引用格式：徐阳，张玉春欣，花嵘. Silicon-Crystal 应用的神威 OpenACC 移植与数据流驱动任务图并行化[J]. 山东科技大学学报(自然科学版)，2019，38(3)：57-64.

XU Yang，ZHANG Yuchunxin，HUA Rong. Sunway OpenACC transplantation and parallelization of task graph based on data stream for Silicon-Crystal application[J]. Journal of Shandong University of Science and Technology (Natural Science)，2019，38(3)：57-64.

Silicon-Crystal 应用的神威 OpenACC 移植与数据流驱动任务图并行化

徐 阳，张玉春欣，花 嵘

(山东科技大学 计算机科学与工程学院，山东 青岛 266590)

摘 要：利用神威 OpenACC 在“太湖之光”上成功移植了 Silicon-Crystal 应用，针对控制流驱动的 OpenACC 无法有效解决访存密集型应用带宽访存优化和跨时间迭代问题，通过数据流驱动的任务图并行化方法挖掘任务迭代间的并行性，利用任务间的错峰访存提高访存带宽。实验结果表明，神威 OpenACC 移植单核组获得 2.26 倍加速；时间步长为 1 时，任务图并行化移植后的该应用可获得 2.52 倍加速，性能较 OpenACC 提升 11.5%；时间步长扩展至 20 时，任务图规模随之增加，任务的乱序调度使错峰访存的优势进一步扩大，整体应用达到 3.2 倍性能加速，性能较 OpenACC 提升 42%。

关键词：太湖之光；神威 OpenACC；数据流；任务图并行；MD 模拟

中图分类号：TP311.52 **文献标志码**：A **文章编号**：1672-3767(2019)03-0057-08

DOI：10.16452/j.cnki.sdkjzk.2019.03.007

Sunway OpenACC transplantation and parallelization of task graph based on data stream for Silicon-Crystal application

XU Yang，ZHANG Yuchunxin，HUA Rong

(College of Computer Science and Engineering，Shandong University of Science and Technology，Qingdao，Shandong 266590，China)

Abstract：In this paper Silicon-Crystal application was successfully transplanted onto the “TaihuLight” by using Sunway OpenACC. Considering the fact that the OpenACC driven by control flow could not effectively solve the problem of memory-intensive applications’ bandwidth-access optimization and cross-time iteration，this research used the method of task graph parallelization driven by data flow to improve the parallelism between task iterations and took advantage of random peak access between tasks to increase the memory bandwidth. Experimental results show that the application through Sunway OpenACC transplantation obtains 2.26 times acceleration on its single-core group. When the time step is 1，the application with transplanted task graph parallelization can obtain 2.52 times acceleration and its performance is 11.5% higher than OpenACC. When the time step is extended to 20，the size of the task graph increases and the out-of-order scheduling of the task further expands the advantage of random

收稿日期：2018-12-10

基金项目：国家重点研发计划项目子课题(2017YFB0202002)

作者简介：徐 阳(1993—)，男，山东济宁人，硕士研究生，研究方向为并行编译、高性能计算.

花 嵘(1969—)，男，江苏常州人，副教授，博士，研究方向为并行编译、高性能计算，本文通信作者.

E-mail：hrfy@263.net

peak access. The total application obtains 3.2 times performance acceleration, which is 42% higher than that of OpenACC.

Key words: TaihuLight; Sunway OpenACC; data flow; task graph parallelization; MD simulation.

分子动力学(molecular dynamics,MD)模拟是指使用数值方法,利用计算机模拟原子核和电子所构成的多体系统的运动过程,已被广泛应用于物理、化学、生物、材料、医学等多个领域,用来研究系统的结构和性质[1]。在材料领域虚拟过程工程中,Silicon-Crystal 应用是研究硅晶体热传导性的 MD 模拟应用,有限的计算能力一直是制约模拟效率的瓶颈[2]。近年来,高性能计算技术的发展为材料领域的虚拟过程工程提供了可能[3]。

"神威·太湖之光"是世界上首台运行速度超过十亿亿次的超级计算机,也是我国第一台全部采用国产处理器 SW26010 构建的超级计算机[4]。清华大学付昊桓等[5]在"神威·太湖之光"上,利用 OpenACC 移植大气模型 CAM 应用,单核组内实现 2 倍加速,但未进行移植后优化;上海交通大学王一超等[6]利用 OpenACC 移植并优化了磁约束聚变领域 GTC-P 应用,单核组内实现 2.5 倍加速,但缺少对访存密集型应用带宽访存优化;中国科学院计算应用研究中心张帅等[7]在 GPU 平台上对 MD 模拟进行访存优化,但未提供模拟中跨时间迭代问题的解决方法。CPU、GPU 架构与 SW26010 架构存在着差异。SW26010 采用片上计算阵列群和分布式共享存储相结合的异构众核体系架构,使得 MD 模拟应用的移植具有更大的灵活性,但也使得移植难度加大,目前对 MD 模拟移植到"神威·太湖之光"超级计算机上的相关研究尚未见到。

本文设计了一种 SW26010 主从计算并行化方案,实现对 Silicon-Crystal 应用的神威 OpenACC 移植与优化;以数据流驱动的任务图并行化方法解决任务间的峰值访存、跨时间迭代问题,针对该应用访存密集型特点进行带宽访存优化。

1 背景介绍

1.1 SW26010 处理器架构及神威 OpenACC 执行模型

"神威·太湖之光"是中国自主研发的超级计算机,峰值性能为 125.4 PFlops,实测峰值约为 93 PFlops。采用新一代的众核异构处理器 SW26010(架构如图 1)。神威 OpenACC 程序的执行模型是在主核指导下,主从核协同工作,其加速执行模型如图 2 所示。

SW26010 异构众核架构中,各核组之间采用片上网络互连,每个核组包含 1 个主核(management processing element MPE)、1 个从核簇(8×8=64 个,computing processing element,CPE)、1 个协议处理单元和 1 个内存控制器。核组内采用共享存储架构,内存与主、从核之间可通过内存控制器传输数据,处理器可通过系统接口与外部设备相连[8]。申威众核处理器旨在用少量具备指令级并行能力的管理核心集成众多面向计算开发的精简运算核心高效处理线程级并行,从而大幅提高芯片性能[9]。

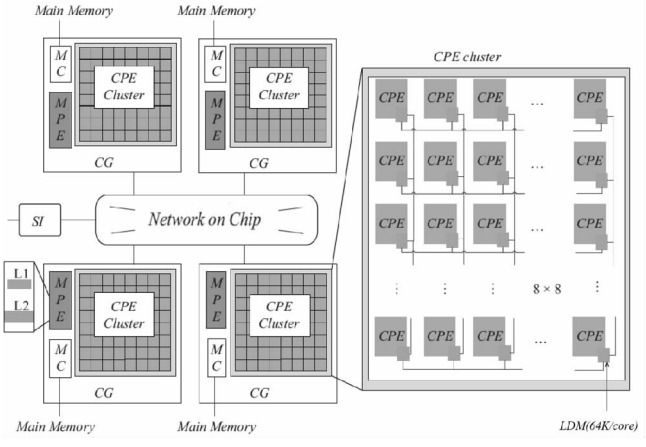

图 1 "SW26010"异构众核架构

Fig. 1 Heterogeneous multi-core processor architecture of "SW26010"

程序首先在 MPE 上启动,以一个主线程串行执行,计算密集区域则在主线程的控制下作为加速任务被加载到加速设备 CPE 上执行[10]。任务的执行过程包括:在 CPE 设备内存上分配所需的数据空间;加载任务代码至 CPE;任务将所需的数据从 MPE 传输至 CPE 内存;等待数据传输完成;CPE 进行计算并将计算结果传送回主存;释放设备上的数据空间等步骤。

MPE 加载一系列任务到加速设备上同时执行,但这种 fork-join 模式在访存带宽有限的 SW26010 处理器上易产生峰值访存问题,使 CPE 之间相互争抢带宽,从而影响计算性能。

1.2　AceMesh 编程框架

AceMesh 编程框架是面向网格应用[11-12]、以数据为中心,应用于多核、众核平台上的数据流驱动并行编程框架。AceMesh 并行编程框架通过底层的任务调度系统[13](运行时库)对网格应用进行任务图并行,其核心设计思想来源于图论中的有向无环图(directed acyclic graph,DAG)。

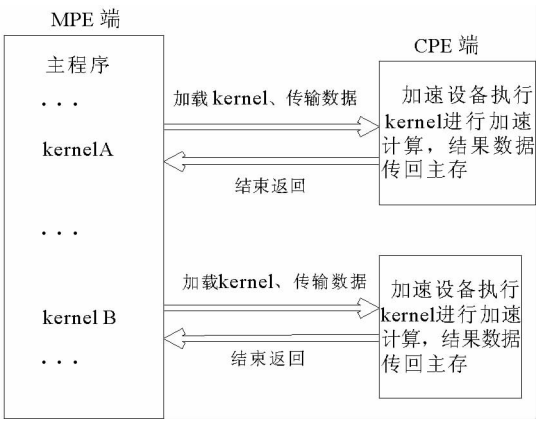

图 2　神威 OpenACC 执行模型

Fig. 2　Execution model of the Sunway OpenACC

任务调度系统采用探测-执行(inspector-executor)两阶段执行的并行模式[14],该模式对并行区域进行代码级调度。探测阶段将代码区域的控制流和数据流信息提交给运行时系统,由运行时系统根据任务间依赖关系建立任务依赖图。执行阶段以构建的任务图为基础,依据资源配置及利用率搭配不同的任务调度策略和算法,动态的调度并行任务。

AceMesh 编程框架任务调度系统结构图如图 3 所示,该调度系统包括四层:

1) 用户接口层,收集任务粒度[15]的描述、数据流信息、任务构造等信息;

2) 任务构建层,根据上层用户提供的信息,在系统内部产生任务、建立依赖和进行任务图管理;

3) 任务调度层,通过静态调度和动静结合调度两种方式提供任务调度支持。静态调度采用轮询法按权值将任务分配至线程;动静结合调度指静态调度策略与任务窃取调度算法[13,16]相结合,提高任务数据重用率和线程间负载均衡性;

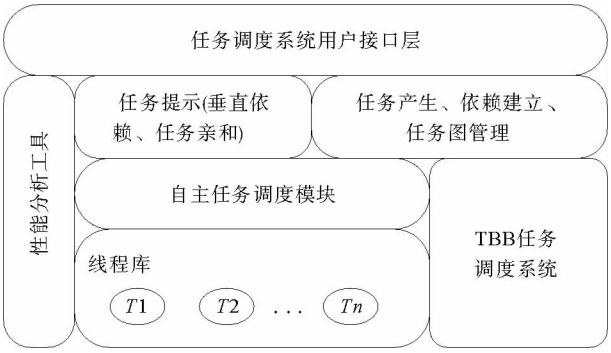

图 3　AceMesh 编程框架任务调度系统

Fig. 3　Task scheduling system of AceMesh programming framework

4) 队列调度层,利用线程库对线程私有并发任务调度队列进行任务级调度。

2　Silicon-Crystal 应用分析及移植方案设计

2.1　应用算法和模拟过程数据特征分析

MD 模拟中,通过差分求解牛顿运动方程可得到系统中原子的一系列位形。由于模拟过程中力的计算工作量很大,常用的龙格-库塔法已不再适用,Silicon-Crystal 应用中的 TP(Tersoff Potent)模块利用 leap-frog 算法[17]模拟原子在 Tersoff 势能作用下的运动轨迹,在所有的线性微分方程的求解器中都有应用。

基于有限差分法 leap-frog 算法,求解线性常微分方程公式如下:

$$r(t+\Delta t) = r(t) + \Delta t \times v(t+\frac{\Delta t}{2}) \ , \tag{1}$$

$$v(t+\frac{\Delta t}{2}) = v(t-\frac{\Delta t}{2}) + \frac{\Delta t}{m} \times F(t) \ 。 \tag{2}$$

其中,r、V、m、F 分别为原子的位置矢量、速度、质量、所受势能力,Δt 为计算时间步长。

2.2　TP 模块并行方案

加速线程库(athread 库)是针对主从加速编程模型所设计的程序加速库,旨在使用户能够方便、快捷地

使用核组内的线程进行控制和调度,从而更好地发挥组内多计算核的性能。本研究使用加速线程库将 TP 模块移植到从核的运算模式如图 4 所示。

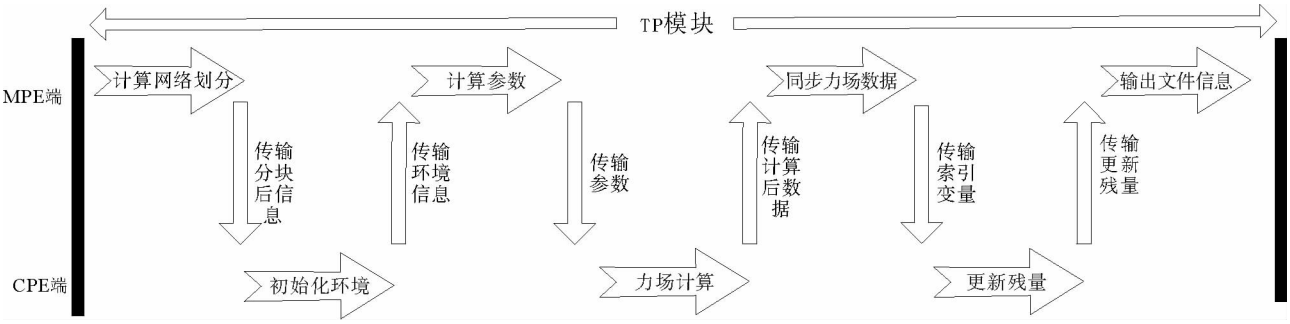

图 4 TP 模块并行方案设计

Fig. 4 Parallel scheme design of TP module

TP 模块的移植主要分为:

1) 计算网络划分。MPE 端沿三维空间 x、y 和 z 三个方向将数据区域按比例分成若干矩形体,每一矩形体计算视为一个任务。这样的划分方式有两个好处:其一,CPE 端得到的数据在空间上是连续的,数据块访问开销比较小;其二,分块内中心原子占比相对较高,减少分块间的原子通信量,提升计算效率。

2) 初始化环境。CPE 端对划分后任务内的原子信息进行初始化,初始化信息包括原子的位置矢量、加速度、速度等。

3) 计算参数初始化。初始化 MPE 端对 Tersoff 势能下的离散计算参数。

4) 力场计算。以任务为基本单位将原子信息加载至 CPE 端进行加速计算,首先进行 MPE 端至 CPE 端的数据拷贝,其次利用 CPE 端的计算阵列群加速核心计算,最后将计算后的各个任务原子信息由 CPE 端传回 MPE 端。

5) 同步力场数据。MPE 端按照邻居关系索引表进行任务间数据更新操作,保证数据全局一致性。

6) 更新残量和输出文件信息。CPE 端进行每个时间步计算后的残量更新,MPE 端将计算范数值输出至文件系统。

3 神威 OpenACC 从核移植

神威 OpenACC 并行编程模型,用编译指示的方式把应用中可并行化的计算循环移植到申威处理器从核以加速计算。具体到 Silicon-Crystal 应用的从核移植,主要分为以下三个步骤:

1) 循环并行化。Silicon-Crystal 应用以任务分片存储的数据为基本单位进行模拟计算,在分块级的 for 循环上添加相应的指导语句 # pragma acc parallel loop,将计算部署在 64 个从核上并行执行。

需要注意的是,gang、worker、vecotr 是 OpenACC2.0 中的 3 层循环设计,由于神威众核架构在物理上并没有分层需求,所以神威 OpenACC 的实现是把 gang 设置成 64,worker 设为 1。

2) 基于计算数据优先的数据管理。神威众核架构中存在访存带宽较小的问题,故从核移植并行化过程最为关键的是将加速计算需要的数据提前拷贝到访问延迟低的 SPM(scratch pad memory)。本研究采用计算数据优先传输策略即将所有计算涉及的数据优先传至 SPM。

计算数据优先传输过程如下:

i) 按循环索引划分传输。若数组的索引变量与循环索引变量紧耦合时,神威 OpenACC 编译器将数组划分为 64 份,然后利用 DMA 的方式将划分后的数据集中传输至各从核 SPM 中,并将任务内的邻居关系表、打包后的计算参数顺序传递给从核。具体使用 copy/copyin/cpoyout 等指导语句完成(如图 5)。

```
①#pragma acc parallel loop\
②local(pi)\                    //变量局存私有化
③copy(ax,ay,az,pox,poy,poz)annotate(dimension(ax(PN,N),ay(PN,N),az(PN,N)))\   //按循环索引传输
④copyin(maplis) annotate(dimension(maplist(4 * N)))\      //邻居关系表传输
⑤packin(deltaT,soma,Mu,Lamda,Beta,nexp,InteriorCutoff,ExteriorCutoff)      //离散计算参数打包传输
⑥for (pi = 0; pi < PN ; pi++){
...
}
```

图 5　数据管理过程的函数指导语言实现

Fig. 5　Implementation of functional instruction language based on data management

ii) 变量局存私有化。对于并行循环索引变量等线程私有变量,既可使用 private 子句也可使用 local 子句将变量私有化,考虑到 private 是线程私有化变量,变量值仍在主存中,而 local 是线程私有化的局存变量,存储在 SPM 中,数据访问更加高效,故采用 local 子句进行变量的私有化。

iii) 离散计算参数打包传输。Silicon-Crystal 应用存在多个离散标量的模拟参数需要传送至从核,若一一传输需要频繁的使用 DMA 方式,会大大增加访存开销。在此情况下,本文利用 pack/packin/packout 等神威定制的指导语句将离散数据打包后一次传递,以更充分有效地利用有限的访存带宽。

综上,得到 Silion-Crystal 应用移植中的数据管理过程的函数指导语言实现如图 5。

3) 加速代码区约束处理。SWACC 编译器进行 OpenACC 并行化过程中,对并行区的代码有一定的要求。如在加速区代码中存在函数调用时,需在函数定义处添加 routine 子句指示,否则生成从核代码将找不到函数的位置。但目前 routine 子句只适用 Fortran 程序,C 代码暂不支持。Silicon-Crystal 应用程序是 C 代码程序,无法利用 routine 子句修饰从核函数。本研究通过利用宏定义实现力场计算的内联函数,来解决移植过程中加速区函数返回值异常的问题。

将运行在 1 个主核上的 Silicon-Crystal 作为测试基准,分别与循环并行化、基于计算数据优先的访存和离散计算参数打包传输 3 个方面在单核组上进行性能测试(图 6)。测试问题规模:回环中存在131 072个粒子,迭代计算次数为 1 000 次。

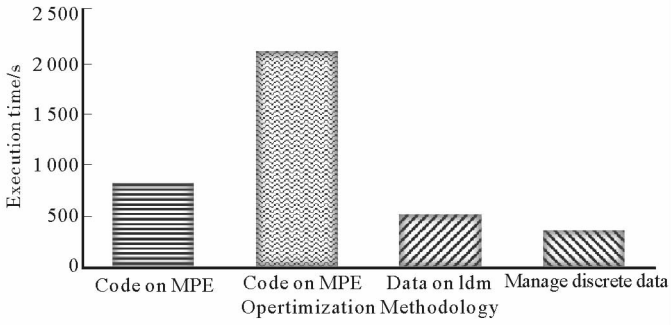

图 6　神威 OpenACC 移植性能数据

Fig. 6　Performance data of the Sunway OpenACC transplant

可以看出,对于访存密集型的应用,仅进行循环并行化将计算过程移至 CPE 端,性能反而会降低;按照计算数据优先方式通过 DMA 方式放入从核 SPM 中,性能开始超越主核;通过 pack 子句对离散计算参数打包后再传输,性能进一步提升,整体应用较主核版实现了 2.26 倍的加速。

4　数据流驱动的任务图并行化

AceMesh 任务调度系统的设计思想来源于数据结构中的有向无环图,即任务依赖图。任务依赖图在图论中是指:如果一个有向图无法从某个顶点出发经过若干条边回到该点,则这个图是一个任务依赖图。任务依赖图中的顶点代表任务,图中的边代表任务间的依赖关系。根据任务依赖图的特点,将并行计算中的大规模计算问题划分为 $N(N\geqslant 1)$个任务,并根据各个任务的依赖关系建立任务依赖图,图中所有没有后继的顶

点都执行完后,任务依赖图的执行完成。

在神威众核处理器上任务图并行化过程分为任务构图期和任务执行期。构图期是任务构建的探测过程,旨在根据注册的数据地址去建立任务间的依赖关系,在不改变串行序结果的情况下以数据流调整执行序列;执行期是指按照构图期间构建的 DAG 图,搭配任务调度系统的不同调度策略执行任务的过程。故任务图并行化总时间等于构图时间(graph time)加执行时间(execution time)。

Silicon-Crystal 应用在太湖之光上使用任务图并行化主要分为以下 3 个步骤:

1)主核构建任务依赖图。根据不同并行区内划分的任务按照对内部访问的数据依赖关系进行地址注册,构建出任务执行序 DAG 图。

图 7 为 Silicon-Crystal 应用 2 线程 4 任务依赖图。其中,每个椭圆代表包装后的一个任务,椭圆中第一个数字为并行区编号,第二个数字为任务编号;箭头代表任务间的数据访问先后的依赖关系;阴影、非阴影圆圈代表执行时不同的线程;实线边是任务垂直后继依赖边,虚线边是普通后继依赖边。在任务执行期间,采用的调度策略使垂直后继任务优先于普通后继任务执行,旨在使任务间的数据重用得到最大化。此外,截断并行区间任务执行的依赖关系,按照与神威 OpenACC 相同的控制流驱动的 fork-join 执行模式,称为任务图单步执行。

2)从核包装任务函数,将应用主要的计算代码包装成任务函数。从核任务函数根据构图期分配的函数参数、循环划分尺寸、数据区划分尺寸等信息,包装任务函数,放入从核阵列并行计算。

3)从核数据管理。SW26010 主从核间的数据传输通过 DMA 实现,DMA 只能由从核发起,主核被动进行数据传输。从核制定传输的模式时,数据传输依据数据在主存数据区内存储地址的连续性和从核计算实际需要的数据尺寸进行传输。DMA 数据传输方式分为跨步式数据传输和非跨步式数据传输。两种传输模式下,软件开销主要体现在传输的启动和对 DMA 传输回答字的处理。本研究采用数据分片存储的数据结构,将任务访问的数据进行分片存储,并以数据块编号为索引划定数据区,通过 athread_get/athread_put 接口进行非跨步传输;对于离散数据访问,在主核代码中对离散数据打包后,使用 DMA 方式进行数据传输,最后在从核代码中对数据进行解包,提高离散数据的访问效率。

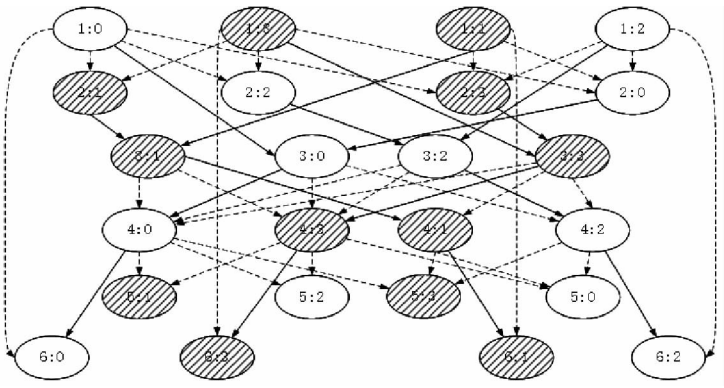

图 7 2 线程 4 任务时 TP 模块任务依赖图

Fig. 7 Task dependency graph of TP module under 2 threads 4 tasks

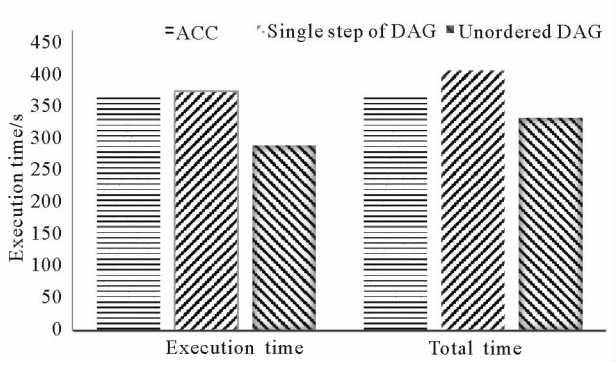

图 8 神威 OpenACC 与任务图并行化性能

Fig. 8 Performance of the Sunway Open ACC and task graph parallelization

本节将第 3 节中 OpenACC 优化后的版本(ACC)作为基础版,同等的优化条件下,与任务图单步版(single step of DAG)、任务图乱序版(Unordered DAG)进行实验对比。迭代时间步长为 1 时,Silicon-Crystal 应用性能如图 8 所示。

可以看出,由于任务图单步版和 OpenACC 采用相同的 fork-join 模式,即并行区开始 spawn 线程,并行区结束 wait 所有线程,故二者的执行时间一致,说明两者具有计算一致性。任务图乱序版比任务单步版提升 27%,验证了任务间的乱序执行,可以错开峰值带宽竞争,充分的利用从核访存带宽。但是,由于神威 OpenACC 采用 fork-join 模式,其执行时间即为总时间;数据流驱动的任务图并行需要在构图期构建任务执行的依赖关系,故其总时间为构图时间与执行时间之和。实验结果表明,Silicon-Crystal 应用的任务图并行化存在相对总时间 8%的构图时间,加上此部分构图开销,总时间上任务图并行比 ACC 性能提升 11.5%。

传统的 fork-join 模式无法扩展多时间步的迭代计算,任务图却可打通迭代时间步间的并行区域,即在构图期依据多个时间步下任务的数据流构建出任务依赖图,执行期按多时间步下任务乱序调度方式执行任务,从而将迭代时间步由单时间步扩展至多时间步,如表 1 所示,随着任务图并行在多时间步的扩展,执行时间进一步降低,构图时间逐渐降低,使得任务图并行性能进一步提升。

表 1　多时间步扩展下任务图并行化构时间

Tab. 1　Times of task graph parallelization based on multiple time steps

multi-time-step	Total time/s	Execution time/s	Graph time/s
1	331.1	302.5	28.6
5	283.1	258.8	24.3
10	260.4	236.7	23.7
20	258.7	235.1	23.6

多时间步下任务图并行化加速比如图 9 所示,以主核版作为基准版,使用神威 OpenACC 移植利用从核加速,实现 2.26 倍加速比;时间步为 1 时任务图并行加速比为 2.52;随着时间步的扩展,任务图规模随之增加,任务的乱序使错峰访存的优势进一步扩大,时间步扩展至 20 时趋于平稳,加速比达到 3.2。

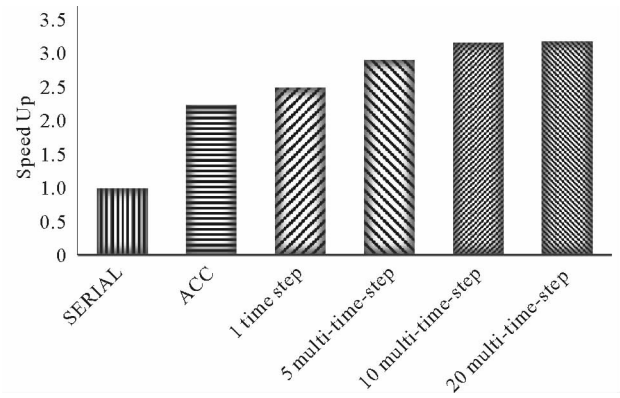

图 9　多时间步下任务图并行化加速比

Fig. 9　Acceleration ratio of task graph parallelization under multiple time steps

5　总结与未来工作

本研究为 Silicon-Crystal 应用设计了一套在 SW26010 上实现主从计算的并行化方案,利用 OpenACC 完成了向目标平台“神威·太湖之光”上的移植,在单核组内实现了 2.52 倍加速;针对该应用访存密集的行为特点,以数据流驱动的任务图并行化方法解决任务间的峰值访存和跨时间迭代问题,结果表明,Silicon-Crystal 应用在数据流驱动的任务图并行在单时间步下性能提升 11.5%,多时间步下性能提升 42%,总体较主核实现 3.2 倍加速。

数据流驱动的任务图并行编程模型采用 AceMesh 任务调度系统中的低级接口对程序源码进行优化,随着本课题组数据驱动的并行调度系统自动转译器的完善,未来将使用指导语言的高级形式对代码进行自动源源变换,从而实现通过指导语言方式对应用任务图并行的自动化过程。

参考文献:

[1]文玉华,朱如曾,周富信,等.分子动力学模拟的主要技术[J].力学进展,2003(1):65-73.
WEN Yuhua,ZHU Ruzeng,ZHOU Fuxin,et al. An overview on molecular dynamics simulation[J]. Advances in Mechanics,2003(1):65-73.

[2]陶晓芳,李晓霞,郭力.MD 模拟 GPU 并行计算现状研究[J].计算机与应用化学,2017,34(5):337-344.
TAO Xiaofang,LI Xiaoxia,GUO Li. Recent progress of molecular dynamics simulations accelerated on graphical processing units[J]. Computers and Applied Chemistry,2017,34(5):337-344.

[3]侯超峰,高国贤,徐骥.纳米材料制备及物性测量的虚拟过程工程初探[J].计算机与应用化学,2016,33(9):1003-1007.
HOU Chaofeng,GAO Guoxian,XU Ji. Explorations to the virtual process engineering offabrication and property measure-

ment of nanomaterials[J]. Computers and Applied Chemistry,2016,33(9):1003-1007.

[4]付昊桓. 全球最强的超级计算机:“神威·太湖之光”[J]. 国防科技工业,2018(5):25.
FU Haoheng. The world's strongest supercomputer:“Sunway TaihuLight” [J]. Defence Science & Technology Industry,2018(5):25.

[5]FU H H,LIAO J F,YANG J Z,et al. The Sunway TaihuLight supercomputer:System and applications[J/OL]. Science China Information Sciences,2016,59(7):072001. DOI. 10. 1007/S11432-016-5588-7.

[6]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.

[7]张帅,徐顺,刘倩,等. 基于 GPU 的分子动力学模拟 Cell Verlet 算法实现及其并行性能分析[J]. 计算机科学,2018,45(10):291-294.
ZHANG Shuai,XU Shun,LIU Qian,et al. Cell Verlet algorithm of molecular dynamics simulation based on GPU and its parallel performance analysis[J]. Computer Science,2018,45(10):291-294.

[8]孟德龙,文敏华,韦建文,等. 神威太湖之光上 OpenFOAM 的移植与优化[J]. 计算机科学,2017,44(10):64-70.
MENG Delong,WEN Minhua,WEI Jianwen,et al. Porting and optimizing OpenFOAM on Sunway TaihuLight system[J]. Computer Science,2017,44(10):64-70.

[9]郑方,张昆,邬贵明,等. 面向高性能计算的众核处理器结构级高能效技术[J]. 计算机学报,2014,37(10):2176-2186.
ZHENG Fang,ZHANG Kun,WU Guiming,et al. Architecture techniques of many-core processor for energy-efficient in high performance computing[J]. Chinese Journal of Computers,2014,37(10):2176-2186.

[10]The OpenACC application programming interface[EB/OL]. [2018-12-06]. https//www. openacc. org/sites/default/files/linline-files/Open ACC20specification. pdf.

[11]王蕾,崔慧敏,陈莉,等. 任务并行编程模型研究与进展[J]. 软件学报,2013(1):77-90.
WANG Lei,CUI Huimin,CHEN Li,et al. Research on task parallel programming model[J]. Journal of Software,2013(1):77-90.

[12]刘颖,吕方,王蕾,等. 异构并行编程模型研究与进展[J]. 软件学报,2014(7):1459-1475.
LIU Ying,LV Fang,WANG Lei,et al. Research on heterogeneous parallel programming model[J]. Journal of Software,2014(7):1459-1475.

[13]王松,花嵘,傅游,等. 数据驱动的任务图执行中并发构图方法[J]. 计算机工程与设计,2018,39(3):758-762.
WANG Song,HUA Rong,FU You,et al. Concurrent composition method for data-driven task graph execution[J]. Computer Engineering and Design,2018,39(3):758-762.

[14]KOELBEL C,MEHROTRA P. Compiling global name-space parallel loops for distributed execution[J]. IEEE Transactions on Parallel and Distributed Systems,1991,2(4):440-451.

[15]XUE W,YANG C,FU 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.

[16]XUE W,YANG C,FU H,et al. Enabling and scaling a global shallow-water atmospheric model on Tianhe-2[C]//IEEE International Parallel and Distributed Processing Symposium. IEEE Computer Society,2014:745-754.

[17]樊康旗,贾建援. 经典分子动力学模拟的主要技术[J]. 微纳电子技术,2005(3):133-138.
FAN Kangqi,JIA Jianyuan. An overview on classical molecular dynamics simulation[J]. MEMS Device & Technology,2005(3):133-138.

(责任编辑:李 磊)