基于OpenMP的磁壳参数 L 值并行计算研究

盛红佼^{1,2}, 张贤国^{1,3,4*}, 朱昌波^{1,3,4}

1. 中国科学院国家空间科学中心, 北京 100190;

2. 中国科学院大学, 北京 100049;

3. 天基空间环境探测北京市重点实验室, 北京 100190;

4. 中国科学院空间环境态势感知技术重点实验室, 北京 100190

* E-mail: zhangxg@nssc.ac.cn

收稿日期: 2019-11-20; 接受日期: 2020-01-18; 网络版发表日期: 2020-06-12

空间科学先导专项(编号: XDA15017100)和空间科学国家实验室培育项目“太阳风暴探测关键技术研究”(编号: Z191100004319003)资助

摘要 辐射带研究中 (L, B) 坐标系把三维问题降为二维问题, 大大提升了辐射带三维空间有限观测数据的利用率。 L 值是其中重要的参量之一, 在粒子特性分析、航天器轨道辐射环境描述以及辐射带建模等研究领域应用十分广泛。 L 值由磁力线追踪积分计算而得, 但地球磁场位型的复杂性导致磁力线追踪步长严格受限, 造成单次追踪计算步数多。此外, 不同空间位置处磁力线追踪路径和追踪距离随时空变化差异显著, 造成多次 L 值计算之间相互独立、复杂度不一且无法简并, 计算速度慢、效率低。为了提高 L 值的计算效率, 本文采用OpenMP对目前广泛应用的 L 值计算程序实现单机多线程并行化, 并分析了并行程序的加速比和效率。结果表明, 并行后 L 值计算效率明显提高, 当线程数量设置与系统最大线程数相等时(本文所用计算机系统线程数最大为8), 并行效果最优, 加速比范围为1.57~2.81, 并行效率范围为19.63%~35.13%。计算过程中发现, 线程间负载不平衡是影响并行效果的主要因素。为了提高并行加速比和效率, 本文提出了一种表征 L 值计算复杂度的排序因子, 并基于此提出了排序和重分配(“sort”和“S”分配)的优化负载算法, 对dynamic, sort+“S”, sort+dynamic, sort+“S”+dynamic四种方式优化负载不平衡的效果进行了比对分析, 结果表明sort+dynamic的效果最优, 加速比范围为2.33~3.30, 效率为29.13%~41.25%, 优化效率提升了6.12%~9.50%。本文研究表明, OpenMP可以成功实现 L 值计算程序的并行计算, 并提高 L 值的计算效率, 针对负载不平衡的问题, 采用sort+dynamic方式调节可获得最好的并行加速比和效率, 为空间物理或其他领域类似大数据分析研究中效率提升提供了有益探索。

关键词 磁壳参数 L 值, 并行计算, OpenMP, 并行加速比, 并行效率, 并行优化

1 引言

在日地空间环境中, 磁层是太阳活动与地球相互作用的主要区域, 是物质和能量交换的重要场所。辐

射带是磁层的核心区域之一, 是磁层俘获高能粒子的主要聚集地, 这些高能粒子既是日地相互作用过程物质能量的载体, 也是威胁航天器安全和效能的重要因素。所以, 辐射带既是空间科学研究的重要领域又是

引用格式: 盛红佼, 张贤国, 朱昌波. 基于OpenMP的磁壳参数 L 值并行计算研究. 中国科学: 技术科学, 2021, 51: 356–370

Sheng H J, Zhang X G, Zhu C B. Research on parallel computing of magnetic L -shell based on OpenMP (in Chinese). Sci Sin Tech, 2021, 51: 356–370, doi: 10.1360/SST-2019-0405

地球轨道卫星和航天器保障的巨大挑战^[1]。

对于时间变化尺度远大于粒子漂移周期的缓变磁场环境, 粒子运动过程漂移壳不变, 并且粒子扩散作用会消除粒子通量等在同一壳层不同位置处的差异。所以, 在辐射带研究中, 把数据组织在 (L, B) 坐标系上可实现把三维问题降为二维问题, 大大提升了辐射带三维空间有限观测数据的利用率。 L 值是其中重要的参量之一, 在粒子特性分析、航天器轨道辐射环境描述以及辐射带建模等研究领域应用十分广泛。

在科学研究方面, L 值作为数据组织和理论研究的坐标系, 在描述粒子的通量分布、加速机制、扩散特征、波粒相互作用机制等数据分析和理论研究中发挥着重要的作用。太阳剧烈活动与地球磁层相互作用, 发生磁暴或亚暴, 辐射带粒子通量具有动态变化特征和区域依赖性。对于区域分布显著的高能粒子, 利用 L 值分区可以突出共性特征、减少干扰误导, 简化通量演化特征以及不同区域的加速加热机制研究。不同的 L 值区域, 粒子通量分布存在显著差异。在 $L < 1.7$ 区域, 高能质子通量相对稳定, $L > 1.7$ 尤其在 $L = 2$ 时通量变化极为显著^[2]。几十keV的电子增强而后迅速减弱的区域为 $2.5 \leq L \leq 6$; 几百keV的电子增强而后逐渐减弱的区域为 $3 \leq L \leq 4$; 大于1 MeV的电子增强占主导^[3]的区域为 $3.5 \leq L \leq 6$, 尤其在 $L = 4.2$ 和 $L = 6.6$ 时电子通量变化极为显著^[4]。行星际激波与磁层相互作用能够在内磁层($L = 1 \sim 8$)激发ULF波, 激发的ULF波与粒子相互作用, 造成电子快速加速, 环向模ULF波在 L 值较大的区域起主导作用, 极向模ULF波在 L 值较小的区域起主导作用^[5]。除了ULF波, 哨声波、合声波、嘶声波、电磁波等发生波粒相互作用, 驱动粒子加速、损失和扩散。不同类型的波在波粒相互作用过程中, 存在明显的区域分布特征。在 $L = 3.5 \sim 5.5$ 区域, 合声波与电子相互作用导致电子加速^[6]。磁声波在等离子体层($L = 4.5$)内侧和外侧分别能加速几keV的质子和几十keV质子^[7]。在 $L = 3 \sim 6$ 区域, 哨声波和电磁波共同作用导致电子损失^[8], 其中, $L = 5 \sim 6$ 时, 电磁波的存在与电子的分布密切相关^[9]。在 $L = 3 \sim 6$ 区域, 电磁波对环电流电子的扩散具有 L 值依赖性, 随着 L 值的增大, 扩散系数不断降低^[10]。综上可得, L 值是粒子运动规律和波粒相互作用机制等研究的重要参数。

在航天应用方面, L 值作为组织坐标参数可以降维数据、简化计算过程, 在模型研建和模拟计算中都发

挥着重要的作用。辐射带俘获的高能粒子会造成航天器单粒子、总剂量、深层充放电、位移损伤等效应, 影响航天器的正常工作, 威胁航天员健康。为了有效减少和规避这些危害, 深入了解辐射带环境, 建立准确的模型, 对辐射带环境进行模拟预测, 提前做好抗辐射加固工作, 是非常有必要的。但在辐射带模型构建过程中, 数据不足是一大难点, 采用 (L, B) 坐标可以将三维空间中有限的数据投射至二维空间中, 增大数据的利用率, 降低建模难度。在模型计算过程中, 由于辐射带粒子的动态变化特征对 L 值具有强依赖性, 为了能够准确描述辐射带环境, 需要根据不同能谱段和不同 L 位置进行模拟计算。以AP-8/AE-8为例, 因辐射带质子/电子分布具有区域分布显著的特征, 不同区域建模需要应用不同理论或计算公式, 所以, 在建模过程中用 L 值将计算区域分为内区、外区及过渡区(AP-8将区间分为 $L = 1.5 \sim 6.6$ 和 $L = 6.6 \sim 10$ 两部分; AE-8将区间分为 $L = 1.2 \sim 2.4$, $L = 3.0 \sim 11$ 及 $L = 2.4 \sim 3.0$ 三部分), 便于模型计算不同区域的粒子通量^[11,12]。此外, 在大部分建模基本公式中, L 值都作为自变量出现, 是计算粒子的能量积分谱的关键输入参数。在模型计算过程中, 由于建模坐标系为 (L, B) 坐标系, 则三维空间任意点辐射粒子通量的计算都必须先把空间坐标转换为 (L, B) 坐标, 此时 L 值承担了空间坐标转换桥梁的作用, 将建立在二维空间坐标基础上的模型可以任意拓展到三维空间进行三维空间任意位置粒子通量的计算。

综上所述, 在科学研究中, 采用 L 值可以直观简洁地描述辐射带粒子不同的运动现象; 在应用中, 采用 L 值便于建立辐射带模型, 计算并预测辐射带环境。所以, L 值在科学研究方面和应用方面都发挥着重要的作用。

然而, L 值计算过程耗时的问题一直存在。 L 值于1961年首次被McIlwain^[13]以坐标的形式提出, 通过对第二绝热不变量 I 进行函数变化进而计算获得。尽管该计算方法得到了广大研究者的认可与沿用, 但仍然存在计算过程复杂、结果不够准确的缺憾。由于 L 值是通过磁力线追踪计算而得, 要获得更加准确的 L 值, 就必须准确地描述磁场位形。鉴于上述目的, 研究人员不断寻找更接近真实的地球磁场模型和外磁场模型描述磁场位形。地球主磁场模型包括IGRF模型、Jensen & Cain模型、GSFC模型、POGO模型、IGS模型、AWC模型、MGST模型、USGS1985模型等^[14]。外磁场模型包括Mead-Fairfield, Olson-Pfitzer, Tsyganenko, Hilmer-

Voigt等^[15]。为了准确地计算 L 值, 需要精准的内外磁场模型, 但精准的模型中磁力线位形更复杂, 导致 L 值计算过程中磁力线的追踪更复杂、计算量更大。此外, 复杂的磁场位形导致追踪步长不能太长、严格受限, 不同空间位置处于不同的磁力线上, 计算过程无法简并, 二者共同作用导致 L 值计算缓慢、耗时严重。

另一方面, 计算机软硬件技术的发展为此类大量、复杂的计算任务提出了解决方案。并行计算的思想于1972年被提出, 但仅停留在理论阶段, 并无实际应用^[16]。经过多年不断的发展进步, 不同的并行计算方式被提出, 并行计算在实践中逐步得到拓展应用^[17,18]。但由于对硬件的要求较高, 多半适用于集群计算机, 并且编程复杂、难以实现, 所以不易于用户接受。直至1997年, 一种共享内存的应用程序接口(application programmers interface, API) OpenMP被提出, 可单机实现程序并行, 并且编程简单、简洁实用, 具有较强的可伸缩性和可移植性^[19,20]。鉴于其存在的优势, 被广泛用于不同的研究领域, 包括激光雷达数据分析^[21]、遥感图像处理^[22]、模型优化设计^[23]、卫星实时轨道估计^[24]、机器学习算法优化^[25]等, 但目前尚未发现将其应用于 L 值计算的报道。

本文采用OpenMP的并行方式, 对 L 值的计算程序实现并行化, 提高 L 值的计算效率, 讨论优化并行效率的方法。

2 研究方法

L 值计算程序的源代码取自国际辐射带环境模型库¹⁾(International Radiation Belt Environment Model, IRBEM-LIB)。从IRBEM-LIB中提取部分源代码, 生成全新的 L 值计算程序。结合 L 值的计算原理, 分析得出程序具有实现并行的条件。采用OpenMP对程序实现并行计算, 以串行、并行计算的结果和耗时差别分析并行后程序的准确性和效率。

2.1 L 值计算原理

早期人们用纯偶极场模型研究辐射带高能粒子在磁场中的运动, 提出了 L 值的概念, 表示粒子所在磁力线与赤道平面交点的地心距离与地球半径的比值。随

着观测数据增多, 人们发现地球磁场并不是纯偶极场, 为解决这一问题, McIlwain^[13]提出了 L_m 的概念, 由非偶极地磁场模型(如IGRF)获得的磁场值代入偶极场中 L 值与第二绝热不变量 I 的关系式计算获得, 并证明了 L_m 沿着磁力线基本守恒。 L_m 仍然可以被近似地认为是磁力线与赤道面交点的地心距离^[26]。

随着观测数据的进一步增多, 能够获取辐射带更远处粒子投掷角的分布和微分能谱信息, 磁场外部的非对称时变源不能再被忽略。为了计算扭曲磁场下的漂移壳, Roederer和Lejosne^[26]提出了 L^* , 采用与McIlwain类似的思路, 利用更接近实际的磁场模型和 L 值与第三绝热不变量之间的关系式计算 L^* , 由于 L^* 的计算过程所用磁场模型不仅考虑了地球磁场, 同时考虑了日地作用诱发的外磁场, 因此 L^* 与 L_m 不同, 被认为是与日地系统关联并且可以适用于外磁场缓变情况的^[26]。

本文中用到的 L 值均指 L_m , 其计算表达式见式(1)~(3)。程序中, 通过式(1)追踪磁力线计算积分不变量 I , 并根据式(3)计算近似值, 从而计算磁壳参数 L 值。

McIlwain^[13]定义积分不变量 I , 通过追踪磁力线积分计算而得, 如式(1)所示:

$$I = \int_A^{A'} \left(1 - \frac{B_t}{B} \right)^{\frac{1}{2}} ds, \quad (1)$$

其中, A 为空间中任意磁镜点, A' 为 A 点的共轭点, B_t 为磁力线追踪过程中各点的磁场强度, B 为 A 点处的磁场强度, ds 为 A 点至 A' 点沿磁力线的积分路径。

在偶极子场中, 磁壳参数 L 值由 I 和 B 的函数关系变换而得^[13], 如式(2)所示:

$$\frac{L^3 B}{M} = F \left(\frac{I^3 B}{M} \right), \quad (2)$$

其中, M 为磁矩($M=8.07 \times 10^{25}$ G), F 为函数变换关系。

对于函数变换关系 F , Hilton^[27]根据镜点粒子的特殊纬度值($0, \pi/2$)假设, 提出了近似式(3), 由此计算的 L 值。

$$\begin{aligned} \frac{L^3 B}{M} = & 1 + 1.35047 \left(\frac{I^3 B}{M} \right)^{1/3} + 0.465376 \left(\frac{I^3 B}{M} \right)^{2/3} \\ & + 0.0475455 \frac{I^3 B}{M}. \end{aligned} \quad (3)$$

1) <https://craterre.onera.fr/prbem/irbem/description.html>

IRBEM-LIB是辐射带环境建模小组(COSPAR Panel for Radiation Belt Environment Modeling, PRBEM)免费发布的一个库, 内含用于辐射带模型的源代码. 本文从IRBEM-LIB中提取部分源代码生成全新的 L 值计算程序, 计算流程如图1所示. 输入待算 L 值的时刻和三维空间坐标, 利用输入参数对计算环境进行初始化, 包括地磁场模型初始化、外磁场模型初始化、坐标转换常数初始化等. 接下来调用不同模块计算输入位置处的磁场 B , 并根据磁场位形复杂程度计算追踪步长 Δs . 以 Δs 为步长沿磁力线追踪计算, 在输入位置点沿磁力线正负两方向各追踪一步并计算磁场 B_1 和 B_2 , 并判断 B_1 、 B_2 和 B 之间的大小关系. 若 B 最小, 利用 B_1 、 B_2 和 B 计算 $B_{\min}$ (其中, 流程图中 aa 、 bb 和 $s_{\min}$ 均为计算 $B_{\min}$ 的中间

量), 进而直接计算 L 值. 否则, 向磁场小的方向以 Δs 为步长继续追踪直至 $B_{\min}$ 并计算积分不变量 I , 进而计算 I 的宗量 XY , 由 XY 计算近似值 YY , 从而计算 L 值. 流程图表示追踪一根磁力线计算 L 值的过程, 针对大批量的磁力线, 需要多次执行该流程进行追踪计算. 追踪一根磁力线的过程十分复杂, 追踪大批量磁力线的过程无法简并, 导致计算缓慢、耗时严重. 但追踪大批量磁力线的过程相互独立、互不干扰, 大批量磁力线的追踪计算可以同时进行. 结合 L 值的计算原理, 得知由IRBEM-LIB中源代码生成的全新程序具有并行计算的条件.

2.2 并行方法

OpenMP是一种共享内存的API, 由编译器命令集组成, 包括一套编译指导语句和库函数. 编译指导语句可以将程序中的循环体自动线程化, 实现程序的并行计算.

为了方便描述并行过程, 本文将大批量磁力线的追踪计算过程称为“任务”, 一条磁力线的追踪计算过程称为“子任务”. 计算一个“子任务”需要6个表征磁力线起始位置的参量, 分别是纬度(lat)、经度(lon)、高度(alt)、年(year)、天(day)和时刻(ut), lat变化范围为 $(-90^\circ, 90^\circ)$, lon变化范围为 $(-180^\circ, 180^\circ)$, alt变化范围为 $(800, 32000 \text{ km})$ (其中, 32000 km 约5个地球半径(Re), 目的是研究整个辐射带区域), year变化范围为 $(1965, 2010)$, day变化范围为 $(1, 365)$, ut变化范围为 $(0, 86400 \text{ s})$. 为了使研究结果更具普适性, 本研究中所有参量均以随机数的形式产生, 在各参量的取值范围内, 共生成了 $N(N=2000)$ 组随机数. 每组随机数包含6个参量, 参量均随机、均匀分布, 以lat, lon, alt, year 4个变量为例分析, 其时空分布如图2所示, 由图可知, 参量在取值范围内随机均匀分布, 表明所选参量具有很好的代表性和典型性. N 组随机数即为 N 个复杂度随机排列的“子任务”, 完整计算 N 个“子任务”的过程称为“任务”. 为了保证研究过程中比对基准一致, 本文所有计算结果均基于该“任务”计算而得.

计算“任务”的程序主体部分如图3所示, 以do循环实现“子任务”分解. 在程序并行过程中, 采用编译指导语句“!\$omp parallel do”使得do循环自动线程化, 将“子任务”均分至 N_{thread} 个线程同时计算, 实现程序并行化.

并行计算流程如图4所示, 读取“任务”中大批量

图1 L 值计算流程

Figure 1 Calculation process of L value.

图 2 各“子任务”参量的时空分布图

Figure 2 Temporal and spatial distribution of parameters of each “subtask”.

图 3 计算“任务”程序的主体部分

Figure 3 The main part of the program that calculates the “task”.

“子任务”的输入参量, 根据设置的线程数 $N_{\text{thread}}=8$, 将“子任务”均分至8个线程. 各线程同时追踪计算 L 值, 直至完成被分配的任务, 计算结果均由主线程输出. 过程中选用坐标系为 GDZ , 内磁场模型为 $IGRF$, 外磁场模型分别为Olson-Pfitzer dynamic ($kext=6$), Tsyganenko short ($kext=2$), Mead & Fairfield ($kext=1$), no external field ($kext=0$). 整个计算过程采用OpenMP实现程序的并行化.

本文采用的计算机为4核8线程($N_{\text{thread}}=8$), 内存为4.0 GB, 处理器为Intel(R) Core(TM) i7-4790S CPU. “任务”在Windows 7操作系统下采用Microsoft Visual Studio 2013编译器进行并行计算.

2.3 分析方法

在OpenMP成功实现程序并行化的基础上, 对比分析串行、并行计算的结果和耗时, 计算串行、并行计算的误差和并行加速比、效率, 验证并行结果的准确性和并行的时效性.

2.3.1 并行计算准确性验证

并行计算结果 L_{pm} 和串行计算结果 L_{m} 间的误差表达式见式(4), 误差计算过程中假设 L_{m} 为真值^[28].

$$\delta = \sqrt{\frac{\sum [(L_{\text{pm}} - L_{\text{m}}) / L_{\text{m}}]^2}{N}}, \quad (4)$$

其中, δ 表示误差, N 表示“子任务”数量($N=2000$).

2.3.2 并行加速比和效率分析

在相同初始条件下, 获取串行计算耗时 T_s 和并行耗时 T_p , 计算并行加速比 S 和效率 E , 表达式见式(5)和(6)^[29].

$$S = \frac{T_s}{T_p}, \quad (5)$$

$$E = \frac{S}{N_{\text{thread}}}, \quad (6)$$

图 4 并行计算流程图

Figure 4 Flow chart of parallel computing.

其中, N_{thread} 为设置的线程数.

由上式可知, S 可用来衡量并行程序的性能和效

果; 并行效率 E 是由加速比 S 派生而来的, 表示在并行计算过程中, 每一个线程的利用率, E 越接近 1, 程序并行越优化.

3 计算结果

针对上述 $N(N=2000)$ 组随机数组成的“任务”, 在 4 种 ($k_{\text{ext}}=0, k_{\text{ext}}=1, k_{\text{ext}}=2, k_{\text{ext}}=6$) 不同的外磁场模型下进行并行计算, 并分析其加速比和效率, 结果表明 OpenMP 可以在不改变计算结果的前提下提高计算效率, 但调研分析发现, 加速比和效率仍有提升的空间. 为了提高加速比和效率, 我们提出了新的“子任务”分配方式, 优化了并行程序并进行了分析研究.

3.1 准确性验证

通过分析“任务”串行、并行的计算结果, 得知在不同外磁场模型下二者差异均为 0, 结果如图 5 所示. 图 5(a)~(d) 分别表示 4 种外磁场模型下的结果对比, 图中每个散点表示一个“子任务”算例, 其横坐标表示串行计算 L_m , 纵坐标表示并行计算结果 L_{pm} , δ 表示串行、并行结果间的误差(式(4)). 由图 5 可知, 在不同外磁场模型下, 所有散点完美排列在斜率为 1 的直线上,

图 5 串行、并行计算结果对比. (a) $k_{\text{ext}}=0$; (b) $k_{\text{ext}}=1$; (c) $k_{\text{ext}}=2$; (d) $k_{\text{ext}}=6$ Figure 5 Comparison of series and parallel computing results. (a) $k_{\text{ext}}=0$; (b) $k_{\text{ext}}=1$; (c) $k_{\text{ext}}=2$; (d) $k_{\text{ext}}=6$.

计算结果差异 δ 均为0, 说明OpenMP并行不会改变计算结果的准确性.

3.2 并行性能分析

在并行过程中, 设置线程数为8, 获取串行、并行计算“任务”所需时间, 计算不同外磁场模型下的并行加速比和效率, 结果如图6所示. 图中折线表示同一“任务”在4种外磁场模型下并行计算的加速比和效率, 横坐标表示不同的外磁场模型 k_{ext} , 纵坐标分别表示并行加速比 S 和效率 E . 由图6可知, 加速比范围为1.57~2.81, 效率范围为19.63%~35.13%. 所以, 采用OpenMP并行后 L 值的计算效率明显提高.

当外磁场模型 $k_{ext}=2$ 时, 设置不同的线程数 N_{thread} , 并行加速比和效率如图7所示. 图中曲线表示设置不同线程数时的并行加速比和效率, 横坐标表示设置的线程数 $N_{thread}=1\sim 16$, 图7(a)纵坐标表示不同线程对应的加速比 S , 图7(b)纵坐标表示不同线程对应的并行效率 E . 由图7(a)可知, 随着线程数的增大, 加速比不断增大直至 $N_{thread}=8$ 最大, $N_{thread}>8$ 时加速比有降低的趋势, 即在并行过程中, 当线程数设置为系统最大线程数8时, 并行加速比最大. 但图7(b)中并行效率 E 并没有展示出明显拐点, 可能是由于随着线程数增加, 每个线程负责

图6 “任务”的并行加速比(a)和效率(b)
Figure 6 Parallel speedup (a) and efficiency (b) of “task”.

图7 (网络版彩图)线程数对并行加速比(a)和效率(b)的影响
Figure 7 (Color online) The effect of the number of threads on the parallel speedup (a) and efficiency (b).

的有效计算量降低, 而线程间的通信、等待花费时间比例增大造成的, 表明此并行程序仍有优化的空间.

3.3 负载不平衡分析

在并行计算过程中, 负载不平衡是影响程序并行加速比的关键因素^[30]. OpenMP并行计算时, 默认的数据分配方式为static, 但static的数据分配方式极易造成线程等待的问题, 导致计算过程负载不平衡, 降低并行加速比. static数据的分配方式如图8所示, 根据“子任务”的输入顺序, 将 N 个“子任务”均分至 N_{thread} 个数据块. 在并行计算过程中, 每个线程负责一个数据块, 各线程同时计算, 实现并行.

并行计算时, 获取各“子任务”耗时, 记作 t_i , 累加不同数据块中所有“子任务”的耗时 t_i , 得到该数据块的总耗时 T . 在8个数据块中, 选择总耗时最低的数据块, 以最低的总耗时对所有数据块总耗时进行归一化处理, 归一化耗时 $T=T/T_1$, 归一化结果如图9所示. 由图可知,

图8 static数据分配方式(8线程)
Figure 8 Data allocation method of static (8 threads).

图 9 (网络版彩图)各数据块的总耗时归一化结果

Figure 9 (Color online) Normalized result of total time-consuming of each data block.

8个数据块的总耗时存在明显差别, 导致并行过程间的线程等待, 引起负载不平衡, 增大并行耗时, 降低加速比和效率。

4 并行优化

负载不平衡常用的调节方式为dynamic, 为了考察dynamic的优化性能, 尝试了新的“子任务”分配方式。将dynamic分配方式与新的分配方式组合, 形成dynamic, sort+“S”, sort+“S”+dynamic, sort+dynamic等4种不同的并行任务分配方式。采用不同的分配方式调节负载不平衡问题, 从而提高并行加速比, 优化并行过程。结果表明sort+dynamic分配是最优的调节方式。

4.1 优化方式

根据负载不平衡分析可知, 并行过程中不存在线程等待时, 理论上并行效果最优。所以, 文中采用dynamic, sort+“S”, sort+dynamic, sort+“S”+dynamic等4种方式降低线程等待时间, 从而调节负载不平衡的问题。

4.1.1 Dynamic调节

针对负载不平衡的调节和优化, dynamic是常用的方法。dynamic的调节方式为, 在并行开始时给 N_{thread} 个线程各分配1或几个“子任务”, 当某一线程执行完被分配的“子任务”时, 立刻给其分配下一个或下一组“子任务”, 不用等待其他线程的任务完成情况, 以减少并行过程中线程等待的时间, 从而优化并行过程。由此可

见, 每次分配“子任务”数越少, 等待时间越少, 效率越高, 最大等待时间是最后一批“子任务”之间的耗时差; 反之, 每次分配子任务越多, 则等待的时间越长。当每次分配的“子任务”数为 $2000/N_{\text{thread}}$ 时, 该分配方式等效于static。

4.1.2 Sort+“S”调节

Sort+“S”调节是本研究团队自主提出的分配方式, 是根据L值计算特点按照减少分配次数、等量分配工作量的原则提出的。Sort是指根据“子任务”的复杂度大小, 对“子任务”进行排序, “S”是指对“子任务”实现特殊的“S”分配。Sort+“S”是指先实现“子任务”的排序, 然后再对排序后的“子任务”进行“S”分配。整个sort+“S”的调节过程在static分配之前实现, 目的是为了在static分配后, 尽可能使各数据块的总复杂度相似, 减少并行过程的线程等待, 减少并行计算时间, 增加并行加速比, 优化并行性能。

为了实现“子任务”的排序, 需要找到可以表征“子任务”复杂度的因子, 针对可能的复杂度因子, 进行如下分析。在实际操作中, 以各“子任务”的耗时 t 表征其复杂度是最理想的选择, 但为了得到复杂度排序因子需要将“任务”中所有“子任务”计算一遍, 这显然与本研究提高计算效率的初衷相矛盾。为了获得易于快速计算、复杂度表征合理的排序因子, 统计分析了“子任务”耗时 t 随磁纬度(mlat)、纬度(lat)、经度(lon)、高度(alt)的变化规律, 结果如图10所示。图10(a)~(d)分别表示“子任务”耗时 t 与mlat, lat, lon, alt之间的关系, 纵坐标均表示“子任务”耗时。由图10(c)和(d)可知, t 值随着lon和alt的变化无明显规律; 由图10(b)可知, t 随lat的变化有一定的关系, 但不是很明显; 由图10(a)可知, t 随mlat的变化具有清晰的函数关系。因此, 我们提出了复杂度因子的概念, 用 t_1 表示, 由malt对 t 进行拟合得到。

为了能够更准确地拟合出 t_1 , 以mlat对 t 进行多项式拟合, 结果如图11所示, 其中, 拟合过程略去了 $t < 10$ ms的情况。由图可知, 在 $[-60^\circ, 60^\circ]$ 区间mlat和 t 区域可得获得很好的拟合关系, 拟合因子 t_1 与mlat的函数表达式见式(7)。

$$t_1 = -6.02 \times 10^{-4} \text{mlat}^3 + 0.06 \text{mlat}^2 - 1.28 \text{mlat} + 51.13, |\text{mlat}| > 0. \quad (7)$$

拟合因子 t_1 与 t 的相关性如图12所示, 由图12可知,

图 10 各“子任务”耗时 t 随4个参量的变化规律. (a) 磁纬度(mlat); (b) 纬度(lat); (c) 经度(lon); (d) 高度(alt)
Figure 10 The variation of time-consuming (t) of each “subtask” with four parameters. (a) mlat; (b) lat; (c) lon; (d) alt.

图 11 (网络版彩图)磁纬度(mlat)与耗时 t 的多项式拟合
Figure 11 (Color online) Polynomial fitting of magnetic latitude (mlat) and time-consuming (t).

t_1 与 t 的相关系数为0.79, 呈强相关关系. 所以, 可以用 t_1 表征“子任务”的复杂度, 并按照 t_1 对“子任务”进行排序.

由于拟合过程略去了 $t=0$ 的情况, 在计算 t_1 的时, 如果lat属于 $[-60^\circ, 60^\circ]$ 区间, t_1 由式(7)计算而得, 否则 $t_1=0$. 根据拟合耗时 t_1 的大小, 对“子任务”进行排序, 实现“子任务”按照其计算复杂度递增/递减的顺序输入.

“S”分配方式如图13所示, 将 $N(N=2000)$ 个“子任务”按照复杂度递增的顺序排序后分至8个数据块(图

图 12 拟合因子 t_1 与耗时 t 的相关性
Figure 12 Correlation between fitting factors (t_1) and time-consuming (t).

中一行“子任务”组成一个数据块), 其中数字顺序表示各“子任务”复杂度的排序, 由于分配方式形似字母“S”, 简称“S”分配. “S”分配可以使得图中每一列相邻“子任务”复杂度相近, 同时不会将每一列中复杂度最大的“子任务”分配至同一行, 从而使得各数据块复杂度总和相似, 减少并行过程中的线程等待, 减少并行

1	16	17		$N-7$	数据块1
2	15	18		$N-6$	数据块2
3	14	19		$N-5$	数据块3
4	13	20		$N-4$	数据块4
5	12	21		$N-3$	数据块5
6	11	22		$N-2$	数据块6
7	10	23		$N-1$	数据块6
8	9	24		N	数据块8

图 13 (网络版彩图)“S”分配方式

Figure 13 (Color online) Distribution method of “S”.

耗时,从而提高并行加速比,优化并行过程.

4.1.3 Sort+“S”+dynamic调节

Sort+“S”+dynamic是在sort+“S”调节的基础上进行dynamic调节.

4.1.4 Sort+dynamic调节

Sort+dynamic是指在对“子任务”按其复杂度排序

之后,直接进行dynamic调节.

4.2 优化结果

采用dynamic, sort+“S”, sort+“S”+dynamic, sort+dynamic 4种方式,优化负载不平衡的问题,4种方式优化后并行加速比和并行效率如图14所示.图中横坐标表示不同的外磁场模型,纵坐标表示加速比 S (图14(a))和效率 E (图14(b)),黑色虚线表示优化前的结果,实线表示采用不同优化方式优化后的计算结果.由图14可知,4种方式优化后的并行加速比和效率排序均为: sort+dynamic>sort+“S”+dynamic>sort+“S”>dynamic,即sort+dynamic优化后并行加速比和效率均最大,加速比范围为2.33~3.30,效率为29.13%~41.25%.所以,sort+dynamic优化后并行效果最好,参与计算的处理器利用程度相对最大,优化效果最佳.

5 讨论

根据并行计算结果得知,并行计算可以明显提高并行加速比,当线程数设置为系统最大线程时,并行

图 14 (网络版彩图)四种优化方式的并行加速比(a)和效率(b)

Figure 14 (Color online) Parallel speedup (a) and efficiency (b) of four optimization methods.

加速比最大. 根据并行的优化结果得知, sort+dynamic 为最优调节方式, 不同外磁场模型具有不同的加速比、不同的优化潜力. 为了解释出现这些结果可能的原因, 我们对不同的结果进行讨论分析.

5.1 线程数与加速比

并行计算过程中, 当线程数设置为系统最大线程时, 并行加速比最大, 产生这一结果可能的原因分析如下. 当设置的线程数小于系统线程数时, 系统存在可用的闲置线程, 若此时增大设置的线程数, 可以提高参与计算的有效线程数, 进而提高并行加速比. 当设置的线程数大于系统线程数时, 系统没有多余的线程执行“子任务”计算, 系统反而需要额外的辅助工作, 创造多线程的工作条件, 增加辅助工作耗时, 从而增加并行过程耗时, 降低并行加速比. 所以, 只有当设置的线程数等于系统最大线程数8时, 并行加速比最大.

5.2 负载不平衡的最优调节方式

在解决负载不平衡的并行优化过程中, 4种方式的优化效果排序为dynamic<sort+“S”<sort+“S”+dynamic<sort+dynamic, 即sort+dynamic为最优调节方式. 产生这一结果的原因分析如下, (1) sort+“S”的优化效果大于dynamic的优化效果, 可能的原因是dynamic分配过程中引入了新的辅助工作(如判断线程进度以及线程间子任务调度等), 目的是多次分配“子任务”以优化并行, 但辅助工作的耗时可能超过了sort+“S”分配的处理时间, 导致dynamic优化后总体耗时大, 并行加速比低. (2) Sort+“S”+dynamic的优化效果大于sort+“S”的优化效果, 可能的原因是dynamic减少了sort+“S”分配后仍然存在的部分负载不平衡, 从而减少了并行过程的总耗时, 增大了加速比. (3) Sort+dynamic的优化结果优于sort+“S”+dynamic的优化结果, 可能原因是“S”分配过程的耗时超过了优化负载不平衡节省的时间, 反而增加了sort+“S”+dynamic的耗时增加. 所以, sort+dynamic的优化效果大于sort+“S”+dynamic, 成为最优的调节方式.

5.3 并行加速比与优化潜力

在并行过程中, 计算同一“任务”, 不同外磁场具有不同的加速比; 在优化过程中, 采用同种优化方式, 不同外磁场具有不同的优化潜力. 产生这一结果的原因

可能是, 不同外磁场模型计算时, 各“子任务”的耗时及其分布不同.

在不同外磁场下, 获取各“子任务”的耗时, 得到的分布直方图如图15所示. 图15(a)~(d)分别表示 $kext=0$, $kext=1$, $kext=2$, $kext=6$ 时的分布结果, 横坐标表示“子任务”耗时, 纵坐标表示频数. 各“子任务”耗时的离散程度用离散系数 v 表示, v_0, v_1, v_2, v_6 分别表示对应外磁场模型下“子任务”耗时的离散系数. “子任务”耗时分布直方图的峰度系数(分布直方图顶端尖峭或扁平的状态)用 k 表示, k_0, k_1, k_2, k_6 分别表示不同外磁场模型下的峰度系数. “子任务”耗时分布直方图的偏度系数(分布直方图偏离对称性的程度)用 y 表示, y_0, y_1, y_2, y_6 分别表示不同外磁场模型下的偏度系数. 各“子任务”的总耗耗时用 T_s (单位: ms)表示, $T_{s0}, T_{s1}, T_{s2}, T_{s6}$ 分别表示不同外磁场模型下的总耗时.

其中, $v=\sigma/\mu$ (σ 表示 t 的标准差, μ 表示 t 的平均值), v 越大, 各“子任务”耗时的离散程度越大, 并行过程中越容易造成负载不平衡, 加速比越小. $k=E(t-\mu)^4/\sigma^4$ ($E(X)$ 表示 X 的期望值), k 越大, 分布直方图顶端尖峭, 并行过程中负载不平衡越严重, 优化过程中可调节的部分越多, 优化潜力越大. $y=E(t-\mu)^3/\sigma^3$, y 越大, 分布直方图偏离对称性的程度越大, 并行过程中负载不平衡越严重, 优化过程中可调节的部分越多, 优化潜力越大.

5.3.1 不同外磁场的并行加速比

在不同外磁场下, 采用同种调节方式, 并行计算同一“任务”, 得到的加速比存在明显差别. 出现这一结果的原因可能是外磁场的存在与否以及不同外磁场模型自身具有不同的复杂度. 在并行过程中, 并行效率不能达到100%是由于信息传递、等待、并行处理等辅助工作占用了大量的计算时间. 针对同一“任务”, 有效计算的复杂度越低, 辅助工作耗时占比越高, 并行加速比越小.

相比于有外磁场的情况, 没有外磁场时($kext=0$)的有效计算复杂度最低, 辅助工作耗时比例最高, 从而并行加速比最小. 当存在外磁场时, 有效计算复杂度较大, 辅助工作耗时比例较低, 从而并行加速比较高. 然而, 不同外磁场模型自身复杂度不同, 导致各“子任务”计算耗时不同, 耗时分布存在明显差别, 结果如图15所示. 由图可知, $v_2 < v_1 < v_6$, 由于 v 越大加速比越小, 所以, 在有外磁场时, 加速比的大小顺序为($kext=2$)>($kext=1$)>($kext=6$). $kext=0$ 时没有外磁场模型, 加速比最小, 所

图 15 (网络版彩图)同一“任务”在不同外磁场模型下各“子任务”耗时 t 的频数分布图。(a) $k_{ext}=0$; (b) $k_{ext}=1$; (c) $k_{ext}=2$; (d) $k_{ext}=6$

Figure 15 (Color online) Frequency distribution of time-consuming (t) of each “subtask” of the same “task” under different external magnetic field models. (a) $k_{ext}=0$; (b) $k_{ext}=1$; (c) $k_{ext}=2$; (d) $k_{ext}=6$.

以, 不同外磁场下加速比大小顺序为 $(k_{ext}=2)>(k_{ext}=1)>(k_{ext}=6)>(k_{ext}=0)$ 。

其中, 需要说明的是当 $k_{ext}=0$ 时, v 最小, 但加速比依然最小, 这可能受“任务”总耗时 T_s 的影响。在4核8线程的计算机中, 并行计算的最大加速比为8, 由加速比的定义可知, 此时并行总耗时 $T_p=T_s/8$ 。在并行优化过程中, 并行总耗时 $T'_p=T_p+T_a$ (其中, T'_p 表示优化后的并行总耗时, T_a 表示优化过程中辅助工作的耗时), 所以, 并行加速比由方程组(8)求解, 结果如式(9)所示。由式(9)可知, 采用同种调节方式(即 T_a 相等), 串行总耗时 T_s 越小, 加速比越小。由图15可知, $k_{ext}=0$ 时“任务”串行总耗时 T_{s0} 最小, 加速比最小。

$$\begin{cases} T_p = \frac{T_s}{8}, \\ S = \frac{T_s}{T_p + T_a}, \end{cases} \quad (8)$$

$$S = 8 - \frac{64T_a}{T_s + 8T_a}. \quad (9)$$

5.3.2 不同外磁场的优化潜力

在并行优化后, 对于不同外磁场模型, 并行加速比

的增加量存在明显差别, 即优化潜力存在明显差别。出现这一结果的原因可能是“子任务”分布的峰度系数和偏度系数不同。在并行优化后, 并行加速比提高的前提是, 并行过程存在可以调节、优化的部分, 可调节部分越大, 并行加速比的增加量越大, 并行的优化潜力越大。由图15可知, 不同外磁场模型下, “子任务”耗时分布直方图的峰度系数大小为 $k_1 < k_2 < k_0 < k_6$, 由于 k 越大优化潜力越大, 加速比的增加量越大, 所以, 不同外磁场模型下, 加速比的增加量大小为 $(k_{ext}=6)>(k_{ext}=0)>(k_{ext}=2)>(k_{ext}=1)$ 。不同外磁场模型下, “子任务”耗时分布直方图的偏度系数大小为 $y_1 < y_2 < y_0 < y_6$, 由于 y 越大优化潜力越大, 加速比的增加量越大, 所以, 不同外磁场模型下, 加速比的增加量大小为 $(k_{ext}=6)>(k_{ext}=0)>(k_{ext}=2)>(k_{ext}=1)$ 。综上所述, 图15中的峰度系数和偏度系数, 均能解释不同外磁场具有不同的优化潜力。

6 结论

本文基于OpenMP并行的方法, 对 L 值常用的计算程序成功实现并行。同一“任务”的串行、并行计算结

果误差为0, 表明并行计算不会改变计算结果的准确性. 分析了并行计算后的加速比和效率, 表明并行计算可以明显提高 L 值的计算效率, 加速比范围为1.57~2.81, 并行效率范围为19.63%~35.13%. 当线程数设置为系统最大线程数8时, 并行加速比最大. 在并行计算过程中发现, 负载不平衡是影响并行加速比的关键因素之一. 为了优化负载不平衡的问题, 采用dy-

namic, sort+“S”, sort+dynamic, sort+“S”+dynamic 4种方式进行调节、优化, 结果表明sort+dynamic调节方式最优, 优化后加速比范围为2.33~3.30, 效率为29.13%~41.25%. 针对不同外磁场模型, 并行加速比存在明显差别, 采用同种调节方式, 并行计算同一“任务”时, 外磁场模型 $kext=2$ 的加速比最大; 外磁场模型 $kext=6$ 的并行潜力最大, 优化后加速比的增加量最大.

参考文献

- 1 Baker D N, Erickson P J, Fennell J F, et al. Space weather effects in the Earth's radiation belts. *Space Sci Rev*, 2018, 214: 1–60
- 2 Xu J Y, He Z H, Baker D N, et al. Characteristics of high-energy proton responses to geomagnetic activities in the inner radiation belt observed by the RBSP satellite. *J Geophys Res Space Phys*, 2019, 124: 7581–7591
- 3 Turner D L, Kilpua E K J, Hietala H, et al. The response of Earth's electron radiation belts to geomagnetic storms: Statistics from the van Allen probes era including effects from different storm drivers. *J Geophys Res Space Phys*, 2019, 124: 1013–1034
- 4 McAdams K L, Reeves G D, Friedel R H W, et al. Multisatellite comparisons of the radiation belt response to the Geospace Environment Modeling (GEM) magnetic storms. *J Geophys Res*, 2001, 106: 10869–10882
- 5 Zong Q G, Yuan C J, Wang Y F, et al. Dynamic variation and the fast acceleration of particles in Earth's radiation belt. *Sci China Earth Sci*, 2013, 56: 1118–1140
- 6 Agapitov O, Mourenas D, Artemyev A, et al. Time scales for electron quasi-linear diffusion by lower-band chorus waves: The effects of ω_{pe}/Ω_{ce} dependence on geomagnetic activity. *Geophys Res Lett*, 2019, 46: 6178–6187
- 7 Fu S, Ni B B, Li J X, et al. Interactions between magnetosonic waves and ring current protons: Gyroaveraged test particle simulations. *J Geophys Res Space Phys*, 2016, 121: 8537–8553
- 8 Mourenas D, Artemyev A V, Ma Q, et al. Fast dropouts of multi-MeV electrons due to combined effects of EMIC and whistler mode waves. *Geophys Res Lett*, 2016, 43: 4155–4163
- 9 Chen Y R, Zhou Q H, He Y H, et al. Global occurrences of electrostatic electron cyclotron harmonic waves associated with radiation belt electron distributions. *Geophys Res Lett*, 2019, 46: 5028–5033
- 10 Wang Q, Fu S, Ni B B, et al. Bounce resonance scattering of ring current electrons by H^+ band EMIC waves. *Phys Plasmas*, 2018, 25: 082903
- 11 Vette J I. The AE-8 trapped electron model environment. Technical Report. NASA Goddard Space Flight Center, 1991
- 12 Heynderickx D, Lemaire J, Daly E J, et al. Calculating low-altitude trapped particle fluxes with the NASA models AP-8 and AE-8. *Radiat Measurements*, 1996, 26: 947–952
- 13 McIlwain C E. Coordinates for mapping the distribution of magnetically trapped particles. *J Geophys Res*, 1961, 66: 3681–3691
- 14 Bai C H, Xu W Y, Kang G F. Earth's main magnetic field model (in Chinese). *Progr Geophys*, 2008, 23: 1045–1057 [白春华, 徐文耀, 康国发. 地球主磁场模型. 地球物理学进展, 2008, 23: 1045–1057]
- 15 Jordan C E. Empirical models of the magnetospheric magnetic field. *Rev Geophys*, 1994, 32: 139–157
- 16 Hoare C A R. Towards a theory of parallel programming. In: Hansen P B, ed. *The Origin of Concurrent Programming*. New York: Springer, 1972. 231–244
- 17 Bal H E, Kaashoek M F, Tanenbaum A S. Orca: A language for parallel programming of distributed systems. *IEEE Trans Software Eng*, 1992, 18: 190–205
- 18 Gropp W, Lusk E, Skjellum A, et al. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge: MIT Press, 1999
- 19 Dagum L, Menon R. OpenMP: An industry standard API for shared-memory programming. *IEEE Comput Sci Eng*, 1998, 5: 46–55
- 20 Li Q, Shao S. The development and future of OpenMP (in Chinese). *Technol Econom Guide*, 2019, 27: 1 [李琦, 邵帅. OpenMP的发展及未来. 科技经济导刊, 2019, 27: 1]
- 21 Liu J J, Tao Z M. Application of OpenMP parallel computing in data processing of side scattering lidar (in Chinese). *Comput Digital Eng*, 2019, 47: 266–271 [刘俊见, 陶宗明. OpenMP并行计算在侧向散射激光雷达数据处理中的应用. 计算机与数字工程, 2019, 47: 266–271]

- 22 Lu D H, Zhao Y J, Zhang D H, et al. Remote sensing image parallel processing technology in single machine multi-core computer environment (in Chinese). *Geospat Inform*, 2018, 16: 55–59 [陆冬华, 赵英俊, 张东辉, 等. 单机多核计算机环境下的遥感图像并行处理技术. *地理空间信息*, 2018, 16: 55–59]
- 23 Liu W, Wang F, Zhou H. Parallel seismic modeling based on OpenMP+AVX and optimization strategy. *J Earth Sci*, 2018, 30: 843–848
- 24 Kuang K F, Zhang S J, Li J C. Real-time GPS satellite orbit and clock estimation based on OpenMP. *Adv Space Res*, 2019, 63: 2378–2386
- 25 Meng Z Y, Koniges A, He Y H, et al. OpenMP Parallelization and Optimization of Graph-Based Machine Learning Algorithms. Cham: Springer, 2016. 17–31
- 26 Roederer J G, Lejosne S. Coordinates for representing radiation belt particle flux. *J Geophys Res Space Phys*, 2018, 123: 1381–1387
- 27 Hilton H H. L parameter: A new approximation. *J Geophys Res*, 1971, 76: 6952–6954
- 28 Liu C B, Wang B, Chen J G, et al. Calculation method and error analysis of magnetic field total tensor measurement (in Chinese). *Spacecr Environ Eng*, 2015, 32: 63–66 [刘超波, 王斌, 陈金刚, 等. 磁场全张量测量计算方法与误差分析. *航天环境工程*, 2015, 32: 63–66]
- 29 Qun N H, Khalib Z I A, Raof R A A. Performance Analysis of OpenMP Scheduling Type on Embarrassingly Parallel Matrix Multiplication Algorithm. Cham: Springer, 2017. 917–925
- 30 Ying S C. Performance analysis of OpenMP parallel program (in Chinese). Dissertation for Master's Degree. Changsha: National University of Defense Technology, 2006 [殷顺昌. OpenMP并程序性能分析. 硕士学位论文. 长沙: 国防科学技术大学, 2006]

Research on parallel computing of magnetic L -shell based on OpenMP

SHENG HongJiao^{1,2}, ZHANG XianGuo^{1,3,4} & ZHU ChangBo^{1,3,4}

¹ National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China;

² University of Chinese Academy of Sciences, Beijing 100049, China;

³ Beijing Key Laboratory of Space Environment Exploration, Beijing 100190, China;

⁴ Key Laboratory of Science and Technology on Environmental Space Situation Awareness, Chinese Academy of Sciences, Beijing 100190, China

In the study of radiation belts, the (L, B) coordinate system reduces the three-dimensional problem to a two-dimensional problem, greatly improving the utilization of the limited three-dimensional observational data on radiation belts. The L -shell is one of the most important parameters in the (L, B) coordinate system. It is widely used in particle characteristics analysis, in the description of spacecrafts' orbital radiation environments, and in radiation band modeling. The L -shell is calculated as the integral of the magnetic line tracing, but the complexity of geomagnetic field configurations results in the strict limitation of the tracking step length, resulting in more steps consisting of a single tracking calculation. In addition, the tracking path and distance of different spatial position lines vary significantly with time and space. This variation results in the independence of many L -shell calculations from one another, with different levels of complexity. In such cases the summation of steps may not be able to be simplified, causing the calculation speed to be slow and the efficiency low. In order to improve the efficiency of the L -shell calculation, this paper uses OpenMP to realize single-machine multithreading parallelization of L -shell calculation programs widely used at present. This paper then analyzes the acceleration ratio and efficiency boost of parallelization. The results show that the efficiency of L -shell calculation is much improved by parallel processing. When the number of threads is set equal to the maximum number of threads in the system (here, 8), parallel performance is optimal. The acceleration ratio ranges from 1.57 to 2.81, and the parallelization efficiency boost ranges from 19.63% to 35.13%. Load imbalance among threads is the main factor affecting parallel performance. In order to improve the parallel acceleration ratio and efficiency boost, this paper proposes a sort factor, representing the complexity of L -value calculation. Based on this, an algorithm of sorting and redistribution (sort, S) is proposed for load optimization. The effects of four methods of optimizing load imbalance are compared and analyzed: dynamic, sort+“S”, sort+dynamic, and sort+“S”+dynamic. The results show that the sort + dynamic method provides the best optimization, increasing the optimization efficiency by 6.12%–9.50%. Using this method, the acceleration ratio ranges from 2.33 to 3.30, and the efficiency boost ranges from 29.13% to 41.25%. This paper shows that OpenMP can successfully parallelize and improve the computing efficiency of L -values. As for the problem of load imbalance, the best parallel acceleration ratio and efficiency boost can be achieved using sort+dynamic adjustment. This investigation provides a useful exploration of efficiency improvement for spatial physics and other fields that leverage similar types of big data analysis research.

magnetic L -shell, parallel computing, OpenMP, parallel speedup ratio, parallel efficiency, parallel optimization

doi: [10.1360/SST-2019-0405](https://doi.org/10.1360/SST-2019-0405)