DOI: 10.16452/j.cnki.sdkjzk.2024.03.011 文章编号:1672-3767(2024)03-0105-09

# 神威·太湖之光平台上宇宙 N 体模拟中 FMM 的并行优化

韩承磊[1],梁建国[2],傅 游[1],叶雨曦[1],花 嵘[1],李倩倩[1]
(1. 山东科技大学 计算机科学与工程学院,山东 青岛 266590;2. 曲阜师范大学 计算机学院,山东 日照 273165)

**摘 要**:宇宙学模拟是典型的 N 体问题,是高性能计算中具有代表性和挑战性的问题之一。本研究在神威·太湖之光平台上对天文 N 体模拟软件 PhotoNs-2 中的计算主体——快速多极子方法(fast multipole method,FMM)进行移植和性能优化。针对目前研究中存在的计算效率不高、通信开销大问题,结合神威·太湖之光 SW26010 处理器架构特点,通过数据重整、超越函数计算重构、设计双缓冲和消息传递接口通信时合并发送树进行优化。相较于优化前,优化后的 PhotoNs-2 在 3 个不同算例规模下均取得约 24 倍的加速效果。提出的优化方案可以为其他高性能应用在神威·太湖之光平台上的移植与优化提供参考。

**关键词**:神威·太湖之光平台;并行优化;数据重整;快速多极子方法;宇宙 N 体
**中图分类号**:TP311 **文献标志码**:A

## Parallel optimization of FMM for cosmic N-body simulations on Sunway TaihuLight platform

HAN Chenglei[1], LIANG Jianguo[2], FU You[1], YE Yuxi[1], HUA Rong[1], LI Qianqian[1]
(1. College of Computer Science and Technology, Shandong University of Science and Technology, Qingdao 266590, China; 2. College of Computer, Qufu Normal University, Rizhao 273165, China)

**Abstract**: Cosmological simulation is a typical N-body problem and one of the representative and challenging problems in high-performance computing. This paper ports and optimizes the performance of the fast multipole method (FMM), the main part of the astronomical N-body simulation software PhotoNs-2 on the Sunway TaihuLight platform. Aiming to increase the computational efficiency and reduce the communication expense in the current research, this paper optimizes the fast multipole method by reorganizing data, reconstructing the transcendental function computation and designing double buffering and merging the sending tree during message passing interface (MPI) communication based on the characteristics of the architecture of the SW26010 processor of Sunway TaihuLight. Compared with the algorithm before opitimization, the optimized PhotoNs-2 achieves a speedup of about 24 times under three different arithmetic cases. The proposed optimization scheme can provide a reference for the porting and optimization of other high-performance applications on the Sunway TaihuLight platform.

**Key words**: Sunway TaihuLight platform; parallel optimization; data reorganization; FMM; cosmic N-body

超算平台凭借超强的计算和存储能力,为多个重点研究领域如分子动力学模拟[1]、大气模拟[2]、地球气

**收稿日期**:2024-03-17
**基金项目**:山东省自然科学基金项目(ZR2022MF274;ZR2021LZH004;ZR2023LZH009)
**作者简介**:韩承磊(1999—),男,山东临沂人,硕士研究生,主要从事高性能计算研究.
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、分布式计算等方面研究,本文通信作者.
E-mail:fuyou@sdust.edu.cn

候模拟[3]、大气动力学[4]、流体力学模拟[5]、宇宙学模拟[6]等计算量巨大的数值模拟计算提供了平台支撑[7]。宇宙学模拟是典型的 N 体问题,主要研究粒子之间的相互作用和运动规律,是高性能计算中具有代表性和挑战性的问题之一。理论上,任意两个粒子之间作用力的计算时间复杂度为 $O(n^2)$。为了提高计算效率,1980 年,Peebles 通过引入非碰撞的波尔兹曼方程,给出了牛顿力学在宇宙学中的近似形式,并以此发展出树算法[8]和粒子网格(particle-mesh,PM)算法[9]。PM 算法将空间网格离散,通过快速傅里叶变换求解泊松方程得到作用力,算法时间复杂度为 $O(n\log n)$,但在计算近程引力时不精确,只适用于计算远程引力。1987 年,耶鲁大学的 Greengard 提出快速多极子方法[10](fast multipole method,FMM),使用树形结构对计算空间进行多层组划分,并利用多极扩展来扩展系统格林函数,将靠近的源作为一个整体以加快 N 体问题中远程引力的计算,该算法兼具精度和速度优势,将粒子引力计算的时间复杂度降低为 $O(n)$。近年来,中国科学院国家天文台提出一种 PM 算法和 FMM 相结合的混合数值算法[11],并开发了 PhotoNs-2 软件,进一步提高了计算效率,能够处理超大粒子数的问题。

随着宇宙学模拟实验计算规模的不断增长,仅从算法角度降低时间复杂度已不能满足模拟实验对大规模计算的要求,而并行化研究可以通过挖掘指令级并行性、数据级并行性以及任务级并行性等,协同多个处理单元进一步提高程序执行效率。随着图像处理单元(graphic processing unit,GPU)的出现,研究者开始使用 GPU 并行加速 FMM。2011 年,李正杰等[12]针对 FMM 在 GPU 上存在的负载不均衡和计算规模受显存大小限制等问题,提出一种新的基于统一计算设备架构平台的实现方法。2014 年,Dang 等[13]提出一种 FMM 与快速傅里叶变换相结合求解大规模电磁问题的方法,在 Nvidia Tesla M2090 GPU 集群上实现了 13 个节点的并行。Wang 等[14]在 GPU 上对 PhotoNs-2 进行优化,针对 FMM 中的 P2P 模块给出了任务划分策略。扶月月等[15]通过优化 FMM 的粒子包参数,减少 CPU 与 GPU 之间的数据传输。近年来,为实现国产超算平台上软件的自主可控,并进一步丰富国产超算软件,研究者们在各种国产超算平台上开展了大规模宇宙学模拟计算。由于体系结构不同,基于 GPU 超算平台提出的优化方法不能直接移植到国产平台上。刘旭等[16]基于 PhotoNs-2 设计了神威·太湖之光平台上的 N 体模拟软件 SwPHoToNs,但仍存在消息传递接口(message passing interface,MPI)通信冗余及访存效率不高等问题。

针对以上问题,在神威·太湖之光平台上完成 PhotoNs-2 的移植后,结合平台体系结构特点,本研究对其中的计算主体——FMM 进行优化:针对在从核(computing processing elements,CPE)上访存不连续问题,通过重整 P2P 计算数据和改进 P2P 的遍历算法,提高在从核上访存的连续性;针对计算效率低下问题,设计双缓冲[17],实现计算与通信的重叠;针对核心算子中超越函数数据获取时主存访问开销较高问题,将其由查表法重构为本地计算,减少从核对主存的访问;针对 MPI 通信耗时随进程规模增大而增加问题,提出合并 K-D 树[18]算法,在不过多占用内存空间的同时减少进程间的通信耗时。相较于优化前,优化后的 PhotoNs-2 在 3 个不同规模算例上的测试均取得约 24 倍的加速效果。本研究提出的优化方案可以为其他高性能应用在神威·太湖之光平台上的移植与优化提供参考。

## 1 神威·太湖之光平台处理器架构和 FMM-PM 混合算法

本节对神威·太湖之光采用的 SW26010 异构众核处理器架构和 PhotoNs-2 软件中的 FMM-PM 混合算法进行介绍。

### 1.1 SW26010 异构众核处理器架构

SW26010 异构众核处理器芯片集成了 4 个核组,共 260 个计算核心,每个核心的工作频率为 1.45 GHz。每个核组包含 1 个主核(management processing element,MPE)、64 个以 8×8 阵列方式排布的从核和 1 个存储控制器(memory controller,MC),每个核组通过 MC 与主存相连,如图 1 所示。

MPE 拥有 L1 和 L2 两级高速缓冲存储器(Cache),其中 L1 Cache 由 32 kB 的数据 Cache 和 32 kB 的指令 Cache 组成,L2 Cache 的大小为 256 kB。每个 CPE 拥有 16 kB 的 L1 指令 Cache 和 64 kB 局部设备内存(local device memory,LDM)。CPE 访问主存数据的方式有两种:一种是全局内存访问,即全局读入/写出(global load/store,gld/gst)离散访问,从核通过全局读入和全局写出指令直接对主存中的数据进行读写操

作,该方式实现简单,但是访问延迟较高;另一种是直接内存访问(direct memory access, DMA)方式,先将内存中的数据传输至 LDM,再访问 LDM 获取内存数据,该方式的访问延迟远低于前者。

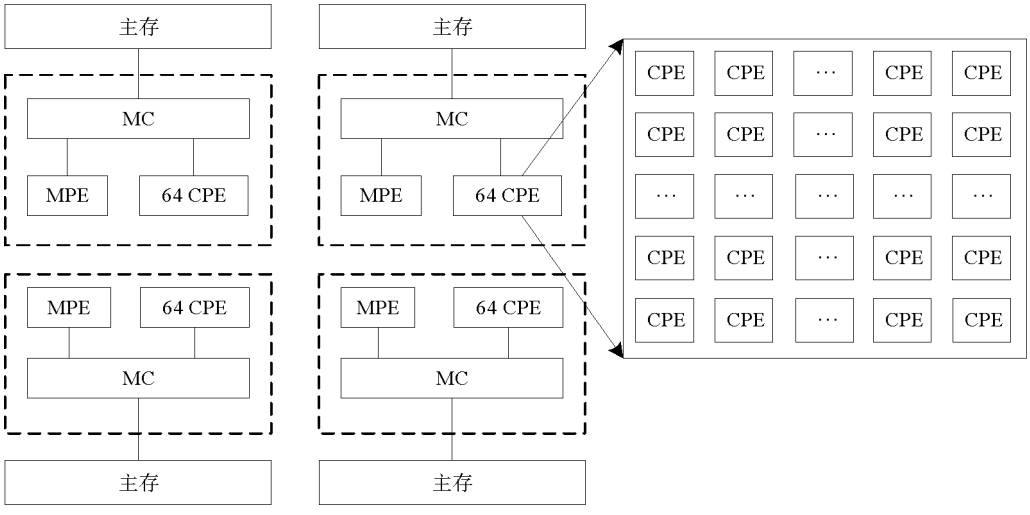


**图 1 SW26010 硬件架构与从核阵列结构**

Fig. 1 SW26010 hardware architecture and slave array structure

### 1.2 PhotoNs-2 软件的 FMM-PM 混合算法

图 2 为 FMM-PM 混合算法的示意图。在宇宙模拟的 N 体问题中,使用 PM 的网格划分和 FMM 的 K-D 树划分两种方式对整个系统进行划分。因此,对于单个粒子而言,所受力分为两个分量:一是 PM 算法计算出的远程力(图 2 中虚线外的力),二是 FMM 计算出的近程力(图 2 中虚线内的力),两部分计算相对独立。

PM 算法根据粒子的分布信息得到均匀网格上的密度分布函数,首先通过快速傅里叶变换方法计算出网格点的势,然后通过有限差分和插值法计算每个网格点处的力。由于整个模拟空间中的网格点数较少,所以 PM 算法部分计算速度较快,耗时仅占整体运行时间的 1%。

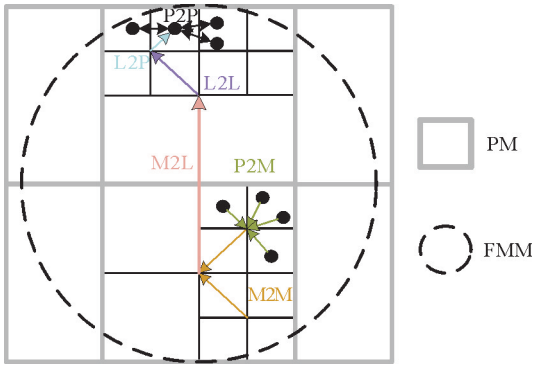


**图 2 FMM-PM 混合算法的示意图**

Fig. 2 Schematic diagram of the hybrid FMM-PM algorithm

FMM 在计算时会将数据分为近程域(相邻的网格)和远程域(不相邻的网格),该方法主要包含 6 个部分:①P2M。粒子多级扩展,将叶节点空间域内的粒子影响力汇聚成空间域整体的泰勒级数多级展开(multipole expansion,ME);②M2M。下层粒子作用力求和,将下一层已经计算出的 ME 汇集到上一层节点上,即父节点 ME 是其所有子节点 ME 的和;③M2L。将 ME 转化为泰勒级数局部扩展(local expansion,LE)的过程,该过程遍历树节点,集中所有交互节点的 ME 生成树节点的 LE;④L2L。对上层 LE 进行累加生成底层叶节点 LE 的过程,上层节点传递 LE 到下层节点 LE;⑤L2P。将 LE 转换成对粒子的作用力,底层叶节点根据 LE 计算远程域内其他节点内粒子对自身粒子产生的作用力;⑥P2P。计算近程域内粒子的作用域,计算同一叶节点内和邻叶节点间的粒子作用力。由于 P2P 部分需要对当前粒子与网格中的其他粒子进行逐一计算,所以主导了 FMM 的计算时间。

## 2 FMM 热点分析及并行优化思路

完成 PhotoNs-2 在 SW26010 处理器的主核移植后,对 FMM 的热点进行测试。

### 2.1 热点分析

为分析不同规模下的程序性能瓶颈以及便于扩展程序规模,将模拟系统空间三维边长均设置为 100 000 个单位长度。每个进程需要处理的粒子数固定为 32 768(即 $2^{15}$)。设进程数为 $N_{\text{proc}}$,则该空间中包含的天体粒子总数可以表示为 $32\ 768 \times N_{\text{proc}}$。表 1 为 3 个不同计算规模算例下的热点测试结果。

**表 1 不同进程规模下的热点测试**

Table 1 Hotspot test in different process scales

| 算例 | 进程数 | 粒子总数 | P2P 计算耗时占比/% | | | MPI 通信耗时占比/% |
|---|---|---|---|---|---|---|
| | | | erfc( ) | exp( ) | P2P 核心 | |
| 算例 1 | 8 | 262 144 | 34.81 | 17.00 | 16.03 | 12.72 |
| 算例 2 | 64 | 2 097 152 | 25.02 | 12.22 | 11.64 | 23.36 |
| 算例 3 | 512 | 16 777 216 | 13.96 | 6.80 | 6.49 | 32.39 |

由表 1 可知,在不同进程规模下 PhotoNs-2 软件程序热点分布不同。当进程规模较小时,程序热点是 P2P 计算部分,占总耗时的 67.84%,主要包含 P2P 核心计算与 exp( )、erfc( )两个超越函数计算,其中

$$\operatorname{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_{x}^{\infty}\mathrm{e}^{-t^2}\,\mathrm{d}t \text{ 。} \tag{1}$$

随进程数增加,P2P 计算耗时占比逐渐减少,进程间的通信耗时占比逐渐增加。当进程数增加到 512 时,进程间的通信耗时占比为 32.39%,超过了 P2P 计算耗时,成为程序热点。

因此,在神威·太湖之光平台上对 FMM 进行性能优化的重点,除了提高 P2P 部分的计算性能外,还需要降低 MPI 通信部分的开销。

### 2.2 FMM 并行优化思路

在 SW26010 处理器上对 FMM 的并行优化具体从以下 4 个方面进行。

1) PhotoNs-2 中的粒子数据存储在连续的空间,但粒子数据的索引与 P2P 计算过程中对于粒子数据的遍历均基于 K-D 树,每个粒子的遍历均需要以 gld/gst 方式进行两次非连续访存,访存效率低下。本研究一方面将主核中的 P2P 核心计算部分从 K-D 树的遍历中分离,交由从核完成,提高主从核的并行性,另一方面通过数据整理提高访存效率。

2) P2P 计算部分包含 exp( )和 erfc( )两个超越函数。神威·太湖之光平台上的数学库采用查表法计算超越函数值,因从核的存储空间小,只能将数据表置于主存中,并通过离散访存获取函数值,大大拖慢计算效率。因此,需要舍弃查表方式,以从核计算代替访存。

3) DMA 通信机制可以将 CPE 计算所需数据批量读取到 LDM 空间,但 CPE 需要等待数据传输完毕才能开始计算,需要设计 DMA 双缓冲通信机制,实现计算与数据传输的重叠。

4) PhotoNs-2 在模拟过程中为系统设置周期性边界条件,导致进程间的通信耗时占比随进程规模的扩大而急剧增加,需要从减少进程间通信次数的角度对 MPI 通信部分进行优化。

## 3 神威·太湖之光平台上的 FMM 并行优化

SW26010 处理器独特的 MPE+CPE 阵列架构与 CPU+GPU 结构不同,导致 GPU 上已有的并行与优化方法无法直接应用于 SW26010 处理器,需要根据 SW26010 的架构特点重新设计。

### 3.1 FMM 的 P2P 计算数据重整

FMM 采用多级级数运算来完成粒子间的直接作用力计算,用级数表达区域内粒子的整体作用,以减少

P2P计算。K-D树划分算法在计算近程域作用力时能够精确到单个粒子的作用力,而在计算远程域作用力时用整个空间的级数代表空间内所有粒子的整体作用,且随着粒子间距离的变化,K-D树划分算法可以灵活地选取最适合的粒子空间大小。

在PhotoNs-2中,所有粒子信息以结构体数组的形式存储。为了与K-D树建立映射关系,每次P2P元计算需要获取两个叶节点的信息。由于P2P元计算任务之间的关联度不高,需要的叶节点数据在原始数组中很难出现连续存储的现象,导致在粒子遍历过程中访问数据时出现大量以gld/gst方式离散访问主存,增加了访存开销。本研究在每次执行P2P计算之前对计算数据进行重整,流程如图3所示,先遍历原数组,找到P2P中第一个元计算所需的两个叶节点——叶节点1和15,并将两个节点分别放入新数组1和新数组2中,对剩下的其他元计算均执行上述操作。

数据重整后,在SW26010处理器上对FMM的并行优化主要从以下3方面进行。

1)查找与计算的分离。数据重新整理前,FMM通过遍历K-D树实现对近程域内的所有粒子叶节点查找和计算,查找和计算过程之间存在数据依赖关系。而在数据重新整理时,在完成全部粒子的K-D树遍历过程中实现了叶节点查找,而P2P核心计算在之后执行,实现了叶节点查找与P2P核心计算的分离:MPE负责数据的整理和K-D树遍历,CPE负责P2P计算。

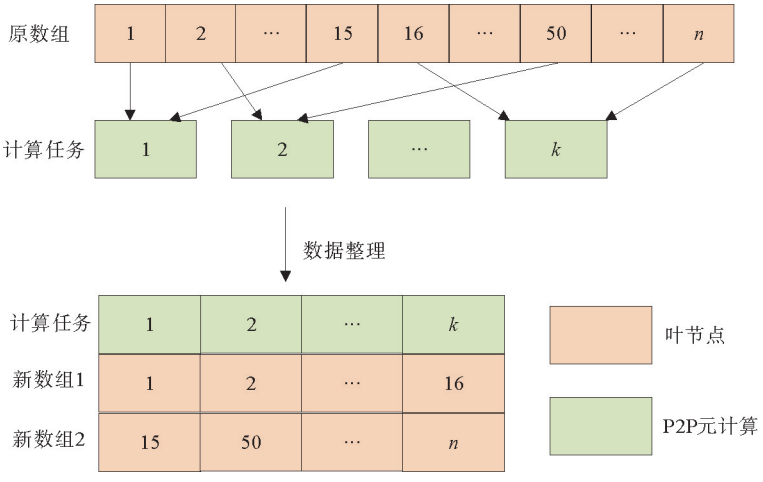


**图3 P2P数据整理流程**

Fig. 3 Process of the P2P data collation

2)CPE的任务划分。SW26010处理器单个CPE的LDM大小仅为64 kB,因此必须将P2P计算所需要的数据进行分块。MPE在整理数据的过程中,获取了整体任务规模,使得在实现CPE任务划分时更容易实现负载平衡。假设每个叶节点的最大粒子数为$L_{max_leaf}$,由于每个粒子需要使用三维坐标表示位置信息,所以输入的两个叶节点需要占用$2\times L_{max_leaf}\times 3$个双浮点数据类型的空间,因此每次通信可以传输的P2P元计算任务数为$S_{task}=\lceil 64\times 1\ 024/(2\times L_{max_leaf}\times 3\times 8)\rceil=\lceil 4\ 096/(3\times L_{max_leaf})\rceil$。将整个P2P计算任务划分为大小为$S_{task}$的任务块后,平均分给64个CPE执行,不能整除的任务块依次按从核编号进行分配,实现CPE负载的相对均衡。

3)LDM的空间复用。通过对原始的叶节点数组进行数据重整,将计算所需要的叶节点存放在相邻位置,可实现P2P计算中对叶节点数据的连续访问,使CPE能以DMA方式访问主存,从而避免离散访存,减少访存开销。但进一步分析P2P的数据流发现,P2P核心计算在为两个叶节点数组单独申请内存空间后,左叶节点的数组元素在被访问过一次后不再被访问。因此,在左叶节点的数组使用以后,可以将P2P计算结果写入该数组的存储空间,实现LDM空间复用,减少对存储空间不必要的长期占用。这样可节约出LDM空间,从而增加每次从主存读取数据的DMA传输数量、减少DMA通信次数,最终达到提高LDM的空间利用率以及访存效率的目的。

### 3.2 超越函数的计算重构

为了满足计算精度要求并减少计算量,神威·太湖之光平台数学库中的超越函数采用查表方式完成[19]。当P2P计算放置于CPE时,多次调用超越函数产生的查表操作会导致CPE进行频繁的gld/gst访存,成为FMM的性能瓶颈。又因从核的LDM空间有限,将超越函数表放入LDM会减少DMA双缓冲的可用空间,影响数据传输效率。因此,本研究对P2P计算中exp( )和erfc( )两个超越函数进行

重构。

对于 exp( )函数通过在 $x=0$ 处的多级泰勒展开进行近似求解得：

$$\exp(x)=2^k\times\exp(r)\text{，} \tag{2}$$

$$\exp(r)\approx 1+r+r^2/2!+r^3/3!+\cdots+r^{10}/10!\text{ 。} \tag{3}$$

式中，$k=\lfloor x\log_2 e\rfloor$，$r=x-k\log_2 e$。为了进一步减少 exp( )函数运算，泰勒展开式中幂函数 $2^k$ 的运算利用移位操作来完成。

对于 erfc( )函数，通过观察可知，当 $x\to+\infty$ 时，计算结果趋近于 0；当 $x\to-\infty$ 时，计算结果趋近于 2。因此可对 erfc( )函数进行近似求解，

$$\operatorname{efrc}(x)=\begin{cases}2, & x<-2;\\ 2-\tau(x), & -2\leqslant x<0;\\ \tau(x), & 0\leqslant x\leqslant 2;\\ 0, & x>2。\end{cases} \tag{4}$$

式中：$\tau(x)=t\times\exp(-x^2+P_0+P_1t+\cdots+P_9t^9)$，$t=1/(1+0.5|x|)$，$P_i$ 为固定数值。

通过对 exp( )和 erfc( )函数重构，可保证在满足计算精度(误差小于 $10^{-14}$)的前提下，减少 CPE 对主存的 gld/gst 次数，同时通过移位操作简化幂函数的计算，提高超越函数的计算效率。

### 3.3 DMA 双缓冲通信机制

经过 3.1 节所述数据重整后，粒子以数组的形式在主存中连续存储，因此 CPE 只需要在计算前获取所需对应数据在主存中的偏移地址，就可以通过 DMA 方式将数据批量读取到 LDM 中，CPE 计算需要在数据传输完成后才能进行。由于 DMA 传输和 CPE 计算使用不同的功能部件，为实现通信与计算的重叠提供了硬件基础。为进一步提高访存效率，本研究基于 DMA 异步传输数据提出 DMA 双缓冲机制，工作流程如图 4 所示。

图 4 的上半部分展示了单缓冲情况下的 P2P 计算与 DMA 访存执行顺序，下半部分展示了双缓冲的 P2P 计算与访存。每次执行 P2P 计算时均需要执行一次 DMA 读取和 DMA 写回，且必须严格按照 DMA 读取、P2P 计算和 DMA 写回顺序执行以确保计算结果的正确性。如果按照正常执行顺序，可以发现 CPE 存在大量空闲时间，而采用双缓冲以后，在执行 P2P 计算的同时执行上一个 P2P 计算结果的写回与下一个 P2P 计算数据的读取，减少了 CPE 的空闲时间，提高了计算效率。

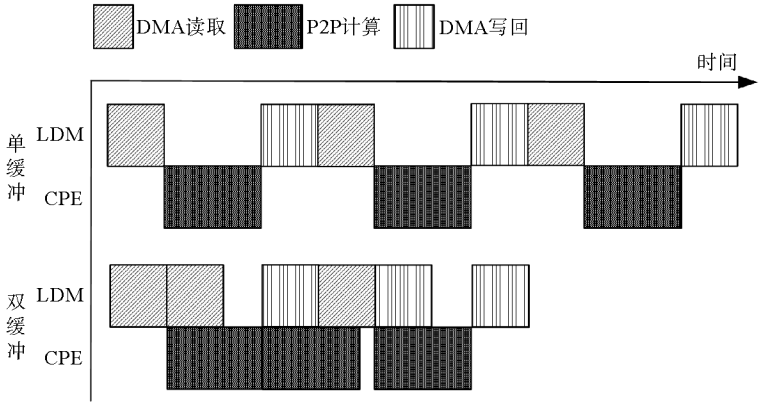


**图 4 DMA 工作流程**

Fig. 4 Working process of DMA

### 3.4 FMM 的通信优化

对比表 1 的不同规模算例发现，P2P 计算耗时的占比会随进程数增加而下降，这是由于随着进程数增加，通信开销在程序整体耗时中的占比增加，因此只针对计算优化带来的性能收益也会随进程数的增加而下降。下面从减少进程间通信次数的角度对 MPI 通信部分进行优化。

PhotoNs-2 软件采用 3d27p 周期性边界条件，其中 3d 表示空间维度为 3，27p 表示每个进程需要与 27 个边界进行通信。进程间通信过程如图 5 所示，为了方便展示，图 5 中只展示了 2d9p 周期性边界条件下的进程间通信，对于源进程而言，其通信对象不仅包含周围边界中的所有进程，还包含当前边界中的其他进程。

当进程数为 $M$ 时，系统内的通信次数为 $27\times M\times(M-1)$ 次。源进程发送给目标进程的 27 棵 K-D 树

均源于同一棵 K-D 树,因此可以将源进程需要发送的 27 棵 K-D 树合并,将合并后的结果发送给目标进程。由于合并后的 K-D 树只包含目标进程所需节点的信息,因此目标进程可以采用相同的方式从合并后的 K-D 树中分离出原 27 棵 K-D 树。通过使用合并后的 K-D 树进行通信,可以将通信次数降低至 $M\times(M-1)$,使得软件在大规模进程并行情况下仍然有很好的加速效果。

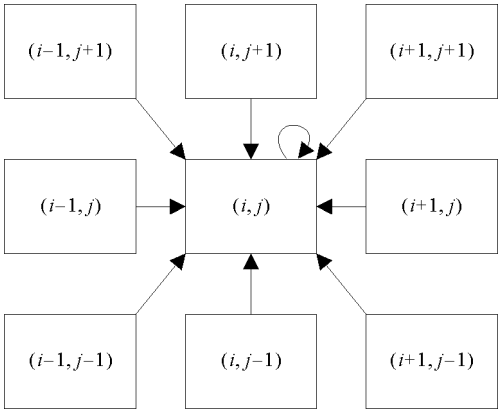


**图 5 二维问题的进程通信示意图**

Fig. 5 Schematic of process communication for a two-dimensional problem

## 4 实验结果与分析

### 4.1 优化后的性能测试

以优化前的算例为基础,表 1 中 3 个不同规模算例经优化后的实验测试结果如表 2 所示。表 2 中,加速比的计算为优化前耗时除以优化后耗时。

**表 2 优化后的程序耗时测试**

Table 2 Time of program after optimization

| 算例 | 优化前/s | 从核以及数据重组优化/s | 超越函数/s | DMA 双缓冲/s | 通信优化/s | 最终加速比 |
|---|---|---|---|---|---|---|
| 算例 1 | 429.48 | 160.58 | 40.36 | 23.45 | 17.90 | 23.99 |
| 算例 2 | 571.02 | 181.83 | 57.64 | 29.70 | 21.42 | 26.66 |
| 算例 3 | 900.53 | 215.39 | 80.92 | 40.61 | 28.62 | 31.47 |

由表 2 可知,优化方案最终在 SW26010 处理器上取得约 24 倍的加速比。其中,数据重整避免了对主存数据的离散访存,使得程序的计算核心可以使用从核进行运算;超越函数的计算重构减少了 CPE 对主存的访问次数,加快了程序的运行速度;DMA 双缓冲减少了 CPE 等待主存数据传输的时间,进一步提高了 CPE 的计算效率;合并 K-D 树减少了进程间的通信次数,缓解多进程环境下的性能瓶颈。

### 4.2 可扩展性测试

为检测优化方法的可扩展性,设计了可扩展性实验,由于测试平台最大可申请核组数为 512,每个核组的计算核心数为 65,因此可进行测试的最大并行核心数为 33 280。

实验中的系统空间三维边长为 100 000,进行强可扩展性实验时,模拟粒子的总数为 524 288,以源程序的 8 进程执行耗时为基准时间,加速比分别为优化后的 8、16、32、64、512 进程除以基准时间;进行弱可扩展性实验时,每个进程中的粒子数为 32 768,实验结果如图 6 所示。在强可扩展性实验中,在进程数较少的情况下,加速比随进程数增加而接近于线性增长,说明本研究的优化方案具有良好的强可扩展性。但当进程数较多时,加速比的变化幅度会随进程数增加而逐渐降低,尤其是当进程数由 64 增加到 512 时,加速比只增加了 1.65 倍,这是由于程序中的计算时间占比会随着进程数增加而降低,导致计算优化带来的性能收益逐渐降低。在弱可扩展性实验中,随着进程数增加,加速比稳定在 24～32,证明优化方法具有很好的弱扩展性。其中,在 512 进程下加速比较高的原因是随进程数增多通信开销占比增大,而针对 MPI 通信的优化会随进程数增多取得更明显的加速效果。

实验结果表明,本研究所提优化方法在不同进程数下加速效果基本稳定,理论上适用于所有基于 FMM 的宇宙学模拟软件优化。

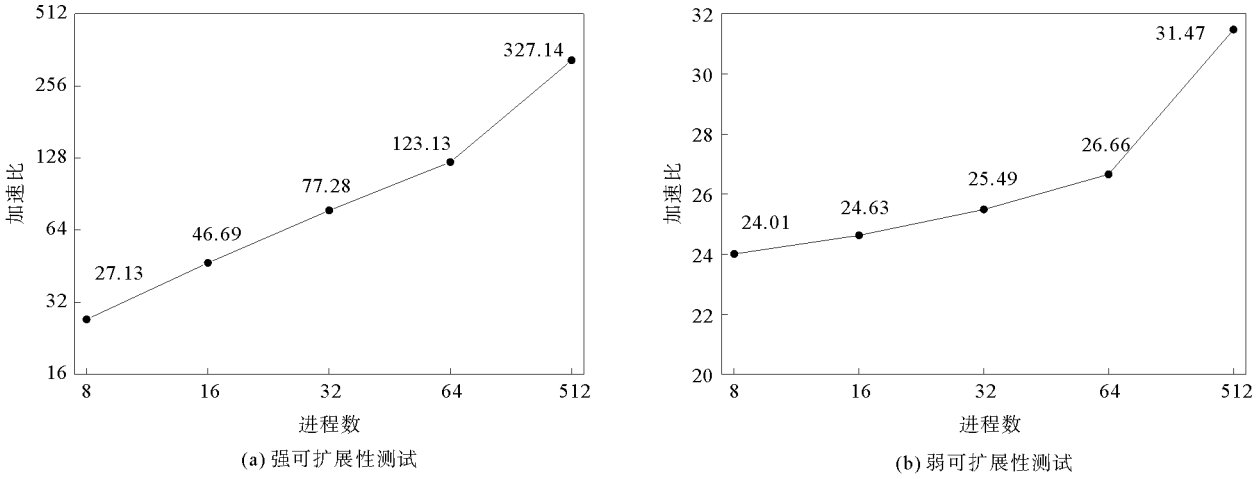


**图 6 强弱可扩展性测试**

Fig. 6 Strong and weak scalability test

## 5 结论

本研究针对 SW26010 处理器架构,首先将 PhotoNs-2 软件中 FMM 的 P2P 计算部分移植到 CPE,然后针对该算法中的访存、超越函数计算以及 DMA 通信进行优化,实现了 CPE 的高效利用,最后通过合并 K-D 树的方式减少了 MPI 进程间的数据通信次数。下一步可针对进程间负载不均衡的情况进行深入研究。此外,新一代神威平台——神威·海洋之光提供了新的通信机制,即 CPE 间的 RMA 通信,为 CPE 提供了更加灵活的通信方式,可以基于此设计更高效的任务并行模式。

**参考文献:**

[1] KRIEGER E, VRIEND G. New ways to boost molecular dynamics simulations[J]. Journal of Computational Chemistry, 2015, 36(13): 996-1007.

[2] COLLINS W D, RASCH P J, BOVILLE B A, et al. The formulation and atmospheric simulation of the community atmosphere model version 3(CAM3)[J]. Journal of Climate, 2005, 19(11): 2144-2161.

[3] LEHNER F, JOOS F, RAIBLE C C, et al. Climate and carbon cycle dynamics in a CESM simulation from 850 to 2100 CE[J]. Earth System Dynamics, 2015, 6(2): 411-434.

[4] 陈道琨,刘芳芳,杨超. 面向新一代神威超级计算机平台的大气动力学问题全隐式求解器研究[J]. 数值计算与计算机应用, 2023, 44(2): 198-213.
CHEN Daokun, LIU Fangfang, YANG Chao. Fully-implicit solver for atmospheric modeling on the next generation Sunway supercomputers[J]. Journal on Numerical Methods and Computer Applications, 2023, 44(2): 198-213.

[5] 柳安军,殷洪辉,王利,等. 基于新一代神威超算的计算流体力学 Palabos 软件的并行优化[J]. 计算机科学, 2022, 49(10): 66-73.
LIU Anjun, YIN Honghui, WANG Li, et al. Parallel optimization of computational fluid dynamics application Palabos based on next generation Sunway supercomputer[J]. Computer Science, 2022, 49(10): 66-73.

[6] JACK R, CARLOS F, ADRIAN J, et al. A high-resolution cosmological simulation of a strong gravitational lens[J]. Monthly Notices of the Royal Astronomical Society, 2021, 501(3): 4657-4668.

[7] SUKHINOV A I, BELOVA Y V, LYAPUNOVA I A, et al. Mathematical modeling of zooplankton productivity in the Azov Sea in summer on a high-performance computer system[J/OL]. Journal of Physics: Conference Series, 2021, 1902(1). DOI: 10.1088/1742-6596/1902/1/012131.

[8] 王武,冯仰德,迟学斌. 树结构在 N 体问题中的应用[J]. 计算机应用研究, 2008(1): 42-44.
WANG Wu, FENG Yangde, CHI Xuebin. Application of tree structures in N-body problem[J]. Application Research of Computers, 2008(1): 42-44.

[9] SINGH J P, HOLT C, TOTSUKA T, et al. Load balancing and data locality in adaptive hierarchical N-Body methods: Barnes-Hut, fast multipole, and radiosity[J]. Journal of Parallel & Distributed Computing, 1995, 27(2): 118-141.

[10] 唐振,张倬,柴亚辉,等. FMM 算法在 Cell/B. E. 处理器上实现的分析与验证[J]. 计算机工程与科学,2011,33(8):79-83.
TANG Zhen, ZHANG Zhuo, CHAI Yahui, et al. Analysis and validation of FMM algorithm based on Cell/B. E. processor [J]. Computer Engineering and Science, 2011, 33(8): 79-83.

[11] WANG Q. A hybrid fast multipole method for cosmological N-body simulations[J]. Research in Astronomy and Astrophysics, 2021, 21(1): 27-34.

[12] 李正杰,徐炜民,柴亚辉,等. FMM 算法中 PP 问题在 GPU 上的研究与实现[J]. 计算机工程与设计,2011,32(9):3050-3053.
LI Zhengjie, XU Weimin, CHAI Yahui, et al. Research and implementation of PP problem in FMM algorithm on GPU[J]. Computer Engineering and Design, 2011, 32(9): 3050-3053.

[13] DANG V, NGUYEN Q M, KILIC O. GPU cluster implementation of FMM-FFT for large-scale electromagnetic problems [J]. IEEE Antennas and Wireless Propagation Letters, 2014, 13: 1259-1262.

[14] WANG Q, MENG C. PhotoNs-GPU: A GPU accelerated cosmological simulation code[J]. Research in Astronomy and Astrophysics, 2021, 21(11): 106-112.

[15] 扶月月,王武,王乔. 基于 FMM-PM 方法的宇宙 N 体模拟在 GPU 上的实现和优化[J]. 数据与计算发展前沿,2020,2(2):155-164.
FU Yueyue, WANG Wu, WANG Qiao. The implementation and optimization of cosmological N-body simulation by FMM-PM method on GPUs[J]. Frontiers of Data & Computing, 2020, 2(2): 155-164.

[16] 刘旭,张曦煌,刘钊,等. 基于神威太湖之光的宇宙学多体模拟[J]. 计算机工程,2020,46(9):35-43.
LIU Xu, ZHANG Xihuang, LIU Zhao, et al. Cosmological multi-body simulation based on Sunway TaihuLight[J]. Computer Engineering, 2020, 46(9): 35-43.

[17] 周倩,梁建国,傅游. 神威·太湖之光上排列熵算法异构并行加速[J]. 计算机工程与设计,2023,44(2):400-406.
ZHOU Qian, LIANG Jianguo, FU You. Heterogeneous parallel acceleration of permutation entropy algorithm on Shenwei Taihu Light[J]. Computer Engineering and Design, 2023, 44(2): 400-406.

[18] BENTLEY J L. Multidimensional binary search trees used for associative searching[J]. Communications of the ACM, 1975, 18(9): 509-517.

[19] 朱文强,傅游,梁建国,等. Silicon-Crystal 应用在 SW26010 处理器上的移植与优化[J]. 小型微型计算机系统,2021,42(6):1313-1320.
ZHU Wenqiang, FU You, LIANG Jianguo, et al. Porting and optimizing of silicon-crystal application on SW26010 processor[J]. Journal of Chinese Computer Systems, 2021, 42(6): 1313-1320.

(责任编辑:齐敏华)