引用格式:傅游,王坦,郭强,等. "神威·太湖之光"上 Tend_lin 并行优化[J]. 山东科技大学学报(自然科学版),2019,38(2):90-99.

FU You, WANG Tan, GUO Qiang, et al. Parallelization and optimization of Tend_lin on Sunway TaihuLight system[J]. Journal of Shandong University of Science and Technology (Natural Science), 2019, 38(2): 90-99.

"神威·太湖之光"上 Tend_lin 并行优化

傅 游[1],王 坦[1],郭 强[1,2],高希然[1,3]

(1. 山东科技大学 计算机科学与工程学院,山东 青岛 266590;
2. 山东省计算中心(国家超级计算济南中心),山东 济南 250101
3. 中国科学院计算技术研究所 计算机体系结构国家重点实验室,北京 100190)

摘 要:大气环流模式是中科院地球系统模式中最为复杂的模式,在当前主流的众核异构平台上开展大气环流模式的众核并行化是高性能计算的热点研究问题。针对 AGCM4.0 热点程序动力框架的适应过程 Tend_lin,利用神威 OpenACC 编程模型在"神威·太湖之光"高性能计算平台上实现并行化,并从循环分布、循环分块、数据传输的表达、函数调用的从核化等方面提升应用性能。详细讨论了不同场景下的数据传输表达,对比测试了不同分块尺寸对程序性能的影响。相比主核串行,两种测试规模下,Tend_lin 应用的单核组多线程并行均获得 6 倍以上的加速;且随着应用分辨率的扩大,众核处理器的性能得到更好发挥,在 C 规模下,多进程获得了 69 倍的全应用加速。

关键词:神威·太湖之光;Tend_lin;神威 OpenACC;众核并行;优化

中图分类号:P595 **文献标志码**:A **文章编号**:1672-3767(2019)02-0090-10

DOI: 10.16452/j.cnki.sdkjzk.2019.02.011

Parallelization and optimization of Tend_lin on Sunway TaihuLight system

FU You[1], WANG Tan[1], GUO Qiang[1,2], GAO Xiran[1,3]

(1. College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China; 2. National Supercomputer Center in Jinan, Jinan, Shandong 250101, China; 3. State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Science, Beijing 100190, China)

Abstract: Atmospheric general circulation model (AGCM) is the most complex model of the Chinese Academy of Sciences' Earth System Model (CAS-ESM) and the many-core parallelization of AGCM on the leading many-core heterogeneous high performance computing (HPC) platform is one of the hotspots in HPC area. In this paper, Tend_lin, the adaptive process of AGCM 4.0 hotspot program, was parallelized on Sunway platform by using OpenACC programming model. Its performance was improved from the aspects of loop distribution, loop tiling, expression of data transfer, and function call. The data transmission expressions under different scenarios were discussed in detail and the effects of different block sizes on program performance were tested. Compared with the master-core serial application, the many-core parallel application of Tend_lin was accelerated more than 6 times in the single core group. With the increase of application resolution, the performance of the many-core processor got better performance. In the C scale, the acceleration ratios of the multi-process application was up to 69.

Key words: Sunway TaihuLight System; Tend_lin; Sunway OpenAcc; many-core parallel; optimization

收稿日期:2018-09-09

作者简介:傅 游(1968—),女,山东茌平人,教授,博士生导师,主要从事高性能计算、并行编译等领域研究.
高希然(1993—),男,山东聊城人,硕士,主要从事高性能计算及并行编译等领域的研究,本文通信作者.
E-mail:gaoxirqn@ict.ac.cn

地球系统模式已成为研究和预测全球变化的有力工具,也是当前大气科学领域的研究热点之一。中科院大气物理所研发的大气环流模式(atmospheric general circulation models,AGCM),利用偏微分方程求解物理问题,是典型的高性能应用,是整个中科院地球系统模式 CAS-ESM(earth system model)中最复杂的部分。其动力框架基于准一致网格[1]实现,在分辨率提高的同时,计算量增大,进程间通信量也随之增加。如何提升较高分辨率下的 AGCM 模式性能,已成为重要的研究课题。

“神威·太湖之光”是由我国自主研发的高性能异构众核处理器“申威(SW)26010”构建的超级计算机系统,其峰值性能超过 100Pflops,曾 4 次蝉联世界超级计算机 Top500 榜首,目前在世界超算平台排名第 3。申威众核处理器支持众核加速编程模型(神威 OpenACC,简称 SWACC)和加速线程库(Athread 库)两种并行编程模型。徐金秀等[2]在神威平台上开展耦合算法的多级并行优化,并使用 SWACC 实现耦合算法的众核并行;李亿渊等[3]基于神威平台,利用 Athread 在线程级层面,对稀疏矩阵向量乘法进行并行算法设计和优化实现。众核异构平台相比传统的高性能硬件平台能提供更强大的计算能力和访存带宽[4],在国内当前主流的异构众核高性能平台上对 AGCM 进行众核并行化,成为目前的重要研究方向之一。

本研究面向国产众核异构平台“神威·太湖之光”,以中科院大气物理研究所研发的 IAP AGCM4.0 为基础,对其热点程序动力框架的适应过程构造 Tend_lin 应用,利用神威 OpenACC 编程模型实现 Tend_lin 应用的众核并行化。通过循环分布、循环分块并行将计算任务分配到计算核心,以高效 DMA 的方式将计算需要的数据批量传输到从核局存,消除计算过程中低效的全局离散访存,实现了 Tend_lin 应用在国产异构众核平台上的 OpenACC 并行化。

1　神威异构众核处理器体系结构

“神威·太湖之光”整机由 40 960 块 SW26010 处理器组成,并通过计算插件板、计算超节点和计算机仓等模式进行系统扩展。“申威(SW)26010”为主从核结构的异构众核处理器,集成 4 个运算核组共 260 个运算核心,采用片上计算阵列集群和分布式共享存储相结合的体系结构(图 1)。

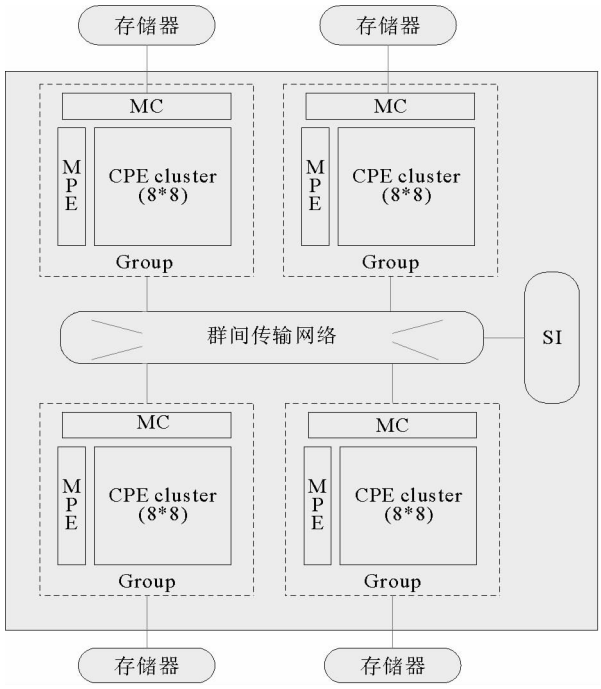

图 1　申威众核处理器结构图

Fig. 1　Architecture of Sunway many-core processor

每个 CPU 包括 4 个异构核组(core-groups,CGs),每个异构核组由 1 个管理核心(management processing element,MPE,主核)、1 个运算核心簇(computing processing elements clusters,CPE Cluster)、1 个存储控制器(memory controller,MC)和系统接口(system interface,SI)组成[5]。

主核在大规模计算过程中通常用于完成管理计算核心以及与其他核组进行通信的任务[7]。CPE Cluster 包含 64 个运算核心(computing processing elements,CPE,从核),以 8×8 的 Mesh 布局通过簇通信网络进行连接。每个从核具备私有的 L1 指令 Cache,一个运算核心簇共享 L2 指令 Cache,从核只能运行于用户模式并且不支持中断,适用于处理逻辑简单而计算相对密集的过程[8]。核组内共享内存。主/从核通过 MC 完成与主存的数据传输。每个从核均包括一个以 SPM(scratch pad memory)方式组织的 64 kB 大小的局部数据存储器(local data memory,LDM),该空间需要用户显式的设计规划使用。

SW26010 处理器从核 LDM 空间有限,往往无法满足使用大数组的科学计算程序存储要求,大部分数据

存储在主存中。在计算过程中,必须通过一定的方式完成主存到从核 LDM 的数据传输。从核支持两种主存访问模式:1)全局读入/写出(gld/gst),直接完成主存到从核寄存器间的零散数据访问;2)直接内存访问(direct memory access,DMA),批量完成从内存到 LDM 的大量数据拷贝,通常延时较大,但带宽利用率较高。相比其他异构平台,从核由于存储空间和带宽限制,数据传输往往成为程序性能瓶颈。Xu 等[9]在主存数据 128 字节对齐的前提下,在一个核组内 64 个从核中进行 Stream Triad,PE 模式下测得 DMA 的带宽可以达 22.6 GB/s,而 gld/gst 方式每个核组带宽利用只有 1.5 GB/s 左右;延迟方面,从核 gld 单核延迟需要 177 个 cycles,而从核访问局存的 1d 只需要 4 个 cycles。可见,DMA 方式比 gld/gst 能够更有效的利用带宽,但启动一次 DMA 需要 300 cycles 左右,开销较大,若 DMA 次数过多会大大降低程序性能。因此在 SW26010 异构众核处理器上,获得更好的数据传输性能的关键在于一次完整数据传输的数据量要尽可能大,并减少 DMA 的启动次数。

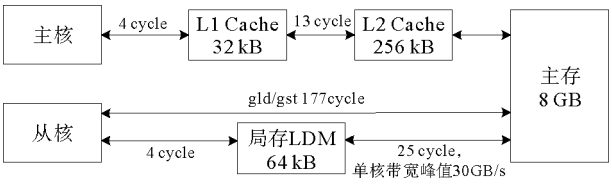

图 2 单核组访存周期

Fig. 2 Access cycles of Sunway core groups

2 AGCM 及其适应过程 Tend_lin 应用的构成及特点分析

2.1 AGCM 动力框架

CAS-ESM 是以耦合器为核心的模块化软件,集成了大气、陆面、陆冰、海冰和海洋等分量模式的复杂系统,AGCM 是其中最复杂的分量模式。

IAP AGCM 主要由物理过程和动力框架两个过程组成。其中,物理过程计算量占比小,并行可扩展性较高;动力框架主要是用来求解关于时间的偏微分方程组,其计算在空间上是三维的,数值方法复杂,是主要的优化部分。IAP AGCM 4.0[12]的动力框架采用均匀经纬网格,并在高纬度采用灵活跳点格式[13]。格点模式中极区计算难以处理,模式引入滤波模块,将纬度划分为 3 个纬度带,不同纬度采用不同滤波方法以增加时间步长,避免计算不稳定。本研究提取 IAP AGCM 4.0 的热点动力框架中的适应过程 Tend_lin 作为研究对象。

2.2 Tend_lin 构成及特点分析

Tend_lin 的计算过程主要包括两个不同的阶段,两个阶段的计算数据来自不同的数组,但都主要由 stencil 和 1 维 fft 两类计算组成,主要包括积分、滤波和平滑等计算类型;通信模式主要是近邻通信和集合通信。其中 stencil 计算涉及 22 组计算循环,占应用串行运行时间的 71%,计算中用到临近若干点,每个计算循环的总访问足迹较大,其中 14 个纬度-高度-经度三维空间全遍历,计算访问 3 维数组较多。滤波和平滑功能的计算分别在 filt 和 smoother 中,不同纬度选择了不同的滤波方法,函数调用层次深(图 3)。该阶段计算循环面对的是同一个高度上的数组区域,包含纬度-经度上遍历的两层循环,每个高度上要进行多阶段处理,不同高度间无数据依赖关系。虽然数据访问量小,但计算循环个数较多。

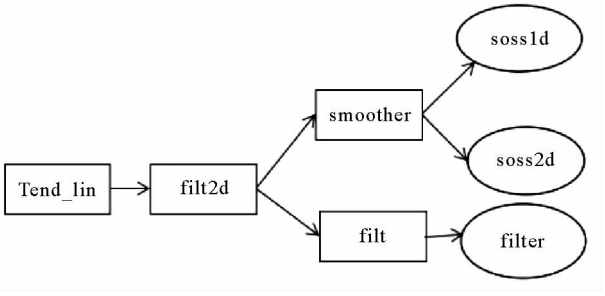

图 3 Tend_lin 应用中 fft 计算函数调用层次图

Fig. 3 Fft computing function call hierarchy of Tend_lin application

Tend_lin 应用涉及 73 个计算循环和单入单出的代码块,数据访问多以数组形式在函数内动态分配内存空间。Tend_lin 程序具有以下特点:①没有热点循环,任务并行代码散落于 73 个并行循环和单入单出的代码块上;②气候模式算法现状导致网格点分辨率不高,每个循环的并行度都不是很大,充分利用众核资源难度大;③循环间大小不一,并行度差异大;④计算/访存比值较低,访存优化为应用性能提升的关键;⑤各进程的计算行为不同构,在纬度方向上,不同进程采用不同滤波方法,且南北极需要特别处理。

为方便并行化,除适应过程的主函数 Tend_lin 之外,本研究增加了纬度和高度方向的区域分解,以及变量初始化部分,并为应用配备 3 个不同的问题规模 A、B、C 以匹配低、中、高不同分辨率,建立了包含 59 个源文件、2 万行左右的 benchmark 程序。

下面,将 AGCM 中的 Tend_lin 串行代码移植到 SW26010 的主核上,并选取网格中典型分辨率即经度×纬度×高度为 1.4°×1.4°×30L(下称规模 B)阐述并行化过程;将 B 规模在经纬度的分辨率扩大 2 倍,得到计算规模是 B 规模 4 倍的 C 规模,用来进行全程序评估。B 规模下,单个紧嵌计算循环最大访问足迹达 62.8 M(stencil_11);2 个计算循环为纬度-经度的二维计算,保持了纬度上的并行度,但缺少高度维,计算量较小;4 个单入单出的代码块,包含经度-高度方向的规约操作;2 个经度方向的单层循环计算。

3　Tend_lin 应用的神威 OpenACC 众核并行化和优化

众核并行化主要目标就是将最耗时的循环迭代分散到多个从核线程上并行执行。下面主要从循环分布、循环分块及并行化、数据传输的优化以及函数调用的从核化 4 个方面描述 Tend_lin 应用在神威平台上的众核并行化方案。

3.1　循环分布

循环分布是在保证应用程序正确性的前提下,将源循环分解为多个独立的目标循环。Tend_lin 应用中,并行循环均为典型的 DOALL 循环,DO 循环中所有的循环实例都可以并行的执行,每个循环实例中的所有语句均串行执行。应用中包含了两类需要特殊处理的并行循环:一类是不同分支的数组读区域,难以被 SWACC 编译器合并,导致编译错误;另一类循环体中包含了多个访问区间不同的子循环且数据足迹较大,其循环体内访问了大量数据重用较低的数组,计算访存比过低。由于 LDM 尺寸有限,若直接并行该类循环,由于分块尺寸较小,数据重用低,会出现较大的数据传输开销,需要在并行化之前进行循环分布。

为避免出现激进的循环分布带来的开销可能会大于并行收益的情况,在对以上两种并行循环进行循环分布的同时要进行空间优化。

1) 分支结构的循环分布

stencil 计算中存在带有分支结构的循环,不同分支间数组的 J 维访问区域冲突,如 GHI(IB:IE, beglev:endlev, JB)和 GHI(IB:IE, beglev:endlev, loc_JB:loc_JE),将其抽象为如图 4 左侧代码。由于并不知道 JB 和 loc_JB:loc_JE 的大小关系,编译器无法给出统一规则完成合并,进而不能完成编译。

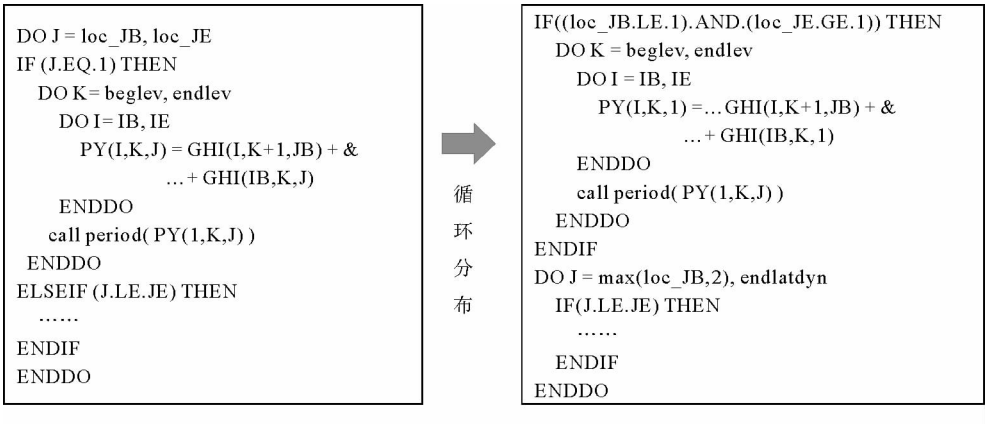

图 4　分支结构分布示意

Fig. 4　Branch structure distribution

经过对该循环的不同分支分布为图 4 右侧所示循环代码,程序能够正常编译且数据传输总量并没有变化。

2) 大数据足迹循环的循环分布

Tend_lin 应用的 02_stencil 阶段存在非紧嵌循环(如图 5 左侧),并且:①最外层 J 循环内,有多个非紧

嵌 K-I 子循环,这些子循环访问变量众多,对从核空间需求较大(约 47 360 kB),即便最外层 J 维分块大小为 1(约 370 kB),仍远超从核 LDM 空间(64 kB);②不同的 K-I 子循环间访问数据足迹大小不一,不同循环间访问的数组不同,且循环间访问的数据无任何依赖或者重用;③K 维访问区间各异,难以合并为紧嵌循环,且 K 维空间过小,不能提供有效的并行度,不适合 K 维并行。

对内层循环进行循环分布时,必须兼顾合法性和收益性问题。分布合法性指循环分布的原有数据依赖语义必须保持,既要保证源循环套并列循环间没有跨迭代依赖,又要保证两个语句组之间没有反向的数据依赖。而为保证循环分布后的收益,有数据重用的尽量放到一个循环套内[14],减少因拆分导致的同一数组的多次读取,避免额外开销。

本研究在保证 J 维并行的前提下进行循环分布,将一个执行次数较多的非紧嵌循环分割成若干个执行次数较少、访问足迹较小的完美紧嵌子循环(如图 5 右侧),以求每个小循环并行时,获得更大的循环分块尺寸,提高 DMA 效率。

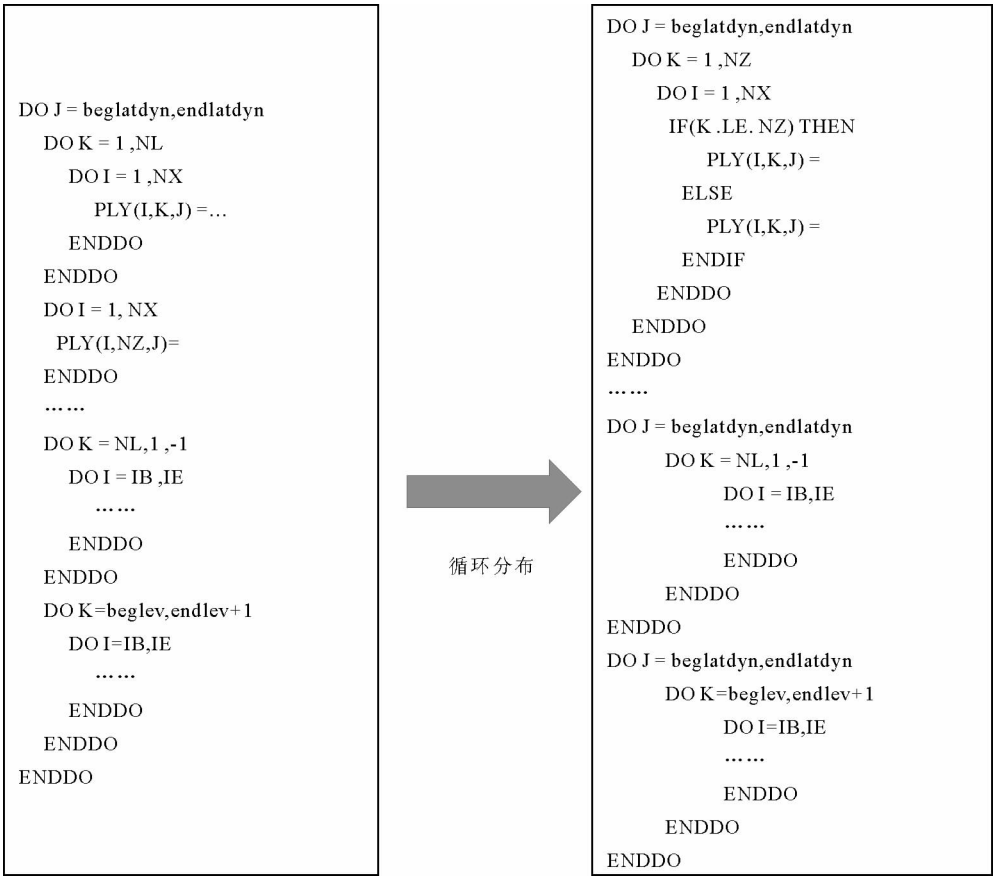

图 5 大数据足迹循环分布示意

Fig. 5 Big data footprint loops distribution sample

3.2 循环分块及并行化

神威 OpenACC 按照 OpenACC 2.0[11]标准进行循环并行化,使用 parallel 和 loop 语句指示加速区代码,加载到从核并行执行。但与标准 OpenACC 不同的是,由于硬件结构的区别,神威上 gang、worker、vector 的三层循环设计没有分层的需求,默认 gang 设置成 64,worker 为 1,vector 也并未做 SIMD 功能支持。将硬件特征与程序特征进行匹配的循环分块可以充分的开发硬件架构的潜能,将串行程序生成适合粒度的并行分块程序。

1) 线程并行维度的选择

Tend_lin 应用中进程间保留了 IAPAGCM4.0 中的纬度-高度二维的进程划分,本研究在线程间使用神

威 OpenACC 进行众核并行化,支持一维的并行。应用中的并行 DOALL 循环除 stencil_09 外,数组索引与循环索引紧耦合,可直接选取循环索引最外层,即选择数组的最高维为并行维,以有效避免低效的低维跨步传输,提升 DMA 传输性能。

应用中并行循环之间访问足迹大小差距较大,导致对 LDM 空间的需求差异大,若使用全局一致的循环分块策略,则导致大部分的并行区 LDM 空间利用率不高,且 OpenACC 作为同步式的并行模型不需要全局一致的分块,故本文采取局部最优划分策略,保证每个循环执行时间最优。由于 LDM 空间有限,对于数据传输量较大的循环,即便并行维最小分块,也无法满足 LDM 空间需求。为了避免低效的跨步传输,本文选取最高维(纬度)和次高维(高度)使用循环分块子句 tile 进行二维划分,即纬度维循环并行分块,高度维做 DMA 流水分块。对于只划分并行维就可以满足空间需求的并行循环,则不再做次高维的流水划分。对于非耦合循环 stencil_09,由于存在最内层 K 维的规约操作,不宜调整循环顺序,仅选取次高维(经度)流水分块。

2) 循环分块尺寸的选择

为充分利用从核 LDM 空间,并保证最少的 DMA 通信次数,编译器在对循环进行分块时,既要保证内循环块中访问的数据总量不超过 LDM 的大小,又要保证使用尽可能大的 LDM 空间。根据文献[14]中的面向异构多核处理器的循环分块方案,以访问足迹最大的紧嵌并行循环 stencil_11 为例,求解循环分块的可行解。分析过程如下:

step 1:计算并行循环的总空间需求

stencil_11 使用的数组数据类型为双精度浮点类型,每个元素 8 个字节,计算该循环用到 8 个三维数组、3 个二维数组和 4 个一维数组,另外还访问 1 个纬度方向跨度为 1 的 3 维数组和 1 个纬度方向跨度为 2 的 2 维数组作为阴影区,则共需要的数组尺寸为((256×128×30×8+256×128×3+256×4+256×1×30×1+256×2×1)×8)=63 774 720 字节=62 280 kB,远超 LDM 空间 64 kB,须进行循环分块后才能够放入局存当中。

step 2:确定最高维纬度 j 层分块大小

令该层的分块大小为 1,这时循环访问的数组所占空间即数据足迹为 558 kB,仍超过 LDM 空间,说明即便使分块大小为 1,仅进行最外层(最高维)的循环分块,仍不能满足 LDM 空间需求的,需要对下一维进行划分。

step 3:确定次高维高度 k 层分块大小

当最外层大小为 1 时,数据足迹是 LDM 空间的 8.7 倍。由于循环访问的数组形状差异大,数据足迹和 k 层分块之间并不是严格成比例。当 k 层循环分块大小取 3(30/8.7=3)时,数据足迹大小 72 kB,超过 LDM 空间;当 k 层循环次外层分块大小取 2 时,数据足迹为 54 kB,为本分块策略下最大可用空间,并为最大次外层分块尺寸。

通过以上 3 步,得到循环分块维度划分方案:最外层(并行维)按分块大小 1 划分,次外层按照分块大小 2 或 1 进行划分,最内层不划分。

3.3　数据传输优化

在第 1 部分介绍的从核线程的两种访存方式中,直接访存比全局离散访存能更好的利用带宽,因此从应用特征出发,最大可能地将计算所需的数据以 DMA 方式批量拷贝到 LDM 存储,消除计算过程中的离散全局访存是性能优化步骤的关键。

本研究采用将计算中所需的数据都存储在从核 LDM 中的数据管理策略,Tend_lin 应用计算过程中访问的数组多为动态数组,编译器无法直接获取其区域信息;循环间访问行为各异,需要对数据传输进行差异化表达。经过 3.1 节循环分布,实现了循环中访问数组索引变量与循环的索引变量紧耦合,这些循环直接使用 tile 子句由编译器自动完成数组划分,搭配 copy/copyin/copyout 语句,在每个并行循环分块计算前,以 DMA 传输方式将所需数据传输到各个从核的 LDM 中,实现数据的提前拷贝。具体进行数据传输时需要解决以下几个问题:

1) 动态数组区域信息的表达

应用中的大量动态数组,在编译时编译器不能获得其维度信息,其形状不能被分析出来,无法在从核 LDM 中申请对应循环分块尺寸的空间,需要使用暗示制导语句 annotate(dimension(array))来指明数组维

度,为编译器划分数组提供依据。如图 6 所示,并行循环中 DGH 和 GHI1 数组为动态数组,利用 copy/copyin 子句搭配 dimension 暗示制导语句的形式实现动态数组的 DMA 传输。

```
! 定义
   real(r8), allocatable :: GHI1(:,:,:)                    ! d(GHI)
   real(r8), allocatable :: DGH(:,:,:)                     ! d(GHI)
! 内存空间分配
   allocate ( GHI1(NX,NZ,beglatdyn:endlatdyn) )
   allocate ( DGH(NX,NL,beglatdyn:endlatdyn) )
! stencil_18  中使用
!$acc parallel loop local(I,K,J) copyin(DGH) copy(GHI1) tile(J:JBLK,K:KBLK) &
!$acc
annotate(dimension(GHI1(1:NX,1:NL+1,beglatdyn:endlatdyn),DGH(1:NX,1:NL,beglatdyn:endlatdyn)))
      数组循环使用
!$acc end parallel loop
```

图 6 Dimension 使用示意

Fig. 6 Dimension usage sample

2) 最低维的传输区域表达

Tend_lin 应用的并行循环中,存在对非并行维非全维度的访问,如图 7 所示,J 维为并行维,VT 为动态分配内存空间的三维数组,在计算中 I 维只访问了区间[IB:IE],占全维度的 98%,并没有访问全维度。

系统提供的 parallel copy 子句仅支持标量和数组,不支持子数组形式,因此 copy 子句只能传输全维度数据,而不是该维度的子区间(IB:IE)。若要实现非全维度的准确低维区间传输,只能利用数组声明进行手工修改。但由于精确传输的 dma_get 为跨步传输,单从核带宽为 6.33 GB/s,64 核带宽为 30.48 GB/s。而直接使用 copy 子句进行 I 维全维度传输为非跨步传输,虽然数据量增加了 2%,但单从核带宽可达 8.14 GB/s,64 核带宽为 30.18 GB/s。导致两个方案在 64 线程时的 DMA 差别不到 1%,即全维度传输虽然多使用了 2%的存储空间,但未明显影响传输时间。因此最终选用 copy 子句实现 I 维全维度传输。

3) 循环不变数组的拷贝

图 8 中,相对于并行维 J,只读数组 A 是循环不变数组,在并行循环中的每个实例是相同的,存在读读重

```
!定义
real(r8), allocatable :: VT(:,:,:)
allocate ( VT (NX,beglev:endlev,beglatdynex:endlatdynex) )
!计算
DO J = loc_JB,loc_JE
  DO K = beglev ,endlev
    DO I = IB, IE
      …= …+…*VT(I,K,J)
    ENDDO
  ENDDO
ENDDO
```

图 7 非并行维非全维度访问示意

Fig. 7 Full dimension access of non-parallel dimension

```
!$acc parallel loop local(I,K,J) &
!$acc copyin(A) tile(J:JBLK,K:KBLK) &
!$acc annotate(entire(A))
DO J
  DO K
    DO I
      ……=A(K)……
    ENDDO
  ENDDO
ENDDO
```

图 8 循环不变数组用法示意

Fig. 8 Loops unchange arrays usage sample

用。循环中数组 A 的访问区域只与流水循环索引有关,编译器依照流水分块尺寸在流水循环内生成的多次 DMA 传输。这时利用 annotate(entire(array))将整个数组在流水循环外以一次 DMA 的方式传输到每个从核 LDM 中,使每个从核的 LDM 中都保有该数组的一个完整副本。但由于 LDM 空间有限,该方式只适用于传输较小的数组。

4) 数组的转置后传输

如图 9 中的 stencil_09 循环,并行循环维度从高到低顺序为 J、I、K,I 循环内存在子循环 K,对应 TT1 数组的最高维,循环内对 TT1 数组的访问是跨步访问,需要利用转置子句 swapin 对原始数据进行重新布局,改善数据传输效率。如图 9 所示,在数据传输前,主核使用 swapin 语句数据进行转置,使循环索引与数组索引的高低维顺序对应,以转置后的数组代替原数组访问,再按照循环分块进行传输。

对于该计算循环,转置前由于不能使用 copyin 将 TT1 拷入 LDM,计算过程只能使用 gld 访存,计算时间占总执行时间的近 80%;使用 swapin 转置后,该数组访问由低效的 gld 变为高效的 ld 操作,计算时间缩短了 13 倍,这样虽然转置时间占到了转置后计算循环总执行时间的 30%,但计算时间的大幅缩短使整个循环执行时间减少 2 倍以上,该循环在整个程序中占计算比重 20%,为 tend_lin 全程序带来 5%左右的影响。

```
real(r8),allocatable :: TT1(:,:,:,:)
allocate ( TT1(IB:IE, beglatdyn:endlatdyn, 2, NL) )
!$acc parallel loop local(I,J,K) copyin(rhos,ksa,...,DpDrX,DpDrY) &
!$acc swapin(TT1(dimension order:4,1,2,3)) tile(j:JBLK,i:IBLK) &
!$acc annotate(dimension(TT1(IB:IE,beglatdyn:endlatdyn,2,NL)))
DO J = loc_JB,loc_JE          //并行维
   DO I = IB, IE              //流水维
      DO K = 1 ,NL
         ......
         D1(K) = TT1(I,J,1,K)
         ......
      ENDDO
      DsaX=·····
      DOK=1, NM
         DWK(K) = ·····+D1*(K)
      ENDDO
      DOK=2, NL
      WSK(K) = ·····-DWK(K)
      ENDDO
   ENDDO
ENDDO
```

图 9　Swapin 子句使用示意

Fig. 9　Swapin clause usage sample

5) 标量的传输

每个并行循环中访问的标量最多有 12 个,平均 5 个左右,按照读写类型分为只写和只读两类。只写标量都为线程私有变量,可直接使用 local 子句本地化。

只读标量多带有 parameter 属性,即便不在制导语句中有任何体现,编译器仍然可以根据其 parameter 属性自动添加到 copyin 和 annotate readonly 子句中,在从核 LDM 中生成对等的 parameter 变量,直接赋值。对于不具备 parameter 属性的只读循环边界量,编译器也可自动添加到 copyin 和 annotate readonly 子句中,在每个从核 LDM 中生成一份拷贝,每个变量以一次 gld 的形式 load 到 LDM 中;对于不具备 parameter 属性的只读循环内部变量,编译器无法分析出该变量是否需要私有化,必须手动添加到 copyin 子句中才能在 LDM 中生成拷贝并以 gld 形式拷入。经过以上处理,将从核计算过程中所需要的数据在计算前存入从核 LDM 中。

为避免计算过程中不存在的低效 gld/gst 操作,利用“神威·太湖之光”上的性能分析工具 penv_slave2_gld_count 和 penv_slave2_gst_count,在编译器生成的中间代码中统计计算过程中从核 gld/gst 请求次数,确保完全消除了计算过程中的全局离散访存。

3.4　函数调用的从核并行

应用中,加速区存在大量 routine 函数调用,且关系复杂,若直接在 routine 函数调用层的循环外使用制

导语句,循环内的 routine 函数调用无法直接生成从核版本,导致生成的从核代码找不到对应被调 routine 函数。

对于加速区的函数调用,在被调用函数内使用函数指示 routine 处理,为加速代码区中被调用的函数自动创建可在从核上执行的版本。

4 并行化方法的测评

本研究以 Tend_lin 应用在 SW26010 处理器 1 个主核上,使用-O3 优化选项的串行版本为测试基准,并以主核串行版本输出为标准,相对误差控制在 10^{-10} 以下。针对第 3 节提出的并行化方案,分别利用 B、C 规模进行实验性能评估。

4.1 不同分块大小对程序性能影响

根据 3.2 节,在 B 规模下,以最高维纬度为并行维,分块大小为 1,次高维高度为流水维,最大访问足迹的并行循环流水分块分别使用 1(36 kB)或者 2(54 kB)对比测试;在最大访问足迹循环流水分块为 2 的基础上,为每个并行循环选取空间需求不超过 LDM 空间的最大流水分块尺寸,记为 n。依此共设置三组对照实验,分别取分块大小为 1×1、1×2 和 1×n,多线程相比主核单线程获得的加速比结果如图 10 所示。

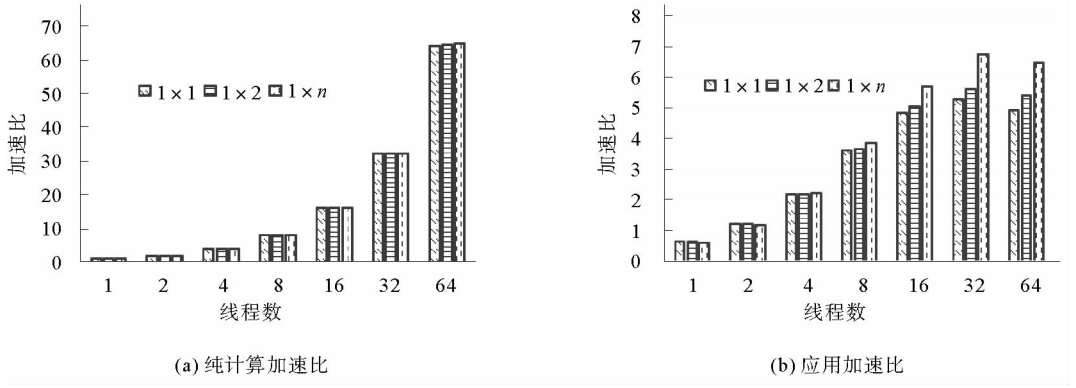

图 10 神威 OpenACC 版多线程不同分块尺寸性能对比

Fig. 10 Performance comparison of different block size of Sunway OpenACC multiple thread

由图 10(a)可看出,随着线程增加,纯计算阶段按比例加速,具有强扩展性,应用性能提升的关键在于优化访存性能。由图 10(b)可知,随着流水分块的增大,应用性能不断提升,说明在满足 LDM 空间限制的前提下,DMA 传输总量不变,随着流水分块尺寸的增大,DMA 传输启动次数减少,降低了 DMA 开销;同时随着流水分块尺寸的增加,每次 dma_get 传输的向量长度增加,带宽利用率增大。以上两方面访存性能的提升使全程序性能随之提升。

4.2 全程序评估

在 B、C 两规模下使用二维进程布局,每个进程选取最优线程数下表现的性能,得到 Tend_lin 应用最终获得的全程序性能(如图 11),在 B 规模下最终获得相比主核串行 25 倍的加速;随着分辨率的进一步扩大,在 C 规模下 64 进程时最高获得 69 倍加速。

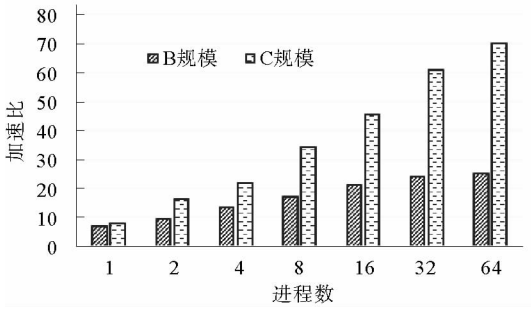

图 11 Tend_lin 应用不同规模多进程下加速比

Fig. 11 Multiple process speedup of Tend_lin application on different scales

5 总结

本研究基于国产异构众核平台“神威·太湖之光”,使用神威 OpenACC 并行编程模型对 Tend_lin 应用进行了并行化,分别从循环分布、循环分块、数据

传输表达和函数调用从核化 4 个方面说明并行化方案。为保证高效的数据传输效率,从核并行选择数组的最高维为并行维,对于最高维并行不能满足 LDM 空间需求的循环,在次高维进行流水划分,并为每个并行循环选取最大的流水分块尺寸;从核并行化使用高效的 DMA 完成主存与 LDM 之间的数据传输,保证计算过程中只访问从核 LDM 空间,并重点对典型数据传输的表达进行讨论。最终应用在国产异构众核平台上,B 规模单核组多线程获得 6.8 倍的全程序加速,多进程获得 25 倍加速;随着分辨率扩大,加速更加明显,C 规模多进程下最多获得 69 倍加速。

参考文献

[1]WANG B,WAN H,JI Z,et al. Design of a new dynamical core for global atmospheric models based on some efficient numerical methods[J]. Science in China Series A:Mathematics,2004,47(1):4-21.

[2]徐金秀,李中华,孙俊,等.基于国产十亿亿次超算系统的近连续过渡流区 N-S/DSMC 耦合算法并行优化研究[C/CD]//青岛:全国高性能计算学术年会论文集,2018.

[3]李亿渊,王欣亮,许平,等.稀疏矩阵向量乘法在申威众核架构上的性能优化[C/CD]//青岛:全国高性能计算学术年会论文集,2018.

[4]YANGG W. A highly efficient GPU-CPU hybrid parallel implementation of sparse LU factorization [J]. Chinese Journal of Electronics,2012,21(1):7-12.

[5]FU H,LIAO J,YANG J,et al. The Sunway Taihu Light supercomputer:System and applications[J]. Science China Information Sciences,2016,59(7):072001.

[6]ZHANG J,LUO J,DONG F. Scheduling of scientific workflow innon-dedicated heterogeneous multicluster platform[J]. The Journal of Systems & Software,2013,86(7):1806-1818.

[7]刘鑫,郭恒,孙茹君,等.“神威·太湖之光”计算机系统大规模应用特征分析与 E 级可扩展性研究[J].计算机学报,2018,41(10):2209-2220.
LIU Xin,GUO Heng,SUN Rujun,et al. The characteristic analysis and exascale scalability research of large scale parallel applications on Sunway Taihulight supercomputer[J]. Chinese Journal of Computers,2018,41(10):2209-2220.

[8]陈德训,刘鑫.神威·太湖之光并行程序设计与优化[M].北京:国家并行计算机工程技术研究中心.2017.

[9]XU Z,LIN J,MATSUOKA S. Benchmarking SW26010 many-core processor[C]//Parallel and Distributed Processing Symposium Workshops (IPDPSW),2017 IEEE International. IEEE,2017:743-752.

[10]刘侃,王欣亮,许平,等.申威众核处理器上的三对角并行求解器[C/CD]//青岛:全国高性能计算学术年会论文集,2018.

[11]FARBER R. Parallel programming with OpenACC[M]. Oxford:Newnes,2016.

[12]张贺.大气环流模式 IAP AGCM4.0 的设计及其数值模拟[D].北京:中国科学院研究生院(大气物理研究所),2009.

[13]张贺,林朝晖,曾庆存.IAP AGCM-4 动力框架的积分方案及模式检验[J].大气科学,2009,33(6):1267-1285.
ZHANG He,LIN Zhaohui,ZENG Qingcun. The computational scheme and the test for dynamical framework of IAP AGCM-4[J]. Chinese Journal of Atmospheric Sciences,2009,33(6):1267-1285.

[14]韩林,徐金龙,李颖颖,等.面向部分向量化的循环分布及聚合优化[J].计算机科学,2017(2):70-74.
HAN Lin,XU Jinlong,LI Yingying,et al. Method of loop distribution and aggregation for partial vectorization[J]. Computer Science,2017(2):70-74.

[15]李雁冰,赵荣彩,赵博, 等.面向异构多核处理器的的循环分块[J].计算机工程与设计,2015,36(1):168-173.
LI Yanbing,ZHAO Rongcai,ZHAO Bo,et al. Loop tiling for heterogeneous multi-core processor[J]. Computer Engineering and Design,2015,36(1):168-173.

[16]ZHANG J,LUO J,DONG F. Scheduling of scientific workflow in non-dedicated heterogeneous multicluster platform[J] Journal of Systems and Software,2013,86(7):1806-1818.

[17]BONATI C,COSCETTI S,D'ELIA M,et al. Design and optimization of a portable LQCD Monte Carlo code using OpenACC[J/OL]. International Journal of Modern Physics C,2017,28(5):1750063.

[18]王一超,林新华,蔡林金,等.太湖之光上利用 OpenACC 移植和优化 GTC-P[J].计算机研究与发展,2018,55(4):875-884.
WANG Yichao,LIN Xinhua,CAI Linjin,et al. Porting and optimizing GTC-P on TaihuLight supercomputer with OpenACC[J]. Journal of Computer Research and Development,2018,55(4):875-884.

(责任编辑:李　磊)