构系统,如何在 CPU 和 MIC 间合理分配计算任务,如何设计两者间通信,这些都是异构系统优化中所必须考虑的重要因素.在异构系统中,CPU 和 MIC 可以同时计算,另外 CPU 和 MIC 擅长的计算有所差异,因此需要合理分配计算任务.任意器件计算与两者间通信可以同时进行,而数据存在位置将直接影响程序正确性,因此需要合理设计通信,在保证程序正确基础上,利用计算通信重叠来掩盖重复开销.

2 最大噪声分数变换热点分析

2.1 最大噪声分数变换

最大噪声分数变换(MNF)根据图像质量进行数据降维和排列,其采用 SNR 和噪声比例对图像质量进行评价.MNF 变换通过寻找变换矩阵,使得变换后的图像按信噪比递减顺序排列,从而达到降维的目的.

对于高光谱影像数据 $X(W \times H \times B)$,宽 W 、高 H 、波段 B , X 可看成是 B 行 S 列($S=W \times H$)的 2 维矩阵,矩阵的 1 行表示 1 个波段的高光谱影像数据.通过 MNF 方法降维,可以获得 m 个成分,其中 $m < B$,从而实现降维目的.

算法 1. 高光谱影像降维 MNF 方法.

Step 1. 对高光谱矩阵 X 滤波;

Step 2. 计算滤波后的噪声协方差矩阵 C_N ;

Step 3. 对 C_N 特征分解, $D_N=U^T C_N U$. (1)

其中 D_N 为降序排列的特征值, U 为所对应的特征向量,得到变换矩阵 $P=U D_N^{-1/2}$; (2)

Step 4. 求原始高光谱 X 的协方差矩阵 C_D ;

Step 5. 对 C_D 变换: $C_{D-adj}=P^T C_D P$; (3)

Step 6. 对 C_{D-adj} 特征值分解, $D_{D-adj}=V^T C_{D-adj} V$. (4)

其中 D_{D-adj} 为降序排列的特征值, V 为所对应的特征向量;

Step 7. 计算变换矩阵 $T=PV$; (5)

Step 8. MNF 变换 $Z=T^T X$. (6)

2.2 热点分析

运行串行高光谱影像 MNF 降维算法,对 $W=614, H=1087, B=224$ 数据降维,测试并统计各步骤占总计算时间的比重(如图 2 所示).图 2 数据显示, Step 2 和 Step 4 中协方差矩阵计算占比最大超过 80%, Step 1 的滤波和 Step 8 的 MNF 变换耗时明显.因此,这些将是 MNF 算法并行移植和性能优化的研究热点.

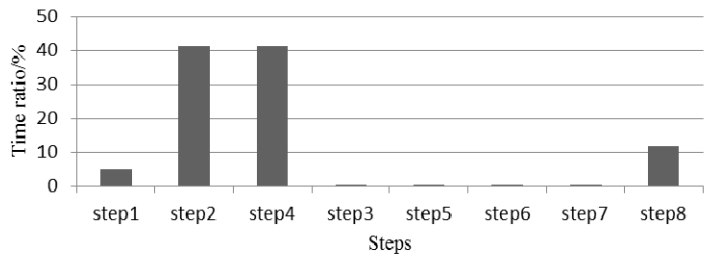


图 2 串行 MNF 时间分布

3 并行滤波与优化

3.1 滤波并行方案设计

滤波种类丰富,均值滤波是其中常见的也是性能较好的滤波方法之一.本文采用的就是均值滤波,求取目标点以及周围 8 个点的平均值,用目标点值与平均值求差的绝对值,即为均值滤波结果.计算公式为

$$FX[i][j]=abs(X[i][j)-(X[i-1][j-1]+X[i-1][j]+X[i-1][j+1]+X[i][j-1]+X[i][j]+X[i][j+1]+X[i+1][j-1]+X[i+1][j]+X[i+1][j+1])/9) \quad (7)$$

从均值滤波公式中可以看出,图像各点的滤波处理相互独立,不存在依赖关系,因此图像中所有像元点都能并行计算,且高光谱影像不同波段也能并行计算.丰富的并行度使其不仅适合 CPU 多核并行计算,还适合在众核协处理器 MIC 上做并行运算.

3.2 向量化优化及其效果

向量化是 x86 架构下多核与众核平台最重要的优化手段之一.通过查看向量化报告(编译选项-vec-report)知道本文初始代码由于存在依赖,导致滤波计算的最内层循环无法向量化.通过分析依赖并调整代码后,达到了向量化目的.

表 1 展示了滤波过程的向量化效果,实验数据为 $W=614, H=1087, B=224$ 的高光谱数据,其中 CPU OMP 是 16 核 CPU 并行运算, MIC OMP 为 57 核并行运算(下同).表中数据显示,向量化效果显著,向量化分别为串行滤波、CPU 多核滤波和 MIC 众核滤波提供了 6.5、4.7 和 1.7 倍加速比.对比初始串行版本滤波器, CPU 多核并行加速 68 倍,而 MIC 众核并行最高加速 12 倍.

表 1 滤波向量化效果 (ms)

224×614×1087	Serial	CPU OMP	MIC OMP
Original	3142.70	214.62	454.66
Vectorization	483.15	45.91	263.74

4 协方差矩阵并行运算与优化探索

4.1 协方差矩阵计算并行方案

协方差矩阵中单一元素的计算公式为 $COV(X, Y) = \frac{1}{s-1} \left[\sum_{i=1}^s (X_i - \bar{X})(Y_i - \bar{Y}) \right]$. 不同矩阵元素的协方差值计算相互独立,因此整个协方差矩阵的所有元素运算都能并行进行.由于协方差矩阵是对称矩阵,仅需计算下三角阵的协方差,然后在对称的上三角阵填入对应的值.对最外层循环进行任务划分时,由于是三角矩阵,将任务调度过程设定为动态调度,当线程数远小于 B 时,能获得较好的并行度和负载均衡;当线程数与 B 相当或比 B 大时,这种方案就会导致负载不均衡问题.

4.2 协方差计算分解及其效果

协方差计算相对比较复杂,也无法直接使用 MKL 库函数.本文设计协方差计算的分解,以达到复杂运算简单化和合理利用 MKL 库的目的.

协方差计算公式有以下变形:

$$COV(X, Y) = \frac{1}{s-1} \left[\sum_{i=1}^s (X_i - \bar{X})(Y_i - \bar{Y}) \right] = \frac{1}{s-1} \left[\begin{array}{l} \left(\sum_{i=1}^{s/N} X_i Y_i + \sum_{i=s/N+1}^{2s/N} X_i Y_i + \dots + \sum_{i=(N-1)s/N+1}^s X_i Y_i \right) \\ - \frac{1}{s} \left(\sum_{i=1}^{s/N} X_i + \sum_{i=s/N+1}^{2s/N} X_i + \dots + \sum_{i=(N-1)s/N+1}^s X_i \right) \\ * \left(\sum_{i=1}^{s/N} Y_i + \sum_{i=s/N+1}^{2s/N} Y_i + \dots + \sum_{i=(N-1)s/N+1}^s Y_i \right) \end{array} \right] \quad (8)$$

通过该变形,协方差运算可转换为求和与求内积运算.经上述转换后,协方差运算可采用 MKL 库实现,具体实现方法见第 4.4 节.

分别实现协方差公式变形前后的串行版本、CPU 多核并行版本和 MIC 众核并行版本,处理 $W=614, H=1087, B=224$ 的高光谱数据,结果见表 2.表中数据显示:协方差计算分解能加速 1.5 倍左右;本文设计的并行方案效果明显,其中 MIC 众核并行版本性能最好,比最初的串行版本加速 46 倍.

表 2 协方差计算分解效果 (ms)

224×614×1087	Serial	CPU OMP	MIC OMP
Before	28 052.98	3 395.79	959.27
After	16 652.45	2 066.82	608.85

4.3 数据类型对计算性能的影响

高光图像数据可用 unsigned char(uchar)类型数据表示,而如果要利用 MKL 库或 Intel 提供的 Intrinsics 指令集进行优化,都要求 float 类型.因此本节分别探讨这两种数据类型在协方差矩阵计算过程中的性能表现.

从理论上分析,unsigned char 类型占 1Byte 存储空间,而 float 类型需要 4Byte 存储空间.无论是访存速度还是内存容量上,unsigned char 类型都较有优势.但 unsigned char 类型无法支持 MKL 等函数库和底层的 Intrinsics 指令集.

下面通过实验对比两者性能表现,分别实现相应的代码,对 $W=614, H=1087, B=224$ 的高光谱数据处理,统计得到表 3 性能数据.表 3 显示,串行时 float 类型性能占优,而 CPU 多核并行和 MIC 众核并行都是 unsigned char 类型性能更佳,特别是众核 MIC 上这种性能提升尤为明显.

表 3 数据类型对性能的影响 (ms)

224×614×1087	Serial	CPU OMP	MIC OMP
Uchar	16 652.45	2 066.82	608.85
Float	9 949.74	4 942.76	1 605.57

这种现象的主要原因可归结为访存带宽的限制.在串行执行时,由于单核计算所需的数据量有限,访存带宽完全能满足需求,而 float 在编译时可能向量化优化更深入,且 uchar 类型参与计算时需要数据转换,因此串行 float 版本更快;而所有核一起运算时,访存带宽无法满足众多 CPU 核心计算时的访存需求,float 型计算比 uchar 型计算需要访问的存储空间更多,因此在并行时 float 版本比 uchar 版本慢.

4.4 协方差矩阵计算的两种MKL库实现及效果

根据第 4.2 节的协方差计算分解公式(8),将协方差计算转化为求和与求内积运算,因此协方差矩阵运算可转化为 MKL(BLAS)库中的 asum 运算(向量加法运算)和 syrkc 运算(可转化为 gemm 运算).

Syrkc 运算($C := \alpha \times A \times A^T + \beta \times C$)转换为 gemm 运算($C := \alpha \times \text{op}(A) \times \text{op}(B) + \beta \times C$)的思想为:将 gemm 运算中 B 矩阵用 A 的转置矩阵替换,即可实现功能上的转换.从计算量的角度看,这种转换是不划算的,因为 syrkc 运算只计算上三角阵或下三角阵,运算量是 gemm 的一半.但运行性能并不仅仅是由计算量决定的,还与其实现代码息息相关.在后文实验结果中可知,MKL 库在 MIC 上执行时,gemm 运算性能远远高于 syrkc 运算性能.

图 3 和图 4 分别是处理两组高光图像数据时统计得到的两个 MKL 库函数的耗时数据和本文书写代码的耗时数据.

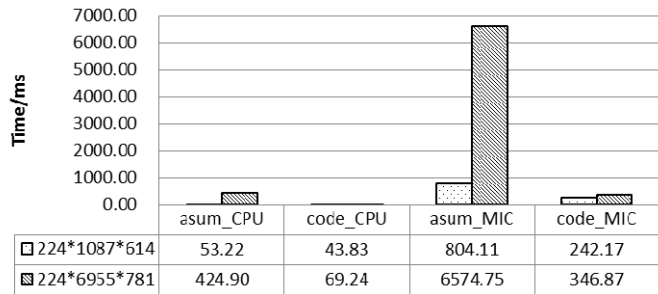


图 3 求和运算耗时

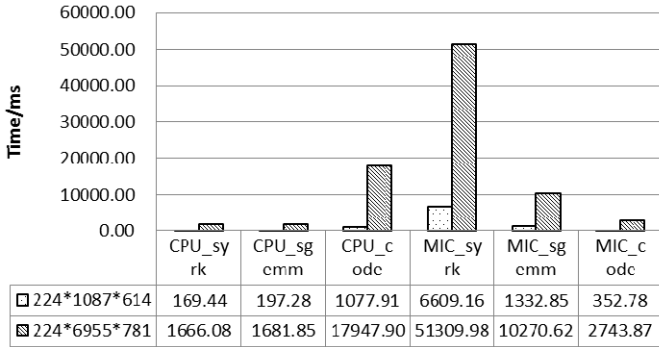


图4 内积运算耗时

函数性能远低于 sgemm 函数,可能是该函数没有针对 MIC 众核架构进行深入优化.MIC 版本以本文开发的代码性能最好.

4.5 深入探究MKL库性能表现

从前文的结果中发现,浮点计算能力强的 MIC 上的 MKL 数学库性能不如 CPU 上的 MKL 库,与 Intel 公布的产品图表数据不符.

针对该问题,本文通过一系列实验进行分析.分别在 CPU 和 MIC 上统计 MKL 库中 sgemm 函数第 1 次调用执行时间和其后连续 3 次调用执行时间,并计算得到相应的浮点计算能力(图 5),其中参与运算的矩阵为方阵.

图 5 数据显示:无论 CPU 还是 MIC,其第 1 次 MKL 库函数性能较差,第 2 次之后 MKL 库函数调用才能发挥正常性能,且这种性能差距在 MIC 上尤为明显;

对比 CPU 和 MIC 的 sgemm 平均性能数据,可知多核 CPU 比众核 MIC 更加适合处理运算量较小的任务,而当计算量超过一定阈值后,MIC 才能发挥出更好的性能.

基于上述结论,在基于 x86 结构的多核 CPU 与众核 MIC 平台上优化程序时,不能盲目使用 MKL 库,需要具体分析其性能表现.比如在某应用中,仅需调用一次 MKL 函数时,程序员需要斟酌库函数首次调用开销与自己开发代码的耗时,力求获得最优解.比如在本文设计的 M-MNF 并行算法中,两个协方差矩阵运算分别在 CPU 和 MIC 执行,每个部件中都存在首次 MKL 函数启动开销,综合考虑图 3 和图 4 的性能对比后,最终选择在 MIC 上采用本文开发的并行代码计算协方差矩阵,而非使用 MKL 库函数.

5 并行 MNF 变换及其优化研究

5.1 并行MNF变换方案

MNF 变换($Z=T^T X$)实质是矩阵乘法运算,因此并行时可采用矩阵乘法并行方案.但是 MNF 变换与一般矩阵乘法又不尽相同,其各个矩阵维度上有些特别,比如 T 矩阵的尺寸是 (B,m) , X 矩阵尺寸为 (B,S) , m 一般为十几或几十,本文数据中 B 为 224, S 非常大,可以达到几百上千万.因此简单的并行矩阵乘法很难为并行程序提供足够的并行度,特别是在集成众核上运算时.因此本文设计了一系列的优化措施来提高并行度、加快运算速度.

图 3 数据显示,在协方差矩阵计算过程中,利用 MKL 库中的 asum 函数在 CPU 和 MIC 上耗时都比本文书写代码长.特别是 MIC 上实现的 asum 函数,明显存在性能漏洞.这主要是由于在协方差计算过程中需要每个波段都求像元和,共 B 次 asum 运算,而并行隐含在 asum 运算中,频繁的 fork-join 会导致性能缺陷.

图 4 数据显示,CPU 版本的 MKL 库性能最好,其中 syrk 函数确实如前文分析比 sgemm 快(计算量少).而 MIC 版本中 syrk

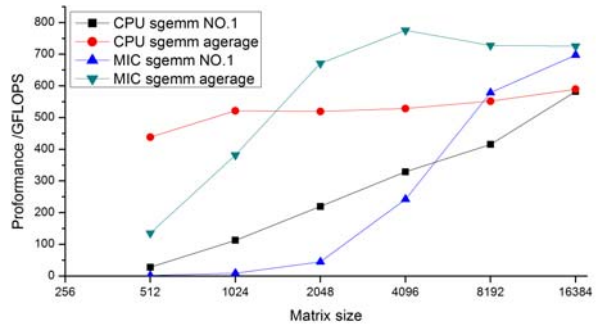


图5 MKL 库函数性能瓶颈

5.2 并行MNF优化探索

针对 MNF 变换过程,本文做了以下优化尝试.

1) 串行 MNF 的连续访存.

串行优化是并行优化的基础,通常来说,只有在每个核上最优执行,所有核心的性能才能最优.原始 MNF 变换过程尽管在编译时的向量化报告显示已经向量化,但由于访存的不连续导致了巨大的性能损失.本文针对这个问题,提出了两种优化方法:①通过转置将访存的数据颠倒,达到顺序访存的目的;②通过循环交换改变数据访问次序,以实现连续访存.

实现上述两种优化方法,测试并统计时间,如图 6 所示,图中 uchar_0(float_0)表示未优化,uchar_1(float_1)表示采用优化方法①,uchar_2(float_2)表示采用优化方法②,图中分别统计了使用 uchar 类型和 float 类型及不同优化方法时串行 MNF 变换的运行时间.由图可知,这两优化方法效果显著,其中方法 1 中引入的转置开销比重较大,而方法 2 引入了置零开销很小,几乎可忽略.从总时间看,采用 uchar 类型和循环交换实现连续访存的方法效果最好.

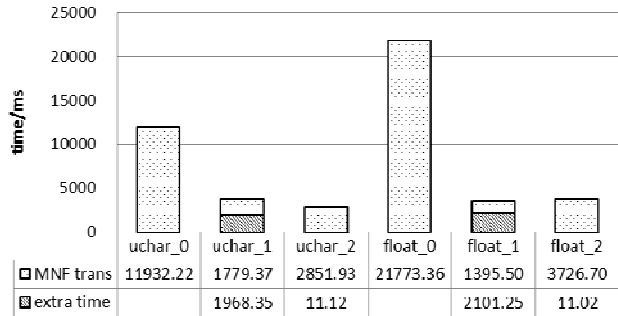


图 6 连续访存优化效果

2) 并行循环合并优化.

由于 MNF 变换问题本身的限制,最外层循环的并行度 m 一般为十几或几十,无法满足众核并行的并行度需求,还会造成 CPU 多核并行的负载不均衡.因此本文采用了循环合并的优化方法.在保持最内层循环不变的前提下(之前的优化已确保向量化和连续访存),合并外层循环.

实现上述优化,并测试对比优化前后性能,发现该优化对 CPU 多核并行性能影响不大,而对 MIC 众核并行性能影响显著,表 4 记录了 MIC 众核并行版本的循环合并优化前后时间消耗,结果数据已包含转置(连续访存优化方法①)和循环交换(连续访存优化方法②)引入的额外开销.表中结果显示循环合并优化效果显著,加速 3.9~5.7 倍不等.

表 4 并行循环合并效果 (ms)

MIC	Transposition①		Exchange loop②	
	Before	After	Before	After
Uchar	2 324.58	592.11	3 894.30	858.29
Float	3 890.59	680.80	3 911.93	850.77

5.3 MKL库实现与性能分析

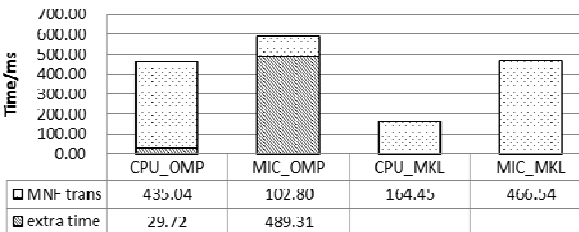


图 7 MNF 变换中 MKL 库性能

MNF 变换过程类似矩阵乘法,可以利用 MKL 库中 sgemm 函数实现.本文利用该函数分别实现 CPU 多核并行和 MIC 众核并行两个版本,并在实验平台上测试.统计实验结果并对本文手写最优代码测试数据(如图 7 所示).图中数据显示,CPU 端 MKL 库性能最佳;而单计 MNF 变换时间,本文手写代码 MIC 版本时间最短,但存在可观的额外开销.

6 多核/众核上并行 MNF 算法

6.1 C-MNF并行算法

在第 3-5 章各热点的并行优化结果对比中,可以知道滤波过程的 CPU 多核版本、协方差矩阵计算的 CPU 多核 MKL 版本和 MNF 变换的 CPU 多核 MKL 版本都是各个热点最快的并行方案.将这几个版本嵌入 MNF 算法中,其中需要加入部分数据类型转换操作以配合 MKL 库的数据类型,此时恰好只用到了多核 CPU 做计算,该版本可视为最快的 CPU 多核并行 MNF 算法,记为 C-MNF.

6.2 M-MNF并行算法

在多核/众核异构系统上设计算法时,一个重要的优化思想是合理利用多核 CPU 和众核 MIC 资源,针对各个计算步骤,选取最佳并行方案,在此基础上还需要合理设计流程,尽可能多地实现计算与通信重叠、计算(多核)与计算(众核)重叠.

CPU 端数据读入、MIC 初始化开销(启动开销和存储分配开销)都比较耗时,且分别在不同设备上进行,可以并行执行.

在 MNF 计算流程中,有两条线路可以并行执行: 1) 对原始数据滤波,求噪声矩阵的协方差矩阵,对噪声协方差矩阵进行特征分解(Step1-Step3);2) 求原始数据的协方差矩阵(Step4).这 2 条线路恰好对应 CPU 和 MIC,本文在 CPU 上执行线路 1, MIC 上执行线路 2.据此,本文基于 CPU/MIC 系统设计了 M-MNF 并行算法,如图 8 所示.

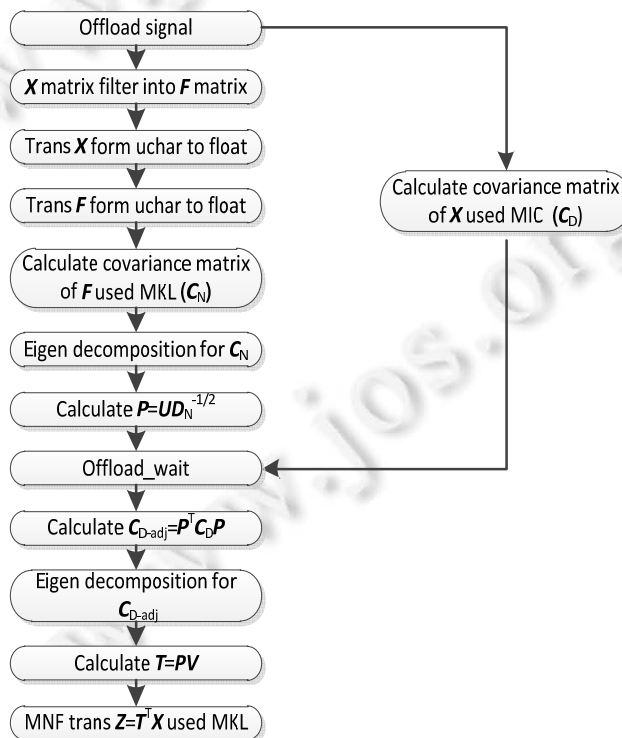


图 8 M-MNF 并行算法流程

7 实验结果与分析

7.1 实验准备

本文所用测试平台包含两个 8 核 Intel(R) Xeon(R) CPU E5-2670 和一个 57 核 Intel Phi 31S1P 协处理器,基于并行方案和优化策略,采用 OpenMP 和 LEO(language extensions for offload)的 C 语言扩展来实现并行算法,采用 Intel C Compiler 13.0.0 编译器编译.

表 5 罗列了本文采用的 AVIRIS 高光谱影像数据基本信息,共 3 组.

表 5 高光谱影像数据详细信息

Data	Width	Height	Band
1	614	1087	224
2	753	1924	224
3	781	6955	224

7.2 并行算法加速比

本文除了实现 C-MNF 和 M-MNF 并行算法外,还实现了一种 O-MNF 原始并行版本.该版本在原始 MNF 基础上利用 OpenMP 并行化加速热点,而未采用文中提及的优化策略.处理 3 组高光谱影像数据,统计 MNF 降维时间(见表 6),计算并行算法加速比(见表 7).

表 6 各版本 MNF 执行时间 (ms)

Data	Serial	O-MNF	C-MNF	M-MNF
1	62 393.56	7 752.66	1 058.68	953.83
2	141 221.63	17 228.51	1 715.13	1480.84
3	538 294.40	64 885.74	5 058.54	3927.96

表 7 并行 MNF 加速比

Data	O-MNF	C-MNF	M-MNF
1	8.0	58.9	65.4
2	8.2	82.3	95.4
3	8.3	106.4	137.0

表 7 数据显示,采用本文并行方案设计的 O-MNF 并行算法可获得 8 倍左右的加速比;而在并行方案基础上,采取文中阐述的优化策略,可最终在多核 CPU 上获得 58.9~106.4 倍不等的加速比;同时使用了多核 CPU 和众核 MIC 的 M-MNF 并行算法性能最好,最高可加速 137 倍.

7.3 并行算法瓶颈分析

统计 M-MNF 降维算法各个步骤消耗时间占总时间的百分比,绘制成图 9(其中 uc2f 表示本文优化引入的 unsigned char 转换为 float 开销).对比图 9 和图 1,发现并行后各步骤耗时比重有所改变,但协方差矩阵计算依旧是算法性能瓶颈.针对这一瓶颈,可以通过扩展计算资源解决,比如使用 MIC 集群,这也是下一步工作的重点.

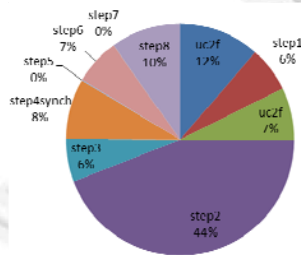


图 9 M-MNF 时间分布

8 结论

本文基于多核 CPU 和众核 MIC 研究高光谱影像降维 MNF 方法,设计了滤波、协方差矩阵计算、MNF 变换等热点并行方案和优化策略;提出和实现了基于多核 CPU 的 C-MNF 和基于 CPU/MIC 的 M-MNF 并行算法.实验结果显示 C-MNF 算法在多核 CPU 上加速 58.9 倍~106.4 倍不等,而基于 CPU/MIC 异构系统的 M-MNF 算法性能最好,最高可加速 137 倍.本文研究显示,高光谱影像 MNF 降维算法可在多核与众核上取得良好的性能,但仍存在性能瓶颈,可考虑进一步将算法扩展到集群以突破瓶颈.

References:

- [1] Green RO, Eastwood ML, Sarture CM, Chrien TG, Aronsson M, Chippendale BJ, Faust JA, Pavri BE, Chovit CJ, Solis M, Olah MR, Williams O. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer(AVIRIS). *Remote Sensing of Environment*, 1998,65(3):227–248. [doi:10.1016/S0034-4257(98)00064-9]
- [2] Green AA, Berman M, Switzer P, Craig MD. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Trans. on Geoscience and Remote Sensing*, 2000,26(1):65–74. [doi:10.1109/36.3001]
- [3] Kaarna A, Zemcik P, Kalviainen H, Parkkinen J. Compression of multispectral remote sensing images using clustering and spectral reduction. *IEEE Trans. on Geoscience and Remote Sensing*, 2000,38(2):1073–1082. [doi:10.1109/36.841986]
- [4] Scott DW, Thompson JR. Probability density estimation in higher dimensions. In: *Computer Science and Statistics: Proc. of the 15th Symp. on the Interface*. Amsterdam: North-Holland, 1983. 173–179.
- [5] Wang HY, Zheng J, Yao ZA, Li L. Application of dimension reduction on using improved LLE based on clustering. *Journal of Computer Research and Development*, 2006,43(8):1485–1490 (in Chinese with English abstract).
- [6] Wikipedia. Multi-core processor. 2015. http://en.wikipedia.org/wiki/Multi-core_processor
- [7] Meuer H, Strohmaier E, Dongarra J. TOP500 supercomputer sites. 2015. <http://www.top500.org/>
- [8] Jeffers J, Reinders J, Wrote; Chen Jian, Li Hui, Yang Kun, *et al.*, *Trans. Intel Xeon Phi Coprocessor High Performance Programming*. Beijing: Posts & Telecom Press, 2014 (in Chinese).
- [9] Valencia D, Lastovetsky A, O'Flynn M, Plaza A, Plaza J. Parallel processing of remotely sensed hyperspectral images on heterogeneous networks of workstations using HeteroMPI. *Int'l Journal of High Performance Computing Applications*, 2008,22(4):386–407. [doi:10.1177/1094342007088377]
- [10] Plaza J, Plaza A, Pérez R, Martínez P. Parallel classification of hyperspectral images using neural networks. *Computational Intelligence for Remote Sensing Studies in Computational Intelligence*, 2008,133:193–216. [doi: 10.1007/978-3-540-79353-3_8]
- [11] Sánchez S, Ramalho R, Sousa L, Plaza A. Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs. *Journal of Real-Time Image Processing*, 2012,10(2):469–483. [doi: 10.1007/s11554-012-0269-2]
- [12] Platoš J, Gajdoš P, Krömer P, Snašel V. Non-negative matrix factorization on GPU. In: *Proc. of the 2nd Int'l Conf. on Networked Digital Technologies*. Heidelberg: Springer-Verlag, 2010. 21–30. [doi: 10.1007/978-3-642-14292-5_4]
- [13] Ramalho R, Tomas P, Sousa L. Efficient independent component analysis on a GPU. In: *Proc. of the 10th IEEE Int'l Conf. on Computer and Information Technology*. Piscataway: IEEE Computer Society, 2010. 1128–1133. [doi: 10.1109/CIT.2010.205]
- [14] Fang MQ, Zhou HF, Shen XL. Multilevel parallel algorithm of PCA dimensionality reduction for hyperspectral image on GPU. *Dongbei Daxue Xuebao/Journal of Northeastern University*, 2014,35(S1):238–243 (in Chinese).
- [15] Fang MQ, Zhang WM, Zhang LL, Wu H, Fang JB. Multilevel parallel algorithms and performance optimization of hyperspectral image dimensionality reduction on MIC. *Dongbei Daxue Xuebao/Journal of Northeastern University*, 2014,35(S1):25–30,36 (in Chinese).
- [16] Gordon, C. A generalization of the maximum noise fraction transform. *IEEE Trans. on Geoscience and Remote Sensing*, 2000, 38(1):608–610. [doi: 10.1109/36.823955]

附中文参考文献:

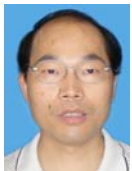
- [5] 王和勇,郑杰,姚正安,李磊.基于聚类和改进距离的LLE方法在数据降维中的应用. *计算机研究与发展*,2006,43(8):1485–1490.
- [8] James J, James R,著;陈健,李慧,杨昆,等,译. *Intel Xeon Phi 协处理器高性能编程指南*.北京:人民邮电出版社,2014.
- [14] 方民权,周海芳,申小龙.基于GPU的高光谱PCA降维多级并行算法. *东北大学学报(自然科学版)*,2014,10,35(S1):238–243.
- [15] 方民权,张卫民,张理论,伍恒,方建滨.面向MIC的高光谱影像降维多级并行算法及性能优化. *东北大学学报(自然科学版)*,2014, 35(S1):25–30,36.



方民权(1989—),男,浙江东阳人,博士生,CCF 学生会员,主要研究领域为高性能计算,异构计算.



高畅(1990—),女,硕士生,CCF 学生会员,主要研究领域为高性能计算,异构计算.



张卫民(1966—),男,博士,研究员,主要研究领域为大气资料同化,并行算法.



方建滨(1984—),男,博士,助理研究员,主要研究领域为异构计算,并行编程模型,性能优化.