

集成众核上快速独立成分分析降维并行算法

方民权 张卫民 周海芳

(国防科学技术大学计算机学院 长沙 410073)

(fmq@hpc6.com)

Parallel Algorithm of Fast Independent Component Analysis for Dimensionality Reduction on Many Integrated Core

Fang Minquan, Zhang Weimin, and Zhou Haifang

(College of Computer, National University of Defense Technology, Changsha 410073)

Abstract There are massive matrix and iterative calculations in fast independent component analysis (FastICA) for hyperspectral image dimensionality reduction. By analyzing hotspots of FastICA algorithm, we design the parallel schemes of covariance matrix calculating, whitening processing and ICA iteration on many integrated core (MIC), implement and validate an M-FastICA algorithm. Further, we present a performance model for M-FastICA. We present a series of optimization methods for the parallel schemes of different hotspots: reforming the arithmetic operations, interchanging and unrolling loops, transposing matrix, using intrinsics and so on. In particular, we propose a novel method to balance the loads when dealing with the lower triangular matrix. Then we measure the performance effects of such optimization methods. Our experiments show that the M-FastICA algorithm can reach a maximum speed-up of 42X times in our test, and it runs 2.2X times faster than the CPU parallel version on 24 cores. We also investigate how the speed-ups change with the bands. The experiment results validate our performance model with an acceptable accuracy and thus can provide a roofline for our optimization effort.

Key words many integrated core (MIC); independent component analysis (ICA); dimensionality reduction of hyperspectral image; performance model; load balancing of lower triangular matrix

摘要 高光谱遥感影像快速独立成分分析(fast independent component analysis, FastICA)降维过程包含大规模矩阵计算及大量迭代计算. 通过热点分析, 面向集成众核(many integrated core, MIC)架构设计了协方差矩阵计算、白化处理和ICA迭代等热点并行方案, 提出和实现一种M-FastICA并行降维算法, 并构建算法性能模型; 基于集成众核研究并程序优化策略, 针对各热点并行方案提出一系列优化策略, 特别是创新性地提出一种下三角阵负载均衡方法, 并量化测试其优化效果. 实验结果显示M-FastICA算法最高可加速42倍, 比24核CPU多线程并行快2.2倍; 探讨了波段数与并程序性能的关系; 实验数据验证了算法性能模型的准确性.

收稿日期: 2014-09-29; 修回日期: 2015-04-16

基金项目: 国家自然科学基金项目(61272146, 41375113); 湖南省研究生创新资助项目(CX2015B030)

This work was supported by the National Natural Science Foundation of China (61272146, 41375113) and the Graduate Innovative Foundation of Hunan Province of China (CX2015B030).

关键词 集成众核;独立成分分析;高光谱影像降维;性能模型;下三角阵负载均衡

中图法分类号 TP391.4

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].但高光谱影像数据有波段多、数据量大、相关性强、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^[2-4].因此,国内外专家学者处理高光谱影像数据过程常常包括降维步骤.降维是通过特定的映射,将高维影像数据转换成低维影像数据,并保持信息量基本不变.

高光谱影像降维方法可分为线性和非线性2类,线性降维有主成分分析(principal component analysis, PCA)^[5]、独立成分分析(independent component analysis, ICA)^[6]等方法;非线性降维包括基于核的方法和基于流行学习的方法,如等距映射法(isometric feature mapping, Isomap)^[7]、局部线性嵌入(locally linear embedding, LLE)^[8]等.由于高光谱影像波段多、数据量大等特征,降维处理过程复杂且耗时,提高降维速度是重要研究热点,王和勇等人^[9]研究基于聚类和改进距离的LLE方法以提高5.5倍降维速度;而并行降维是加快降维过程的另一个重要研究方向.

自Intel公司2012年推出集成众核(many integrated core, MIC)产品Phi^[10],集成众核已成为目前高性能计算领域的研究热点,搭载MIC的天河2号超级计算机已连续6届登临TOP500榜首^[11].

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^[12]基于异构MPI在网络机群系统研究了高光谱影像处理的技术;Plaza等人^[13]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^[14]对高光谱解混进行了GPU移植;Platoš等人^[15]和Ramalho等人^[16]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还没有相关研究.本文将面向集成众核架构探讨加速高光谱影像降维的可行性.

本文以基于负熵最大的快速独立成分分析(fast independent component analysis, FastICA)算法^[17-18]为对象,面向集成众核架构展开并行算法研究,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,特别是创新性地提出一种下三角阵负载均衡策略;3)提出并实现了一种并行降维算法M-FastICA;4)设计了该并行算法的性能预测模型,并验证了其有效性.

1 集成众核体系结构与优化策略

1.1 集成众核体系结构

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个通道,峰值带宽可达350 GBps^[10].

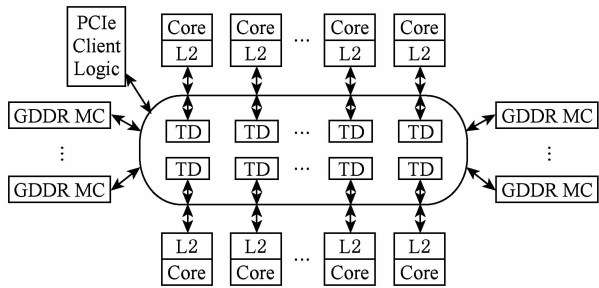


Fig. 1 Many integrated core architecture.

图1 MIC架构

MIC的每个计算核心拥有1个512b的向量处理单元(vector processing unit, 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,但发射1条指令需要2个时钟^[10].

MIC搭载的嵌入式Linux uOS使其具有5种不同的工作模式:CPU原生模式、CPU为主MIC为辅模式、对等模式、MIC为主CPU为辅模式和MIC原生模式^[10].本文使用CPU为主MIC为辅模式.

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,可以采取的一般优化策略包括并行度、内存空间、数据通信、Cache访问、向量化、负载均衡、线程扩展性等方面优化.对于CPU/MIC异构系统,如何合理分配CPU和MIC的计算任务、如何设计host端和device端的通信这些都是异构系统优化过程中所必须考虑的重要因素.

2 独立成分分析算法与加速热点

2.1 基于负熵最大的快速独立成分分析

FastICA 算法有基于负熵最大、基于似然最大、基于峭度 3 种形式,本文研究基于负熵最大的 FastICA 算法.该算法以负熵最大为搜寻方向,实现独立源的顺序提取,充分体现传统线性变换中的投影追踪思想,该算法还采用定点迭代的优化算法,使收敛更加快速、稳健^[17-18].

对于高光谱影像数据 $\mathbf{X}(W \times H \times B)$,宽 W 、高 H 、波段 (band, B), \mathbf{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.通过独立成分分析方法降维,可以获得 m 个 IC 图像,其中 $m < B$,从而实现降维效果.

- 算法 1. FastICA 高光谱影像降维方法^[17-18].
- 输入:高光谱影像数据 \mathbf{X} ;
- 输出:独立成分 \mathbf{Y} .
- Step1. 计算 \mathbf{X} 的协方差矩阵;
- Step2. 计算协方差矩阵的特征值及特征向量,并根据阈值 T 选取最终 IC 图像数量 m ;
- Step3. 计算白化矩阵 $\mathbf{M} = \mathbf{D}^{-1/2} \mathbf{V}^T$,其中 \mathbf{D} 为 \mathbf{X} 的特征值矩阵, \mathbf{V} 为对应的特征向量矩阵;
- Step4. 高光谱影像数据白化处理 $\mathbf{Z} = \mathbf{MX}$,这里的 \mathbf{X} 为减去均值之后的 \mathbf{X} (即零均值处理);
- Step5. 迭代开始:
- Step5. 1. 初始化 \mathbf{W}_i ;
- Step5. 2. $\mathbf{W}_i = E\{\mathbf{Z}g(\mathbf{W}_i^T \mathbf{Z})\} - E\{g'(\mathbf{W}_i^T \mathbf{Z})\}\mathbf{W}_i$,其中 g 为非线性函数,经测试,取 $g(y) = y^3$ 可更稳定、更快地使 ICA 迭代收敛;
- Step5. 3. 正交化 $\mathbf{W}_i = \mathbf{W}_i - \sum_j (\mathbf{W}_i^T \mathbf{W}_j) \mathbf{W}_j$;
- Step5. 4. 归一化 $\mathbf{W}_i = \mathbf{W}_i / \|\mathbf{W}_i\|$;
- Step5. 5. 判断是否收敛,不收敛则转 Step5. 2,若收敛,如果所有的 \mathbf{W}_i 已经计算结束 (即 $i = m$),则退出,否则令 $i = i + 1$,转到 Step5. 1 计算新的 \mathbf{W}_i ;
- Step6. 计算独立成分 $\mathbf{Y} = \mathbf{WZ}$.

2.2 加速热点分析

实现串行高光谱影像 FastICA 降维算法,对

$W = 614, H = 1087, B = 224$ 的高光谱影像降维,测试并统计计算部分 (不含 I/O) 各步骤占总计算时间的比率如表 1 所示.表 1 数据显示,Step1, Step3, Step4, Step5 比重较大,共占总计算时间的 99%,是本文并行研究的关键热点.分析算法过程,计算量主要集中在 Step1 中协方差矩阵计算、Step4 中白化处理时的矩阵运算和 Step5. 2 中 \mathbf{W}_i 计算中的大量向量运算.此外 Step2 中特征值计算取决于 B ,变化不大可不考虑并行;Step3 白化矩阵很小,计算量有限,不必做并行考虑;而 Step6 中的 IC 变换与 \mathbf{W}, H, m 正相关,作为次要热点考虑其并行化.

Table 1 Time Distribution of FastICA
表 1 FastICA 时间分布

Step	Running Time Ratio/%
Step1	19.8
Step2	0.5
Step3+Step4	60.7
Step5	18.5
Step6	0.5

3 FastICA 集成众核并行算法

3.1 协方差矩阵任务划分

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.传统的并行方案是对最外层循环进行任务划分,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.本文采用的高光谱影像数据的 $B = 224$,与集成众核的硬件并发线程总数 228 相当,即采用传统方案并行计算协方差矩阵可能存在负载不均衡问题.为解决这一问题,本文提出一种下三角阵负载均衡方法,在本文第 5 节性能优化部分详细描述.

3.2 白化处理与 IC 变换任务分配

白化处理包括白化矩阵计算和白化处理过程,其中白化矩阵维度较小、计算量也较小、无需并行,主要考虑白化处理 $\mathbf{Z} = \mathbf{MX}$.针对白化处理过程,首先抽象并构建白化处理宏观模型 (对算法核心计算的宏观描述),如图 2 所示,其中 \mathbf{X} 需要进行零均值处理,构建宏观模型时暂不考虑.

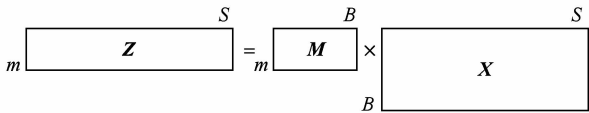


Fig. 2 Macroscopic model for whitening processing.
图 2 白化处理宏观模型

上述宏观模型可视为矩阵乘法,传统的并行矩阵乘法策略是在最外层循环做并行,但针对高光影像降维过程,降维后取得的波段数目 m 较小,而使用了硬件多线程并发技术的集成众核并行度远大于 m ,将导致严重的负载不均衡.因此,本文对像元素 S 进行任务划分,由于 $S=W \times H$ 足够大,可满足大量线程的并行均衡负载要求;也可采用循环合并策略进一步扩大并行度,具体优化策略参见本文第 5 节性能优化部分.

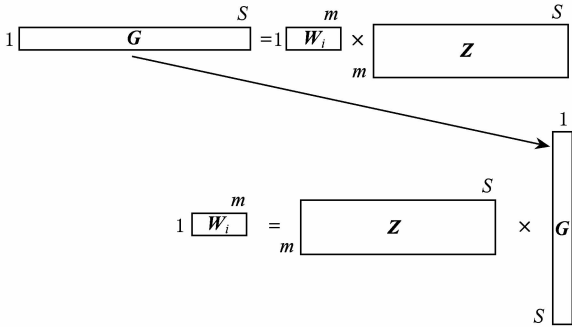


Fig. 3 Model of key step for ICA iteration.
图 3 ICA 迭代关键步骤模型

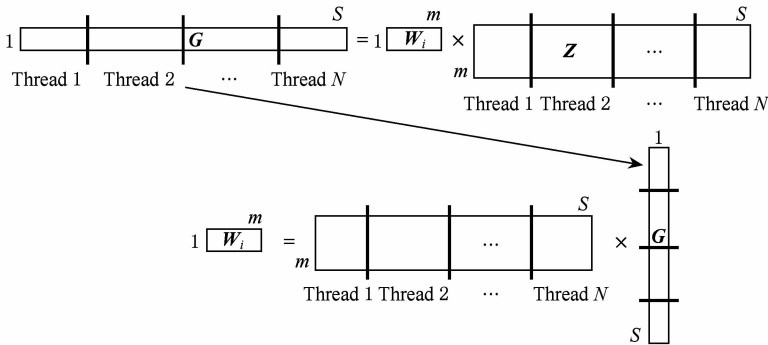


Fig. 4 Parallel model of ICA iteration.
图 4 ICA 迭代并行模型

3.4 M-FastICA 算法

基于上述热点并行任务划分策略,结合串行算法,对白化处理和 ICA 迭代并行宏观模型进行演绎,获得热点详细并行方案.通过第 5 节对比各热点并行方法在 CPU 与 MIC 上执行性能和 CPU+MIC 协同工作策略,提出 M-FastICA 并行降维算法,如图 5 所示,其中由于 W_i 正交化、归一化及判断等运

IC 变换($Y=WZ$)的模型与白化处理($Z=MX$)模型基本一致,因此采用与白化处理相同的并行任务划分方案和优化策略.

3.3 集成众核上的 ICA 迭代

ICA 迭代的 5 个步骤中,计算量主要集中在 Step5.2 中.该运算比较复杂,首先对其抽象,得到如图 3 所示的宏观模型,然后针对该模型进行并行任务分割研究.

基于 ICA 迭代关键步骤宏观模型,对任务进行切分,如图 4 所示. $W_i^T Z$ 实质是求长度为 m 的 2 个向量内积,共求了 S 次,由于 $S \gg m$,对 S 分割可满足集成众核架构中众多线程的负载均衡需求. $Zg(W_i^T Z)$ 实质是循环 m 次求长 S 的向量内积, $S \gg m$,可采用并行循环归约的方法,通过 m 次 fork-join,多线程协同归约计算向量内积.

算量较小,ICA 迭代过程可完全 Offload 到 MIC.算法将协方差矩阵计算、ICA 迭代 Offload 到 MIC 上进行并行计算;白化处理、IC 变换、转置等 CPU 计算占优势的步骤仍在 CPU 上并行处理;另外充分利用了 MIC 与 CPU 协同计算、传输计算重叠等关键异构优化策略,图 5 中纵向可视时间为轴,横向重叠部分可同时执行.其原理将在第 5 节详细讨论.

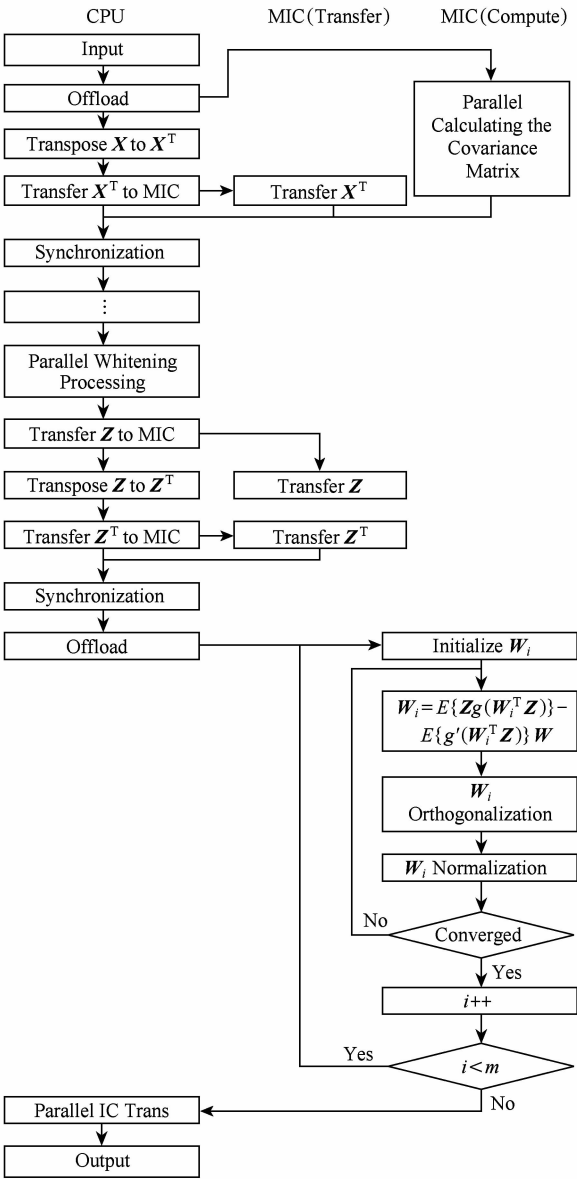


Fig. 5 Flow chart of M-FastICA.

图 5 M-FastICA 流程图

4 算法性能模型

本节旨在构建 M-FastICA 算法性能模型,通过输入高光谱影像数据参数(W, H, B, m 等),经性能模型分析,预测出 M-FastICA 并行算法的降维时间,如图 6 所示:

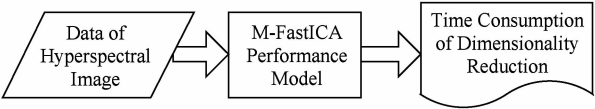


Fig. 6 Performance model of M-FastICA.

图 6 M-FastICA 算法性能模型

研究 M-FastICA 降维算法,建立性能模型:

$T = T_{COV} + T_{eigen} + T_{whiteprocess} + T_{ICAiter} + T_{ICtrans}; \quad (1)$

$T_{COV} = T_{COVc} + T_{COVt} = (BWH + B(B+1)WH)v_{COVc} + (BWH + 4BB)v_{PCI-e}; \quad (2)$

$T_{whiteprocess} = (2mBWH)v_{whiteprocess}; \quad (3)$

$T_{ICAiter} = T_{Z^T} \times 2 + T_{everyICA} \times n_{iter} \approx 2(4mWH)v_{PCI-e} + (4mWH)v_{ICAiter} \times n_{iter}; \quad (4)$

$T_{ICtrans} = (2mmWH)v_{ICtrans}. \quad (5)$

- T_{COV} : 协方差矩阵计算时间;
- T_{COVc} : MIC 并行计算协方差矩阵时间;
- T_{COVt} : 高光谱影像数据和协方差矩阵通信时间;
- v_{COVc} : MIC 计算协方差矩阵速率(图 7);
- v_{PCI-e} : PCI-e 总线传输速率,v1 版本取 3 GBps;
- T_{eigen} : 特征值和波段选择时间,为 0.28 ± 0.05 s;
- $T_{whiteprocess}$: 白化处理时间;
- $v_{whiteprocess}$: CPU 白化处理速率,如图 7 所示;
- $T_{ICAiter}$: ICA 迭代时间;
- T_{Z^T} : 白化结果矩阵 Z 及其转置矩阵的传输时间;
- $T_{everyICA}$: 每次 ICA 迭代时间;
- n_{iter} : 迭代次数;
- $v_{ICAiter}$: MIC 上并行 ICA 迭代性能,如图 7 所示;
- $T_{ICtrans}$: IC 变换时间;
- $v_{ICtrans}$: CPU 处理 IC 变换的速率,如图 7 所示;

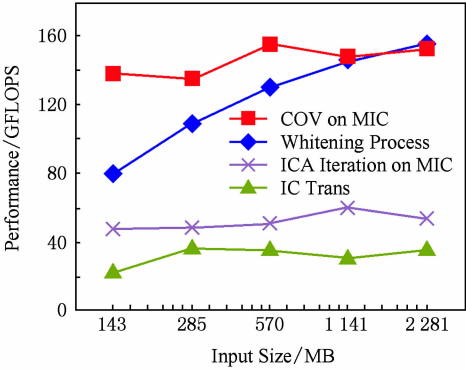


Fig. 7 Performance of different steps.

图 7 各步骤计算性能

MIC 是高度并行计算协处理器,对并行度非常敏感,且在运算规模较大时能获得较好的吞吐率;问题规模影响并行度,对 MIC 性能发挥有着较大的影响.本文通过高光谱影像数据缩放,获得成比例的数据规模,测试并统计计算单元(CPU, MIC)性能.在利用该模型预测性能时,根据高光谱影像数据规模,从统计曲线中获得估计速率,再根据算法模型公式预测并行降维时间.

5 面向集成众核的性能优化

本节首先对协方差矩阵计算、白化处理、ICA 迭代等展开研究集成众核性能优化,然后研究了并行算法在 CPU+MIC 异构系统中的协同计算。

5.1 协方差矩阵计算优化

针对协方差矩阵计算过程,面向集成众核架构进行优化研究,下列优化方法可加速协方差矩阵计算:

1) 数据类型选择. 高光谱影像属于图像数据,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,可降低协处理器访存开销和主机端设备端间通信开销。

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. 协方差矩阵中每个元素的计算:

$$Cov(\mathbf{X}, \mathbf{Y}) = \frac{1}{s-1} \left[\sum_{i=1}^s (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{Y}_i - \bar{\mathbf{Y}}) \right] = \frac{1}{s-1} \left[\sum_{i=1}^s \mathbf{X}_i \mathbf{Y}_i - \frac{1}{s} \sum_{i=1}^s \mathbf{X}_i \times \sum_{i=1}^s \mathbf{Y}_i \right]. \quad (6)$$

通过式(6)的协方差计算公式变形,将协方差的计算变形为累加和点乘运算;在此基础上还能将所有协方差矩阵元素的累加提取出来单独计算,即将累加和矩阵乘运算分开进行. 该计算分解过程还可减少计算量,使计算量从 $2B(B+1)WH$ 降为 $BWH + B(B+1)WH$,减少了近半. 算法 2 和算法 3 分别描述了计算分解前后协方差矩阵计算的详细过程。

算法 2. 计算分解前的协方差矩阵计算.

输入: 高光谱影像数据 \mathbf{X} ;

输出: 协方差矩阵 cov .

```
parallel for  $i=0$  to  $(B-1)$  /*  $B$  为波段数 */
  for  $j=0$  to  $i$  /* 下三角 */
    for  $k=0$  to  $S$  /*  $S=W \times H$  像元数 */
       $x = X[i][k]$ ;
       $y = X[j][k]$ ;
       $m += x$ ;  $n += y$ ;
       $r += x \times y$ ;
    end for
     $cov[i][j] = (r - m \times n / S) / (S - 1)$ ; /* 计算协方差 */
```

```
 $cov[j][i] = cov[i][j]$ ; /* 上三角矩阵赋值 */
```

```
end for
```

```
end parallel for
```

算法 3. 计算分解后的协方差矩阵计算.

输入: 高光谱影像数据 \mathbf{X} ;

输出: 协方差矩阵 cov .

```
parallel for  $i=0$  to  $(B-1)$ 
```

```
  for  $k=0$  to  $S$ 
```

```
     $m[i] += X[i][k]$ ; /* 计算累加 */
```

```
  end for
```

```
end parallel for
```

```
parallel for  $i=0$  to  $(B-1)$ 
```

```
  for  $j=0$  to  $i$ 
```

```
    for  $k=0$  to  $S$ 
```

```
       $r += X[i][k] \times X[j][k]$ ; /* 计算矩阵乘 */
```

```
    end for
```

```
     $cov[i][j] = (r - m[i] \times m[j] / S) / (S - 1)$ ; /* 计算协方差 */
```

```
     $cov[j][i] = cov[i][j]$ ; /* 上三角矩阵赋值 */
```

```
  end for
```

```
end parallel for
```

3) 下三角阵负载均衡. 协方差矩阵是对称阵, 仅需计算下三角阵, 由于 MIC 核心较多, 2.2 节的并行方案无法合理地做到下三角阵的负载均衡, 如线程 0 执行 1 个协方差计算, 而线程 223 需要执行 224 个协方差计算. 为实现下三角阵的负载均衡, 本文设计了一种新的负载均衡方法, 如图 8 所示. 其基本思想是将三角形状的 2 维计算任务映射为 1 维计算任务, 然后对得到的 1 维计算任务进行并行划分, 最终实现均衡负载的目的. 本文通过图 8 中行③和

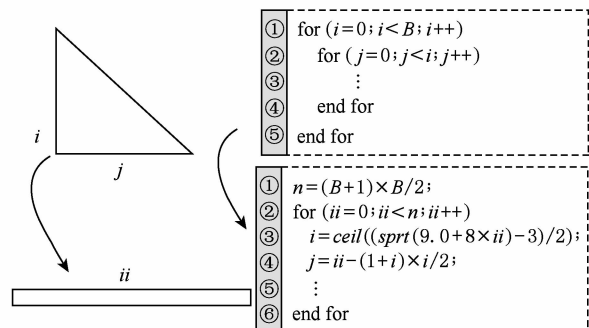


Fig. 8 Load balancing for lower triangular matrix.

图 8 下三角阵负载均衡

行④的公式,将 1 维索引映射到 1 维数组中,把 2 层循环合并为 1 层循环,不仅可以实现负载均衡,还可增大并行度. 算法 4 采用了下三角阵负载均衡策略.

算法 4. 下三角负载均衡后的协方差矩阵计算.

输入: 高光谱影像数据 \mathbf{X} ;

输出: 协方差矩阵 \mathbf{cov} .

```
parallel for i=0 to (B-1)
    for k=0 to S
         $m[i] += X[i][k];$  /* 计算累加 */
    end for
end parallel for
parallel for ii=0 to (n-1)
    /*  $n = (B+1) \times B/2$  */
     $i = \text{ceil}((\text{sqrt}(9.0 + 8 \times ii) - 3)/2);$ 
     $j = ii - (1+i) \times i/2;$ 
    for k=0 to S
         $r += X[i][k] \times X[j][k];$  /* 计算矩阵乘 */
    end for
     $\mathbf{cov}[i][j] = (r - m[i] \times m[j]/S)/(S-1);$ 
    /* 计算协方差 */
     $\mathbf{cov}[j][i] = \mathbf{cov}[i][j];$  /* 上三角阵赋值 */
end parallel for
```

程序经上述优化方法优化后进行实验测试,其效果如图 9 所示,图 9 中 0 表示未采取优化手段,1~3 分别表示相继采用优化手段 1~3 后的时间消耗.

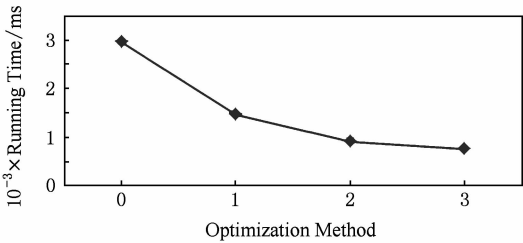


Fig. 9 Optimization effect of COV calculation.

图 9 COV 计算优化效果

5.2 白化处理集成众核优化

针对白化处理过程,本文采取的优化方法包括:

1) 并行循环选择. 高光谱影像白化处理过程有其独特的特点,最外层循环量是根据贡献率所取得的波段数目 m ,该值往往为十几到几十不等,该层循环并行度较小,不适合在众核处理器(MIC 上有 50 多个核,每个核可并发 4 个线程)上并行. 选择适合众核架构核数的并行度的循环开发并行至关重要.

本文选择第 2 层循环开发并行,其并行度($S=W \times H$)巨大.

2) 数据类型选择. 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,将导致大量的内存访问开销;最内层循环无法向量化时,MIC 的重要部件向量处理单元(vector process unit, VPU)得不到有效利用,将大大影响性能. 若程序存在上述 2 方面情况,需要循环交换. 白化处理由于存储访问不连续而采取循环交换措施,通过交换内 2 层循环使数据连续访问.

4) 除最内层的嵌套循环展开. 白化处理共 3 层循环,最外层取值 m (降维后取得的波段数)较小而不适合 MIC 并行,通过将外 2 层嵌套循环展开,增加最外层循环次数,使其适用于 MIC 等众核架构,还能减少并行区域 fork-join 开销. 在嵌套循环展开的同时保持最内层循环不变,以此来保证向量化和连续存储访问.

5) 高光谱影像矩阵转置. 白化处理对高光谱影像矩阵的访问是按列的,为了使访问存储连续,可以采用矩阵转置的策略. 通过转置,原始矩阵行列交换,实现存储连续访问. 该方案将引入转置开销,若转置开销大于性能收益,则不可取(效果统计时已包含转置开销).

优化方法 3 和优化方法 5 无法重复使用,因其目的是一致的,即连续访存. 图 10 所示 6 种不同优化方法组合效果,其中组 0 没有优化,组 1 用了方法 1,组 2 用了方法 1 和方法 2,组 3 采取方法 1 至方法 3,方法 1~4 组成组 4,组 5 用了方法 1、方法 2、方法 5 优化方法,组 6 采用方法 1、方法 2、方法 4、方法 5. 通过一系列的优化,组 6 获得最佳性能.

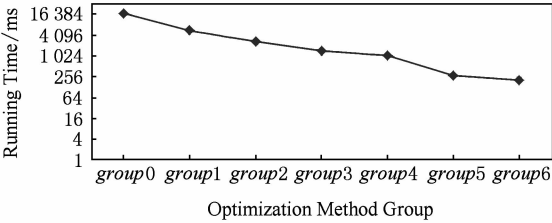


Fig. 10 Optimization effect of whitening process.

图 10 白化处理优化效果

5.3 ICA 迭代优化

ICA 迭代部分包含大量的小规模迭代计算,本文对单次迭代展开优化研究,主要采取 3 种优化策略:

1) 矩阵转置以保证连续访存. $\mathbf{W}^T \mathbf{Z}$ 计算中,矩阵 \mathbf{Z} 的访问是以列为主的,C 代码中数据是按行存

储的,无法连续访存.通过建立 Z 的转置矩阵参与运算,实现存储连续访问.

2) ICA 迭代过程中,参与计算的 W_i 是 W 矩阵的 1 列,迭代过程中 W_i 参与大量计算,因此本文采用中间数组存储 W_i 中元素参与计算.C 语言中矩阵是按行存储的,列访问导致 Cache 命中率下降.利用中间数组 $tmp[]$ 存储 W_i 的元素,那么运算时访存是顺序的,Cache 命中率高,从而提升程序性能.

3) 使用较底层的 intrinsics 函数,显式地指定数据向量化处理过程,以充分发挥 MIC 中向量处理单元的性能.intrinsics 函数是类汇编函数,指定程序中的向量运算,比如向量内积($c = A \cdot B$)运算时,普通 C 程序代码和 intrinsics 代码如下所示:

```
C 版向量内积:
for(i=0;i<n;i++)
    c+=a[i]+b[i];
endfor
intrinsics 版向量内积:
_mm512_a, b, c;
for(i=0;i<n;i+=16)
    _a=_mm512_load_ps(&a[i]);
    _b=_mm512_load_ps(&b[i]);
    _c=_mm512_add_ps(_c,_mm512_mul_ps
        (_a,_b));
endfor
c=_mm512_reduce_add_ps(_c);
```

其中 intrinsics 代码中,1 条指令完成 16 个浮点数的运算.

在程序中分别实现上述优化方法,测试出各优化方法使用前后 ICA 迭代步骤的时间消耗,获得图 11 所示的优化效果,其中横坐标 0 表示未采取优化措施,1~3 分别表示相继采用优化方法 1~3 后的时间.

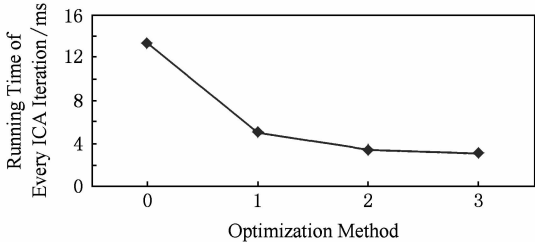


Fig. 11 Optimization effect of ICA iteration.

图 11 ICA 迭代优化效果

5.4 CPU/MIC 异构优化

1) 异构协同计算优化.CPU + MIC 架构是典

型的异构系统,CPU 和 MIC 可同时进行计算或通信,通过协同计算可掩盖部分步骤处理时间.本文算法中,协方差矩阵计算(COV)和高光谱影像数据 X 转置的过程可同时执行,如图 12 所示;白化结果矩阵 Z 的传输可与矩阵 Z 转置同时执行,如图 13 所示.通过上述 CPU 与 MIC 协同工作,可以掩盖 X 和 Z 的转置开销.

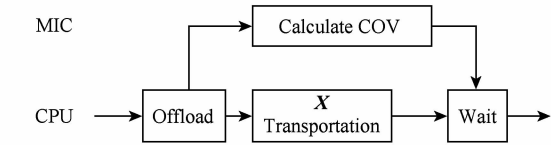


Fig. 12 Cooperative computing between COV and X transposition.

图 12 COV 与 X 转置协同计算

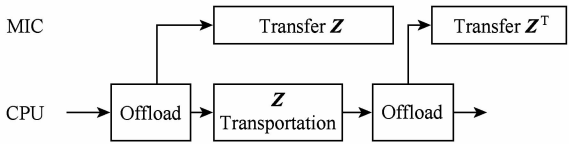


Fig. 13 Transpose Z and transfer Z at the same time.

图 13 Z 转置与通信重叠

2) CPU 与 MIC 计算的权衡.CPU 和 MIC 都是异构系统中重要的计算资源,由于体系结构的不同,其擅长领域也不同.本文通过实验统计,获得各步骤 MIC 与 CPU 上执行时间比值(图 14 中横轴 1~6 分别表示 COV、 X 转置、白化处理、 Z 转置、ICA 迭代和 IC 变换).比值大于 1 时 CPU 计算有优势,小于 1 则 Offload 到 MIC 能获得更好的性能.观察图 14,当数据较大时,COV 过程 Offload 到 MIC

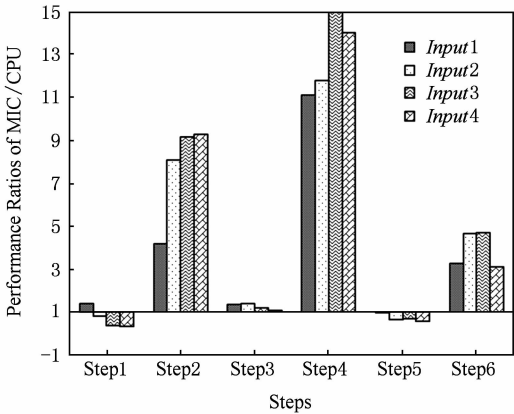


Fig. 14 Performance ratios of MIC/CPU for different steps.

图 14 各步骤 MIC/CPU 性能对比

合适,ICA 迭代过程 Offload 到 MIC 可获得更好的性能,其他过程在 CPU 端处理更划算.

6 实验结果与分析

6.1 实验准备

本文测试平台是天河 2 号超级计算机. 天河 2 号单个节点搭载 2 个 12 核 Xeon E5-2692v2 CPU 和 57 核 Intel Phi 31S1P 协处理器. 根据热点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 M-FastICA 并行算法,采用 Intel C Compiler 2013 sp1. 1. 106 编译器、O3 编译选项,最后在天河 2 号单节点上测试性能.

表 2 罗列了本文采用的 AVIRIS 高光谱影像数据的详细信息,共 4 组.

Table 2 Information of Hyperspectral Image Data			
表 2 高光谱影像数据详细信息			
Input	Width	Height	Band
Input1	614	1 087	224
Input2	753	1 924	224
Input3	781	6 955	224
Input4	1 562	6 955	224

由于高光谱影像数据和 ICA 迭代过程中取得的随机数不同,导致 ICA 迭代次数不同,不同的迭代次数将导致消耗的时间也不同. 为了获得准确的性能对比数据,本文约定:同一数据取固定的迭代次数. 通过多次实验求平均值的方法,确定各组数据的迭代次数,如表 3 所示:

Table 3 Iteration Number of Hyperspectral Image Data	
表 3 高光谱影像数据迭代次数	
Input	Iterations
Input1	334
Input2	809
Input3	559
Input4	378

6.2 并行算法加速比

本实验分别在 MIC 和 CPU 上测试并行程序计算时间,与打开最佳优化开关(O3)的串行程序对比,获得加速比绘制成图 15. 其中 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 15 中处理高光谱影像数据 1 时,

CPU 上的加速比高于 MIC,可知 CPU 比 MIC 更擅长于处理小规模运算;高光谱影像数据 2~4 中, MIC 获得了比 CPU 并行更高的性能,说明 MIC 适合规模较大的计算;在处理高光谱影像数据 4 时, M-FastICA 算法可获得 42 倍加速比,比 24 核 CPU 并行快 2.2 倍.

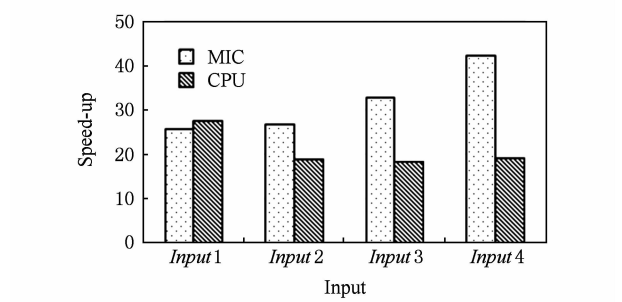


Fig. 15 Speed-up of MIC and CPU.

图 15 MIC 与 CPU 加速比

6.3 波段数与性能的探讨

本文选用的 AVIRIS 高光谱影像数据中 $B=224$,刚好与 Intel Phi 31S1P 协处理器的硬件线程数量 228 接近,因此会提出疑问,如果 B 变化时是否会影响并程序性能?

当 B 发生变化时,各热点的计算量也会产生变化,因此并程序性能会有所变化,但总体稳定. 究其原因,本文对各热点的并程序进行了大量深入优化,比如下三角矩阵均衡负载、循环合并等,已经将受 B 限制的并行度进行了扩展,并行粒度不会受 B 变化而发生较大变化.

本文通过控制变量法验证 B 与性能的关系. 选取输入高光谱影像数据 3,取其中前 112,140,168,196,224 波段的影像数据作为输入数据,通过实验统计并行算法对最优串行的加速比,如图 16 所示. 图 16 中 $B=112$ 和 $B=140$ 时,其加速比较小,原因是这 2 组输入降维时最终 IC 图像数量为 5,6,而其他几组输入降维时最终 IC 图像数量为 13,14,这会

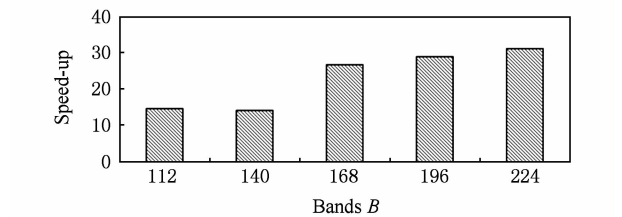


Fig. 16 Speed-ups with different bands.

图 16 波段数与加速比的关系

及性能. 而当 IC 图像数量不变时, 程序性能基本稳定, 不会因 B 的变化而影响程序性能.

6.4 算法性能模型验证

将高光谱影像数据的参数 (B, W, H, m) 输入 M-FastICA 算法性能模型, 通过式 (1)~(5) 计算可得到预测时间; 执行优化后的并程序, 获得实际运行时间, 如图 17 所示. 对比预测时间和执行时间的误差, 图 18 所示为误差百分比. 图 17 显示预测时间与真实运行时间基本一致, 图 18 显示预测性能和实测性能最大误差小于 5%, 说明本文提出的 M-FastICA 算法性能模型能较准确地预测算法性能.

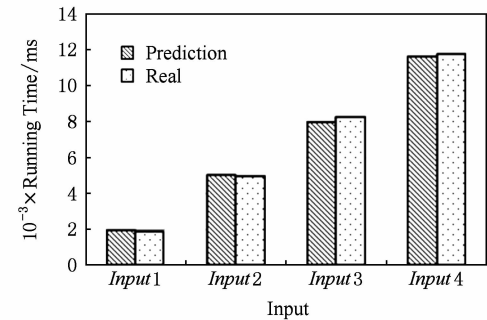


Fig. 17 Time of real and perdition for M-FastICA model.

图 17 M-FastICA 模型预测与实测时间

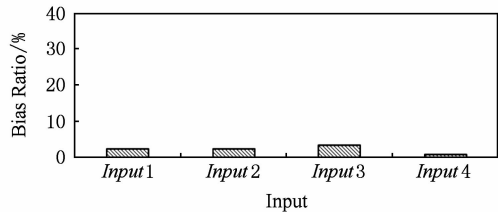


Fig. 18 Bias ratio of the M-FastICA model.

图 18 M-FastICA 性能模型预测误差

6.5 并行算法瓶颈分析

统计独立成分分析降维算法各个步骤消耗时间占总时间的百分比, 绘制成图 19 (横轴 1~5 分别表示独立成分分析的 5 个关键步骤: 协方差矩阵计算、特征值特征向量计算、白化处理、ICA 迭代、IC 变换). 对比图 19 和表 1, 我们发现并行后各步骤耗时

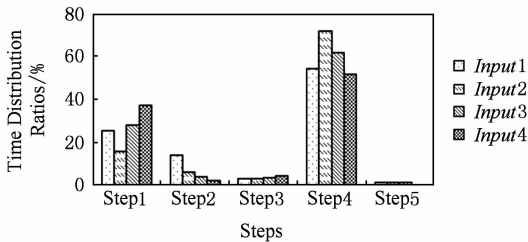


Fig. 19 Time distribution of M-FastICA.

图 19 M-FastICA 时间分布

比重有所改变, 但协方差矩阵计算和 ICA 变换依旧是算法性能瓶颈. 要解决这一瓶颈, 可以通过扩展计算资源来解决, 比如使用 MIC 集群, 这也是下一步工作的重点.

7 结 论

本文基于集成众核 MIC 研究高光谱影像 FastICA 降维方法, 设计了协方差矩阵计算、白化处理、ICA 迭代、IC 变换等热点并行方案和优化策略; 提出和实现了基于 MIC 的 M-FastICA 算法, 给出算法性能模型并验证其准确性. 实验结果显示 M-FastICA 算法获得了良好的性能提升, 对比最优串行程序最高可加速 42 倍, 比 24 核 CPU 多线程快 2.2 倍.

本文研究显示, 高光谱影像 FastICA 降维算法可在集成众核架构上获得良好的性能, 但仍存在性能瓶颈, 可考虑进一步将算法扩展到 MIC 集群以突破性能瓶颈.

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 Sensing of Environment, 1998, 65(3): 227-248

[2] Green A 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 Trans on Geoscience and Remote Sensing, 2000, 26(1): 65-74

[3] Kaarna A, Zemcik P, Kalviainen H, et al. Compression of multispectral remote sensing images using clustering and spectral reduction [J]. IEEE Trans on Geoscience and Remote Sensing, 2000, 38(2): 1073-1082

[4] Scott D W, Thompson J R. Probability density estimation in higher dimensions [C] //Proc of the 15th Symp on the Interface of Computer Science and Statistics. Amsterdam: North-Holland, 1983: 173-179

[5] Jolliffe I T. Principal Component Analysis [M]. Berlin: Springer, 2002

[6] Hyvärinen A, Oja E. Independent component analysis: Algorithm and applications [J]. Neural Networks, 2000, 13 (4/5): 411-430

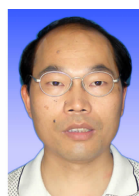
[7] Tenenbaum J, De Silva V, Langford J C. A global geometric framework for nonlinear dimensionality reduction [J]. Science, 2000, 290(5500): 2319-2323

[8] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000, 290(5500): 2323-2326

- [9] Wang Heyong, Zheng Jie, Yao Zheng'an, et al. Application of dimension reduction on using improved LLE based on clustering [J]. Journal of Computer Research and Development, 2006, 43(8): 1485-1490 (in Chinese)
(王和勇, 郑杰, 姚正安, 等. 基于聚类和改进距离的 LLE 方法在数据降维中的应用[J]. 计算机研究与发展, 2006, 43(8): 1485-1490)
- [10] Jeffers J, Reinders J. Intel Xeon Phi Coprocessor High Performance Programming [M]. Amsterdam: Elsevier, 2013 (in Chinese)
(James J, James R. Intel Xeon Phi 协处理器高性能编程指南[M]. 陈健, 等译. 北京: 人民邮电出版社, 2013)
- [11] Meuer H, Strohmaier E, Dongarra J, et al. TOP500 supercomputer sites [EB/OL]. 2014 [2014-09-26]. <http://www.top500.org/lists/2014/06>
- [12] Valencia D, Lastovetsky A, O'Flynn M, 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
- [13] Plaza J, Plaza A, Pérez R, et al. Parallel classification of hyperspectral images using neural networks [J]. Computational Intelligence for Remote Sensing Studies in Computational Intelligence, 2008, 133: 193-216
- [14] Sánchez S, Ramalho R, Sousa L, et al. Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs [J/OL]. Journal of Real-Time Image Processing, 2012 [2014-09-26]. <http://link.springer.com/article/10.1007/s11554-012-0269-2>
- [15] Platoš J, Gajdoš P, Krömer P, et al. Non-negative matrix factorization on GPU [C] //Proc of the 2nd Int Conf on Networked Digital Technologies. Berlin: Springer, 2010: 21-30
- [16] Ramalho R, Tomas P, Sousa L. Efficient independent component analysis on a GPU [C] //Proc of the 10th IEEE Int Conf on Computer and Information Technology. Piscataway, NJ: IEEE, 2010: 1128-1133
- [17] Hyvriinen A. Fast and robust fixed-point algorithms for independent component analysis [J]. IEEE Trans on Neural Networks, 1999, 10(3): 626-634
- [18] Hyvriinen A. The fixed-point algorithm and maximum likelihood estimation for independent component analysis [J]. Neural Processing Letters, 1999, 10(1): 1-5



Fang Minquan, born in 1989. PhD candidate. Student member of China Computer Federation. His main research interests include parallel computing and data assimilation.



Zhang Weimin, born in 1966. Professor and PhD supervisor. His main research interests include parallel computing and data assimilation.



Zhou Haifang, born in 1975. Professor. Member of China Computer Federation. Her main research interests include parallel computing and image processing.