面向异构众核超级计算机的大规模稀疏计算性能优化研究

胡正丁，薛巍
清华大学计算机科学与技术系，北京 100084

摘要
随着超级计算机技术的发展，大数据应用中大规模稀疏问题的求解成为可能，而稀疏问题的不规则计算和访存特性又给应用实现和性能优化带来了挑战。异构众核是超级计算机系统中的常见架构，其设计向应用开发者提出了高要求，如何发挥其强大的计算能力成为一个难题。分析了稀疏计算的性能优化挑战，介绍了基于典型异构众核计算机系统的3种大规模稀疏处理类应用设计和性能优化案例，以期为在新一代异构众核系统上开展大规模稀疏计算问题求解提供借鉴。

关键词
大数据应用；稀疏问题；高性能计算；性能优化

中图分类号：TP31 文献标识码：A doi: 10.11959/j.issn.2096-271.2020032

Research on performance optimization for large-scale sparse computation over many-core heterogenous supercomputer

HU Zhengding, XUE Wei
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract
With development of supercomputer technique, it is possible to solve extra-scale sparse problems in big data applications. However, irregular feature in computation and memory access of sparse problems brings challenges to implementation and optimization of applications. Many-core heterogenous architecture is popular in supercomputer design, which advances a higher requirement for application developers. How to utilize its extraordinary computing ability becomes a very difficult problem. Challenges in optimizing sparse computing problems were analyzed, and three cases of implementation and optimization based on typical many-core heterogenous computer system were introduced, which of all achieve very high performance. Experiences in those successful cases were summed up, to better solve extra-scale sparse computing problems on many-core heterogenous system of new generation.

Key words
big data application, sparse problem, high performance computation, performance optimization

1 引言

近年来，随着计算机系统和大数据技术的发展，大规模数值计算、科学计算等在大型异构并行系统上的应用逐渐深入。从对自然现象的模拟和预测到工程学设计和产品研发，超级计算（以下简称超算）在这些领域发挥着不可或缺的作用。与此同时，应用的需求反过来也促进了超级计算机的发展，更大型超级计算系统的构建使得更多富有挑战性的任务的解决成为可能。

大规模计算往往与大数据，特别是大规模稀疏数值问题紧密相连。数值天气预报通过数值计算求解描述天气演变过程的流体力学和热力学的方程组，以预测未来的大气运动状态和天气现象。若在全球采取千米级分辨率，会产生百亿规模的计算网格，相应的联立方程组规模会达到千亿级别。7天左右的天气预报需要约6万步迭代，而涉及气候预测的时间跨度甚至多达数年，其中的计算规模和数据规模是难以想象的。在线网络欺诈分析结合大数据和人工智能技术检测网络欺诈行为，需要保证预测结果的准确性和实时性。全球的中文网页约有2 700亿个，链接数量达12万亿个，相应网页图存储规模达到137 TB，这无疑对数据存取和算法运行效率提出了很高的要求。

稀疏问题的计算核心（如稀疏矩阵运算和图遍历等）在大规模计算中广泛存在。天气预报、地震分析等自然现象模拟过程需要对大规模偏微分方程进行求解，其中涉及频繁的稀疏矩阵运算操作。而蛋白质交互、基因工程和脑科学等科学研究工作需要对大规模稀疏图进行生成、遍历和处理。

超级计算机系统由于具有强大的存储和计算能力，成为解决大规模稀疏问题的有效选择。而由于其访存和计算模式的特殊性质，稀疏问题在并行和分布式计算机系统上的求解成为一个难题，具体体现在任务划分、计算调度、存储管理和功耗管理等多个方面。超级计算机给稀疏问题求解带来了全新的机遇和挑战。

因此，本文针对基于异构众核的超级计算机——“神威·太湖之光”的大数据稀疏问题解决和优化方案进行阐述，探讨异构众核计算机架构下大规模稀疏计算性能优化的一般性方法，为在新一代异构众核系统上开展大规模稀疏计算问题求解提供借鉴。

2 稀疏问题的计算挑战

因为稀疏问题具有非规则的计算与访存特征，所以其在大规模超级计算机中的求解面临严峻的挑战，主要包括以下几点。

（1）不规则的主存储器访问

随着CPU主频的提高和处理器计算能力的不断增强，CPU运算速度与主存带宽不匹配的问题越来越严重。与计算密集型应用不同，稀疏计算核心的计算访存比往往较低。典型的稀疏计算问题（如稀疏矩阵向量乘法、基础向量矩阵运算、模板计算等）只有常数级别的计算访存比，而其余典型算术核心（如涡格法、快速傅里叶变换、粒子法）的计算访存比会达到$O(\log N)$甚至$O(N)$的级别。因此对于稀疏计算型应用而言，存储器访问的开销可能远远超过计算本身带来的开销，使得访存问题成为应用开发过程中需要重点关注的部分。

计算机系统的存储器架构往往是多级的。靠近处理器的存储层级一般存取速度快，但容量较小；反过来，远离处理器的存

储层级容量大、速度慢。对于大规模计算机系统而言，这种容量和计算速度的对比往往更加夸张。因此，要解决应用的访存问题，需要解决大规模稀疏数据的存储管理策略问题，尽量将频繁使用的数据放在高层级，减少低层存储器的访问次数，同时做好数据的分块和搬运策略，增强访存的连续性和一致性。

稀疏型计算问题的访存模式是不规则的。稀疏计算问题的数据局部性较差，可能存在离散化、随机化、不规则访存的问题，随机化访存对数据分块和局部化并不友好，而细粒度访存会导致不同节点的竞争，增大存储总线的压力。稀疏计算问题的这种特性给开发者的存储管理策略带来了许多困难。

此外，传统的动态随机存取存储器（dynamic random access memory，DRAM）价格昂贵、能耗高、性能不稳定，给许多大数据稀疏问题的解决带来了限制。近年来出现了大量的新型非易失性存储器（non-volatile memory，NVM），它们具有价格和能耗较低、容量大、性能高的特点，给内存存储与计算模式带来了巨大的变革，新型内存计算技术正在蓬勃发展[1]。为了充分利用NVM容量大和DRAM读写性能好的优势，并且最大限度地避免各种存储介质的缺陷，DRAM-NVM异构内存系统的设计与优化成为研究的热点。这种异构系统的实现面临体系结构、系统软件、编程模型等多个层面的挑战，相关研究工作已经提出了具有针对性的解决方案[2]。例如，相变存储器（phase change memory，PCM）就是非易失性存储器的一种，其存储密度较高、持久性强，有学者通过将PCM与DRAM结合来构建优势互补的混合存储架构[3]。

（2）可并行化与负载均衡

部分稀疏计算核心可能存在非规则的计算模式。比如，在求解线性方程组用到稀疏矩阵LU分解（LU factorization）与稀疏三角矩阵方程求解（sparse triangular solver，SpTRSV）的过程中，不同位置的数据具有计算依赖关系，存在求解的先后顺序。在模板计算中，每个进程需要等待halo区，也就是由其部分邻居进程负责计算的数据区域完成后才可开始下一步计算。这种基于数据依赖的非规则计算模式使得传统分块并行方法不再适用，开发者需要最大限度地挖掘应用中可并行化的部分。

同时，多核计算机系统中每个处理器的负载也是需要考虑的问题。稀疏矩阵中的非零元排布如**图1**所示，其中b和x分别表示矩阵的行和列两个维度。在一个稀疏矩阵中，不同行/列间的非零元分布密度可能存在巨大差别。如果采用静态分块方法，会导致不同处理器负责计算的非零元数目不均衡。这种不均衡不仅会大大降低应用的性能，还可能造成部分处理器资源的浪费，增大应用运行的功耗和成本。这对大规模稀疏问题的问题划分和任务分配提出了更高的要求。

（3）数据传输与通信

在大规模异构计算机系统上，稀疏问题的求解往往涉及频繁的进程间/节点间通信。这种通信给I/O和节点间网络带来了巨大压力。随着众核架构的广泛使用，处理器主频和单核的计算能力受限，原有的超算基础软件（如MPI通信库）主要面向进程通信开发，其中的大量计算功能依赖单核的计算能力，已经无法满足新的架构需求；同时，随着管理进程数的增加，超算基础软件本身的内存开销成为一个不可忽视的问题，这些都成为限制大数据稀疏应用性能提升的关键因素。

随着超算规模的增大，相对固定的系统配置无法与多种多样的应用计算和通信模式有效匹配，同时在通信、I/O层面的应

用相互干扰问题愈加突出。通信瓶颈往往会对应用性能的可扩展性与稳定性造成影响。因此，如何解决数据的传输和通信问题，对于应用开发者来说是一项挑战。

在大数据时代，应用的问题规模和相应的数据规模呈爆炸式增长，大量非结构化数据的出现使得提取信息的难度越来越大，也对外存储器的访问效率提出了更高的要求。显然，基于磁盘的存储系统已经难以满足日益增长的访问需求。与磁盘相比，闪存（flash memory）具有体积小、能耗低、带宽高、时延低、抗震性强、可靠性高等特点，研究人员正着力于构建大规模闪存存储系统，以充分发挥闪存优势，适应大数据环境的发展[1]。

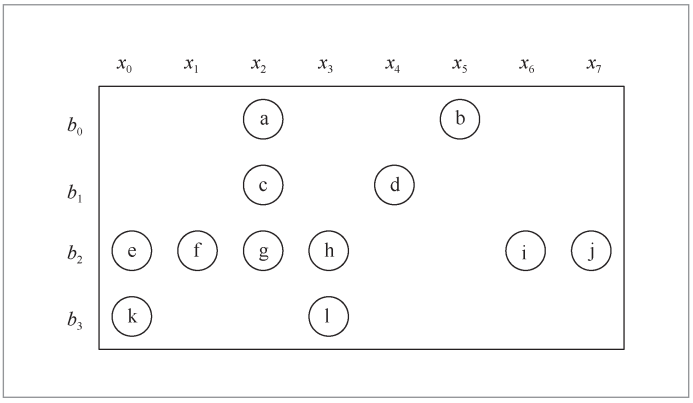


图1 稀疏矩阵中的非零元排布

3 异构众核架构及挑战

本文以典型的异构众核超级计算机——“神威·太湖之光”中的申威26010众核处理器（SW26010）为例，介绍异构众核架构及其应用开发的挑战。

3.1 异构众核架构设计

图2所示为典型的采用异构众核架构的申威26010众核处理器，每个处理器包含4个核组，每个核组通过片上网络互联，并通过PCI-E 3.0对外连接。每个核组独

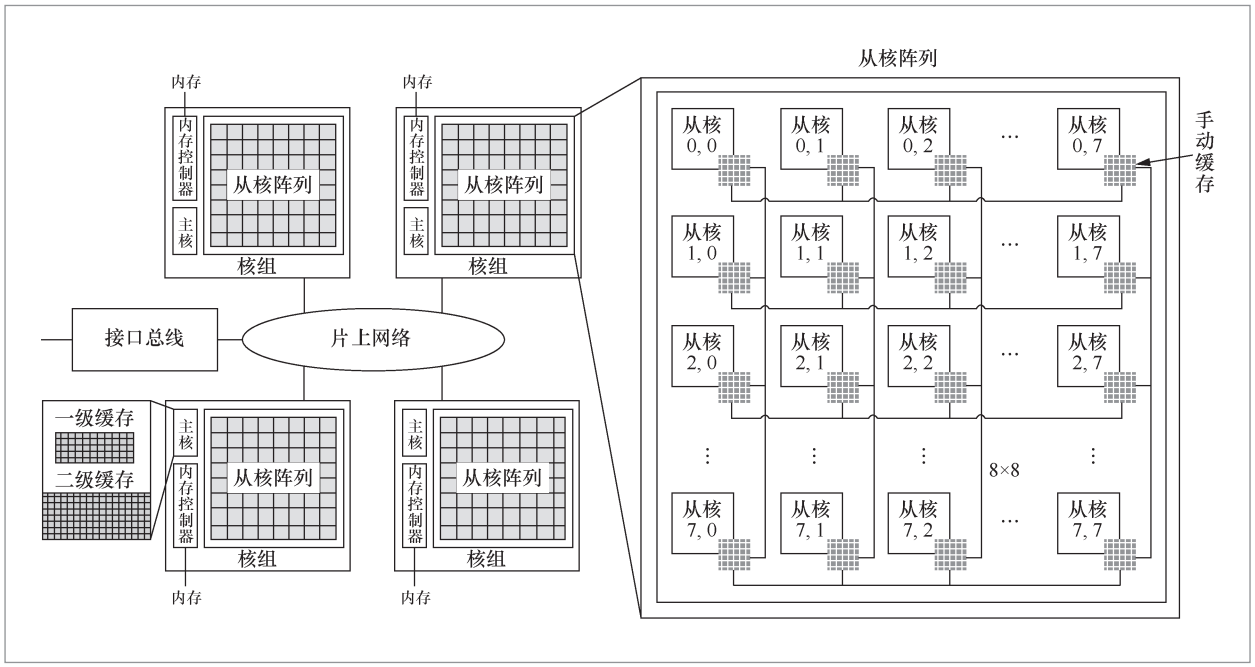


图2 SW26010架构

立运行，包含一个控制核心（主核）、64个运算核心（从核）和一个内存控制器。整个处理器可以提供3.06 TFlops的双精度浮点计算峰值性能和136 GB/s的理论总内存带宽。

主核拥有常规的两级Cache系统，通常被用于执行管理和通信任务。从核具有很高的浮点运算性能，通常被用于执行计算任务。与常规的缓存方式不同，每个从核包含一个大小为64 KB的便笺存储器（local data memory, LDM）。LDM由静态随机存取存储器（static random-access memory, SRAM）设计，与主存DRAM的地址空间分离，并对用户可见，用户需要显式地控制数据在主存和LDM之间的传输。每个核组的64个从核构成8×8的网格阵列，每两行从核共享一条连接到内存控制器的总线。从核访问主存的方式有两种：一种是通过全局读入（gload）和写出（gstore）指令实现内存-寄存器的数据传输，这种方式粒度较细，更加灵活，但带宽只能达到1.5 GB/s；另一种是通过直接内存访问（direct memory access, DMA）实现内存-LDM的数据传输，再通过访问LDM来获取数据。DMA是一种粗粒度的访存模式，根据StreamTriad测试[4]，64个从核同时通过DMA访存可以获得22.6 GB/s的带宽。

SW26010另一个独特的设计是从核阵列上的寄存器通信技术。根据StreamTriad测试[4]，寄存器的通信时延仅11个指令周期，集合带宽超过600 GB/s。在8×8的网格阵列中，同一行或同一列的从核可以高速互传数据。每个从核都有一个发送缓冲区、一个行接收缓冲区和一个列接收缓冲区。在寄存器通信中，硬件会将发送缓冲区内的数据放到目标从核的行/列接收缓冲区中。这个过程以阻塞方式自动进行，直到发送缓冲区为空或者接收缓冲区已满。

3.2 异构众核架构的挑战和开发技巧

异构众核架构拥有与常规并行程序开发不同的编程和优化模式。这种不同为大规模并行程序带来巨大性能潜能的同时，也给程序开发者提出了更多的要求。“神威·太湖之光”把并行度推进到千万核级别，因而也对数值型应用和优化方法的可扩展性提出了挑战[5]。因此，在开发过程中应当注意以下几个方面。

（1）充分发挥从核运算性能

SW26010每个核组内的从核可以使用SunwayOpenACC或Athread实现并行执行。根据性能指标计算，SW26010上的主核浮点性能约为23.2 GFlops，而从核浮点性能达到了742.4 GFlops。由于这种浮点性能上的巨大差距，要提升计算密集型程序的运行效率，就需要尽可能充分地发挥从核的运算性能，充分发掘应用内部的并行性。

由于从核的数目和物理拓扑关系相对固定，以及从核LDM大小和内存带宽的限制，应用内部的子问题划分需要具有足够的局部性，同时也要考虑从核阵列的排布特点。非同行/列的从核间无法直接进行寄存器通信，可能需要其他从核参与，这会显著增加从核间寄存器通信的开销，因此最好将相邻的任务分配到相同的行/列上。由于每两行从核之间共享一条内存总线，要想提升内存带宽，就需要充分利用4条内存总线，将内存访问均匀地分配到每条总线上。对于一些计算和访存不规则的应用，简单的分块方法可能造成从核间负载不均衡，因而无法完全发挥处理器的性能。这些都对并行问题的划分提出了较高的要求。

（2）充分利用LDM，减轻主存压力

SW26010主存和局部存储器的访问

性能差异尤其明显，从核进行离散化访存的开销是高昂的，全局离散存/取（gload/gstore）指令需要超过200个时钟周期，而访问局部存储器LDM仅需4个时钟周期。因此，要提升并行程序的运行效率，就要充分利用LDM局部存储器，减少全局内存访问，设计好的缓存策略。SW26010的独特架构将缓存策略的设计交给开发者，这一做法更加增加了这一问题的重要性和难度。SW26010的从核LDM大小为64 KB，显然无法满足所有应用对局部数据的需求。在一些程序中，频繁的主存-LDM交换是可能存在的，而减少交换次数、提高交换效率是开发者需要考虑和实现的。合理的LDM管理策略是提高申威架构下程序运行效率的关键点之一。

从核DMA的带宽高达22.6 GB/s，而gload/gstore指令的带宽只有不到1.5 GB/s。另外，DMA可以在数据传输过程中解放CPU，实现计算-访存重叠模式，缩短时延。因此，连续化、聚合化的DMA访存可以有效提升访存效率，而部分应用的不规则访存模式增大了使用DMA的难度。

值得注意的是，申威架构的DMA效率在特定情况下可达到峰值。对于随机化访存，DMA操作性能会在256 B及以上的粒度下达到峰值[6]。另外，由于DMA是以128 B大小的块为单位进行访问的，因此数据需要按照128 B对齐，以充分发挥其性能[7]。稀疏计算型应用常常涉及对主存中多个数组的离散化、细粒度的访问，这种访问模式很难充分发挥DMA操作的性能，因此需要对数据布局进行调整。对于多个具有相似的访问模式的数组，可以将其合并，即将包含多个数组的结构体（structure of arrays，SOA）转化为一个大的包含多个元素的结构体的数组（arrays of structures，AOS）。如果合并后的结构体不满足内存对界要求，可以适当地加入空位（padding）进行填补。

对于难以合并的、访问模式独立的数组，可以对数组的数据分布进行调整。比如，在地震模拟应用中[8]，在主存中开辟出额外的存储空间，用于存储每个进程需要访问的划分后的包含halo区的数组部分，这样可以保证DMA操作的连续性，减少内存访问操作的频率。

数据结构的调整可以有效解决稀疏问题的细粒度访存问题，提升稀疏型应用在申威架构下的访存带宽。

（3）充分运用从核间通信

从核间高效的寄存器通信接口为数据通信和共享提供了有效的方法。寄存器通信的时延为7~11个时钟周期，远小于DMA（超过25个时钟周期）和全局存取（超过600个时钟周期）的开销。因此，一个通用的方法是将从核LDM中或寄存器中的数据通过寄存器通信发送给其他从核，实现数据共享，减少对全局内存的访问频率。

寄存器通信的编程模式给开发者带来了挑战。由于实际应用中可能存在复杂的核间通信和同步关系，阻塞式的通信接口会显著增加程序设计的难度，开发者需要谨慎考虑核间数据传输关系，排除死锁的可能性。由于从核接收缓冲区大小有限，如果发送从核（即发送数据的从核）传送的数据规模较大，则需要保证目标从核在自身阻塞前能够完成接收，否则可能出现级联阻塞现象。另外，在多发送者-单接收者的模式下，可能会存在数据乱序的问题，需要额外考虑程序的正确性。这些都给程序设计和优化带来了很大难度。

（4）SIMD向量化的使用

单指令多数据流（single instruction multiple data，SIMD）是SW26010的一个扩展的功能模块。SW26010提供了256位的寄存器，每个寄存器可以存放8个整型数或4个浮点数。使用这些寄存器进行向

量化运算，可以达到一条指令得到多个结果的效果。SIMD从源操作数的数组空间将数据装载到256位SIMD寄存器，并通过SIMD运算指令完成计算，最后将结果存储到目标操作数的数组空间。SIMD不仅降低了功耗，而且显著提高了性能，定点和浮点的理论峰值性能为单部件的8倍或4倍。循环展开（loop unwinding）作为一种牺牲程序的尺寸来加快程序的执行速度的优化方法，可以由程序员完成，也可由编译器自动优化完成。对于拥有多个计算部件的SW26010，SIMD可以被看作一种指令形式的循环展开，SIMD向量化寄存器为多个运算器提供了指令级并行。SW26010编译器提供了简洁的SIMD编程指令来显式地开发指令级并行，开发者不再需要对代码进行手动展开或依赖编译器的自动优化。

向量化为申威架构下的程序提供了巨大的性能机遇，但其实际应用存在一些困难。向量化适用于连续型数据访问和运算，对于非连续型（如AOS类型）数据，其装载和存储过程带来的开销可能超过计算优化本身带来的收益。因此，开发者应当注意SIMD使用的可行性，要合理使用向量化，需要时可对数据排布进行调整。

例如，分子动力学应用[6]需要按前文所述的方法将数据转化为AOS形式的粒子数据包，以最大限度地提升DMA性能。但这种AOS形式的数据并不适合向量化。为此，对于局部获取的数据，需要进行类似矩阵转置的转化，使得相同数组的元素在存储空间中连续，如**图3**所示，将一个粒子包内的数据转化为每种元素连续的形式，这样可以用向量寄存器存储，并开展计算。这种转换操作可以使用SW26010支持的指令（如simd_vshulff）高效地完成。参考文献[6]中的一个粒子数据包包含4个粒子的数据，转换完成后的数据刚好按照4个浮点数对齐，放在一个4浮点数向量寄存器内。

4 大规模稀疏计算问题的性能优化实践

4.1 高分辨率大气模拟中的隐式求解

大规模大气动力模拟对于天气预报和预测气象灾害有重大意义，该领域的应用往往涉及对大规模网格的计算和求解。此前，国内相关研究实现了基于CPU-GPU和CPU-MIC加速的显式时步全球浅水波（shallow water）模式，它们分别在天河-1A和天河2号上取得了800 TFlops[9]和1.63 PFlops[10]的性能，扩展到半系统级别。此后，以上工作被扩展到3-D非静力模式，在天河2号上取得8%的峰值浮点运算效率[11]。然而，这些工作只关注了显式求解过程，在高分辨率的大气模拟中，传统的大气动力学方程显式求解方法面临计算步长过小的问题，因此隐式求解成为可能的解决方法。但隐式求解方法又面临收敛性和稀疏线性方程组求解低效的问题，如何在隐式求解算法上开发千万核并行是待解决的问题。

三维非静力大气模拟过程主要涉及对完全可压缩欧拉方程的求解。在超大规模方程组求解中如何保证鲁棒性较强的收敛率是一个问题，为此，浅层区域分解多重网格（domain decomposition-multigrid，DD-MG）算法[12]被提出。**图4**展示了一个3层的DD-MG算法，在每个k-cycle的MG层级，一层RAS方法被作为区域分解的预条件，从而在处理器层级最大限度地开发并行性。DD-MG算法保证了求解过程的收敛性，同时，作为一种粗粒度的并行，其保证了核组间的负载均衡。

大规模隐式方程求解的性能取决于局部求解的性能，为此，参考文献[12]提出并实现了高局部性、细粒度和无同步的

本地求解器。对于指定的重叠子区域，基于低秩的7点空间偏导构建近似的雅可比矩阵，并在每个网格点中对未知数进行排序。该过程不破坏原有矩阵物理成分的联系。在DD-MG的框架下，可以用不完全LU（incomplete LU，ILU）分解方法对子区域开展求解。传统的LU分解由于矩阵非零元的相互依赖和可能的不规则分布，很难有效通过并行算法进行求解。为此，在适用于众核架构的并行ILU（parallel incomplete LU，PILU）方法[13]的基础上进行改进，几何流水化ILU（geometry-based pipelined incomplete LU，GP-ILU）算法被提出，这种方法在保持数据依赖关系的基础上很大程度地开发了片上并行性。

在整体算法实现上，参考文献[12]在处理器、线程及指令层级上开展了不同程度的优化，在隐式求解器的关键运算核心上取得了有效的性能提升。

考虑到SW26010的特性，参考文献[12]针对不同计算核心提出了3种不同的划分策略，如**图5**所示。这里假设主存内的三维AOS数据按照$z-x-y$的维度顺序存储，core(i, j)表示处理器阵列中第i行第j列的从核。右端相关运算核心中，相应的模板计算有13个依赖点，整个求解区域被分为内部区域（inner）和halo区，halo区是不同节点计算区域的邻接部分，由顶部、底部和东西南北6个面组成，这些部分都涉及数据通信。不需要通信的内部区域采用2.5D分块与双缓冲策略结合的方法，如**图5(a)**所示，分块大小由LDM大小、向量化程度、双缓冲占用率和DMA效率综合考虑决定，最终采用4×4的大小。MAT运算核心没有halo区，因此沿轴按“柱”方向进行1D分块，如**图5(b)**所示。这里的分块大小应当是4的倍数，以方便向量化。ILU核心实现了线程间和线程内部的并行，分块方式如

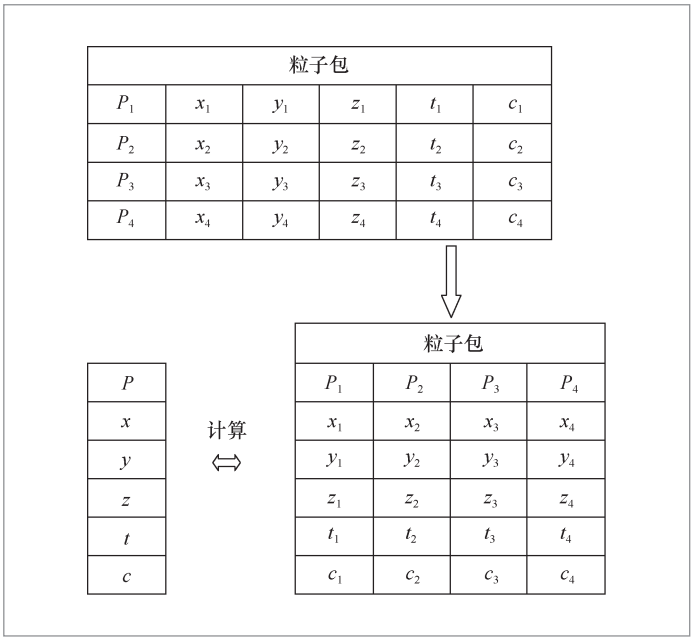


图3 分子动力学中的数据布局变换[6]

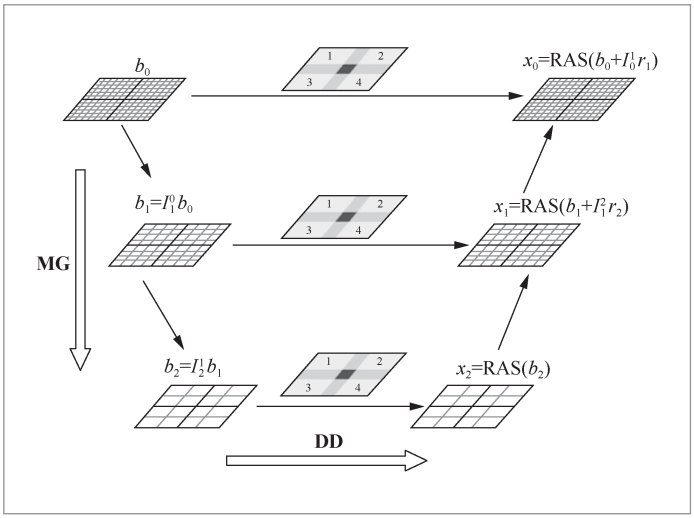


图4 DD-MG算法示意图[12]

图5(c)所示。在xy平面上，分块把整个求解区域划分成8×8的子区域，每个子区域中沿z轴的一“柱”刚好对应8×8=64个SW26010处理器众核。在这种粒度的划分下，求解流水线开始/结束时从核间的负载不均衡可以被最小化，水平和竖直方向上的两层流水线可以高效地工作。类似地，

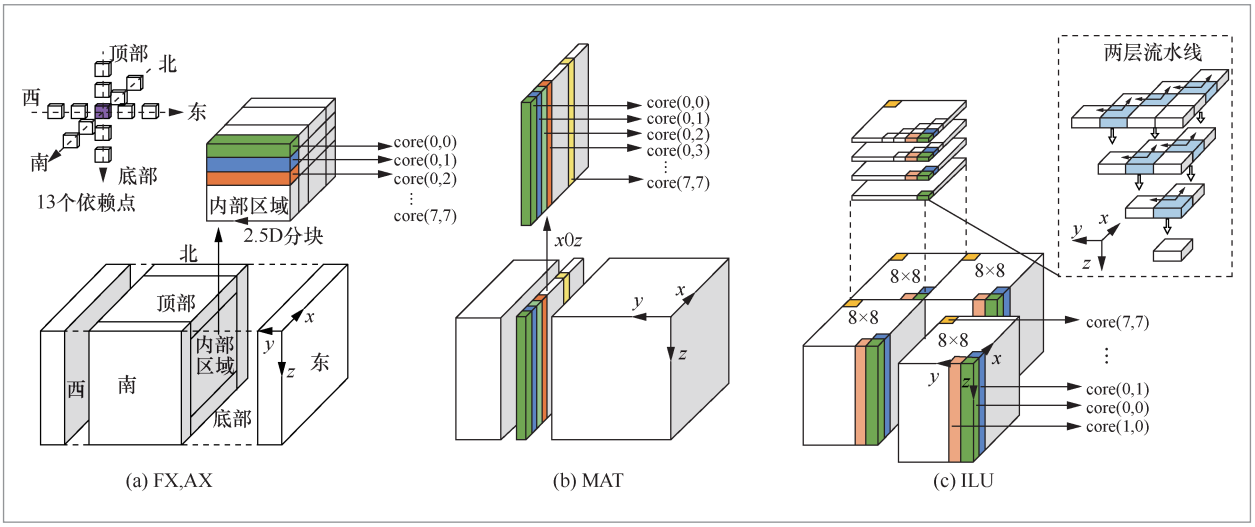


图 5 针对不同运算核心的数据划分策略[12]

前代/回代过程（下三角/上三角矩阵求解）采取类似的划分方法。

在2.5D分块中，每个从核对内存的访问存在一定间隔，导致内存带宽的不充分利用。一种利用寄存器通信的在线数据共享方法可以有效解决该问题。如图6所示，该方法将4个从核分为一组，通过3个步骤完成数据共享，在第一步分解操作中，对于求解的内部区域，组内的从核从内存读入计算区域和两层halo区，共4×4+2×2=20个元素的数据；在第二步复制操作中，每个核上对应的数据区域被扩展，开辟冗余的halo区，从而形成4×(4+2×2)=32个元素区域；在第三步交换操作中通过快速的寄存器通信在从核间传递计算所需数据，这种数据交换不涉及LDM与内存的数据传输，减轻了内存带宽的负担。图6中的4个从核通过3个步骤完成数据交换，每一步之后都需要进行同步。一般来说，增加每组包含的从核个数可以显著地提升数据重用效率，但相应的同步开销会增大。实验表明，4个从核分为一组最好地平衡了两者。

为了更好地实现向量化，参考文献[12]中实现了高效的AOS和SOA转换接口。这里使用SW26010的shuffle指令，可以在十几个时钟周期内将结构内的AOS数据装载到256位向量寄存器中。另外，部分数据操作（如BLAS-1向量更新和halo区交换）需要在SW26010上得到实现和优化。基于申威架构的xMath数学运算加速库是为了高性能数学运算开发的，提供了BLAS、LAPACK和FFT操作接口。调用该库并添加一些手动优化，可以在BLAS-1向量操作上取得20倍以上的加速。

以上提及的完全隐式方程求解器应用已被成功地扩展到整个“神威·太湖之光”超级计算机的超过100万个的异构众核上，在双精度求解下性能达到了7.95 PFlops。实验中，在488 m水平分辨率（超过7 700亿个非零元）条件下，该应用依然能够实现快速而精确的大气模拟，成为世界上较大规模的完全隐式模拟之一。

4.2 非线性大地震模拟中的显式求解

我国是受地震灾害影响严重的国家，

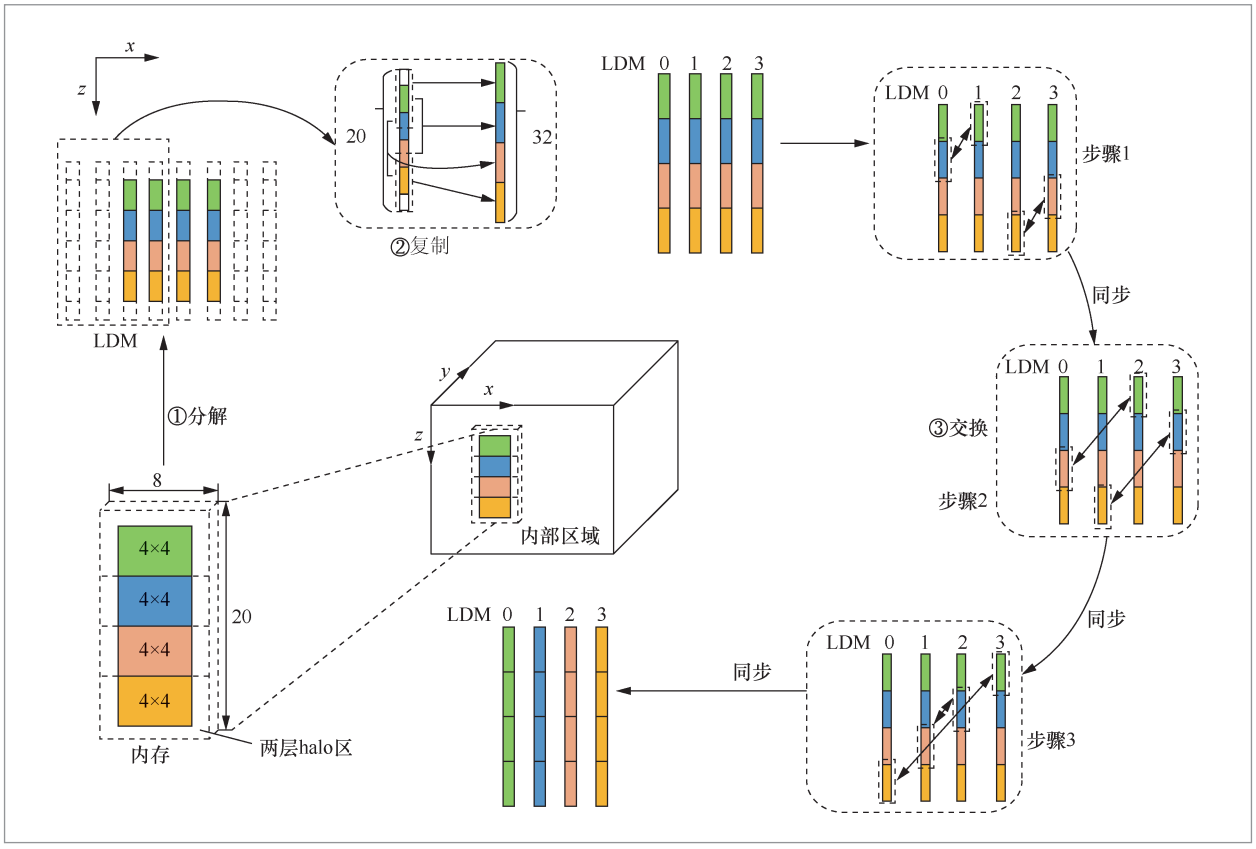


图 6 在线数据共享方法过程 [12]

分布有23条地震带，7度以上的高烈度区域约占国土面积的50%。对地震的模拟和预测可以有效减少地震灾害带来的损失。很多与地震模拟相关的应用已经开始在大规模并行计算机系统上寻找答案，以开源软件AWP-ODC（anelastic wave propagation by Olsen, Day and Cui）为例，该软件自2008年起开始推进在千兆级计算机系统上的应用[14]，2016年该应用完成了对非线性效应模拟的支持，并在“泰坦”超级计算机上取得1.6 PFlops的性能，扩展到半系统级别[15]。

由于计算过程中每个网格点都需要对超过20个变量进行读写，传统的分块策略并不适用。为此，参考文献[16]提出了一种自定义的多级计算区域分解策略（如**图7**所示），包括MPI分解、CG核组分块划分和CPE从核分块划分。在LDM空间利用策略的计算过程中，计算区域和完成区域不断向前推进，未完成区域逐渐缩小，缓冲区域用来存储计算所需的邻接区。这里假设主存内的三维AOS数据按照$z-x-y$的维度顺序存储，该方法先沿着xy平面对求解区域进行2D划分，并分配到每个MPI进程中。这是由于在该应用中竖直（z轴）方向的长度要远小于水平（x轴和y轴）方向的长度，这种划分方法可以有效减少MPI进程间的通信量。第二层沿着zy平面进行块划分，并将该层的每个块分配给一个核组。最终，第三层依然沿着zy平面把核组中的块划分成数个不同的区域，将每个区域分配给一个SW26010处理器从核，每个从核线

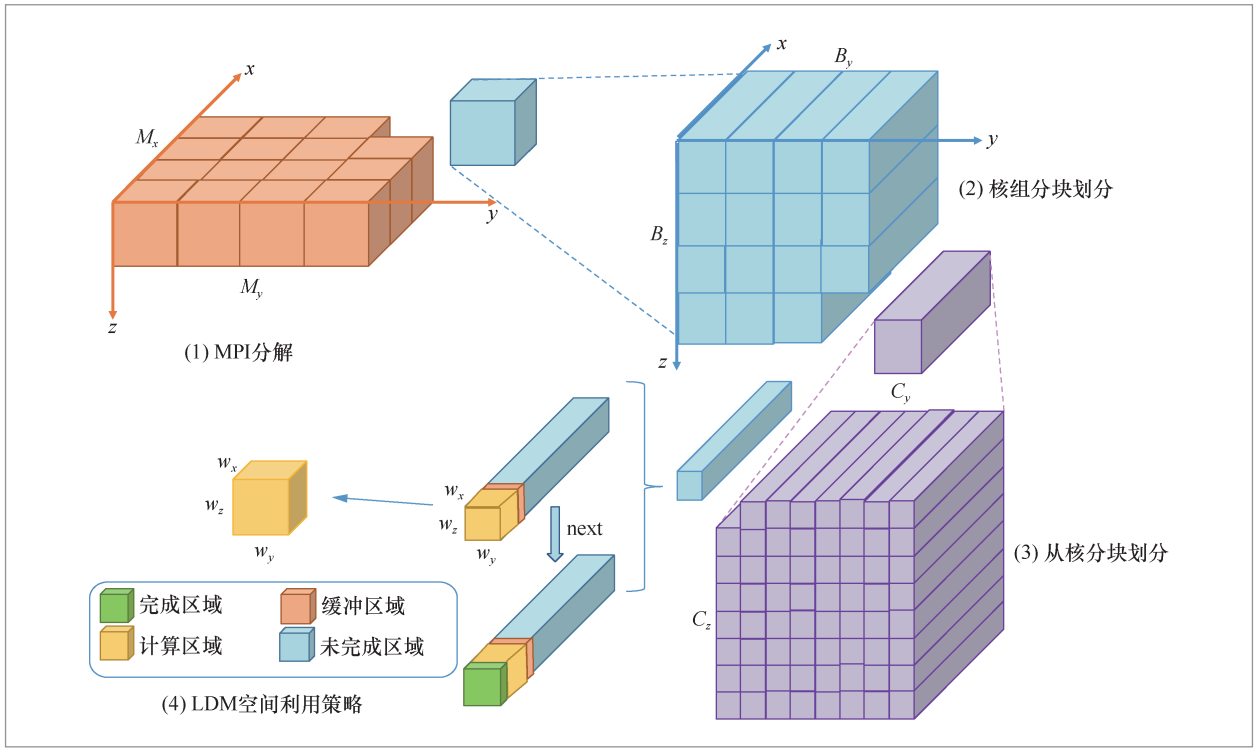


图 7 非线性大地震模拟中的多级区域划分策略[16]

程沿着x轴方向开展迭代，从而确保快速访问内存。考虑到每个LDM空间大小有限，每个从核一次通过DMA载入适当大小的计算区域，包括内部计算区域和halo区。随着计算的进行，从核缓存区域会沿x轴方向向后推进。DMA被设置为异步的，以达到与计算重叠的效果。这里各层级的分块大小可以根据问题规模、LDM空间大小及单个网格点的变量数目等因素动态计算得出，实际应用中采取计算分析得到的最优值。

为了在同样的内存带宽和存储空间大小的限制条件下取得更高的性能，参考文献[16]还提出一种有损压缩策略，有效解决了在线压缩和解压缩开销与整体应用有效性能提升的矛盾，也有效保证了应用的工程计算精度。如图8所示，每个参与计算的从核（CPE）先从主核内存，也就是主存中通过DMA读操作（dma_get）将压缩后的16位数据读入LDM，并解压缩为32位数据，然后进行32位数据上的实际计算，并将

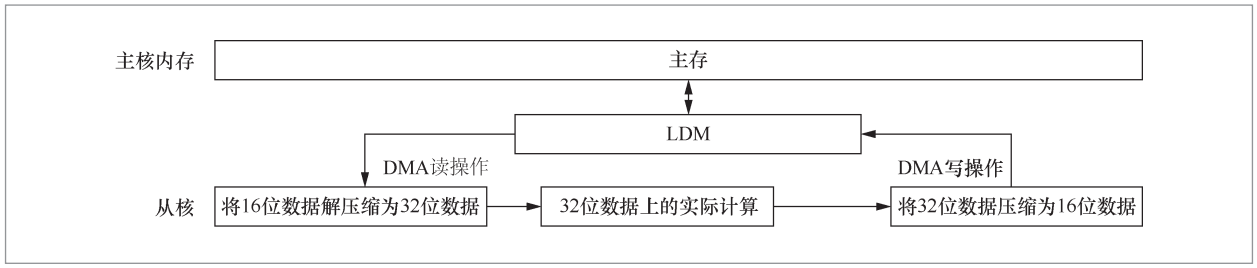


图 8 有损压缩工作流程[16]

计算结果重新由32位压缩为16位的数据，通过DMA写操作（dma_put）存入内存。

图9展示了3种不同的有损压缩方法，其中用sign exp表示指数，frac表示尾数。计算所用数据在压缩前固定为32位浮点数，压缩后的16位数据可以采取不同的表示方法。方法1进行IEEE 754标准32位到16位浮点数的转化，直接将压缩后的数据定义为IEEE 754标准的半精度浮点数，包含固定的5位指数和10位尾数。编译器内置的对半精度浮点数的支持使得压缩前后的数据转换效率很高，但由于指数位数少，数值分布范围较大的变量可能出现溢出，进而引入数值精度问题。而对于数值分布范围很小的变量而言，5位的指数可能是一种浪费。针对这一问题，方法2使用动态方法定义指数位数。对于每个参与计算的变量，计算其一定范围内的数值范围分布，并根据范围动态分配不同变量压缩后的指数位数，在保证能覆盖大范围指数分布的同时，也能为小范围数值分布的变量保留更多的尾数位数。但这一方法的转换效率和计算效率较低。方法3被用于模拟程序速度和压力变量的压缩，它将数组中的元素规格化到1和2之间，并采用16位定点小数的表示方法。这种方法平衡了性能和精度，因此在实际应用中具有最好的效果。在地震波传播核心部分采用有损压缩策略，最终能取得约24%的性能提升。

该地震模拟应用经过以上优化，可以在“神威·太湖之光”超级计算机上达到超过15%的系统峰值性能，超过了类似应用在“泰坦”超级计算机上的表现（11.8%），且其具有强可扩展性，几乎可以线性扩展到全机上千万核。在18 Hz、8 m分辨率的超大规模地震模拟中，该应用可以达到18.9 PLlops的持续性能。

4.3 “神图”图计算框架

图是数值科学领域应用频繁的概念之一，随着大数据处理问题规模的增大，图数据结构的大小也相应增大，需要高效可扩展的图处理系统来解决图计算问题。比如，人类基因研究目前需要对拥有超过50亿个点/边的布鲁因图（de Bruijn graph）进

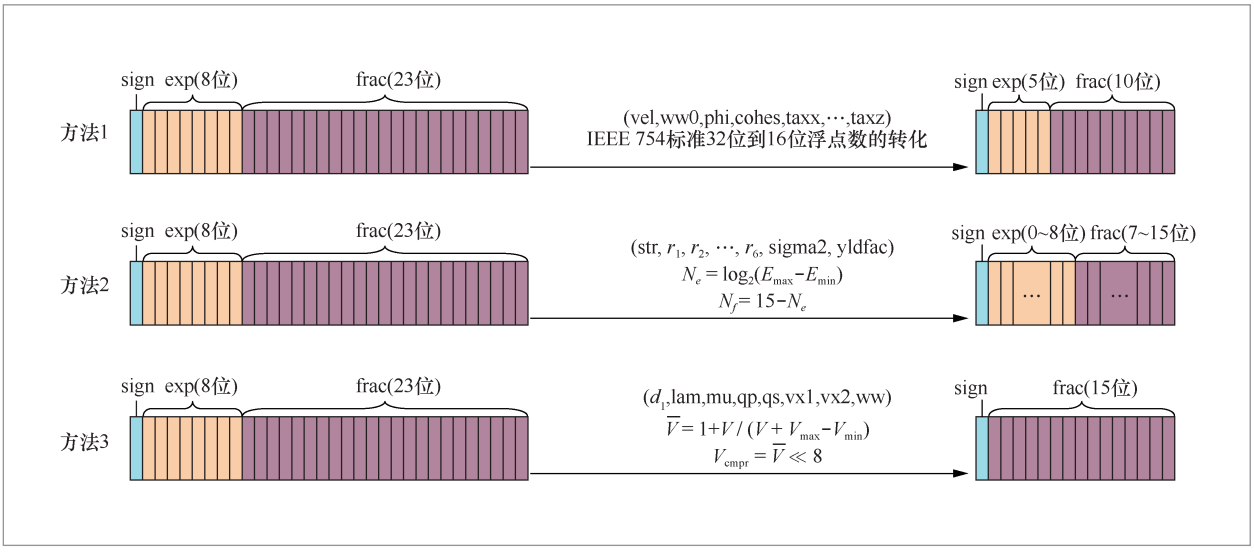


图 9 3 种不同的有损压缩方法[16]

行处理[17]，类似地，人脑建模分析要考虑超过1 000亿个神经元以及每个神经元的平均7 000个突触连接[18]。图计算是典型的大数据稀疏处理类问题，浮点运算少，访存随机性大，对数据存储和管理提出了很高要求。同时，幂律分布造成通信和计算负载不均衡，对于复杂图而言，计算过程中存在大量的核间和节点间通信，通信次数多，通信量少，非常低效，给系统的效率和可扩展性提出了巨大挑战。

“神图”[19]是首个运用千兆级系统解决百万规模图处理问题的通用框架。

针对申威处理器的异构特性，“神图”在不同层级对硬件功能进行划分。在粗粒度的层级上，每4个核组被分为一个节点，分别具备4种不同的功能：一是生成，读入当前组分配的节点数据，识别待处理图中的“活跃”点，并生成通信消息；二是转发，路由聚合后的消息，提供高吞吐率的组间通信；三是粗排序，实行第一阶段的初步桶排序，每个桶可以适应性地放入从核LDM中，为下一阶段做准备；四是更新，图处理过程的最后一步，对每个桶进行排序，并更新目标节点。

“神图”引入了超节点和处理器上的两级路由机制与高效的专用数据排序策略。超节点路由方法解决了小型消息过多及通信节点对过多带来的通信开销问题。

“神图”将数个节点划分为一个组，数个组属于一个超级节点。每个节点将属于相同目标组的通信消息聚合为一条，发送给相应组内的一个节点。该节点中负责转发的核组会将消息解包并发送给其他核组。**图10**展示了超节点多级路由的工作过程，超节点中的一组包含4个核组，超节点X中A节点作为生成节点，发送消息给超节点Y中的排序和更新节点C，中途通过超节点Y中的转发节点B进行转发[20]。

大部分图计算应用受限于内存带宽，细粒度的随机内存访问会对性能造成影响。为此，“神图”提出了一种片上排序的方法。**图10**中节点C可能会按随机顺序接收图中节点更新的消息，片上排序把更新消息的不同目标点进行划分和排序，同时合并对相同目标点进行更新的消息，显著减少了内存总线负载和同步开销。如**图11**所示，每个用于排序的核组中的众核又被分为3类：p为消费者，负责读入数据；r为路由者，负责传递数据；c为消费者，负责使用数据进行计算，剩下的从核被用于其他任务。初始输入是无序输入，经过两步片上洗牌操作，数据变为有序，可开展后续处理。核组3完成第一阶段的初步桶排序操作，把数据放在不同的桶中，使数据

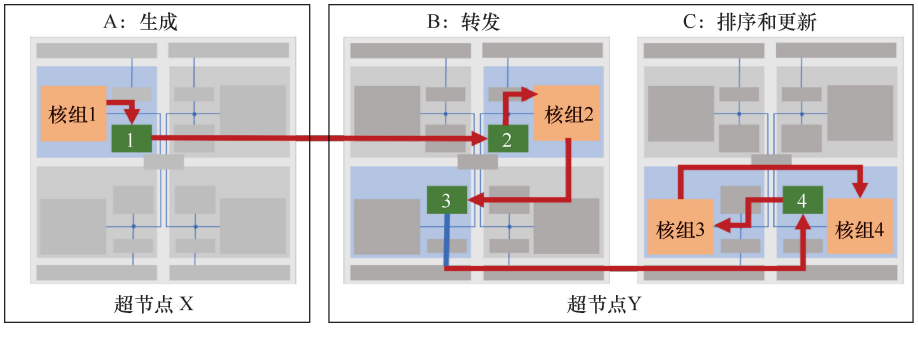


图 10 “神图”的超节点多级路由[19]

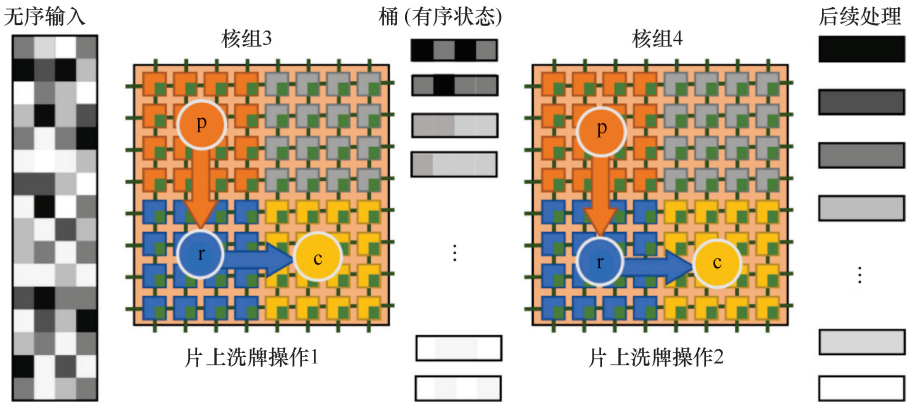


图 11 片上排序过程[19]

成为半有序状态，核组4利用其结果完成对整个数据的排序[20]。

在真实应用的图中，点的入度/出度往往呈现指数级增长。在分布式图处理系统中，度数高的点会产生大量的数据通信，涉及系统中的大部分节点，给通信网络带来巨大负载。“神图”将这种度数高的点复制到每一个计算节点中，原本负责该点的计算节点存储的是原件，其他计算节点存储的是镜像。高出度的点需要向外发送的消息很多，为了避免大规模地发送更新消息，每一个计算节点通过简单的MPI_Bcast接口协作更新所有镜像，再根据镜像来更新其对本地点的影响。对于高入度的点，“神图”采用类似的方法，计算节点先对本地镜像进行更新，最后使用MPI_Gather或MPI_Reduce接口更新原件。这种与节点度相关的通信优化模式显著减少了通信量，减轻了并行系统互联网络的压力；同时，镜像的存在将高度数点的处理工作平均分配给每一个计算节点，均衡了系统负载。

“神图”图计算框架可在分钟级完成对搜狗中文网页图的处理，每次迭代仅需8.5 s，解决了过去由于机器规模和计算框架限制而无法解决的问题。

5 结束语

目前超算发展进入E级阶段，新的超大规模异构并行计算机在解决富有挑战性的计算问题方面的潜力是值得期待的。异构众核并行系统的设计已经成为高端超算系统的重要构建方式。但其给大规模稀疏处理问题带来了挑战。稀疏问题具有非规则的计算与访存特征，对并行应用的存储管理、负载均衡、数据通信等提出了更高的要求，需要开发者依据软硬件特点开展设计和优化，兼顾性能、成本、功耗等多方面的约束。异构众核系统的架构设计具有巨大的性能潜力，但也给应用实现和优化带来更高的难度。本文总结了基于“神威·太湖之光”超级计算机的大规模隐式/显式求解器和“神图”图计算框架的性能优化经验，涵盖任务划分、存储访问、数据压缩、数据共享与通信等多方面，为新一代异构众核计算系统的稀疏问题求解提供了借鉴。实际上，基于异构众核架构的大规模计算问题的求解和优化案例还有很多，在应用和算法设计层面，动态稀疏问题的高效求解算法设计依然是急需解决的问

题。同时，许多实际科学与工程问题中的大规模应用性能优化方法还期待着更多的开发者投入研究。

参考文献:

[1] CHEN Y M, LI F, SHU J W. Building storage systems in big data era: challenges, methods and trends[J]. Big Data Research, 2019, 5(4): 27–40.

[2] WANG X Y, LIAO X F, LIU H K, et al. Big data oriented hybrid memory systems[J]. Big Data Research, 2018, 4(4): 15–34.

[3] LI X, CHEN X, HUANG Z Q. Analysis on hybrid memory architecture for big data application[J]. Big Data Research, 2018, 4(3): 61–80.

[4] XU Z G, LIN J, MATSUOKA S. Benchmarking SW26010 many-core processor[C]// 2017 IEEE International Parallel & Distributed Processing Symposium Workshops. Piscataway: IEEE Press, 2017.

[5] FU H H, LIAO J F, YANG J Z, et al. The Sunway TaihuLight supercomputer: system and applications[J]. Science China Information Sciences, 2016, 59(7): 1–16.

[6] ZHANG T J, GAN L, FU H H, et al. SW_GROMACS: accelerate GROMACS on Sunway TaihuLight[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. New York: ACM Press, 2019: 1–14.

[7] DUAN X H, GAO P, ZHANG T J, et al. Redesigning LAMMPS for peta-scale and hundred-billion-atom simulation on Sunway TaihuLight[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2018.

[8] CHEN B W, FU H H, WEI Y W, et al. Simulating the Wenchuan earthquake with accurate surface topography on Sunway TaihuLight[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2018.

[9] YANG C, XUE W, FU H H, et al. A peta-scalable CPU-GPU algorithm for global atmospheric simulations[C]// The 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. New York: ACM Press, 2013.

[10] XUE W, YANG C, FU H H, et al. Enabling and scaling a global shallow-water atmospheric model on Tianhe-2[C]// The 28th IEEE International Parallel & Distributed Processing Symposium. Piscataway: IEEE Press, 2014: 745–754.

[11] XUE W, YANG C, FU H H, et al. Ultra-scalable CPU-MIC acceleration of mesoscale atmospheric modeling on Tianhe-2[J]. IEEE Transactions on Computers, 2015, 64(8): 2382–2393.

[12] YANG C, XUE W, FU H H, et al. 10m-core Scalable fully-implicit solver for nonhydrostatic atmospheric dynamics[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2016.

[13] CHOW E, PATEL A. Fine-grained parallel incomplete LU factorization[J]. Siam Journal on Scientific Computing, 2015, 37(2): 169–193.

[14] BURSTEDDE C, GHATTAS O, GURNIS M, et al. Scalable adaptive mantle convection simulation on petascale supercomputers[C]// The 2008 ACM/IEEE Conference on Supercomputing. New York: ACM Press, 2008: 1–15.

[15] ROTEN D, CUI Y F, OLSEN K B, et al. High-frequency nonlinear earthquake simulations on petascale heterogeneous supercomputers[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2016: 1–12.

[16] FU H H, YIN W W, YANG G W, et al. 18.9 PFlops nonlinear earthquake simulation on Sunway TaihuLight:

enabling depiction of 18 Hz and 8 meter scenarios[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. New York: ACM Press, 2017: 1–12.

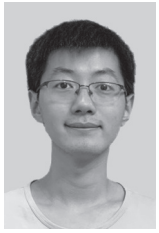
[17] MUSTAFA H, SCHILKEN I, KARASIKOV M, et al. Dynamic compression schemes for graph coloring[J]. Bioinformatics, 2018, 35(3): 3.

[18] PAKKENBERG B, GUNDERSEN H J G. Total number of neurons and glial cells in human brain nuclei estimated by disector and fractionator[J]. Journal of Microscopy, 1988, 150(Pt 1): 1–20.

[19] LIN H, ZHU X W, YU B W, et al. ShenTu: processing multi-trillion edge graphs on millions of cores in seconds[C]// International Conference for High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2018.

[20] SHUN J, BLELLOCH G E. Ligra: a lightweight graph processing framework for shared memory[J]. ACM SIGPLAN Notices, 2013, 48(8): 135–146.

作者简介



胡正丁（1997– ），男，清华大学计算机科学与技术系硕士生，主要研究方向为高性能计算。

薛巍（1974– ），男，博士，清华大学计算机科学与技术系副教授，高性能计算研究所所长，中国计算机学会高级会员，主要研究方向为大规模科学计算、量化不确定分析。

收稿日期： 2020-05-09
基金项目： 国家电网公司科技项目（No. XT71-19-022）
Foundation Item: Science and Technology Project of State Grid Corporation of China(No. XT71-19-022)