文章编号： 1007-4627(2017) 02-0204-07

MPI在蒙特卡罗程序GMT中的应用和发展

许建亚[1,2]，杨 磊[2,†]，张延师[2]，张勋超[2]，付 芬[2]，张雅玲[2]，杨 琼[1,2]

（1. 中国科学技术大学，合肥 230026；
2. 中国科学院近代物理研究所，兰州 730000）

摘要： 针对ADS颗粒靶概念的研究和设计，中国科学院近代物理研究所自主研发了蒙特卡罗模拟软件GMT。为了提高GMT程序的计算效率，研究了MPI在GMT中的应用和发展，实现了大规模随机数在进程中的随机分配，并采用快速读写文件的方式替代了MPI相关数据通信函数，极大地提高了计算效率。并研究了不同规模计算实例进程数、加速比、效率之间的关系，确定了最大加速进程数及并行效率最高时的进程数，为科研工作者在计算资源和计算效率之间选择最优计算方案提供了科学依据。MPI在GMT中的成功应用使计算资源得到了充分、高效的利用，极大地提高了计算效率，解决了蒙特卡罗方法中大规模事件模拟计算时间长、计算不稳定等问题，在散裂靶大规模扫描计算中发挥了重要的作用。

关键词： ADS颗粒靶；MPI；GMT；随机数；数据传输；加速比
中图分类号： TL329　　**文献标志码：** A　　**DOI：** 10.11804/NuclPhysRev.34.02.204

1 引言

加速器驱动次临界系统(ADS)[1]由强流质子加速器、次临界反应堆、散裂靶及其它部件构成的系统，可用于乏燃料嬗变、核燃料增殖及产能。在该系统中，散裂靶装置是持续产生中子的中子源，是联系加速器和次临界堆的重要耦合环节，通过加速器产生的高能质子轰击重金属靶，使其发生散裂反应释放出大量中子，从而使整个ADS系统在次临界条件下持续工作。目前，被广泛采用的固体靶受材料导热性等的限制，只能在极有限的空间内提升靶的功率，如果热沉积可以离线处理，将可以使靶功率的提升上限得以很大的增加，基于这个概念，MEGAPIE和SNS项目中选择了有窗液态重金属靶，运行功率达到约1 MW，但是事实证明MEGAPIE选用铅铋液态重合金靶也存在某些缺陷。基于液态金属靶放射产物毒害性高、温度-材料腐蚀效应严重等特点，Yang等[2]提出了颗粒流靶方案。颗粒流靶是一种流化固体的设计，能有效地解决固体靶热移除限制问题。颗粒流靶以固体颗粒为靶介质的同时也作为冷却介质将束流沉积热带出靶区，具有比常用的液态金属高出一倍的热移除能力，兼具固态靶和液态靶的优点，而密集颗粒流避免了普通流体的流体力学不稳定性。常用的颗粒靶模拟软件主要是MCNPX[3]，Geant4[4]，Fluka[5]，上述软件模拟大规模无规则排布颗粒靶的几何建模和统计上存在着不足。

针对这些问题，中国科学院近代物理研究所基于流化固体颗粒靶设计的需求，自主研发了基于蒙特卡罗方法[6]的粒子输运程序GMT(GPU-based Monte Carlo Transport Program)[7]，主要用来快速计算散裂靶中子学、能量沉积、散裂产物等。程序主要包括五个模块：主程序、几何模块、输运模块、数据模块和后处理模块。其中，主程序完成整个事件模拟过程中模型、变量初始化、模块调用、数据流控制；几何模块提供了靶的几何模型及粒子与靶关系的计算函数，输运模块完成每一次事件过程，靶内部发生的各种反应判断、函数调用及反应产物、粒子运行轨迹、能量沉积等输运结果的存储；数据模块提供了粒子输运过程中用到的各种材料模型、反应模型及数据模型；可视化模块是程序的统计分析模块，完成相应的数据统计和分析工作。另外，程序中加入了颗粒几何分布的批量读入以及颗粒边界的优化处理，同时还对粒子在颗粒介质中的输运过程进行特殊处理，能够满足颗粒靶计算的需求。

蒙特卡罗方法要得到较高精度的解，需要投入大量的实验次数，即使把庞大数目的随机实验交由计算

收稿日期： 2016-06-20；　**修改日期：** 2016-08-24
基金项目： 中国科学院战略性先导科技专项(XDA03030100)
作者简介： 许建亚(1988–)，男，甘肃定西人，研究实习员，从事高性能计算研究：E-mail: xujianya@impcas.ac.cn
† **通信作者：** 杨磊，E-mail: lyang@impcas.ac.cn。

机完成，巨大的计算量依然会需要很长的计算时间，因此GMT需要并行计算[8,9]来减少计算时间。GMT 进行大规模计算模拟时，每一次事件的计算过程都是独立的，并与其他事件的计算过程、计算参数及计算结果不存在相互依赖的关系，所以不同的事件可以由不同的进程独立完成。结合目前并行软件编程模型及中国科学院近代物理研究所超算中心的软硬件资源，MPI+CUDA 并行模型[10]成为了 GMT 并行化的首选方案。其中，MPI(Message Passing Interface) 是粗粒度 (进程级) 并行，CUDA(Compute Unified Device Architecture) 是细粒度 (线程级) 并行，本文主要研究 MPI 并行模型在GMT中的实现。

2　MPI架构及实现

2.1　MPI简介

并行编程模型主要分为相同操作同时作用于不同数据的数据并行模型、使用共享变量实现并行进程间通信的共享变量模型、以及本文采用的消息传递并行模型。

MPI(Message Passing Interface) 是一种消息传递并行编程标准[11]，由全世界工业、科研和政府部门联合建立，为并行应用程序设计提供的一个高效、统一的编程环境，是目前最通用的并行编程方式。其中最流行的版本有 MPICH、LAMMPI，支持目前绝大多数并行计算系统。本文的研究基于中国科学院近代物理研究所超算中心深腾 7 000 G 高性能集群，MPI 是 LAMMPI 版本 OpenMPI-1.3.2。深腾 7 000 G 服务器结点机采用基于 Intel 架构的商用服务器，通过高速通信网络实现结点间互连，系统峰值浮点计算能力为每秒 200 万亿次。全机由 1 个接入控制台、100 个计算节点、1 个存储 I/O 节点以及计算网络、存储网络、管理网络共同组成，外部文件存储由 20 TB 容量的磁盘阵列组成。每个计算节点均包含 2 个 Xeon E5504CPU 处理器 (主频 2 530 MHz，4 核) 及 2 个 TeslaC1060GPU 处理器，8 GB 内存，500 GB 硬盘，最多可以同时运行 200 个 CUDA 程序。操作系统为 Red Hat Enterprise Linux Server release 5.6，编译器为 GCC-4.9.0，提供给用户的并行计算环境有 MPICH、Intel MPI 等。深腾 7 000 G 服务器能够很好地支持大规模科学工程计算，具有结点选择丰富灵活、机群域网专业高效、基础架构完备可靠与机群管理简洁实用等技术优点。

2.2　MPI架构

GMT 主要由主程序、输运模块、几何模块、数据库、可视化模块等构成。如图1，MPI 进程及任务管理主要在主程序中实现，计算节点核数和进程数一一对应 (N 是进程数)，其中主进程负责所有模型及变量的初始化、确定子进程和事件的执行关系；子进程调用随机数生成函数、颗粒靶模型函数、输运模型函数及相关数据库等完成每一次事件计算，并将计算结果输出在子进程相关文件。所有子进程完成各自计算任务后，由主进程负责收集子进程计算结果完成可视化处理。在并行过程中，进程数目的初步选择主要考虑了各个子进程之间的负载均衡问题，使事件数目和进程数能够整除，保证每个子进程都执行相同数目的事件，避免某一进程计算时间过长而影响整个计算效率。

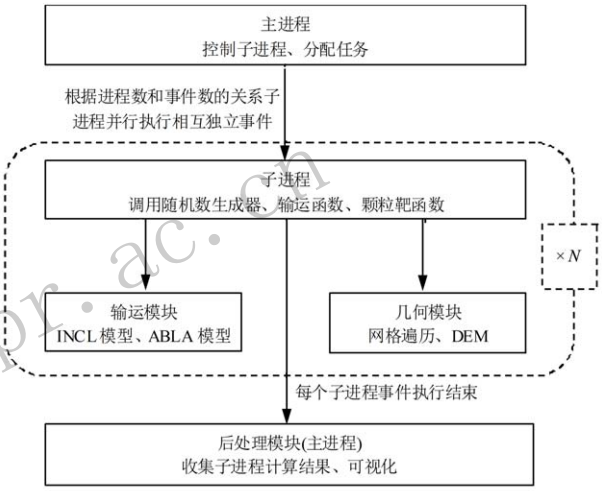

图 1　MPI进程及任务管理

2.3　MPI算例

本文采用 MPI 并行的 GMT 程序模拟了质子与圆柱容器内钨颗粒靶的反应过程。图2 是颗粒靶模型图。入射质子能量为 1 GeV，圆柱体直径为 10 cm，高为 100 cm，颗粒直径分别为 1 cm, 5 mm, 3 mm 三种尺

图 2　(在线彩图) 颗粒靶模型

度，颗粒数最多达到30多万。随机颗粒的空间分布由中国科学院近代物理研究所自主研发的分子动力学软件生成。每次模拟计算均模拟10万次事件。分别用1, 2, 4, 8, 10, 20, 40, 80, 100个进程对三种尺度的颗粒靶进行模拟计算，并对计算时间和相关结果进行了详细的分析和比较。

2.4 MPI实现

在GMT中MPI实现的主要难点在于两个方面，一是蒙特卡罗方法对随机数序列分段与分配，二是大规模数据传输对并行性能的影响。

2.4.1 随机数

GMT中要求每次事件及单个事件中各个模型调用的随机数不能出现重复，并行之后，每个子进程需要拥有与其他子进程不同的随机数序列，且并行时每次事件的随机数序列要与串行时完全一致。基于蒙特卡罗模拟方法对随机数要求的严格性，GMT随机数的实现借鉴了文献[12]中随机数的实现算法将文献中随机数的实现应用到了并行进程中，并在2.3节算例中，定义一个变量，记录每次事件模拟过程中调用随机数的次数，然后统计不同情形下随机数调用次数，找出其中最大值，以该值的10倍为随机数预估值，以防止模拟过程中随机数的调用次数超过预估值而程序中断的情形。如果出现超越预估值的状况，程序会给出警告。这样的方式保证了百万量级输运模拟中每个进程对随机数的要求。

2.4.2 数据传输

GMT中的数据结构极为复杂，包含了各种数据类型，且每个数据类型对应多个大尺度二维数组。子进程中每一次事件结束之后都会产生大量的数据，这些数据都需要由子进程利用MPI通信函数传递给主进程。

程序最初采用MPI自定义结构体对数据类型进行封装，保证了每次传输过程中数据的完整性，在用阻塞通信和非阻塞通信模式分别进行了大量测试之后，发现随着进程数的增加不管采用那种通信模式，通信本身都会占用大量的时间，对整个并行效率造成严重的影响，造成这种影响的主要原因是大规模的数据发送和接收本身会占用大量时间，同时，也会占用大量的内存，当进程数增加时，对内存的占用情况更加严重，使得每个进程的计算能力都受制于内存的不足，不能发挥出最佳计算能力，尤其在采用非阻塞通信时，随着进程数不断增加，等待接收的数据不断增加耗尽内存使计算停止。如图3，结合MPI模型在GMT中的架构及蒙特卡罗方法本身的特性，本文对程序数据流进行了深入分析，发现数据通信传输都在每次事件的计算结束后进行，在计算过程中，各个进程的事件完全独立，不需要进行数据和信息交换；也就是说，在GMT并行中，MPI通信函数的主要作用是在各个子进程事件计算结束后将计算结果传递给主进程。所以，本文采用磁盘阵列快速读写文件的方式代替了MPI通信函数，即在子进程每一次事件计算结束后，不进行MPI数据传输，计算结果由各自独立的文件存储，文件的命名与进程号相关。当所有的子进程计算结束后，由主进程对各个子进程的计算结果进行统一的读取和相关处理。这样的处理模式避免了大规模次数的MPI数据通信传输，再加上磁盘阵列强大的数据读写能力，极大地提高了计算效率。

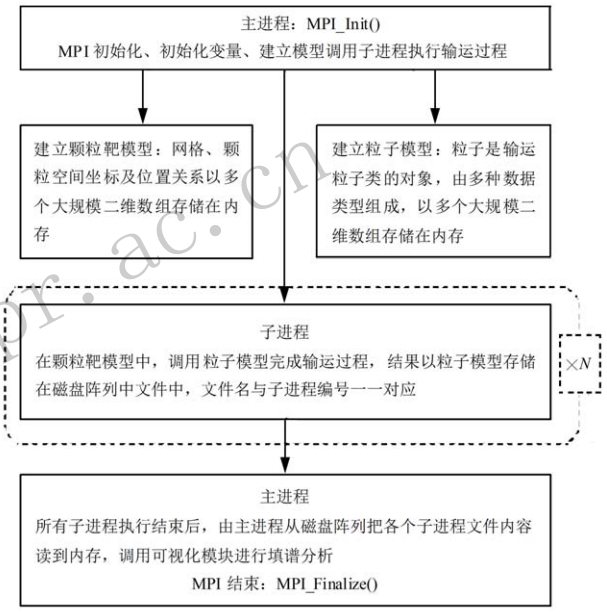

图3 MPI数据传输

在2.3节算例比较了阻塞通信、非阻塞通信、磁盘阵列快速读写三种方法的加速性能，在阻塞通信和非阻塞通信时，数据类型定义为一个结构体，每次事件模拟的返回值为该结构体对象，每个对象由事件模拟过程中产生的次级粒子、粒子质量数、电荷数、能量、位置信息等组成，大小约2 GB，每次通信，均发送接收对象首地址，传递类型为该结构体类型。图4为1 cm颗粒靶中三种通信模式的比较结果，在阻塞通信时，从进程在主进程未接收其发送的数据之前都处于等待状态，当模拟的事件数越大时，这种等待时间的累加导致通信会占用大量时间，且随着进程数的增加，这种效果越加明显，当进程数超过40时，通信时间已经大于因并行加速而减少的计算时间，使整个计算时间随着进程数的增加逐渐增加。在非阻塞通信时，从进程只要完成了通信发送的命令，便继续进行下一次事件的模拟，已经发送

尚未接收的数据存储在内存中，因为每次传递的数据类型约2 GB，随着进程数的增加，当主进程接收速度小于从进程计算发送速度时，等待接收的数据会逐渐耗光内存，使计算终止，如图4，当进程数超过10 时，内存耗光，计算终止 (计算时间无限大)。

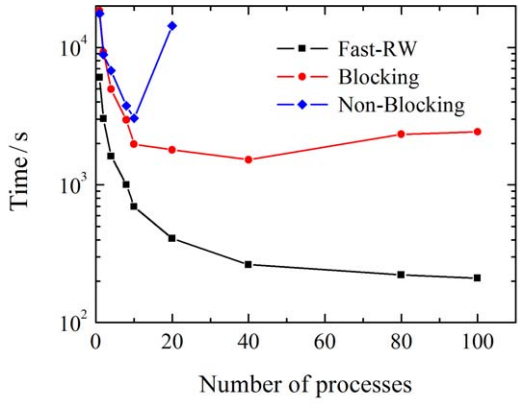

图 4　(在线彩图) 1 cm颗粒靶-通信模式比较

鉴于每次事件返回的数据对象占用存储空间大、模拟事件规模大等原因，为了避免使用阻塞和非阻塞通信时出现的问题，本文采用了磁盘阵列快速读写文件的方式替代了MPI 通信模式，使计算和通信分离，互不影响，完全发挥了每个进程的最佳计算能力，极大地提高了计算性能。

3　性能分析

本文计算模拟均在深腾7000 G 集群下进行。

3.1　准确性分析

本文用不同的进程数分别计算了三种尺度的颗粒靶，分别在各种尺度下，对串行结果与并行结果及不同进程数的计算结果进行了比较分析，在中子产额、能量沉积、气体产物等方面都取得了较好的吻合，并行计算结果的准确度达到了阶段性预期目标。

图5 是颗粒直径为1 cm 时，10 进程计算结果和串行计算结果的中子泄漏能谱比较，从图中可以看出两者的结果吻合较好，误差在1‰ 之内。

图6 是颗粒直径为5 mm 时，20 进程计算结果和串行计算结果侧边泄漏中子在轴向分布的比较，从图中可以看出两者的结果吻合较好，误差在1‰ 之内。

由于GMT 程序中，粒子输运的过程会调用多个子模块来完成输运的完整过程，这些模块中都有随机数的产生、调用 (如INCL 模块、ABLA 模块等)，在串行和并行时，子模块中随机数的序列并不完全相同，所以需要对所有随机数产生器及相关调用过程进行修改，由于

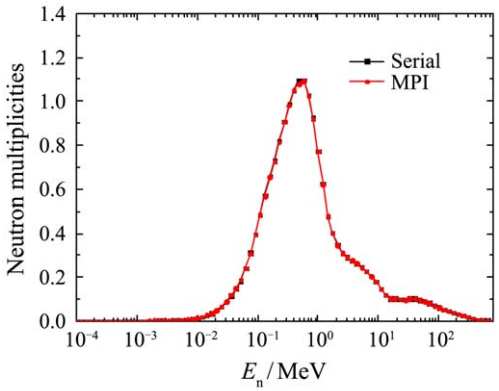

图 5　(在线彩图) 串-并行中子泄漏谱

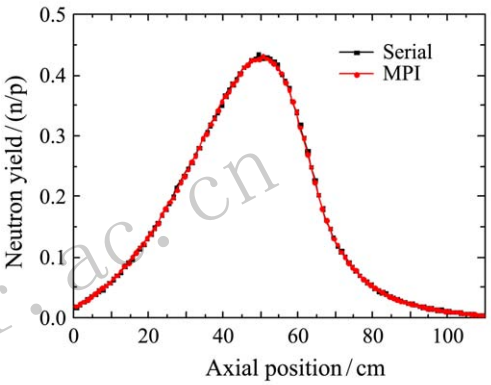

图 6　(在线彩图) 串-并行侧面泄漏中子轴向分布

部分子模块实现的代码量大、结构复杂，这一部分工作目前正在进行，文中提到的11‰ 误差便是由子模块串并行随机数序列不完全一致所引起的。

3.2　加速分析

MPI 在GMT 中的应用使计算资源得到了充分、高效的利用，极大地提高了计算效率，解决了蒙特卡罗方法中大规模事件模拟计算时间长、计算不稳定等问题，促进了散裂靶的设计工作。并行的主要目的是提高计算速度、节省计算时间，加速比及效率则是衡量并行性能的重要指标。其计算公式计分别是：

$$S_{\mathrm{p}} = \frac{T_1}{T_{\mathrm{p}}}\ , \tag{1}$$

$$E_{\mathrm{p}} = \frac{S_{\mathrm{p}}}{P}\ , \tag{2}$$

其中 P 指参与并行计算的CPU核数，T_1 指一个核计算所用的时间，T_{P} 指 P 个节点并行计算所用的计算时间，公式(1)中，S_{P} 表示加速比，公式(2)中，E_{P} 表示效率，通过加速比可知并行时计算速度提高的倍数 (即计算时间减小的倍数)，效率则反映特定进程数时的并行性能，结合加速比及效率，针对具体的计算规模和计算资源，可以确定最优的计算方案。

图 7 给出了颗粒直径为 5 mm 时，随着进程数增加计算时间及加速比的变化趋势，其中加速比及进程数的比值为并行效率。由图可知：进程数在 10 以内时，加速效率在 80% 以上 (并行性能良好)，进程数在 40 以内时，加速效率在 30% 以上，随着进程数的增加，加速比增加，但加速效率在一直降低，当进程数达到 100 时，加速比达到最大值 15.208(计算速度最大可提高 15.208 倍)，再增加进程数时，计算所用时间反而会变长，加速比减小，同时加速效率也持续降低。

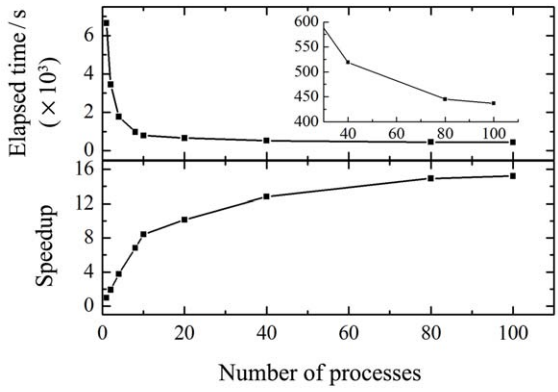

图 7 (在线彩图) 5 mm颗粒-进程与加速

图 8 给出了颗粒直径为 3 mm 时，随着进程数增加计算时间及加速比的变化趋势。由图可知：进程数在 10 以内时，加速效率在 80% 以上，进程数在 20 以内时，加速效率在 60% 以上，随着进程数的增加，加速比增加，但加速效率在一直降低，当进程数达到 40 时，加速比达到最大值20.486(计算速度最大可提高 20.486 倍)，再增加进程数时，计算所用时间反而会变长，加速比减小，同时加速效率也持续降低。

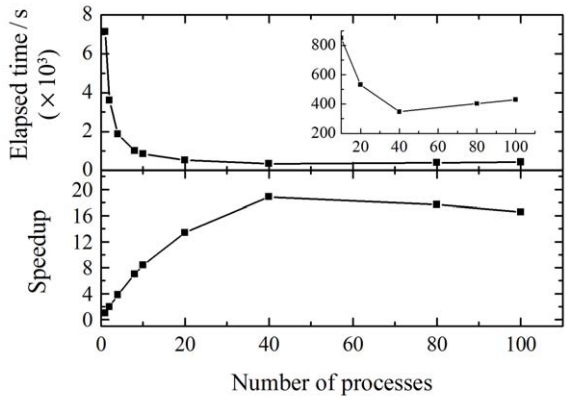

图 8 (在线彩图) 3 mm颗粒-进程与加速

上述加速趋势与并行本身的算法实现及本文计算环境深腾 7000 G 集群的结构相关，如图 9，在 3 mm 颗粒靶中比较了计算时间与通信时间对加速性能的影响。

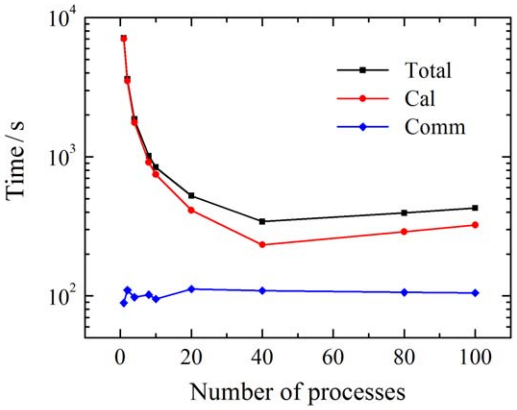

图 9 (在线彩图) 3 mm颗粒靶—计算通信时间比较

当采用不同进程模拟时，通信时间 (文件读写时间) 基本稳定在 100 s 左右，原因在于所有事件运行结束产生的数据规模大小基本一定，主进程计算读写性能一定，使得整个通信时间稳定。理想情况下，随着进程数的增加，计算时间不断减小，通信时间不变，并行计算花费的总时间会无限接近通信时间 (100 s)，所以，加速存在上限。

本文模拟采用了集群 16 个计算节点 (共 128 核)，当进程数为 10 时(小于节点数 16)，每个计算节点只运行一个进程 (单核运行)，此时,每个进程占用的内存充足,能发挥最佳计算能力，加速比呈现近似线性加速 (此时计算时间远大于通信时间)。由于每个节点内存大小一定，随着进程数不断增加 (大于节点数 16)，每个计算节点上运行的进程不断增加 (多核运行)，则每个进程占用的有效内存不断减小，进程数超过某一临界值时，因内存不足，节点的计算性能反而开始下降。所以本文中，进程数超过 40 时，会出现计算时间反而开始增加的情况。

由上文结果可知：

(1) 在 GMT 程序并行化过程中，不能并行化的部分 (通信时间) 是整个并行算法的瓶颈，也是加速上限。当计算时间远大于通信时间时，随着进程数的增加，能够出现近似线性加速的情形，当计算时间接近通信时间或小于通信时间时，加速比快速降低，因为通信时间在整个并行时间中的占比越来越大。理想情况下，通信时间也是GMT并行最大加速时间；

(2) 程序运行的软硬件环境，会对加速性能造成影响。在本文中，每个计算节点的内存大小是影响节点上进程加速性能的另一主要原因，在并行计算的过程，需要通过多次模拟，确定合理的进程数，才能实现最大加速；

(3) 在不同规模的计算模拟过程中，需要结合计算资源、任务进度，在加速比和加速效率之间做出权衡，在提高计算速度的同时降低计算资源的浪费。

4　结论

本文研究了MPI在GMT中的发展和应用，实现了蒙特卡罗随机数在并行进程中的随机分配，并采用磁盘阵列快速读写文件的方式代替MPI通信函数，极大地提高了计算效率，并行的准确性和加速比都取得了较好的结果。并用大量的进程计算了不同尺度颗粒靶，对进程数、加速比及效率之间的关系进行了分析，为最优计算方案的选择提供了科学依据。采用MPI并行的GMT程序已经在散裂靶大规模扫描计算中发挥了重要的作用，尤其在颗粒靶设计方面有突出的贡献。在下一步的工作中，需要进一步对随机数分配、并行任务管理、多尺度颗粒、网格遍历方法等方面做深度优化，以期取得更好的加速比、更高的并行效率。

参考文献：

[1] ZHAN Wenlong, XU Hushan. Bulletin of National Academy of Sciences, 2012, **27**(3): 375. (in Chinese)
(詹文龙, 徐瑚珊. 中国科学院院刊, 2012, **27**(3): 375)

[2] YANG Lei, ZHAN Wenlong. Science China Technological Sciences, 2015, **58**: 1.

[3] DENISE B, PELOWIT Z. MCNPX^{TM} Users's manual. Version 2.6.0, LA-CP-07-1473. US, Los Alamos National Laboratory, 2008: 1.

[4] AGOSTINELLI S, ALLISON J, AMAKO K, *et al*. Nucl Instr and Meth A, 2003, **506**(3): 250.

[5] ALFREDO F, PAOLA R S, ALBERTO F, *et al*. FLUKA: a Multi-particle Transport Code (Program version2005), in: CERN 2005-10 (2005), INFN/TC 05/11, SLAC-R-773.

[6] DENG Li, LI Gang. Chinese Journal of computational Physics, 2010, **27**: 791. (in Chinese)
(邓力, 李刚. 计算物理, 2010, **27**: 791.)

[7] WANG Yi, YANG Pingli, ZHU Weijie, *et al*. Nuclear Electronics& Detection Technology, 2001, **21**(1): 31. (in Chinese)
(王义, 杨平利, 朱伟杰, 等. 核电子学与探测技术, 2001, **21**(1): 31.)

[8] WANG Lei, WANG Kan, YU Ganglin. Nuclear Electronics& Detection Technology, 2008, **28**: 163. (in Chinese)
(王磊, 王侃, 余纲林. 核电子学与探测技术, 2008, **28**: 163.)

[9] LU Fengshun, SONG Junqiang, YING Fukang, *et al*. Computer Science, 2011, **38**(3): 5. (in Chinese)
(卢风顺, 宋君强, 银福康, 等. 计算机科学, 2011, **38**(3): 5.)

[10] YANG Bo. Research on CPU/GPU Synergetic Algorithm for Monte Carlo Deep Penetration Particle Transport[D]. Changshai: Graduate School of National University of Defense Technology, 2011. (in Chinese)
(杨博. 深穿透粒子输运蒙特卡罗模拟的CPU/GPU协同算法研究[D]. 长沙: 国防科学技术大学研究生院, 2011.)

[11] DU Zhihui. High Performance Parallel programming-MPI Parallel Programming[M]. BeiJing: Tsinghua University Press, 2001. (in Chinese)
(都志辉. 高性能并行编程技术—MPI并行程序设计[M]. 北京: 清华大学出版社, 2001.)

[12] FORREST B. The MCNP5 Random Number Generator, LA-UR-07-7963. US, Los Alamos National Laboratory, 2002: 1.

Application and Development of MPI in Monte Carlo Code GMT

XU Jianya[1,2], YANG Lei[2,†], ZHANG Yanshi[2], ZHANG Xunchao[2], FU Fen[2],
ZHANG Yaling[2], YANG Qiong[1,2]

(*1. University of Science and Technology of China, Hefei 230026, China;*
2. Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, China)

Abstract: For the research and design of the ADS granular-flow target concept, the Institute of Modern Physics, CAS has developed a Monte Carlo simulation software (GPU-accelerated Monte Carlo Transport program, GMT). In order to improve the computational efficiency of the GMT program, development and application of MPI in GMT were studied, to realize random distribution of the large-scale random number in the sub processes. Rapid reading and writing files were employed instead of the MPI data communication function, which greatly improves the computational efficiency. Different scale calculations were performed to study the relationship of process instance number, speedup to find the maximum acceleration process number and the number of processes when parallel efficiency is highest, which provides a scientific basis for researchers to optimize the computational program between computational resources and computation efficiency. The successful application of MPI in GMT, utilizes the computing resources fully and efficiently, improves the computational efficiency, solve the long time cost and unstable problem of Monte Carlo method in large-scale event simulations, plays an important role in the large-scale scanning calculation of the spallation target.

Key words: ADS granular-flow target; MPI; GMT; random number; data transmission; speedup

Received date: 20 Jun. 2016; **Revised date:** 24 Aug. 2016

Foundation item: Strategic Priority Research Program of Chinese Academy of Sciences(XDA03030100)

† **Corresponding author:** YANG Lei, E-mail: lyang@impcas.ac.cn.