计算机科学
COMPUTER SCIENCE

基于国产众核架构的非结构网格分区块重构预处理算法研究

叶跃进，李芳，陈德训，郭恒，陈鑫

引用本文

叶跃进，李芳，陈德训，郭恒，陈鑫. 基于国产众核架构的非结构网格分区块重构预处理算法研究[J]. 计算机科学，2022, 49(6): 73-80.

YE Yue-jin, LI Fang, CHEN De-xun, GUO Heng, CHEN Xin. Study on Preprocessing Algorithm for Partition Reconnection of Unstructured-grid Based on Domestic Many-core Architecture[J]. Computer Science, 2022, 49(6): 73-80.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210900045

基于国产众核架构的非结构网格分区块重构预处理算法研究

叶跃进[1] 李 芳[1] 陈德训[2] 郭 恒[2] 陈 鑫[1]
1 国家超级计算无锡中心 江苏 无锡 214000
2 清华大学计算机科学与技术系 北京 100084
(ye_ddr@foxmail.com)

摘 要 如何高效地解决非结构网格离散访存问题一直是科学与工程计算并行算法和应用领域关注的核心热点问题之一。基于国产申威异构众核架构而设计的分布式区块重连的优化算法,在解决应用课题中的非结构稀疏问题时能始终保持高效的计算性能。通过深入分析众核架构片上的通信机制来设计高效的消息分组策略,以提高从核片上阵列带宽的利用率,同时结合无栅栏数据分发算法充分发挥国产异构众核体系架构网络的性能。通过建立性能模型与实验测试分析可知,该算法在不同访存特征下平均内存带宽能达到理论值的70%以上,与主核串行算法相比具有平均10倍和最高45倍的加速性能。同时通过对多个不同领域的应用进行测试分析也证明了该算法的普适性。

关键词:国产众核架构;非结构网格;片上通信;消息分组;无栅栏数据分发
中图法分类号 TP311

Study on Preprocessing Algorithm for Partition Reconnection of Unstructured-grid Based on Domestic Many-core Architecture

YE Yue-jin[1],LI Fang[1],CHEN De-xun[2],GUO Heng[2] and CHEN Xin[1]
1 National Supercomputing Center in Wuxi,Wuxi,Jiangsu 214000,China
2 Department of Computer Science and Technology,Tsinghua University,Beijin 100084,China

Abstract How to efficiently solve the discrete-memory-accessing problem of unstructed-grid is one of the hot-spot issues in the field of parallel algorithms and application in scientific and engineering computing. The distributed block reconnection optimization algorithm,which is designed on the basis of domestic Sunway heterogeneous many-core architecture,can maintain high computing performance when solving the problem of unstructured sparsity in applications. After deeply analyzing the on-chip communication mechanism of the many-core architecture,an efficient message grouping strategy is designed to improve the bandwidth utilization of on-chip array on the slave core. At the same time,a barrier-free data distribution algorithm is combined to give full play to the network perfor-mance of the domestic heterogeneous many-core architecture. Through the establishment of performance models and experimental analysis,the average memory bandwidth of the proposed algorithm can reach more than 70% of the theoretical value under different memory access situations. Compared with the serial algorithm on the master core,it has an ave-rage of 10 times and a maximum of 45 times performance acceleration. At the same time,the universal applicability of the algorithm is proved by application tests in different fields.

Keywords Domestic many-core architecture,Unstructed-grid,On-chip communication,Message grouping,Barrier-free data distribution

1 引言

非结构网格(或非结构化网格)指网格区域内的内部点不具有相同的毗邻单元,即与网格部分区域内的内点相连的网格数目不同[1]。在实际应用中,几何形状太复杂的模型生成结构网格费时费力,这时采用非结构网格可以在较短的时间内完成计算前处理[2]。因此,非结构网格技术在众多科学与工程计算领域中被广泛地使用,如航空航天、船舶工程、核模拟等。但此类应用存在着大量的细粒度访存特点,严重制约了访存带宽的有效利用。如何高效地解决非结构网格的离散访存问题一直是科学与工程计算并行算法和应用领域关注的核心热点问题之一[3-4],并且随着网格

到稿日期:2021-09-05 返修日期:2022-02-21
基金项目:国家重点研发计划"高性能计算"重点专项(2020YFB0204804,2016YFB0201100)
This work was supported by the National High Performance Computing Foundation of China(2020YFB0204804,2016YFB0201100).
通信作者:李芳(lifang56@163.net)

规模的增加,“访存墙”问题进一步加剧,对计算效率产生的负面影响随之增加,尤其是国产异构众核架构访存瓶颈问题更加突出[5]。

本文针对国产异构众核架构特性而设计的离散访存优化算法对不同访存特性的非结构稀疏问题具有较强的适应性。通过使用基于非结构网格单元映射机制而设计的消息分组策略、基于国产众核架构片上通信机制而设计的无栅栏数据分发算法、基于国产异构架构特有的主从异步并行模型等优化技术可充分发挥国产众核处理器的性能,并以西北核技术研究所结构力学软件 MYDYNA、计算流体力学软件 OpenFOAM(Open Source Field Operation and Manipulation)[6-7]以及稀疏矩阵向量乘核心部件 SpMV[8-9]等为例对算法进行验证,通过对不同数据规模下的核心离散访存代码进行测试分析可知,该算法在不同网格模型和规模下平均有效内存带宽利用率达 70%,与主核串行算法相比具有平均 10.8 倍和最高 45 倍的性能加速。对不同领域应用的测试结果更加充分地证明了该方法的通用性与高效性,同时也为在其他架构下研究非结构网格离散访存问题提供了借鉴意义。

2 背景

2.1 国产众核处理器架构介绍

“神威·太湖之光”系统由国产高性能处理器 SW26010 构建,处理器的计算能力可达 3TFlops,理论带宽为 134 GB/s[10]。如图 1 所示,该众核处理器由 4 个异构群(Core Group,CG)构成,每个异构群由一个主处理器(Main Processing Element,MPE 或主核)和 64 个协处理器(Computing Processing Element,CPE 或从核)及其对应系统接口和控制单元共同组成。主核为控制核心,具备计算、通信、I/O 等多核处理器所具备的功能,负责从核的任务调度和消息控制[11],其包含一个 32 kB 的一级数据缓存、一个 32 kB 的一级指令缓存和一个 256 kB 的二级缓存。从核为计算核心,负责细粒度的计算任务,每个从核具有 64 kB 空间的本地高速数据缓存(Local Data Memory,LDM)。从核上的向量化长度对于单双精度处理并不相同,单精度浮点向量长度为 128 bits,双精度浮点向量长度为 256 bits。从核可以直接离散访问主存或直接连续访存(Direct memery access,DMA),从核阵列内可以通过寄存器通信方式进行高效通信[12-13]。

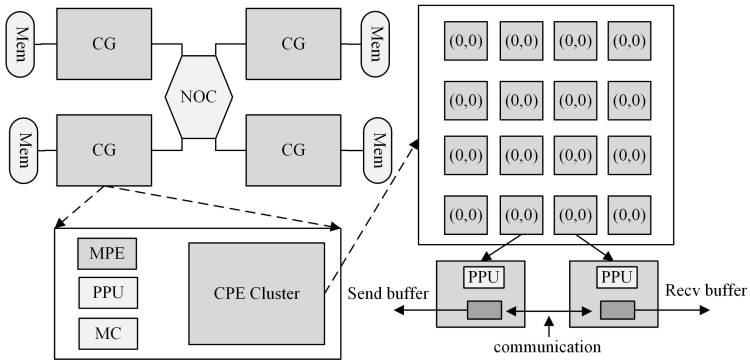


图 1 国产众核处理器

Fig. 1 Domestic many-core processor

2.2 常用离散访存优化

现阶段针对非结构网格的离散访存优化方法主要是用于解决大型稀疏线性方程组为代表的稀疏类问题。针对该类问题的优化方法众多,同时也取得了不少的研究成果。

2.2.1 基于空间压缩的优化

Kourtis 等[14]提出了 CSX(扩展压缩格式),该方法基于共享内存架构的设计思路,利用矩阵内的子结构来压缩元数据。Sun 等[15]在申威众核架构上设计了 SWCSR-SpMV 算法,该算法基于 CSR 存储格式对矩阵进行切分,以保证每行数据可以保存在从核 LDM 空间中,最大限度地提高数据复用。此外,由 Ashari 提出的自适应格式分块行存储格式(Blocked Row-Colunm,BRC)基于矩阵的稀疏特征,使用密集结构的二维分块策略,避免了线程发散和冗余计算,同时也实现了负载均衡[16]。Liu 等[17]在申威架构上也用类似的分块方法,更多地挖掘数据并发性,提升局部性。

2.2.2 基于数据重排的优化

将传统重排方法加以改造,设计与众核架构特性相适应的众核并行排序算法[18],该算法的基本原理是:首先对数据分块并对 LDM 空间进行适配,以有效保证各从核负载平衡;然后对局部数据重排序,以提高局部数据访存的连续性;最后将完成排序的数据分块导入 CPE 内。该方法的优点是能够针对非结构网格应用全局离散局部连续的计算特点,充分利用众核线程的异步并行性,结合数据压缩、变量预处理等其他技术实现高效的优化性能。

2.2.3 基于片上通信的优化

基于片上数据传输的离散访存优化算法是解决稀疏类问题的另一种常用算法。该方法采用任务并行和流水线并行技术,将众核线程按照不同的功能模块划分,利用片上通信技术实现数据流水作业。例如,Lin[19]在“神威·太湖之光”系统上建立的“生产—路由—消费”模型将国产众核处理器中的从核阵列按照数据存储、坐标映射以及计算等不同工作流进行划分,实现流水式并发作业。

上述几种方法在特定应用场景中能发挥出高超的性能,但缺乏普适性与通用性。面对广泛而复杂的科学计算领域,稀疏类问题始终难以找出一种既能满足高效率的计算性能同时又具备良好的通信性、可靠性与稳定性的方法。并且为了满足特定应用场景的性能要求,极有可能需要开发人员进行大量针对性的优化,导致工作量巨大。本文提出的离散访存

优化算法采用非结构网格单元重排序算法，结合了分布式区块重连技术对网格单元遍历过程进行预处理，使网格单元的遍历过程由离散转化为连续，且与计算无关，可广泛适用于不同离散访存的应用场景。同时为了提高文章的可读性，针对该算法设计了一套基于国产众核架构的离散访存库（Discrete Access-memory Library on SunWay，SWDCL）。

3 非结构网格分区块重构预处理算法设计

该算法以单元索引表预建模为基础，分为 3 个模块：单元索引表建模、区间数据重链接机制与分区块数据高通量互连。首先对需要访问的网格单元进行预处理，创建基于众核阵列的坐标访问映射表，并对该索引表进行分组重排，提高数据在不同从核之间流动的吞吐率。在网格单元遍历时先将单元数据分块拷入从核阵列（考虑到 LDM 容量的限制，可根据实际内存需求进行分块），并在从核阵列上以分布式存储方式均匀存储在各从核 LDM 中，并使用片上通信网络进行数据互通。对于非结构化静态网格，网格无需重构，此时对单元索引表的预处理也只需一次即可。该算法的预处理机制省去了计算过程中许多额外的数据处理开销，为其他众核架构处理器提供了很好的参考。但分区块数据高通量互连技术主要针对国产申威众核处理器架构而设计，因此该算法在国产申威众核架构上具有更高的访存性能，同时通过第 4 节也充分验证了其对科学计算领域非结构类型应用均有很好的访存性能。

3.1 基于从核阵列的单元索引表建模

利用分布式数据存储技术将从核阵列视为整体组成共享存储空间（理论上最大容量可为 64×64kB），并将参与计算的数据均匀分布在各 CPE 内，通过建立坐标映射确定数据对应的远程位置，在计算过程中利用该模型进行核间数据传输。对于超大规模网格，可将数据分批次导入并添加全局数据块编码作为当前数据对应全局数据块的编号，如图 2 所示。

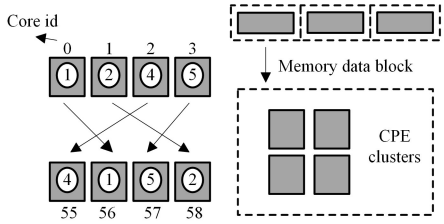


图 2 坐标重映射

Fig. 2 Coordinate remapping

3.2 基于单元遍历规则的数据重链接机制

坐标映射模型可实现从核阵列数据共享，但从核间的数据访问仍然不连续，为充分发挥片上网络带宽性能，建立消息队列模型，将本地数据分块打包后批量传输，以提高核间数据的流水传输效率。本文将该模型定义为 RBA（Remote Block Access）通信方法。基于单元遍历规则的数据重链接机制的算法设计过程大致如下：假设 0 号 CPE 需要访问 1 号 CPE 中数据的坐标依次为(0,1)，(1,9)，(1,8)，(0,7)，(0,4)，(1,2)((x,y)，x 表示全局数据块编码，y 表示在本地 LDM 内的下标编号)，将具有相同 x 值的数据进行归类，再对相同 x 值下的 y 值排序，最终生成坐标有序集：{(0,1)，(0,4)，(0,7)}，{(1,4)，(1,8)，(1,9)}，如图 3 所示。

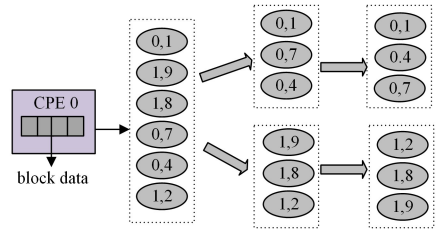


图 3 消息队列

Fig. 3 Message queue

3.3 基于从核片上通信的无阻塞消息互通技术

远程异步并发读数据由于受到操作系统调度的影响，不同从核任务进度不同步，导致在进行核间通信时无法保证数据的完备性。引入无栅栏消息接收技术能在保证数据完备的同时减小从核同步所引起的额外开销。其工作原理如下：当从核完成数据准备后，依次向其他从核发送“完成”的信号，在本地使用一个信号包存放其余从核发送的信号，并进行循环轮转接收。从核会对信号包中的每一个信号进行判断，若为“未完成”，则跳过，否则继续判断下一个信号；当判断为“完成”时，则下达“传输”命令，并进行数据筛选（见图 4），直至完成所有数据的筛选。

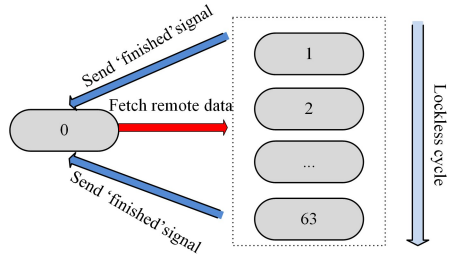


图 4 无栅栏的消息接收技术示意图

Fig. 4 Schematic diagram of fenceless message reception technology

远程异步并发写数据创建多副本数据流水作业模式，在提高并行度的同时避免了数据写冲突问题。在本地建立副本以存放远程消息，并设置一组标识符作为消息应答。当本地完成计算时，通过坐标映射表找到数据对应的远程从核编号，并将信息发送至远程从核对应数据副本中，将远程消息应答字设置为“启动”状态。同时在远程从核上循环轮转查询应答表，如果应答消息为“启动”，则对实际数据进行更新并设置消息应答为“未启动”，以此类推直至完成所有数据更新，如图 5 所示。

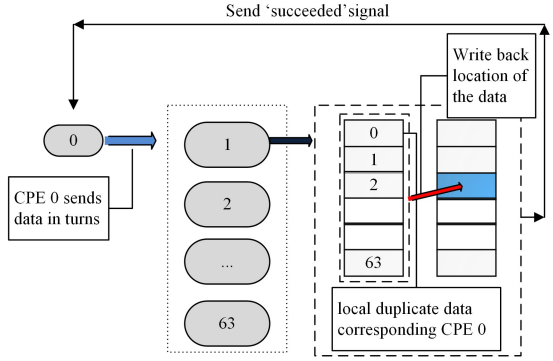


图 5 流水线式消息传输

Fig. 5 Pipelined message transmission

4 实验与分析

4.1 编程接口

为提升易用性，本文为用户封装了简洁的离散访存库编程接口。图 6 给出了部分编程接口内容。

```
1. void SWDCL_prePosmod
2. (
3.   int * d,int N,int M,
4.   int allsize,
5.   struct DCL_TYPEDATA * pos2global
6. );
7. void SWDCL_prePosmod2D
8. (
9.   int * r,int * d,int N,int M,
10.  int allsize,
11.  struct DCL_TYPEDATA * pos2global
12. );
13. void SWDCL_setArea
14. (
15.  struct DCL_TYPEDATA * pos2global,
16.  int * slave_loop_size
17. );
18. void SWDCL_readD
19. (
20.  double * C,double * s_C,
21.  int i_block,
22.  int blocksize,
23.  int vec_size,
24.  struct DCL_TYPEDATA * pos2global
25. );
...
```

图 6 部分离散访存库编程接口

Fig. 6 Appplication programming interface of part of SWDCL

为了便于用户理解上述接口，以简单的涉及离散访存计算核心段为例进行描述，如图 7 所示。

```
1. for I 0->N
2.   A[i] += B[i] * C[d[i]]
3. endfor
```

图 7 离散访存 demo 示意

Fig. 7 Demo of SWDCL

使用离散访存库设计对该热点函数进行众核加速运算，使用方法如图 8 所示。图 8 左边是坐标预处理过程，在计算前调用 swDCL_prePosmod 接口创建坐标映射模型，右边是对循环体众核的并行化改造。swDCL_read 与 swDCL_write 接口通过消息队列模型基于寄存器通信轮转方式实现远程离散数据批量读写功能，是集片上通信与数据重排为一体的功能化接口，并与计算过程相互独立，因此在实际编程中无法利用双缓冲机制实现计算与通信重叠，但高效的异步并发算法可充分发挥片上带宽性能，减小通信开销。离散访存库使申威架构下细粒度访存高延迟低带宽的访存瓶颈问题尽可能得到缓解。离散访存库由于主要解决应用程序中的数据离散访存问题，与实际计算过程无关，因此可适用于绝大多数离散访存类型的应用，并且结合了多种先进的优化算法，可在多数离散访存应用中保持较高的性能。

Master core	Slave core
1. DCL_TYPEDATA buf; 2. SWDCL_prePosmod(&d[0], N, M, DCL _ REAL8, &buf); 3. SWDCL_pre_join();	1. SWDCL_setArea(&buf, &block_n); 2. For I 0 -> block_n 3. SWDCL _ getLocInfo(i, &bsize, &mst, &med); 4. s_C = alloc; 5. SWDCL_read(&C[0], I, bsize , 1, & buf, &s_C [0]); 6. for j 0 -> bsize 7. s_A[j] = s_B[j] * s_C[j]; 8. endfor 9. SWDCL_write(&s_A[0], I, bsize , 1, & buf, &A[0]); 10. endfor

图 8 众核优化代码

Fig. 8 Many-core optimizatedcode

4.2 性能模型和分析

在建立性能模型前简要给出以下性能测试的硬件及软件环境：以下性能模型全部基于“神威·太湖之光”的 SW26010 异构众核架构处理器，主频为 1.45 GHz，从核私有缓存空间大小为 64 kB，DMA 内存带宽的实测内存带宽为 25 GB/s，单核组内存 DDR3 容量为 8GB，编译器使用国产 swgcc 指令集。

假设本地从核与其余从核 RBA 通信的次数为 K^i（i 表示核号）。设从核主频为 FR_{spe} GHz，从核访 LDM 带宽为 BW_{ldm} GB/s，平均筛出一个正确数据需要 C_{ldm} 次 ldm 访问，设置 2 个表示离散程度的变量 L_m^i 和 β_m^i，用于表示第 i 号从核从其余从核进行第 m 次 RBA 的通信量（单位为 byte），那么 $\sum L_m^i$ 就表示第 i 号从核的消息总长，β_m^i 表示第 i 从核从其余从核进行第 m 次 RBA 的数据实际利用率，$\sum(L_m^i \times \beta_m^i)$ 表示第 i 号从核实际的有效数据量。假设 RBA 的延迟为 T_{delay} 拍（拍数即为 CPU 时钟周期数），k^i 表示本地从核与其余从核的通信总次数，则从核调用一次离散访存库的时间 T_{dcal}（单位为拍）满足：

$$\begin{aligned} T_{dcal} &= t_0 + t_1 + \max_i(t_2^i) \\ &= T_0 + \frac{FR_{spe}M}{BW_{dma}} + \max_i\{\sum_{m=1}^{k^i}(\frac{FR_{spe}L_m^i}{BW_{rba}} + (FR_{spe} \cdot C_{ldm} \times \\ &\quad L_m^i \times \beta_m^i)/BW_{ldm}) + k^i \cdot T_{delay}\} \end{aligned} \tag{1}$$

针对不同访存规模对性能模型(1)进行验证。访存总量从 60 kB 依次增加到 4000 kB（单核组规模），分别记录调用一次离散访存库的时间（实测时间），并计算性能模型(1)得到的理论时间，两者的对比结果如图 9 所示，其中偏差率 = |实测时间-理论时间| /理论时间。

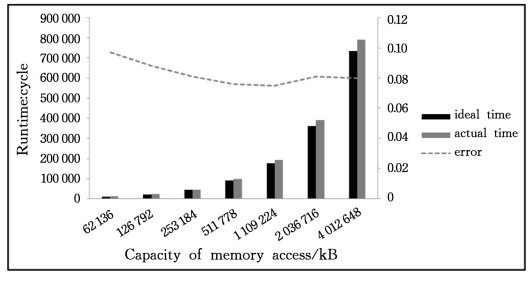


图 9 不同访存规模理论时间与实测误差对比

Fig. 9 Error comparison between theoretical time and actual time with different access memoryies

由图 9 可知，在不同访存规模下离散访存库的性能模型理论时间与实际运行时间接近，误差均在 10% 以内。当访存总量较小时，由于运行时间较短，消息队列、通信延迟、指令延迟等将导致实测时间与理论值误差较大；随着访存总量增大，这些系统误差对实验的影响逐渐减小，理论时间与实测误差逐渐缩小；但是，随着访存总量继续增加，尤其当数据规模在 1 MB 以上时，从核间的带宽竞争加剧，从而导致误差增大，但误差范围仍不超过 10%。通过上述实验验证了性能模型的正确性。

通过对模型(1)的大量测试分析发现，影响离散访存库性能的因素主要是消息队列模型的性能，而消息队列模型性能主要与实际访存总量和片上消息总长的比值有关，该比值在一定程度上反映了数据离散程度。假设：

$$\alpha=\frac{\text{有效数据总量}}{\text{片上消息总长}}$$

由于数据离散性导致访存频率过于密集，因此访存带宽的性能决定了稀疏类问题的整体计算性能，提高稀疏类问题的最主要方法就是提高访存带宽利用率。本文采用 DMA 实际访存带宽利用率来说明计算性能，并采用众核与纯主核的并行加速比的测试方法补充验证离散访存库的性能。

使用不同访存总量 MBytes 分别测试不同 α 值下的实际内存带宽利用率和对应主核性能加速比，结果如图 10 所示。

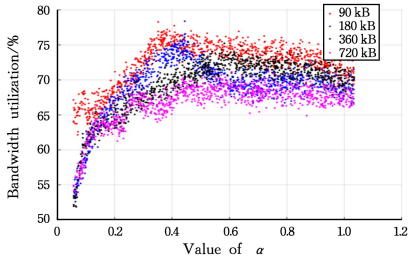


图 10 不同访存总量实际内存带宽利用率随 α 的变化趋势

(电子版为彩图)

Fig. 10 Utilization of actual memory bandwidth changes with α with different access memories

图 10 中，横坐标为 α，纵坐标为实际访存带宽利用率，红色标记表示访存总量为 90kB 时实际访存带宽利用率 α 值的变化趋势，蓝色标记表示 180 kB 访存总量时实际访存带宽利用率 α 值的变化趋势，黑色标记表示 360kB 访存总量，紫色标记表示 720 kB 访存总量。从图 10 中可看出，当 α 的值在 0.4 至 1 区间时，不同访存总量下的实际访存带宽利用率均在 70% 及以上，当访存总量一定时，随着 α 的增加(即消息总长减小)，访存带宽均呈上升趋势。由于消息总长越小，数据的实际利用率越高，数据就越集中，因此调用一次离散访存库的时间越短。当 α 继续增加至 1.035 时出现了带宽下降趋势，这是由于数据过于集中在某些从核，导致 RBA 带宽竞争加剧，从而影响了访存带宽。因此，影响离散访存库性能的因素除了 DMA 访存带宽受限之外，还与从核片上消息队列模型的性能有关，整体访存性能会随着消息队列模型的性能增大而增大。为保证整体性能高效与稳定，在下一阶段将会深入优化片上消息队列模型，建立更加高效的消息分类算法，减小消息总长，提高数据利用率。上述实验测试分析也充分说明了离散访存库在不同网格模型和网格规模下始终保持着较高的性能。

4.3 应用实例

OpenFoam(Open Source Field Operation and Mainpulation)是一款通用开源计算流体力学软件[20]，涵盖了众多大型线性方程组求解器，其中预共轭梯度法(Pre-conditioned Conjugate Grandient，PCG)是常用的几种迭代方法之一。本文以某真实应用场景为例，详细分析 PCG 求解器中使用本文所述的分区块重构预处理算法在国产"神威·太湖之光"系统架构下的性能。PCG 算法的原理如图 11 所示，该算法主要的性能瓶颈在于进行预处理后的梯度向量以及稀疏矩阵向量乘法运算的过程，但在 OpenFOAM 中原有的预条件子包括 Gauss-Seidel 迭代法、不完全 Cholesky 分解(DIC)、不完全 LU 分解(DILU)，以及代数-几何多重网格(GAMG)迭代法都不利用众核并行，而对角预条件子(DIA)的收敛性较差，因此为了在申威异构众核架构进行更好地适配，本文选取原始矩阵作为条件矩阵，同时将求逆过程进行分解，经简化后得到 1 阶 Neumann 多项式[20]迭代法，并将其作为 PCG 的预条件子，迭代公式如下：

$$\begin{cases} z_0=\boldsymbol{D}^{-1}r \\ z_{n+1}=(r-(M-\boldsymbol{D})\cdot z_n)\cdot D^{-1},(M=\boldsymbol{A}) \end{cases} \tag{2}$$

$x_0=0$

$r_0:=b-Ax_0$

$z_0:=M^{-1}r_0$

$p_0:=z_0$

$k:=0$

repeat

$\alpha_k:=\dfrac{r_k^T z_k}{p_k^T A\, p_k}$

$x_{k+1}:=x_k+\alpha_k p_k$

if r_{k+1} is sufficiently small then exit loop end if

$z_{k+1}:=M^{-1}r_{k+1}$

$\beta_k:=\dfrac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$

$p_{k+1}:=z_{k+1}+\beta_k p_k$

$k:=k+1$

end repeat

$x_{k+1}\rightarrow exit$

图 11 PCG 代码示意图

Fig. 11 Schematic algorithm of PCG code

上述公式中，矩阵 $\boldsymbol{A}$ 为方程组系数矩阵，矩阵 $\boldsymbol{D}$ 是由系数矩阵 $\boldsymbol{A}$ 的主对角线元素组成的对角矩阵。选用上述条件子后，PCG 算法中稀疏矩阵向量乘法(Sparse Matrix-Vector Multiplication，SpMV)耗时占总耗时的 70%，因此主要性能瓶颈为稀疏矩阵向量乘运算，众所周知，SpMV 为典型的离散访存问题，本节将会对 SpMV 进行优化前后的性能分析。优化方法为本文提出的分区块重构与处理算法，使用依据该算法所封装的离散访存库与原始纯主核程序进行对比，分析在

实际应用场景中的优化前后的性能。原始 OpenFOAM 是以 Coordinate(COO)为存储格式,将矩阵分解为 $\boldsymbol{L}$,$\boldsymbol{D}$,$\boldsymbol{U}$ 这 3 个矩阵进行存储(见图 12),众所周知,无论是 COO 格式还是其他压缩格式,稀疏矩阵的非零元素在内存中是连续存储的,同时向量在内存中也是连续的。但在实际计算时,由于非零元对应矩阵位置不连续,因此与非零元对应的向量元的访问位置是不连续的,故在 SpMV 中离散访存问题实际上是对向量部分的访问。

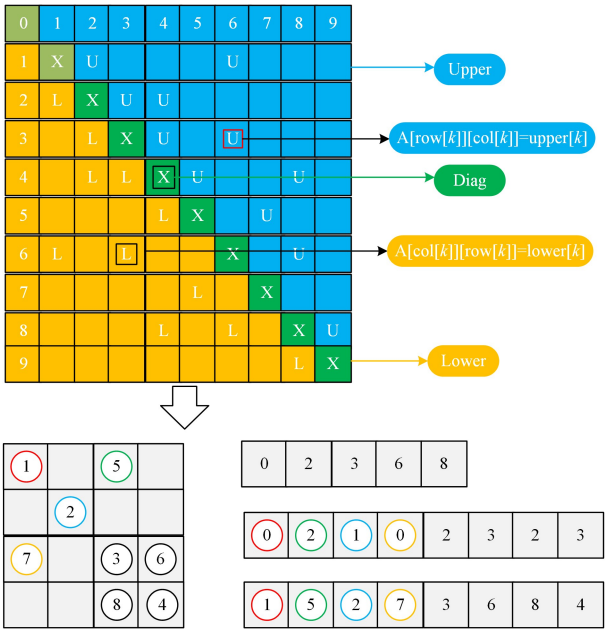


图 12 COO 格式与 CSR 格式压缩示意图

Fig. 12 COO data format convert to CSR format

在进行矩阵向量乘法运算时,每一次乘加运算需要两次非规则离散访存,即一次离散读加一次离散写,且离散读与离散写的访存局部性较差,难以发挥访存带宽的性能,通过对该真实应用场景进行分析发现,矩阵的非零元主要集中在主对角线区域,因此为了提高访存的局部性,充分发挥访存带宽的性能,我们采用 Compressed Sparse Row (CSR)行压缩作为矩阵的压缩格式进行存储。如图 12 所示,通过 CSR 压缩后,矩阵中 3,6,8,4 元素在内存中连续,与之对应的向量部分的访问也连续。同时,采用本文的分区块重构预处理算法所设计的离散访存库测试了该应用在不同算例规模下的性能。

表 1 列出了该应用算例在不同网格规模下原始主核程序与使用离散访存库后的各部分运行时间,同时也列出了不同网格规模下的稀疏矩阵非零元个数。表 1 中,第一列数据表示平均单核组的网格规模数,第二列数据表示平均单核组非零元个数,第三列数据表示原始纯主核程序的 SpMV 运行时间,第四列数据表示使用 SWDCL 的 SpMV 预处理时间,第五列数据表示使用 SWDCL 的 SpMV 实际运行时间,最后一列数据表示原始主核程序与使用 SWDCL 的 SpMV 性能加速比值。从测试数据可以看出,该算法的预处理过程在不同网格规模下的耗时占比普遍较小,其原因是非结构静态网格的网格模型在计算前已确定且不发生变化,因此只需进行一次预处理。从表 1 可看出,预处理的时间占比远小于其他操作,因此可忽略不计。使用离散访存库的 SpMV 性能相比原始主核程序有了较大提升,单核组 1 万矩阵规模(即表中的网格规模),2.1 万个非零元素,SpMV 众核加速比为 13.5,当矩阵规模上升至单核组 16 万时(44.3 万个非零元)众核加速比上升至 23.8 倍。随着单核组网格规模的增加,使用本文提出的离散访存预处理算法的性能随之增加,原因是在小规模算例中由于非零元素总量较低同时矩阵规模较小,因此对应向量的长度较短,从第 3 节可知,该算法会将需要所有非零元和离散访问的数组(即 SpMV 的向量部分)以分布式存储方式均匀存放在众核阵列内,此时向量长度较短,因此每个从核存放该向量的数据量小,影响了从核内的数据局部性,导致无法发挥片上阵列的通信带宽。当算例规模增大至 16 万时,向量的长度也随之增长至 16 万,非零元数量为 44.3 万。向量长度的增大在一定程度上提高了从核内部的数据局部性,此时的片上通信带宽利用率最高。但随着算例规模增加,向量长度的增长超过众核阵列一次性可存放的私有缓存空间,则需要将向量与矩阵非零元分块存放至从核 LDM 中,将向量的一部分从内存中 DMA 至众核阵列上,访问结束后再更新下一块。由于计算时需要对各块进行遍历访问,因此需要多次反复 DMA 向量块,影响了整体众核加速性能。如表 1 所列,在 16 万的算例规模之后加速比略有下降,佐证了上述分析。表 1 所列的测试结果不仅展示了本文提出的优化算法具有良好的加速性能,同时也指出了该算法目前仍存在的问题,给后续的工作提供了良好的思路。

表 1 原始主核程序与调用离散访存库各部分总耗时分布

Table 1 Total time consumption distribution of original main core program and call SWDCL

The grid number of per CG	The number of non-zero	SpMV runtime of original code	SpMV preprocess runtime of using swDCL	SpMV runt-ime of using swDCL	Speed-up of SpMV
1×10^4	3.15×10^4	149.42	0.0015	11.07	13.5
2×10^4	6.75×10^4	296.59	0.002	16.57	17.9
4×10^4	13.3×10^4	573.34	0.004	26.42	21.7
8×10^4	27.2×10^4	1196.24	0.007	53.64	22.3
16×10^4	56.3×10^4	2455.27	0.013	103.16	23.8
32×10^4	119.9×10^4	5202.72	0.057	232.11	22.4

为了进一步体现本文算法的先进性,单独对 SpMV 进行测试,将上述实际应用所给的算例数据作为测试的输入数据,测试单个 intel E5-2680V3 处理器与单个国产 SW26010 处理器下 SpMV 的运行时间及加速比,结果如图 13 所示。其中在国产 SW26010 上使用本文算法对从核阵列进行了适配,而在 intel 上使用原始串行程序进行测试。通过分析表 1 可知,采用本文算法时,在实际应用中 SpMV 的预处理时间远短于实际计算时间,因此下文的 SpMV 测试时间不包括预处理部分。如图 13 所示,横坐标表示单处理器下不同数据的规模,对应 SpMV 中的矩阵列数,纵坐标主轴(左轴)表示不同数据规模下的 SpMV 的运行时间(单位为 s),纵坐标次轴(右轴)表示使用 SW26010 处理器相比 intel E5-2680v3 的加速比。由图 13 可知,在 8 万数据规模时,SW26010 的加速比接近

intel E5-2680V3 的 5 倍，单数据规模上升至 64 万时加速比最高接近其 9 倍，同时在 64 万规模之后加速比均大于 7。因此 SpMV 计算在国产 SW26010 处理器下采用本文设计算法对众核进行适配后单 CPU 的性能普遍优于 intel E5-2680V3 单 CPU 的性能，由此进一步体现了本文算法的先进性。

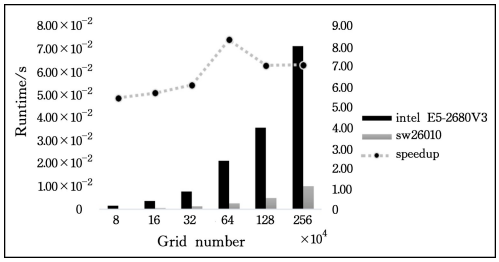


图 13 不同处理器之间运行时间及加速比

Fig. 13 Runtime and speedup between difference processor

为了更好地展示本文算法在实际应用场景发挥的作用，进一步测试该应用在不同算例规模下的求解器(PCG)的运行时间和优化后求解器的整体运行时间及加速比，结果如图 14 所示。

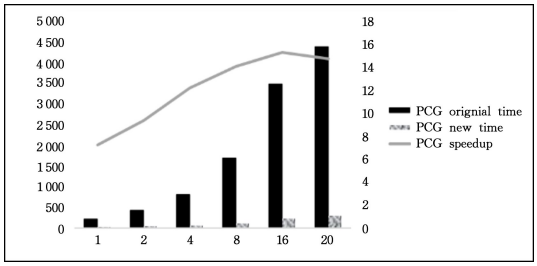


图 14 优化前后 PCG 的整体运行时间及加速比

Fig. 14 Overall runtime and speedup of PGG before and after optimization

图 14 中，黑色柱形为原始纯主核程序，灰色柱形为调用离散访存库之后的优化程序。横坐标表示不同算例下的网格规模，纵坐标的主轴(左边)表示整体 solver 的运行时间，次轴(右边)表示优化前后 solver 的加速比值。从图中可看出，在不同算例规模下使用本文算法优化后性能较原始纯主核程序性能提升显著。随着网格规模增加，solver 整体的加速比从 1 万网格规模的 7 倍上升至 16 万网格规模的 15.4 倍，当网格规模上升至 20 万时加速比略有下降，这是由于经过算法改造之后的 PCG 计算热点 SpMV(根据上述内容可知该部分占整体 PCG 求解时间的 70%)在 16 万网格规模之后的加速比略有下降，影响了整体 solver 的加速性能。同时通过进一步测试可知，优化后该实际应用的整体性能提升了 400% 以上。

本文又测试了其他应用领域的非结构网格应用课题，以中国空气动力研究发展中心计算流体力学软件 AHL3D-uns 为例，优化前程序运行时间占整体运行时间的 20%，优化后下降至 1.8%；以西北核技术研究所结构力学软件 FEMDYNA 为例，测试原始纯主核程序与使用离散访存库后的众核程序加速比核心段最高加速达 45 倍，并且在不同规模下应用算例整体加速比均在 30 倍以上，最高接近 35 倍。法国电力集团开发的一款通用开源计算流体力学(CFD)软件 Code_Saturne 通过结合多个真实算例与数据规模测试可知，以离散访存为主的计算热点优化前后占整体运行时间的比值由原来的 60%下降至 8%，并使课题整体性能提升了 100%以上。并且对西北核技术研究所三维 PIC 应用算例进行测试分析发现，使用离散访存库之后主从核加速比达 40 倍，课题整体性能提升了 10 倍以上。因为工作量的原因，未进行其他离散访存优化方法在上述实际应用上的适配和性能对比分析，但我们从性能模型以及与 CPU 的性能对比等多个方面证明了本文方法的先进性，由此充分说明该离散访存优化算法在不同网格规模和不同应用场景下都具有良好的加速性能。

结束语 本文基于国产申威众核架构而设计的非结构网格分区块重连预处理算法，以分布式存储技术为基础，对单元索引进行预建模，减少网格单元遍历过程中冗余查表操作；通过对单元索引表分组重排，提高数据在不同从核之间流动的吞吐率；基于从核片上通信机制，设计无阻塞消息互通策略，实现通信与计算的重叠。最终通过实验分析以及多个应用实例测试表明，该优化方法在众多非结构网格应用中均能保持较高的加速性能。为了进一步提高性能，后续将深入研究消息分组策略，在提高数据利用率的前提下更大程度地减少片上通信复杂度，降低各从核间带宽竞争。同时进一步增强软件适应性和鲁棒性，丰富应用场景，提高离散访存库在不同领域的运用范围。

参 考 文 献

[1] LI YY, XUE W, CHEN D X, et al. Performance optimization of sparse matrix vector multiplication on Sunway many-core architecture[J]. Chinese Journal of Computers, 2020, 43(6): 1011-1020.

[2] ZHENG F, LI H L, LV H, et al. Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture[J]. Journal of Computer Science and Technology, 2015, 30(1): 145-162.

[3] GUNNELS J A, HENRY G M, VAN DE GEIJN R A. A Family of High-Performance Matrix Multiplication Algorithms[C]// Proceedings of the International Conference on Computational Sciences-Part I. London, UK, UK: Springer-Verlag, 2001: 51-60.

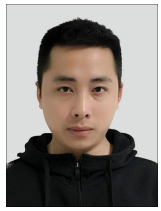
[4] GOTO K, VAN DE GRIJN R. High-performance Implementation of the Level-3BLAS[J]. ACM Transaction on Mathematical Software, 2008, 35(4): 1-14.

[5] CHECCONI F, PETRINI F, WILLCOCK J, et al. Breaking the speed and scalability barriers for graph exploration on distributed-memory machines[C]// International Conference on Storage Anal & High Performance Computing Networking. SC12, 2012.

[6] UENO K, SUZUMURA T, MARUYAMA N, et al. Exreme scale breath- first search chon super computer[C]// Big Data (Big Data). IEEE International Conference, 2016: 1040-1047.

[7] BEAMER S, BULUC A, ASANOVIC K, et al. Distributed memory breadth-first search revisited: Enabling bottom-up search [C]// Parallel and Distributed Porcessing Symposium Work-

shops. IEEE International Conference, 2013: 1618-1627.
[8] CHECCONI F, PETRINI F. Traversing trillions of edges in real time: Graph exploration on large scale parallel machines[C]// International Conference & International Parallel and Distributed Processing Symposium. IEEE International Conference, 2014: 425-434.
[9] BISSON M, BERNASCHI M, MASTRONSTEFANO E. Parallel Distributed Breadth First Search on the Kepler Architecture [J]. IEEE Transaction on Parallel and Distributed System, 2016, 27(7): 2091-2102.
[10] LIAO J F. Redesigning CAM-SE for Peta-Scale Climate Modeling Performance on Sunway TaihuLight[D]. Beijing: Tsinghua University, 2017.
[11] LI F, LI Z H, XU J X, et al. Research on Adaptation of CFD Software Based on Many-core Architecture of 100P Domestic Supercomputing System [J]. Chinese Journal of Computers, 2020, 47(1): 1-8.
[12] AO Y L. Research on Key Optimizations of Sparse Matrix and Stencil Computation for the Domestic Large Many-core System [D]. Hefei: University of Science and Technology of China, 2017.
[13] AN H, YU Y, CHEN J S, et al. Pipelining Computation and Optimization Strategies for Scaling GROMACS on the Sunway Many-core Processor[C]// International Conference on Algorithms and Architectures for Parallel Processing. 2018: 134-137.
[14] KOURTIS K, KARAKASIS V, GOUMAS G, et al. Csx: An extended compression format for spmv on shared memory system [J]. ACM SIGPLAN Notices, 2011, 46(2): 247-256
[15] SUN Q, ZHANG C Y. Bandwith reduced parallel SpMV on the SW26010 many-core platform[C]// Proceedings of the 47th International Conference on Parallel Processing Eugence. USA, 2018: 1-10.
[16] ASHARI A, SEDAGHATI N, EISENLOHR J, et al. An efficient two-dimensional blocking strategy for sparse matrix-vector multiplication on GPUs[C]// Proceedings of the 28th ACM International Conference on Supercomputing. ACM, 2014: 273-282.
[17] LIU C X, XIE B W, LIU X, et al. Towards efficeient SpMV on sunway many-core architectures[C]// Proceedings of the 2018 International Conference on Supercomputing. Portland, USA, 2018: 363-373.
[18] NI H, LIU X. Many-core Optimization Technology Of Unstructured-grid On SunWay TaihuLight[J]. Computer Engineering, 2019, 45(6): 51-57.
[19] LIN H. Extreme-scale graph analysis on heterogeneous architecture [D]. Beijing: Tsinghua University, 2017.
[20] APHU E S, BRANTSON E T, ADDO B J, et al. Development of Finite Difference Explicit and Implicit Numerical Reservoir Simulator for Modelling Single Phase Flow in Porous Media[J]. Earth Science, 2018, 134: 2-10.



YE Yue-jin, born in 1991, master, engineer, is a member of China Computer Federation. His main research interests include high performance computing and so on.

LI Fang, born in 1980, postgraduate, Ph.D, associate professor. Her main research interests include high performance computing and so on.

（责任编辑：喻藜）