

面向 MIC 的高光谱影像降维多级并行算法及性能优化

方民权¹, 张卫民¹, 张理论¹, 伍恒², 方建滨^{1,3}

(1.国防科学技术大学 计算机学院, 湖南 长沙 410073; 2.德州理工大学 计算机系, 德克萨斯 拉巴克 79409; 3.代尔夫特理工大学 电气电子数学与计算机学院, 代尔夫特 2628CD)

摘要: 高光谱影像降维是经典的计算密集型和存储密集型应用, 本文旨在研究经典的主成分分析降维方法, 面向集成众核架构提出新的并行算法加速高光谱降维过程。通过分析加速热点, 研究协方差矩阵计算和 PCA 变换两个加速热点的并行方案和优化策略, 提出一种单 MIC 的 M-PCA 并行算法和一种多 MIC 的多级并行 Ms-PCA 算法, 并提出并行算法性能模型。实验结果表明, 两种算法都获得了显著的加速效果, Ms-PCA 算法对比 CPU 串行最高可获得 133 倍的加速比, 比 24 核 CPU 多线程加速 6.2 倍; 结果验证了 M-PCA 和 Ms-PCA 两种性能模型是可靠的。

关键词: 集成众核; 多级并行; 高光谱影像降维; 主成分分析; 高性能计算; 性能优化

中图分类号: TP391

文献标志码: A

Multilevel Parallel Algorithms and Performance Optimization of Hyperspectral Image Dimensionality Reduction on MIC

FANG Min-quan¹, ZHANG Wei-min¹, ZHANG Li-lun¹, Wu Heng², Fang Jian-bin^{1,3}

(School of Computer Science, National University of Defence Technology, Changsha, 410073, China; Department of Computer Science, Texas Tech University, Rabac, 79409, USA; EEMCS, Delft University of Technology, Delft, 2628CD, Netherlands. Corresponding author: FANG Min-quan, E-mail: 877086820@qq.com)

Abstract: Dimension reduction for hyperspectral image is computationally expensive and memory accessing intensive. Based on the classical Principal Component Analysis (PCA) method, we propose novel algorithms for Many Integrated Core(MIC). By analyzing the accelerating hotspot and researching the parallelism schemes and optimizing strategies of COV and PCA transform, We implemented and validated A M-PCA algorithm for a single MIC and a Ms-PCA algorithm for multiple MICs. Further, we presented performance models for M-PCA and Ms-PCA. Results show that the algorithms can remarkably enhance the performance: the Ms-PCA algorithm can reach a maximum speed-up of 133X comparing with the CPU serial version and a speed-up of 6.3X comparing with the CPU multithreaded version on 24 cores. The results validate our models with an acceptable accuracy.

Key words: Many Integrated Core; Multilevel Parallel; Hyperspectral Image Dimensionality Reduction; Principal Component Analysis; High Performance Computing; Performance Optimization

高光谱遥感技术广泛应用于军事、农业、环境科学、地质、海洋学等领域, 这些领域大都要求及时处理^[1-2]。但高光谱数据有波段多、数据量大、相关性高、冗余多等特点, 直接处理将导致样本类别训练困难、维数灾难、空空间现象等严重的计算问题^[3-5]。因此, 降维是高光谱处理过程中必不可少的步骤。由于高光谱数据的特点, 降维过程复杂且耗时。

Intel 公司于 2012 年推出集成众核 (Many Integrated Core, MIC) 产品, MIC 已成为目前高性能

计算领域的研究热点, 搭载 MIC 的天河 2 号超级计算机目前占据 TOP500 榜首^[6]。

高光谱影像并行处理在传统并行系统已有成熟的应用, David^[7]基于异构 MPI 在网络集群系统研究了高光谱影像处理的技术; Javier 等^[8]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, Sergio Sanchez^[9] 和 Mahmoud ElMaghrbay^[10] 对高光谱解混进行了 GPU 移植, He Yang 等^[11]在多 GPU 实现高光谱影像快速波段选择, 但基于 CPU+MIC

收稿日期: 2014-03-30

基金项目: 国家自然科学基金项目(41375113)、国家自然科学基金项目(61272146)

作者简介: 方民权 (1989 -), 男, 浙江东阳人, 国防科技大学博士研究生; 张卫民 (1966 -), 男, 浙江诸暨人, 国防科技大学研究员, 博士生导师。张理论 (1975 -), 男, 河南南阳人, 国防科技大学研究员。

这一新型高性能体系结构还没有研究。因此，本文面向集成众核架构探讨加速高光谱降维的可行性。

主成分分析 (Principal Component Analysis, PCA) [1] 是经典的高光谱遥感影像降维方法。本文面向集成众核架构展开研究，提出并实现一种单 MIC 的 M-PCA 并行算法和一种多 MIC 的 Ms-PCA 多级并行算法，提出这两种算法的性能预测模型。

本文结构如下，第 1 部分分析 PCA 算法加速热点；第 2 部分阐述各加速热点的并行策略，提出 M-PCA 和 Ms-PCA 两种并行算法；第 3 节提出并行算法性能模型；第 4 节阐述集成众核优化策略及效果；第 5 节通过实验分析算法性能，验证并行算法性能模型；最后是总结和展望。

1 PCA 降维算法与加速热点分析

1.1 高光谱影像 PCA 降维算法

设 $\mathbf{X}=(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_S)=(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_B)^T$ 表示高光谱遥感图像，其中 S 表示高光谱遥感图像的象元数目， B 表示波段数目。下面简单对 PCA 降维算法进行描述：1) 首先计算 \mathbf{X} 中各波段间的相互协方差，计算其协方差矩阵；2) 求解协方差矩阵的特征值和特征向量。根据特征值计算贡献率 v ，并根据给定的阈值 R (本文取 99%)，按特征值从大到小选取主成份个数 m ，使得累计贡献率 $\sum v \geq R$ ；3) 对选中的主成份进行 PCA 变换。令 $\mathbf{A}=\mathbf{T}^T$ ，PCA 变换结果矩阵 $\mathbf{Y}=(\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_m)^T=\mathbf{A}\mathbf{X}$ 。通过 PCA 降维，高光谱遥感影像数据从 B 个波段降为 m 个波段，大大减少了后续高光谱影像处理的计算量。

1.2 加速热点分析

测试串行 PCA 降维过程中各步骤的时间消耗，计算其占总时间的百分比 (图 1)。协方差矩阵计算 (cov) 和 PCA 变换 (PCA trans) 的比率最大，本文主要对这两个步骤展开 MIC 并行化研究。

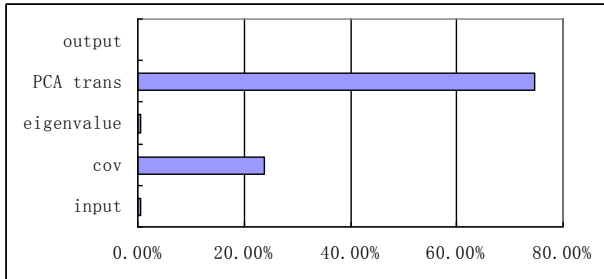


图 1 PCA 时间分布

Fig.1 Time distribution of PCA

2 PCA 集成众核并行算法

2.1 协方差矩阵任务划分

协方差计算公式有以下变形：

$$\begin{aligned} \text{COV}(\mathbf{X}, \mathbf{Y}) &= \frac{1}{s-1} \left[\sum_{i=1}^s (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{Y}_i - \bar{\mathbf{Y}}) \right] \\ &= \frac{1}{s-1} \left[\left(\sum_{i=1}^{s/N} \mathbf{X}_i \mathbf{Y}_i + \sum_{i=s/N+1}^{2s/N} \mathbf{X}_i \mathbf{Y}_i + \dots + \sum_{i=(N-1)s/N+1}^s \mathbf{X}_i \mathbf{Y}_i \right) \right. \\ &\quad \left. - \frac{1}{s} \left(\sum_{i=1}^{s/N} \mathbf{X}_i + \sum_{i=s/N+1}^{2s/N} \mathbf{X}_i + \dots + \sum_{i=(N-1)s/N+1}^s \mathbf{X}_i \right) \right. \\ &\quad \left. * \left(\sum_{i=1}^{s/N} \mathbf{Y}_i + \sum_{i=s/N+1}^{2s/N} \mathbf{Y}_i + \dots + \sum_{i=(N-1)s/N+1}^s \mathbf{Y}_i \right) \right] \end{aligned} \quad (1)$$

上述变形将协方差的计算任务转换为 $\sum \mathbf{X}$ 和 $\sum \mathbf{X}\mathbf{Y}$ ，又可分割为 $\sum \mathbf{X}_i$ 和 $\sum \mathbf{X}_i \mathbf{Y}_i$ ，将其分配给不同的进程。各进程完成计算后将结果发送给 root 进程计算最终协方差矩阵。整个过程仅需 1 次通信 (图 2)。

每个进程可创建多个线程，协同计算 $\sum \mathbf{X}_i$ 和 $\sum \mathbf{X}_i \mathbf{Y}_i$ 。有两种不同粒度的并行方案：1) 粗粒度并行方案，以 $\sum \mathbf{X}_i$ 数组 (用 x 表示) 和 $\sum \mathbf{X}_i \mathbf{Y}_i$ 矩阵 (用 mul 表示) 元素为单位将任务均衡分发给线程；2) 细粒度并行方案，多个线程协同计算 x 和 mul 中的每个元素。

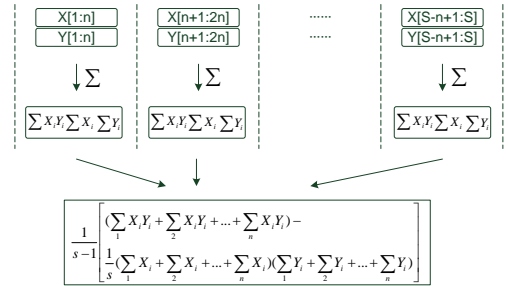


图 2 多进程协方差计算

Fig.2 COV calculation of progresses

2.2 PCA 变换任务分配

$\mathbf{Y}=\mathbf{A}\mathbf{X}=\{\mathbf{A}\mathbf{X}_1, \mathbf{A}\mathbf{X}_2, \dots, \mathbf{A}\mathbf{X}_N\}$ ，通过分解，PCA 变换过程可分为 N 个 $\mathbf{A}\mathbf{X}_i$ 计算任务，即每个进程计算 $\mathbf{A}\mathbf{X}_i$ 。每个进程计算时均需要 \mathbf{A} 矩阵 (特征向量矩阵)，故需要 1 次广播将数据分发给所有进程。最终结果输出时，不同进程可以同时输出，通过指针控制输出数据位置，整个过程仅需要 1 次通信。

$\mathbf{A}\mathbf{X}_i$ 的计算还能拆成 $\{\mathbf{A}_1\mathbf{X}_i, \mathbf{A}_2\mathbf{X}_i, \dots, \mathbf{A}_n\mathbf{X}_i\}$ ，对 \mathbf{A} 矩阵均匀分割，并将子计算任务分配给多个线程完成计算，各线程的计算任务是独立的。

2.3 M-PCA 算法和 Ms-PCA 算法

根据上述任务划分方案，将线程级的分配方案 offload 到 MIC 上，可以得到基于单 MIC 的 M-PCA 算法，图 3 为其简单流程。主机端的流程与串行基本一致，在 COV 计算和 PCA 变换时，开发 OpenMP 并行版本，并将其用 offload 映射到 MIC 完成计算。

针对 M-PCA 算法进行多 MIC 扩展，将任务均衡分配给多个进程，各进程将任务 offload 到 MIC，可以得到基于多 MIC 的 Ms-PCA 算法 (图 4)。整

个流程需要两次进程间通信。

3 算法性能模型

3.1 M-PCA 算法性能模型

由于算法各步骤有严格的依赖，提出 M-PCA 算法性能模型：

$$T = T_{\text{cov}} + T_{\text{eigenvalue}} + T_{\text{transposition}} + T_{\text{pcatrans}} \quad (2)$$

其中 T_{cov} 表示协方差矩阵计算时间，包括 MIC

上计算时间和数据通信时间：

$$T_{\text{cov}} = (B*W*H + B*(B+1)*W*H) / v_{\text{cov}} + (B*W*H* \text{size}(\text{data}) + B*B* \text{size}(\text{cov})) / v_{\text{pcie}} \quad (3)$$

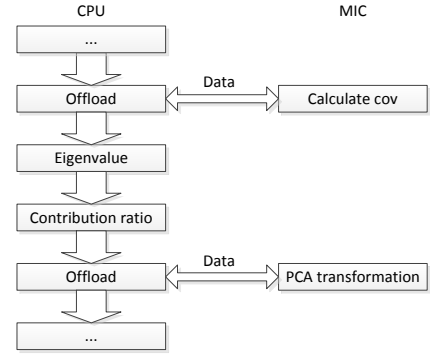


图 3 M-PCA 流程图

Fig.3 Flow chart of M-PCA

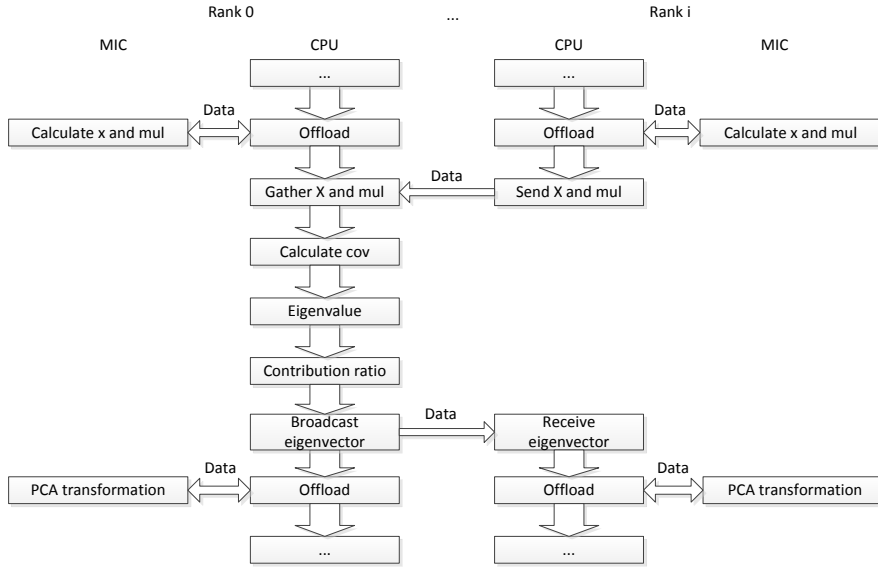


图 4 Ms-PCA 流程图

Fig.4 Flow chart of Ms-PCA

B 为波段数， W 为图像宽度， H 是高度， v_{cov} 为 MIC 上协方差矩阵计算速率， $\text{size}(\text{data})$ 表示高光谱数据类型长度， $\text{size}(\text{cov})$ 为输出的协方差矩阵元素数据类型长度， v_{pcie} 是 PCI-e 接口传输速率。

$T_{\text{eigenvalue}}$ 为特征值计算和贡献率统计时间，此步骤在 CPU 上执行，一般可取 250ms~400ms。

转置开销： $T_{\text{transposition}} = B*H*W / v_{\text{transposition}}$

T_{pcatrans} 是 PCA 变换开销，用 v_{pcatrans} 表示 MIC 上进行 PCA 变换的速率， $\text{size}(\text{trans})$ 表示结果矩阵元素的数据类型长度，其计算公式为

$$T_{\text{pcatrans}} = m*B*W*H*2 / v_{\text{pcatrans}} + (B+m*W*H* \text{size}(\text{trans})) / v_{\text{pcie}} \quad (4)$$

v_{cov} 和 v_{pcatrans} 的取值见图 5，天河 2 号采用了 PCI-e v1 标准， v_{pcie} 约为 3GB/s；转置速率见图 6，单位 Ge/s 表示每秒转置 10 亿个元素。

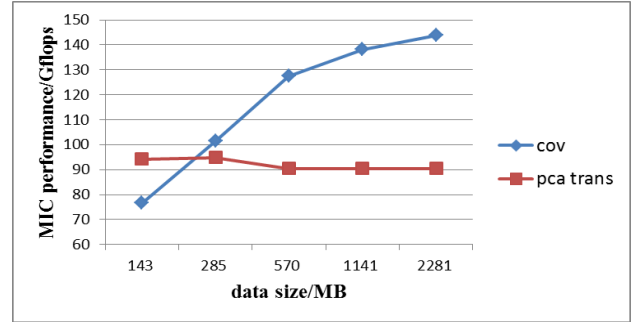


图 5 MIC 上 COV 计算与 PCA 变换速率

Fig.5 MIC performance of COV and PCA trans

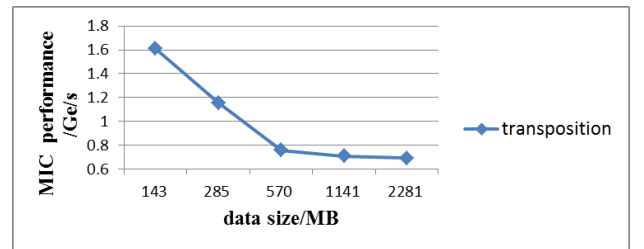


图 6 MIC 上转置速率

Fig.6 MIC performance of transposition

3.2 Ms-PCA 算法性能模型

深入研究 Ms-PCA 算法，建立其性能模型：

$$T = T_{ncov} + T_{eigenvalue} + T_{transposition} + T_{npcatrans} \quad (5)$$

其中 $T_{ncov} = T_{mul} + T_{pcie1} + T_{communication1} + T_{cov}$

T_{mul} 表示计算 ΣX 和 ΣXY 耗费的时间：

$$T_{mul} = \frac{(B*W*H + B*(B+1)*W*H)}{N * v_{mul}} \quad (6)$$

T_{pcie1} 是主机端与设备端交互时间：

$$T_{pcie1} = \frac{(\frac{B*W*H * size(data)}{N} + B*B * size(cov))}{v_{pcie}} \quad (7)$$

$T_{communication1}$ 是进程间的通信时间，3 个进程时平均取值 0.73ms； T_{cov} 是 root 进程计算协方差矩阵的时间，约为 31.4ms； v_{mul} 表示 MIC 计算 ΣX 和 ΣXY 的速率，其取值参考图 7。

转置时间（ $v_{transposition}$ 取值见图 8）：

$$T_{transposition} = B*W*H / N / v_{transposition} \quad (8)$$

多进程协作 PCA 变换时间为：

$$T_{npcatrans} = (T_{communication2} + T_{pcatrans} + T_{pcie2}) * a \quad (9)$$

$$T_{pcatrans} = \frac{m*B*W*H*2}{N * v_{pcatrans}} \quad (10)$$

$$T_{pcie2} = (B + \frac{m*W*H * size(trans)}{N}) / v_{pcie} \quad (11)$$

$T_{communication2}$ 是进程间广播特征向量矩阵的时间， $T_{pcatrans}$ 是 MIC 上 PCA 变换的时间， $v_{pcatrans}$ 是速率，取值见图 7； T_{pcie2} 是 CPU 和 MIC 的通讯开销； a 是多进程协作开销参数，针对 PCA 过程，取 1.15。

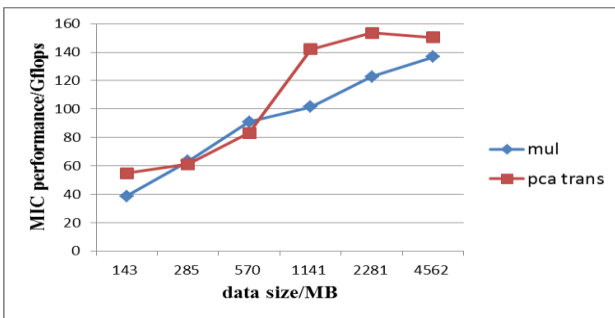


图 7 MIC 上 mul 计算和 PCA 变换速率

Fig.7 MIC performance of mul and PCA trans

4 PCA 集成众核优化

协方差矩阵计算优化研究（效果见图 9）：

1) 数据类型转换：高光图像属于图像数据，由 0~255 等像素信息组成，可以采用 unsigned char

类型（替代原始的 float 类型）数据存储信息，由于单位数据的存储减少，可降低访存开销。

2) 计算分解：相对于 CPU 强大的事务处理能力，作为协处理器的 MIC 更擅长于较简单的计算。将复杂的计算任务分解为数个简单的计算，使其适用于 MIC 轻量级协处理器。COV 计算分解还可减少计算量，计算量从 $(B*(B+1)*W*H*2)$ 降为 $(B*W*H+B*(B+1)*W*H)$ ，减少了近半。

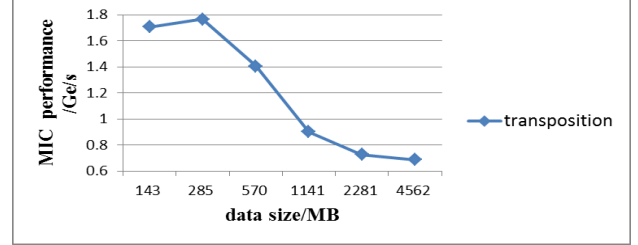


图 8 MIC 上转置速率

Fig.8 MIC performance of transposition

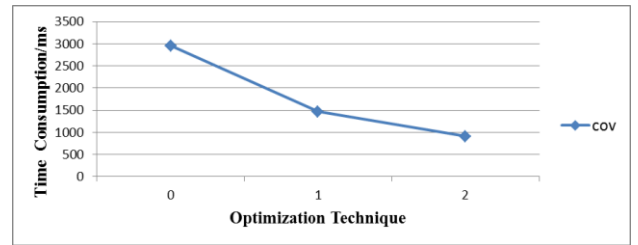


图 9 COV 计算优化研究

Fig.9 Optimization of COV calculation

PCA 变换过程优化措施：

1) 并行循环选择：若循环小则不适合于在众核处理器（MIC 上有 50 多个核，每个核可并发 4 个线程）上并行。需要选择大循环开发并行。

2) 数据类型转换，同协方差矩阵计算过程。

3) 循环交换：当存储访问不连续时，cache 命中率下降，将导致大量的内存访问开销，需要循环交换。PCA 变换过程访存不连续，需要循环交换。

4) 除最内层的嵌套循环展开：通过循环展开的方法扩大单层循环量，使其适用于集成众核架构。在嵌套循环展开的同时，保持最内层循环不动，以此来保证向量化和连续存储访问。

5) 转置：矩阵转置是为了使访问存储连续（效果统计时已包含转置开销）。

措施 3 和 5 无法重复使用，其目的是一致的。

图 10 是几种不同优化方法组合效果，其中组 0 没有采取优化措施，组 1 用了措施 1，组 2 用了措施 1 和 2，组 3 采取措施 1-3，措施 1-4 组成组 4，组 5 用了 1,2,5 优化方法，组 6 采用措施 1,2,4,5。通过一系列的优化，效果最好的组合 6 获得较优化前的 MIC 并行 83 倍的性能提升。

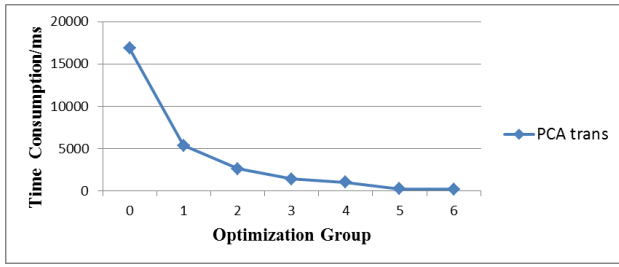


图 10 PCA 变换优化研究
Fig.10 Optimization of PCA trans

5 实验结果与分析

5.1 实验平台

本文数据均在天河 2 号超级计算机上测得。天河 2 号每个节点搭载两颗 12 核 Xeon CPU 和 3 个 57 核 MIC 协处理器。本文代码采用 ICC 编译。

5.2 实验数据

高光谱遥感图像数据来自 AVIRIS 官网，共 6 组，详细信息见表 1。

表 1 高光谱数据详细信息
Tab.1 Information of hyperspectral data

num	Width	height	band
1	614	512	224
2	614	1087	224
3	753	1924	224
4	781	6955	224
5	1562	6955	224
6	1562	13910	224

5.3 性能评价指标的选择

本文采用两个性能评价标准：最优 CPU 串行程序和 24 核 CPU 满载的 openmp 多线程并序程序。

首先选取串行程序优化开关。表 2 是打开优化开关 O0、O1、O2、O3 时，处理高光谱遥感图像（数据 2）的时间。其中打开优化开关 O2 时，串行程序执行时间最短，以此衡量并行算法性能。

表 2 串行程序优化执行时间表
Tab.2 Run-time list of serial programs

option	00	01	02	03
Time/s	158.08	53.88	44.57	44.91

5.4 对 CPU 串行的加速比

测试并行程序执行时间，与打开最佳优化开关的串行程序对比，分别协方差矩阵计算、仅 PCA 变换、PCA 变换和转置开销、所有的计算时间等 4 个模块加速比，绘制成图 11（M-PCA）和图 12（Ms-PCA）。由于 MIC 启动时间和 MIC 上存储分配时间是一次性开销，不计入本文时间统计范畴。

图中数据显示本文设计的热点并行方案能获得良好的性能提升；M-PCA 算法能获得 50 倍左右加速比，Ms-PCA 算法可最高加速 133 倍（不含 I/O）。

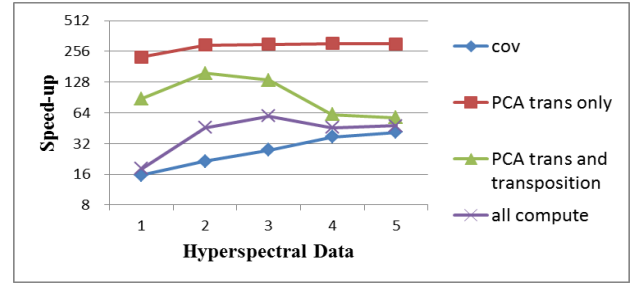


图 11 M-PCA 算法对串序程序加速比
Fig.11 Speed-up between M-PCA and serial PCA

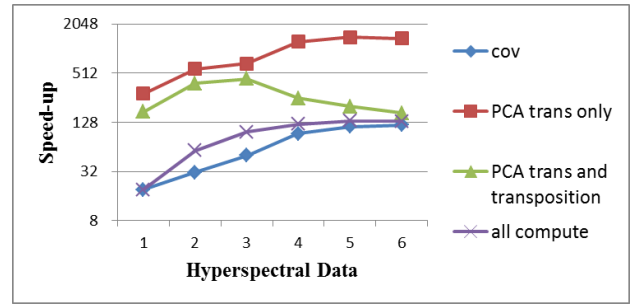


图 12 Ms-PCA 算法对串序程序加速比
Fig.12 Speed-up between Ms-PCA and serial PCA

5.5 对 CPU 多线程的加速比

MIC 与 Intel Xeon CPU 都采用 Xeon 架构，因此 MIC 上的优化策略也可在 CPU 多线程程序中应用。测试 CPU 多线程程序（天河 2 号采用 24 线程）与 M-PCA、Ms-PCA 的性能。图 13 是单 MIC 对 24 核 CPU 多线程的加速比，数据量大时，可获得 2.2 倍的总计算加速比。图 14 显示 Ms-PCA 算法能发挥 3 个 MIC 的性能，最高比 CPU 多线程提速 6.2 倍。

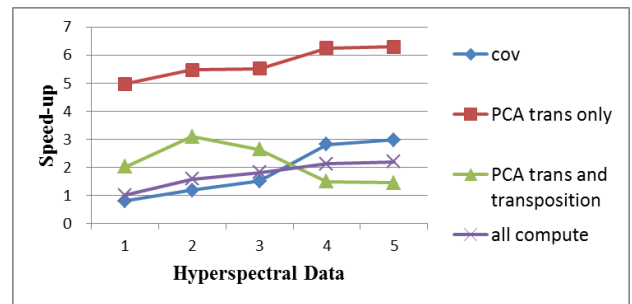


图 13 M-PCA 对 CPU 多线程的优势
Fig.13 contrast between M-PCA and multithread PCA

5.6 并行算法性能模型验证

将本文的高光谱数据参数信息输入到本文提出的 M-PCA 算法性能模型，可得到一组预测时间；执行优化后的程序，获得实际运行时间（图 15）。统计对比预测时间和执行时间的误差，不超过 7%，证明本文提出的基于 M-PCA 算法的性能模型能较

准确地预测算法性能。

将高光谱数据参数及 MIC 数量等输入 Ms-PCA 性能模型，预测程序执行所需的时间，计算与真实执行时间的误差百分比（图 16）。图中数据显示最大误差约为 10%，说明 Ms-PCA 性能模型是可靠的。

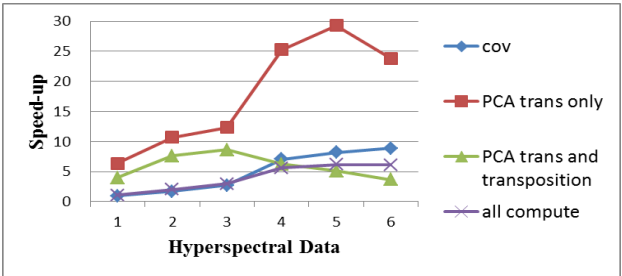


图 14 Ms-PCA 对 CPU 多线程的优势

Fig.14 contrast between Ms-PCA and multithread PCA

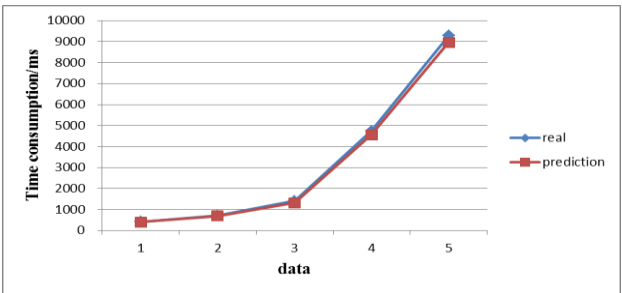


图 15 M-PCA 模型预测与实测时间

Fig.15 Time of real and perdition for M-PCA model

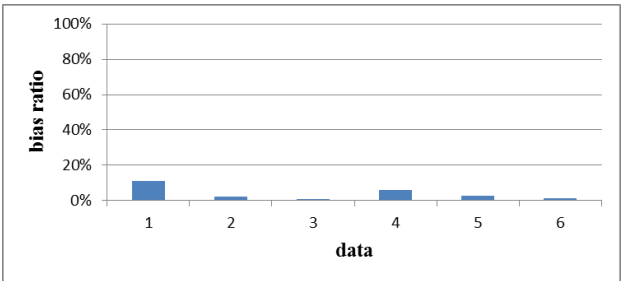


图 16 Ms-PCA 性能模型预测误差

Fig.16 bias ratio of the Ms-PCA model

6 总结与展望

本文基于集成众核研究高光谱遥感影像 PCA 降维方法，提出并实现基于单 MIC 的 M-PCA 并行算法和基于多 MIC 的 Ms-PCA 算法。实验结果表明本文设计的并行算法均可获得良好的加速效果，Ms-PCA 算法在 3 个 MIC 上执行比串行 CPU 快 133 倍，比 24 核 CPU 多线程快 6.2 倍。提出并验证 M-PCA 和 Ms-PCA 算法的性能预测模型。

本文研究显示，高光谱遥感影像 PCA 降维方法适合于基于 MIC 架构的处理平台。基于 GPU 或 MIC 等异构系统开发其他高光谱遥感影像降维方法是下

一步研究计划。

参考文献

- [1] Green R O, Eastwood M L, Sarture C M, *et al.* Imaging spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer(AVIRIS)[J]. *Remote Sens. Environ.* 1998,65:227–248.
- [2] Tom G T, Radhakrishna M. High-Speed Neural Network Filtering of Hyperspectral Data[J]. *SPIE, Bellingham, WA*, 2003:245-254.
- [3] Green A, Berman M, Switzer P, *et al.* A transformation for ordering multispectral data in terms of image quality with implications for noise removal[J]. *IEEE Transaction on Geoscience and Remote Sensing*, 2000, 26(1):65-74.
- [4] Kaarna A, Zemcik P, iainen H, *et al.* Compression of multispectral remote sensing images using clustering and spectral reduction[J]. *IEEE Trans. on Sci. Remote Sensing*, 2000, 38(2): 1588-1592.
- [5] Scott D, Thompson J. Probability density estimation in higher dimensions[C]. In *Proceedings of the 15th Symposium on the Interface Computer Science and Statistics*. 1983: 173–179.
- [6] TOP500. TOP500 Supercomputer Sites[EB/OL]. <http://www.top500.org/>
- [7] David V, Alexey L, Maureen O, *et al.* Parallel Processing of Remotely Sensed Hyperspectral Images On Heterogeneous Networks of Workstations Using HeteroMPI[J]. *International Journal of High Performance Computing Applications*. 2008, 22(4): 386-407.
- [8] Javier P, Antonio P, Posa P, *et al.* Parallel Classification of Hyperspectral Images Using Neural Networks[J]. *Computational Intelligence for Remote Sensing Studies in Computational Intelligence*, 2008, 133: 193-216.
- [9] Sergio S, Rui R. Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs[J]. *Real-Time Image Proc.* 2012.
- [10] Mahmoud E, Reda A. Fast GPU Algorithms for Endmember Extraction from Hyperspectral Images[J]. *IEEE*, 2012:631-636.
- [11] He Yang, Qian Du. Unsupervised Hyperspectral Band Selection Using Graphics Processing Units[J]. *IEEE journal of selected topics in applied earth observations and remote sensing*, 2011:660-668.