

面向天河超算的回旋动理学模拟代码异构开发及优化

李青峰¹ 李跃岩¹ 栾钟治² 张文禄³ 龚春叶^{1,4} 郑刚¹ 康波¹ 孟祥飞¹

¹(国家超级计算天津中心 天津 300457)

²(中德软件技术联合研究院(北京航空航天大学) 北京 100191)

³(中国科学院物理研究所 北京 100086)

⁴(国防科技大学计算机学院 长沙 410073)

(liqf@nsc-tj.cn)

Heterogeneous Programming and Optimization of Gyrokinetic Simulation Code on Tianhe Supercomputer

Li Qingfeng¹, Li Yueyan¹, Luan Zhongzhi², Zhang Wenlu³, Gong Chunye^{1,4}, Zheng Gang¹, Kang Bo¹, and Meng Xiangfei¹

¹(National Supercomputer Center in Tianjin, Tianjin 300457)

²(Sino-German Joint Software Institute (Beihang University), Beijing 100191)

³(Institute of Physics, Chinese Academy of Sciences, Beijing 100086)

⁴(College of Computing Science and Technology, National University of Defense Technology, Changsha 410073)

Abstract The magnetic confinement fusion particle in cell (PIC) gyrokinetic simulation code, VirtEx, has been capable of studying the confinement and transport of the fusion product Alpha, which is the key to fusion energy realization. Alpha particle simulation relies heavily on the computational code of the kinetic ion, which has more complex memory access than the electron, and contains both non-regular accesses and atomic write-back operations, belong to memory-intensive application. MT-3000 as a new heterogeneous acceleration device provided by Tianhe's new-generation supercomputing platform, has powerful computational performance with its extremely high computational density. Heterogeneous porting of Alpha particle simulations for this device is a great challenge. In order to fully exploit the computational power of the acceleration array in MT-3000, we combine application characteristics and propose some optimization methods, such as recalculation of intermediate variables, customized software cache design, memory locality optimization, and hotspot function merging, which are designed and implemented to reduce the total amount of memory accesses in the program. The medium scale benchmark with gyrokinetic ion shows an overall speedup of 4.2 times, with 10.9, 13.3 and 16.2 times of speedup on hotspot functions *Push*, *Locate* and *Charge*, respectively, meanwhile it shows a good scaling of scalability with 88.4% efficiency with 5 898 240 accelerator cores in 3 840 nodes.

Key words particle in cell (PIC); gyrokinetic simulation; memory intensive; high arithmetic intensity system; heterogeneous programming

摘要 磁约束核聚变粒子网格法 (particle in cell, PIC) 回旋动理学模拟代码 VirtEx 具备研究聚变产物 Alpha 粒子约束及输运的能力, 是聚变能实现的关键. Alpha 粒子回旋动理学模拟相比电子模拟, 访存量更

收稿日期: 2023-10-31; 修回日期: 2025-03-19

基金项目: 国家重点研发计划项目 (2021YFBO300101); 国家自然科学基金项目 (62032023, 42104078, 61902411)

This work was supported by the National Key Research and Development Program of China (2021YFBO300101) and the National Natural Science Foundation of China (62032023, 42104078, 61902411).

通信作者: 孟祥飞 (mengxf@nsc-tj.cn)

大、更复杂,同时包含非规则访问和原子写操作,对访存性能依赖较高,在面向高计算密度特征的天河新一代超算平台异构处理器 MT-3000 进行移植及性能优化时具有巨大挑战.考虑到异构加速器架构和 PIC 算法特性,设计并实施了一些优化方法,如中间变量的即时计算、定制化的软件缓存设计、缓存空间局部性优化、热点函数合并,显著地提高了热点函数的计算访存比.通过中等规模的回旋动理学离子基准算例测试显示了在热点函数 *Push*, *Locate*, *Charge* 上分别有 10.9, 13.3, 16.2 倍的速度提升,同时在扩展性测试中的 3 840 个节点的 5 898 240 个加速核上显示了良好的扩展性,并行效率为 88.4%.

关键词 粒子网格法;回旋动理学模拟;访存密集型;高计算密度系统;异构开发

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202330872 **CSTR:** 32373.14.issn1000-1239.202330872

随着人类日益增长的能源需求和不可再生资源的枯竭,核聚变能源由于其清洁性和安全性作为解决长期能源需求的解决方案,越来越受到人类社会的关注,目前正在建设中的国际热核实验反应堆(international thermonuclear experimental reactor, ITER)是实现核聚变能和平应用的重要里程碑.磁约束核聚变是产生热核聚变能的最重要方法之一^[1-2].在反应堆中实现和维持等离子体聚变过程具有巨大的科学和技术挑战,其中针对等离子体稳定性的研究有助于理解、预测、控制和减轻等离子体破坏的威胁,是优化燃烧等离子体运行模式、改善等离子体约束和输运的重要保障,是设计和制造先进的核聚变装置的重要依据.

数值模拟是等离子体稳定性研究中的关键方法之一,相比理论研究,它能够分析复杂的物理过程,而相比实验研究,它更加经济和灵活.在等离子体物理数值模拟研究中,回旋动理学理论经常被用来研究在拉莫尔半径空间尺度下的动理学不稳定性 and 湍流传输^[3-5].在回旋动理学理论中,通过回旋平均方法将描述分布函数的方程维度从 6 降低到 5,使得其特别适用于研究更长时间尺度下的等离子体不稳定性 and 湍流传输物理过程.

粒子网格法(particle in cell, PIC)由于其良好的可扩展性、物理守恒性、波粒相互作用描述准确性等优势,在众多回旋动理学模拟算法中具有广泛适用度和应用前景^[6-8].基于 PIC 算法的突出特点,科研学者在解决特定时空尺度物理问题的同时,逐步向多时空尺度耦合的非线性复杂物理模拟演进.其对磁约束核聚变高性能数值模拟中涉及的程序架构、计算性能、算法优化、并行效率都提出了前所未有的挑战.许多科研学者尝试借助异构平台的计算性能满足回旋动理学 PIC 代码日益增长的算力需求,在移植优化和数值算法上作出了诸多努力.

GTC 代码是早期受益于异构并行计算的代码之一,基于 CUDA 在天河一号上展示 2~3 倍的加速^[9].基于 OPENACC 在 Titan 上展示了 2~3 倍的加速,在 Summit 上展示了 3~4 倍的加速^[10].基于 Intel Xeon Phi 加速器,在天河二号上展示了 2~5 倍的加速^[11].ORB5 代码基于 OPENACC,在 Tesla P100 GPU 和 Tesla V100 GPU 的 Summit 中分别获得了 4 倍和 5 倍的加速^[12].

在上述研究中,通常着重考虑了等离子体中电子对模型的贡献,针对电子的模拟,凭借访存规则等优势可以获得较高的计算性能加速.而聚变产物 Alpha 粒子与动理学离子类似,回旋半径较大,必须在回旋运动轨迹上进行回旋平均,从而带来大量非规则的网格数据访存,对访存性能提出了很高的要求.文献[13]显示在只有动理学离子和绝热电子的情况下,异构移植给整体性能带来了负面的优化.考虑到聚变产物 Alpha 粒子的约束和输运是磁约束聚变能否成功的关键.本文重点聚焦于以 Alpha 粒子为代表的回旋动理学代码的异构移植和性能优化.

1 实验平台:天河新一代超算系统

本文的移植优化及分析测试在天河新一代超级计算机上进行.天河新一代超级计算机使用异构处理器 MT-3000^[14],它包含 16 个 CPU、4 个加速集群(簇)、96 个控制核心和 1 536 个加速核心,理论计算密度高达 145 FLOPB.每个加速核心以超长指令字(very long instruction word, VLIW)方式工作,每 16 个加速器核心和 1 个控制核心被组织成 1 个加速阵列,以 SIMD 指令控制.MT-3000 具有混合的存储器层次结构,包括每个集群的 GSM(6 MB),HBSM(48 MB),DDR(32 GB)存储器,每个加速阵列的 AM(768 KB)和 SM(64 KB)片上存储器为加速核供给数据.其架构如图 1 所示.

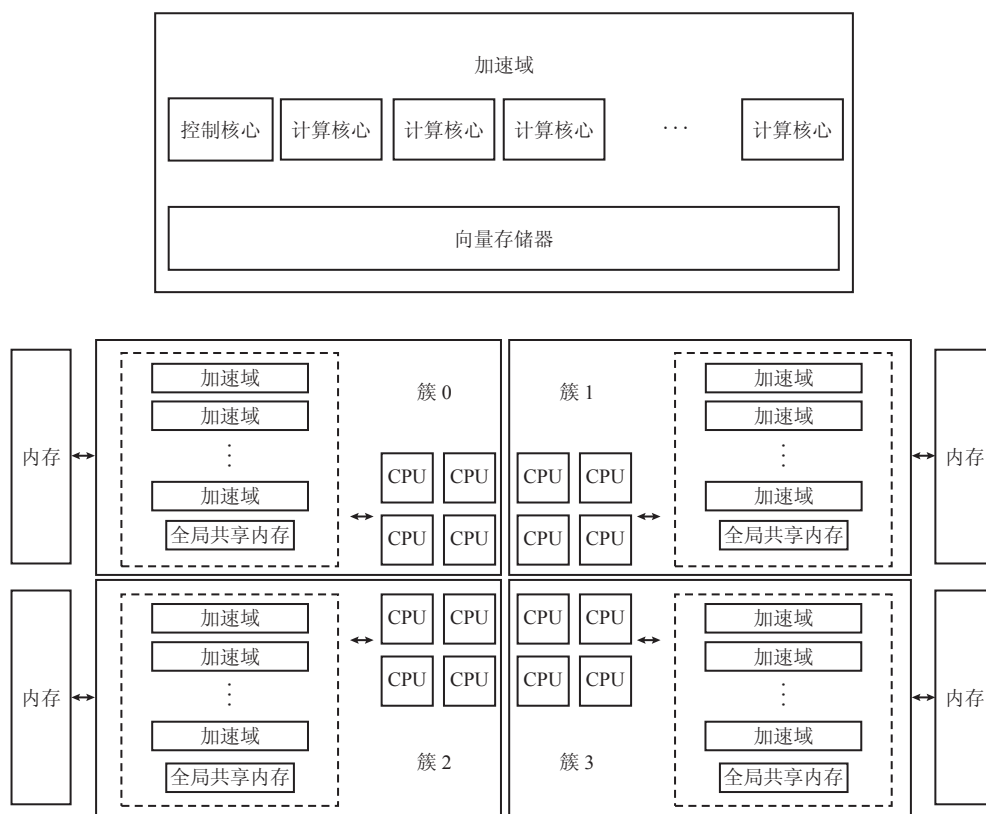


Fig. 1 Architecture of MT-3000

图1 MT-3000 架构

在异构处理器 MT-3000 上移植程序时有 2 个挑战: 1) 如何高效使用复杂的内存结构将数据传递到加速阵列; 2) 如何充分发挥高计算密度特性. 这 2 方面的挑战需要在程序移植优化时打破传统基于 CPU 的程序设计结构, 更多地强调计算性能的充分发挥, 从而实现整体性能的提高.

2 VirtEx 代码热点分析及异构开发

VirtEx 是基于 PIC 算法开发的回旋动力学模拟代码, 已成功用于分析线性电阻撕裂不稳定性^[15]. 代码按照 PIC 算法, 将带电粒子以拉格朗日法描述, 对应连续相空间的分布函数采样点; 而场信息以欧拉法描述, 采用结构化网格描述平衡场, 采用非结构化网格描述扰动场^[16]. VirtEx 代码的并行化策略是通过在环形方向上将模拟区域划分为不同的子域以实现空间并行化, 每个子域由 1 组进程管理. 该组中的每个进程拥有子区域内的场信息副本, 并在该子域内将粒子按照进程编号进行并行划分.

VirtEx 代码的主要结构如图 2 所示, 其主循环使用 2 阶龙格-库塔算法, 在每个循环中, 通过函数 *Push* 更新粒子在相空间的位置, 其可以被更加细致地分

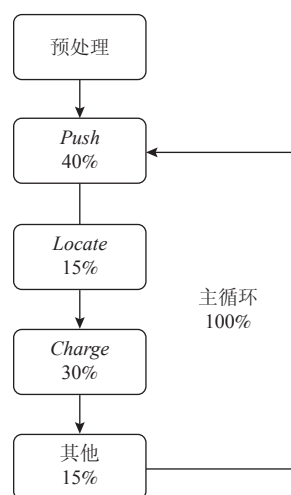


Fig. 2 Main structure and hotspot distribution of VirtEx code

图2 VirtEx 代码的主要结构及热点分布

为在粒子运动轨迹上的扰动场回旋平均函数 *PG* (push gather) 和粒子位置更新函数 *PI* (push interpolation); 通过函数 *Locate* 计算粒子位置和扰动场网格之间插值的权重系数; 通过函数 *Charge* 计算在非结构化扰动网格上的分布函数矩. 而其他热点部分主要是对非结构化网格上的扰动场更新和粒子 MPI 通信等操作. 其中 3 个函数 *Push*, *Locate*, *Charge* 为代码

的热点,共占主循环时间的85%以上.3个热点函数中涉及的算法如下.

算法1. 函数 *PG* 回旋平均算法.

输入: 环向格点权重 *wzpart*, 径向格点权重 *wppart*, 极向格点权重 *wtpart*, 格点编号 *jtpart*, 扰动电场 *gradphi*;
输出: 回旋平均扰动场 *wpgc*.

```
① for (mp=0, mp<mpmax, mp++)/*粒子循环*/
②   for(igyro=0, igyro<ngyro, igyro++)/*回旋平均循环*/
③     读取粒子所在的格点权重及索引;
④     以索引读取 gradphi;
⑤     计算临时变量 e;
⑥   end for
⑦   累加计算 wpgc, 供函数 PI 使用;
⑧ end for
```

算法2. 函数 *PI* 粒子位置更新算法.

输入: 相空间坐标 *zpart*, 历史相空间坐标 *zpart0*, 回旋平均扰动场 *wpgc*;
输出: 相空间坐标 *zpart*.

```
① for (mp=0, mp<mpmax, mp++)/*粒子循环*/
②   读取粒子信息 zpart, wpgc;
③   插值获取网格信息、电场、磁场等;
④   计算场对粒子的作用;
⑤   推动粒子更新速度位置信息;
⑥ end for
```

算法3. 函数 *Locate* 粒子到场的插值权重系数算法.

输入: 相空间坐标 *zpart*;

输出: 环向格点权重 *wzpart*, 径向格点权重 *wppart*, 极向格点权重 *wtpart*, 格点编号 *jtpart*.

```
① for (mp=0, mp<mpmax, mp++)/*粒子循环*/
②   for(igyro=0, igyro<ngyro, igyro++)/*回旋平均循环*/
③     读取粒子信息 zpart;
④     读取网格信息;
⑤     计算粒子插值权重;
⑥   end for
⑦ end for
```

算法4. 函数 *Charge* 非结构化扰动网格上的分布函数矩算法.

输入: 环向格点权重 *wzpart*, 径向格点权重 *wppart*, 极向格点权重 *wtpart*, 格点编号 *jtpart*;
输出: 电流密度 *density*.

```
① for (mp=0, mp<mpmax, mp++)/*粒子循环*/
②   插值获取网格信息、电场、磁场;
```

```
③   for(igyro=0, igyro<ngyro, igyro++)/*回旋平均循环*/
④     读取粒子插值权重;
⑤     计算粒子对于周围格点的扰动量;
⑥     粒子信息向网格上规约到 density;
⑦   end for
⑧ end for
```

上述3个热点函数中的4个算法的外层循环体均围绕粒子展开,且粒子间具有良好的独立性,面向异构处理器 MT-3000 的异构移植工作主要围绕粒子循环的向量指令集改写展开.

同时,为了更好地适配向量指令集的访存特性,重构程序的数据结构,将粒子数据使用 SOA(struct of array)数据结构,网格数据使用 AOS(array of struct)数据结构.粒子数据具有数量多、独立性好的特性,配合 SOA 数据结构更适用于发挥向量指令运算的优势;而网格数据数量远远小于粒子数量,访存量巨大, AOS 的数据结构能够充分发挥内存局部性.针对数据结构的改写工作为后续程序的性能优化提供了重要的保障.

3 面向高计算密度异构设备的性能优化策略

基于上述对于程序热点函数的分析,回旋动力学 PIC 数值模拟算法涉及粒子与网格数据间的大量访存,尤其在面向扰动场网格数据的访存操作中存在非规则访问和原子写操作,二者对于访存性能提出了艰难的挑战,4个热点函数的访存与计算量统计如表1所示.

Table 1 Initial Calculated Density Statistics of VirtEx Hot Spot Function

表1 VirtEx 热点函数的初始计算密度统计

函数	浮点计算量/FLO	访存量/B	计算密度/FLOPB
<i>PG</i>	269 <i>mp</i>	232 <i>mp</i>	1.15
<i>PI</i>	462 <i>mp</i>	224 <i>mp</i>	1.98
<i>Locate</i>	238 <i>mp</i>	200 <i>mp</i>	1.17
<i>Charge</i>	158 <i>mp</i>	200 <i>mp</i>	0.75

注: 变量 *mp* 表示粒子数量, 变量前系数为热点函数中每个粒子计算访存量的统计值.

如何将 1~2 FLOPB 计算密度的访存密集型函数通过性能优化策略发挥高计算密度型异构设备的计算性能,是关键性的研究内容,也是本文的研究重点.在本节中通过中间变量的即时计算,基于 SM 片上存储的软件缓存设计,热点函数合并3种优化方法展开介绍.

3.1 中间变量的即时计算

在传统基于 CPU 的程序设计中, 开发者更倾向于主动寻找公用数据预先计算并暂存于内存中, 利用多级高速缓存, 通过索引获取数据, 通过增加访存量换取计算量的减少. 然而, 这种优化方法并不适合于基于宽向量计算的高计算密度型异构设备, 大量引入访存会限制计算能力的发挥, 同时使用索引的非规则访存模式也不适用于向量计算. 因此, 考虑到新架构的特点, 本文采用了与传统方法截然相反的优化方法来提高计算性能.

在 VirtEx 中, 磁场、温度、密度、安全因子等中间变量可以将预计算转换为即时计算, 引入热点函数, 按照每个粒子对中间变量的需求完成计算. 该操作可以有效减少热点函数中的规则访存和非规则访存, 降低流水线中断次数, 避免索引访问带来的向量重组操作.

通过热点函数分析, 可以进行优化的中间变量主要分为 2 类. 一类以每个径向网格上的极向网格点数 $m\theta$ 为例, 该函数可以在热点函数中完成即时计算. 其中, $m\theta_i$ 为粒子 i 所在的径向网格上的极向网格点数.

$$m\theta_i = 2\text{Floor}\left(\frac{\pi r_i}{\Delta l} + 0.5\right), \quad (1)$$

其中, r_i 和 Δl 分别表示第 i 个径向网格的半径和极向网格的宽度.

另一类中间变量却难以直接解析化表达, 例如粒子在非结构化扰动场网格中的位置索引信息 $igrid$. 其中, $igrid_i$ 为粒子 i 所在扰动场网格中的位置索引信息, 其形式为

$$igrid_i = 1 + \sum_{j=0}^{i-1} m\theta_j, \quad (2)$$

$$m\theta_i = \frac{2\pi r}{\Delta l} + \delta_i = ai + b + \delta_i. \quad (3)$$

如式(2)所示, 变量 $igrid$ 的计算基于变量 $m\theta$ 的累加式, 而由于函数 Floor 引入的不连续性, 导致变量 $igrid$ 的数学公式不能通过简单的变换和积分得出.

但考虑到极向格点数远大于 1, 且径向格点在 r 坐标描述下是均匀的, 当残差 $\delta_i \ll 1$, $igrid$ 同样可以表示为

$$igrid_i = ai^2 + bi + c + r_i, \quad (4)$$

其中残差 r 远小于二次函数部分. 为了能够构建 $igrid$ 的解析表达式, 采用多项式来拟合二次函数的部分, 而残差可以通过周期函数 f 来降低到 0.5 以下, 如图 3 所示. 从而 $igrid$ 的解析表达式可以表示为

$$igrid_i = \text{Round}[ai^2 + bi + c + f(i)]. \quad (5)$$

得益于对平衡剖面信息的解析化表达和即时计

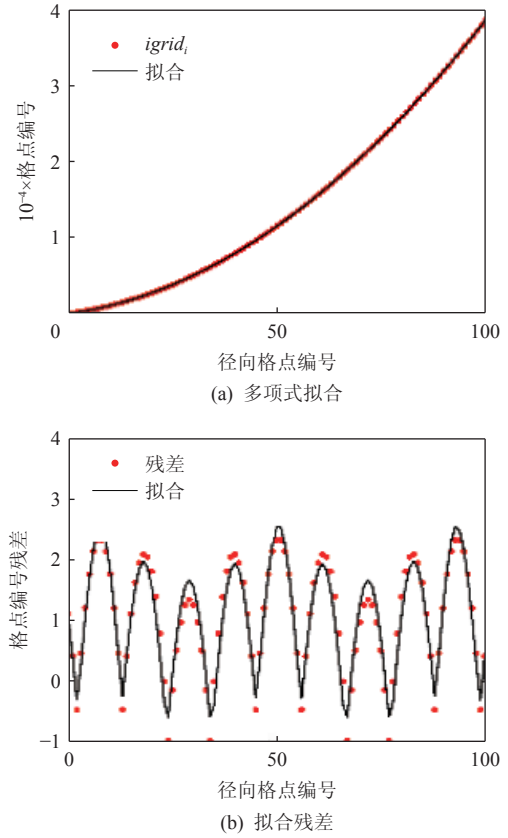


Fig. 3 Comparison of the real value and numerical fitting of the location index variable $igrid$

图 3 位置索引变量 $igrid$ 真实值与数值拟合的对比

算, 函数 PI 和函数 $Locate$ 中的随机内存访问得到减少. 只有热点函数 PG 中存在针对扰动场回旋平均的随机内存访问, 在下面的章节中会论述相应的优化方法.

3.2 基于 SM 片上存储的软件缓存设计

在基于 CPU 的通用架构中, 内置的缓存机制允许开发者在编程时无需关注高速缓存. 而在 MT-3000 处理器中, 内存和 SM/AM 之间、SM/AM 和向量寄存器之间的数据交换需要由程序员手动控制. 在处理内存的随机访问时, 依赖 DMA 的接口操作需要依赖索引和数据, 造成了内存带宽的浪费. 为了解决这个问题, 本文针对加速阵列内部片上存储 SM 设计软件缓存机制, 充分发挥内存结构和内存局部性的优势.

在 VirtEx 热点函数中有 2 个非规则访问, 其中一个是在函数 $Push$ 中涉及到对扰动场网格数据的非规则访问, 另一个是在函数 $Charge$ 中涉及到对扰动场网格数据更新的原子写操作.

函数 $Charge$ 通过累加操作 ($+=$) 将粒子信息累积到网格上, 由于粒子分散在子域内的多个进程, 且网格数远小于粒子数, 这将涉及原子操作. 读/写锁是 MT-

3000 处理器中解决数据竞争的重要方法,因此基于读/写锁设计了一种多级同步的软件缓存机制.首先在 SM 中进行细粒度(如单字)更新,不涉及任何同步操作;其次,使用读写锁保证缓存块在被换出时不会受到数据竞争,同时完成缓存块从 SM 到主存储器的累加操作.

函数 *PG* 主要通过 4 点回旋平均算法获取粒子在回旋运动轨迹上的扰动场信息.由于片上缓存空间有限,回旋平均算法的随机访问性质会对主存带来巨大的访存开销.因此基于片上 SM 存储设计了一种软件缓存机制,该机制通过粒子索引将网格数据按照缓存块读入.如果向量宽度内所有粒子的索引均在缓存块内命中,将组装网格数据向量传到向量寄存器完成向量计算;如果索引未在缓存块命中,按照所需索引完成缓存块数据的更新.同时考虑到性能和局部性的平衡,设计 64 个缓存块并使用哈希作为缓存块的标识.

在软件缓存机制实施后,非规则访存被有效转化,访存带宽的压力得到了缓解.为缓存命中问题,进一步地,考虑到回旋平均算法需获取轨迹上每一点的扰动场信息,由于粒子在速度空间分布的随机性,在更新粒子位置后,极坐标方向的粒子分布会被分散,从而扰乱粒子在非结构化扰动场网格上的分布. *VirtEx* 程序采用基于径向网格点的排序算法修正粒子分布,由于加速阵列中的片上存储空间有限,该算法不足以支撑高计算密度的异构设备,导致缓存命中率的降低.

图 4 显示了排序算法优化前后,粒子编号与相应的非结构化格点编号之间的关系,其中 *psi* 排序是原始的径向排序算法, *igrid* 排序是改进的排序算法,按照粒子所在的网格点排序,增强了空间局部性.优化后的排序采用桶式排序算法,每个桶对应于粒子所

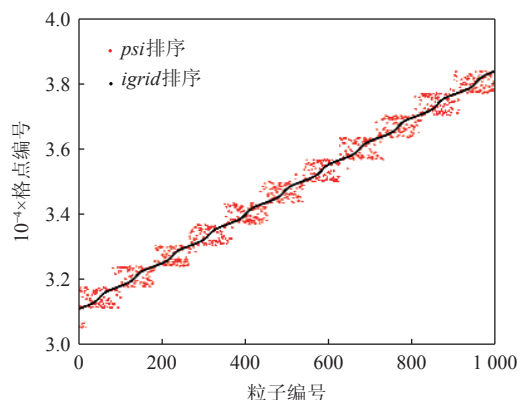


Fig. 4 Comparison of particle lattice numbers under different sorting algorithms

图 4 不同排序算法下的粒子格点编号对比

属的网格点,由于粒子运动的对称性,每个桶的容量总是与每个网格的粒子数同序,因此该算法的复杂度与原来的 *psi* 排序同样是 $O(N)$.

不同排序算法下针对扰动场变量 *gradphi* 的缓存命中率,如表 2 所示.表中,在 64 个缓存块和 1 024 B 缓存块大小的情况下,扰动场变量 *gradphi* 在没有粒子排序的情况下的缓存命中率为 77.99%,接近于 *psi* 排序下的 84.47%,而采用 *igrid* 排序可以获得 99.15% 的缓存命中率,得益于超高的缓存命中率,针对变量 *gradphi* 的非规则访问可以被近似视作规则访问.

Table 2 Cache Hit Rate for Disturbance Field Variable *gradphi* Under Different Sorting Algorithms

表 2 不同排序算法下针对扰动场变量 *gradphi* 的缓存命中率

排序算法	缓存命中率/%
不排序	77.99
<i>psi</i> 排序	84.47
<i>igrid</i> 排序	99.15

3.3 热点函数合并

通过热点函数面向异构加速器 MT-3000 的移植以及上述优化方式的应用,非规则访存操作已经被近似消除,减轻了访存带宽的压力.在经过优化后,热点函数 *PG*, *PI*, *Locate* 的浮点计算量、访存量以及计算密度的统计数据如表 3 所示.从表 3 可以看出,由于函数 *PG* 中的回旋平均操作主要涉及内存访问,其计算密度仅为 1.39;而时间占比最高的函数 *PI*,考虑到基于粒子的计算特点,计算密度仅为 12.4;而函数 *Locate* 在经过变量即时计算优化后,计算密度达到 56.3.综上所述,时间占比高达 40% 的函数 *Push* 的计算密度需要进一步提高计算访存比.

函数 *Locate*, *PG*, *PI* 在 PIC 算法中是计算粒子运动的 3 个相关函数,函数 *Locate* 负责计算插值系数,函数 *PG* 负责获取网格数据,函数 *PI* 负责推动粒子,三者算法上具备可合并性.将函数 *Locate* 引入到函

Table 3 Hot Spot Function is Merged and Optimized to Calculate the Density Statistics

表 3 热点函数合并优化后计算密度统计

函数	浮点计算量/FLO	访存量/B	计算密度/FLOPB
<i>PG</i>	277mp	198.64mp	1.39
<i>PI</i>	1 888mp	152mp	12.4
<i>Locate</i>	12 161mp	216mp	56.3
<i>PushOpt</i>	14 326mp	134.64mp	106.4

注:变量 *mp* 表示粒子数量,变量前系数为热点函数中每个粒子计算访存量的统计值; *PushOpt* 是函数 *Push* 的优化.

数 *Push* 中, 并将函数 *PG* 和 *PI* 合并, 合并后输入仅为粒子信息和网格信息, 输出为粒子信息, 减少了对于大量中间变量的读写. 优化后的函数 *PushOpt* 的计算密度达到 106.4 FLOPB, 更契合高计算密度型计算设备.

4 优化性能测试及分析

4.1 中等规模基准算例性能测试

在中等规模基准算例测试中, 我们用 1 个 MPI 进程控制 1 个 MT-3000 加速集群(簇), 在天河新一代超算系统上使用 120 个节点上的 480 个 MPI 进程和 480 个簇. 该基准测试使用了 1.23×10^6 个网格, 模拟了 2.5×10^9 个粒子.

表 4 显示了 CPU 版本和优化版本在热点函数上的性能对比, CPU 版本在 3 个主要的热点函数的占比达到 86.06%. 结果显示, 基于 MT-3000 处理器的优化后版本的应用加速效果良好, 总体速度提高了 4.2 倍, 其中函数 *Push* 和函数 *Locate* 分别实现了 10.9 倍和 13.3 倍的加速, 具有原子操作的函数 *Charge* 实现了 16.2 倍的性能提升.

Table 4 Performance of Benchmark Examples

表 4 基准算例的性能表现

热点函数	CPU 版本		优化后版本		加速比
	计算时间/s	占比/%	计算时间/s	占比/%	
主循环	845.63	100	201.46	100	4.2
<i>Push</i>	323.86	38.30	29.64	14.71	10.9
<i>Locate</i>	128.69	15.22	9.67	4.80	13.3
<i>Charge</i>	275.19	32.54	16.98	8.43	16.2

注: “占比”指各个函数在主循环中的时间占比.

4.2 扩展性测试

本节展示了优化后的 VirtEx 程序的弱扩展性测试结果. 在弱扩展性测试中, 基准测试为 120 个节点, 使用了 3.86×10^5 个网格, 模拟了 3.7×10^9 个粒子. 随着节点数增加至 3 840 个, 模拟的粒子数也相应地增加到了 1.18×10^{11} . 经过多轮测试取平均后的并行效率如图 5 所示, 在天河新一代超算系统的 3 840 个节点 5 898 240 个加速器核心上, 其并行效率为 88.4%, 展示了良好的弱扩展性.

5 结 论

基于天河新一代超算系统的异构加速器 MT-3000 对大规模并行磁约束聚变回旋动理学模拟代码

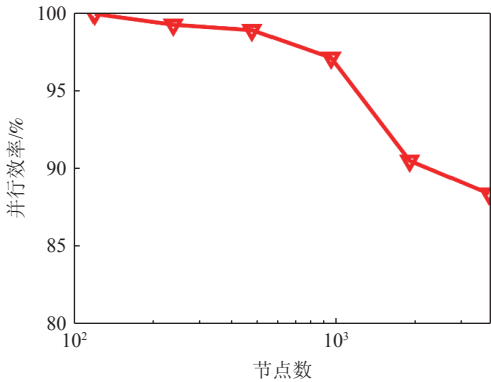


Fig. 5 Weak scalability test results from 120 to 3 840 nodes

图 5 120 个节点到 3 840 个节点的弱扩展性测试结果

VirtEx 进行代码移植和性能优化, 围绕高计算密度型系统和访存密集型应用间存在的矛盾. 通过中间变量的即时计算、定制化的软件缓存设计、空间局部性优化、热点函数合并等优化策略, 并通过数据分析验证了优化的合理性. 同时在基准测试中, VirtEx 的优化显示了良好的加速效果, 其中函数 *Push* 提速 10.9 倍, 函数 *Locate* 提速 13.3 倍, 函数 *Charge* 提速 16.2 倍, 从而使整个程序提速 4.2 倍. 并且在 3 840 个节点的 5 898 240 个加速器核心上展示了良好的可扩展性, 并行效率为 88.4%.

作者贡献声明: 李青峰负责程序设计、移植、测试, 并撰写论文; 李跃岩设计并实现优化算法; 梁钟治负责程序瓶颈分析和提供解决方案; 张文禄提供了针对程序原理和算法方面的指导; 龚春叶提供了针对异构加速设备的优化指导; 郑刚提供了系统测试环境及保障工作; 康波提供了共性技术的指导; 孟祥飞负责设计研究方案并把控研究进度.

参 考 文 献

[1] Sheffield J. The physics of magnetic fusion reactors[J]. Review of Modern Physics, 1994, 66(3): 1015–1103

[2] Lawson J D. Some criteria for a power producing thermonuclear reactor[J]. Proceedings of the Physical Society: Section B, 1957, 70(1): 6–10

[3] Catto P J, Tang W M, Baldwin D E. Generalized gyrokinetics[J]. Plasma Physics, 1981, 23(7): No. 639

[4] Frieman E A, Chen Liu. Nonlinear gyrokinetic equations for low-frequency electromagnetic waves in general plasma equilibria[J]. The Physics of Fluids, 1982, 25(3): 502–508

[5] Brizard A J, Hahm T S. Foundations of nonlinear gyrokinetic theory[J]. Review of Modern Physics, 2007, 79(2): 421–468

[6] Lee W W, Tang W M. Gyrokinetic particle simulation of ion

temperature gradient drift instabilities[J]. The Physics of Fluids, 1988, 31(3): 612-624

- [7] Parker S E, Lee W W. A fully nonlinear characteristic method for gyrokinetic simulation[J]. Physics of Fluids B Plasma Physics, 1993, 5(1): 77-86
- [8] Lin Zhihong, Tang W M, Lee W W. Gyrokinetic particle simulation of neoclassical transport[J]. Physics of Plasmas, 1995, 2(8): 2975-2988
- [9] Meng Xiangfei, Zhu Xiaoqian, Wang Peng, et al. Heterogeneous programming and optimization of gyrokinetic toroidal code and large-scale performance test on th-1a [C] //Proc of the ISC 2013. Berlin: Springer, 2013: 81-96
- [10] Zhang Wenlu, Joubert W, Wang Peng, et al. Heterogeneous programming and optimization of gyrokinetic toroidal code using directives[C]//Proc of the 5th Int Workshop on Accelerator Programming Using Directives. Cham: Springer, 2019: 3-21
- [11] Noé O, Gheller C, Lanti E, et al. Gyrokinetic simulations on many- and multi-core architectures with the global electromagnetic particle-in-cell code ORB5[J]. Computer Physics Communications, 2021, 262: 107208
- [12] Wang Endong, Wu Shaohua, Zhang Qing, et al. The gyrokinetic particle simulation of fusion plasmas on TIANHE-2 supercomputer[C]// Proc of the 7th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems. Piscataway, NJ: IEEE, 2016: 25-32
- [13] Dannert T, Marek A, Rampp M. Porting large HPC applications to GPU clusters: The codes gene and vertex[J]. Advances in Parallel Computing, 2013, 25: 305-314
- [14] Lu Kai, Wang Yaohua, Guo Yang, et al. MT-3000: A heterogeneous multi-zone processor for HPC[J]. CCF Transactions on High Performance Computing. 2022, 4(2): 150-164
- [15] Madduri K, Ibrahim K Z, Williams S, et al. Gyrokinetic toroidal simulations on leading multi- and manycore HPC systems[C/OL]// Proc of the 11th SC. Piscataway, NJ: IEEE, 2011[2024-02-25]. <https://xplore.staging.ieee.org/document/6114454/metrics#citations>
- [16] Feng Hongying, Zhang Wenlu, Dong Chao, et al. Verification of linear resistive tearing instability with gyrokinetic particle code vortex[J]. Physics of Plasmas, 2017, 24(10): 1-9



Li Qingfeng, born in 1993. Master, engineer. His main research interest includes heterogeneous high performance computing.

李青峰, 1993年生. 硕士, 工程师. 主要研究方向为异构高性能计算.



Li Yueyan, born in 1993. PhD, engineer. His main research interest includes heterogeneous high performance computing.

李跃岩, 1993年生. 博士, 工程师. 主要研究方向为异构高性能计算.



Luan Zhongzhi, born in 1971. PhD, associate professor. His main research interests include distributed computing, high performance computing, parallel computing, computer architecture, cloud computing, and big data.

栾钟治, 1971年生. 博士, 副教授. 主要研究方向为分布式计算、高性能计算、并行计算、计算机系统结构、云计算、大数据.



Zhang Wenlu, born in 1975. PhD, professor. His main research interest includes theory and numerical simulation of plasma physics.

张文禄, 1975年生. 博士, 教授. 主要研究方向为等离子体物理理论和数值模拟.



Gong Chunye, born in 1982. PhD, associate professor. His main research interests include parallel algorithms and automatic performance optimization.

龚春叶, 1982年生. 博士, 副研究员. 主要研究方向为并行算法、自动性能优化.



Zheng Gang, born in 1988. Master, senior engineer. His main research interests include high performance computing, parallel computing, and large-scale parallel programs.

郑刚, 1988年生. 硕士, 高级工程师. 主要研究方向为高性能计算、并行计算、大规模并行程序.



Kang Bo, born in 1986. PhD, senior engineer. His main research interests include intelligent computing and high performance computing applications.

康波, 1986年生. 博士, 正高级工程师. 主要研究方向为智能计算、高性能计算应用.



Meng Xiangfei, born in 1979. PhD, senior engineer. His main research interests include heterogeneous high performance computing and generative artificial intelligence, responsible for the research and development of the "Tianhe" series of supercomputers.

孟祥飞, 1979年生. 博士, 正高级工程师. 主要研究方向为异构高性能计算与生成式人工智能, 长期负责我国重大自主创新成果“天河”系列超级计算机应用技术研发与开拓.