

MIC 异构直线加速器束流动力学模拟并行软件开发^①

董晓彤, 杜克伟, 赵瑞峰, 徐 进

(中国科学院软件研究所 并行软件与计算科学实验室, 北京 100190)

摘 要: 现代高能物理研究需要使用高能量的粒子加速器, 加速器束流动力学模拟软件具有重要的实用意义. 介绍了一个 3 维基于 MIC 的异构直线加速器并行束流动力学模拟软件 NEWBEAM-MIC 的开发进展. 目的是使用最新的超级异构计算机提高束流动力学模拟软件的性能, 更好地完成加速器的设计和优化工作. 这个软件模拟了 DTL 和 SOLENOID 加速器装置中粒子的运动过程. NEWBEAM-MIC 是在 NEWBEAM-CPU 软件基础上, 将粒子推进部分分配到 MIC 卡上运行, 从而利用 MIC 多线程的优势使计算加速的. 通过实际测试, 这个软件在天河二号上使用 100 CPUs 和 100 MICs 可以模拟 10^9 个粒子, 其中 DTL 场力计算、SOLENOID 场力计算和粒子推进三个部分均可以比仅使用 100 CPUs 的 NEWBEAM 软件有 100 倍以上的加速效果. 再考虑 MIC 卡上的多线程, 对同样规模的粒子, 使用 100 CPUs 和 100 MICs, 当 MIC 线程数开到最大(224)时, NEWBEAM-MIC 可以比单线程串行计算方式加速 10000 倍以上. 这表明本文开发的基于 MIC 的异构软件可以很好地加速原有的 CPU 软件, 发挥现有 MIC 异构超级计算机的潜在性能.

关键词: 粒子追踪; 异构并行计算; 直线加速器; 集成众核; FFTW

Parallel Development for Beam Dynamic Simulation in Linear Accelerator Based on MIC Heterogeneous Software Environment

DONG Xiao-Tong, DU Ke-Wei, ZHAO Rui-Feng, XU Jin

(Laboratory of Parallel Software and Computational Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: High energy particle accelerator is necessary in the research of modern high energy physics, software of beam dynamic simulations is of great importance. This paper introduces the development status of a 3D heterogeneous parallel beam dynamic simulation solver, NEWBEAM-MIC. The purpose of this research is to speed up the beam dynamic simulation software and help to realize design and optimization of the accelerator by latest heterogeneous supercomputers. The motion of the particles through DTL and SOLENOID devices are simulated by this software. This software is called NEWBEAM-MIC, which is developed on the foundation of previous CPU version, NEWBEAM-CPU. It assigns simulation work of particle tracking on MIC, thus accelerates the procedure by taking advantage of MIC's multiple threads. Benchmarks verify that NEWBEAM-MIC can simulate 109 particles through DTL and SOLENOID devices on TIANHE-2 using 100 CPUs and 100 MICs. It can achieve 100 times acceleration than the CPU version software, NEWBEAM-CPU. Considering using multi-threads on MIC, for the same number of particles, using 100 CPUs and 100 MICs with MIC's maximum threads (224) opening, NEWBEAM-MIC can achieve 10000 times acceleration than CPU version with only one thread. It proves that, the software developed in this paper can accelerate CPU version quite well by making use of the great power of the MIC supercomputer.

Key words: particle tracking; heterogeneous parallel computing; linear accelerator; MIC; FFTW

1 背景介绍

现代科学对物质组成的研究已经深入到原子核、夸克等微观粒子层次. 对这些微观粒子的研究需要使

用粒子加速器产生的高能粒子进行探测. 粒子加速器是非常复杂的科学装置, 主要涉及电磁场、束流、控制、测量等领域的技术. 微观粒子的尺度越小, 需要

^① 收稿时间:2015-12-28;收到修改稿时间:2016-02-25 [doi:10.15888/j.cnki.csa.005345]

使用的探测粒子的能量就越高^[1,2]。粒子加速器就是利用某些特定的电磁场对微观粒子进行加速的装置。中国科学院正在进行的一个战略先导专项——“未来先进核裂变能的关键技术”中的核心部分是 ADS(Accelerator Driven System) 嬗变系统。这一系统的关键是设计和制造出高能量的重离子直线粒子加速器^[3,4]。完成这一任务需要有能够进行强电流模拟的大规模、高性能的并行软件来进行数值模拟研究。数值模拟已经在粒子加速器的设计和优化中发挥了越来越重要的作用。ADS 项目中直线加速器的峰值电流大约是 20mA。每个加速粒子束团中有 7.7×10^8 个粒子, 这些粒子需要在数百米的装置中同时进行加速模拟。

模拟这样数量的带电粒子对于现有的传统的串行计算机的计算能力是一个巨大的挑战, 因此只能用一个粒子来模拟多个真实粒子^[5]。为此, 我们已经开发了基于 CPU 并行的束流模拟软件, 但是它们仍然不能满足模拟实际数目粒子的要求^[6,7]。由于基于 MIC 的天河二号超级计算机是目前世界上最快的机器, 使用天河二号将可以大大增加模拟粒子的数目和计算速度, 更加精确地模拟带电束团的运动。这对于加速器的设计和控制运行都有非常重要的意义。因此研究 MIC 编程环境下的数值模拟方法, 将模拟过程尽量加速将是利用这一技术的关键, 本文主要就是进行这方面的研究。

直线加速器(LINAC)中大规模带电粒子的运动模拟对强电流直线加速器的设计和优化至关重要。束流模拟主要是研究带电粒子束团在复杂的外部电场和相互之间的空间电荷效应作用下运动的状态, 改进加速器的设计, 提高粒子束团的质量。粒子束团的质量主要是指束团的分布集中程度, 即散射度的大小。为使束团的散射度尽可能小, 必须采用合理的加速器件设计、优化的加速器的结构, 即使粒子束团获得尽可能多的能量, 又让它能够保持比较集中的形状。模拟尽可能多的粒子对于逼近真实束团运动状态时有益的, 这对于完成优化设计也是有益的。带电粒子束团一旦聚焦性能不好, 就会出现晕区, 也就是粒子分布稀少的区域。这些晕区的粒子将会与金属壁面碰撞而损失掉, 这对于加速器的运行是非常有害的。只有模拟了真实粒子数目的束团, 才能够准确地模拟出这些晕区的位置和大小。使用天河二号的意义之一正在于此。另外, 加速器的设计往往具有非常多的参数, 不同参

数的组合可能成千上万。对每一种参数组合需要进行束流模拟, 研究束团在达到高能量后的形状。这就需要进行成千上万次的模拟, 采用天河二号对不同的参数组合进行并行计算可以大大加快计算的速度, 提高设计的质量, 这对于加速器的制造具有重要的意义。

束流动力学模拟软件已经由许多加速器物理学家开发并使用, 例如美国的很多加速器物理国家实验室(阿贡国家实验室、斯坦福加速器中心、费米国家实验室等)和许多国外大学都有自己开发的模拟软件。基于 CPU 版本的并行束流动力学模拟软件也已经被广泛使用, 但是目前尚没有基于 MIC 版本的束流模拟软件。本文的研究正是在这方面的尝试。

2 物理模型和数值方法

2.1 物理模型

束流模拟需要求解粒子的动力学运动方程, 本文采用的是 PIC 算法, 并采用泊松方程描述空间电荷效应。这种算法采用宏粒子来代替真实的粒子, 并且不直接计算不同宏粒子间的相互作用, 而是将所有粒子的电荷分配到一个固定的网格上, 然后在这个网格上求解描述静电场的泊松方程, 最后将得到的电场力反作用到带电粒子上。

粒子的运动由牛顿运动定律给出:

$$\mathbf{F} = \frac{d\mathbf{p}}{dt} = m \frac{d(\gamma\mathbf{v})}{dt} \quad (1)$$

其中 $\mathbf{p} = \gamma m \mathbf{v}$ 是一个粒子的相对动量, γ 是相对质量因子。在电场 $\mathbf{E}$ 和磁场 $\mathbf{B}$ 中一个带电 q 和速度为 $\mathbf{v}$ 的粒子的洛伦兹力 $\mathbf{F}$ 由下面公式给出 $\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$

粒子的坐标 $\mathbf{x}$ 和速度的关系为 $d\mathbf{x}/dt = \mathbf{v}$, $\mathbf{E}$, $\mathbf{B}$ 和 $\mathbf{v}$ 都是粒子坐标变量 $\mathbf{x}$ 和时间变量 t 的数学函数。不考虑 LINAC 设备和带电粒子之间的电磁相互作用力, 作用在粒子上的电磁场是外场 $\mathbf{E}_{ext}$, $\mathbf{B}_{ext}$, 和粒子束的空间电荷场, $\mathbf{E}_{sc}$, $\mathbf{B}_{sc}$ 的叠加结果 $\mathbf{E} = \mathbf{E}_{ext} + \mathbf{E}_{sc}$, $\mathbf{B} = \mathbf{B}_{ext} + \mathbf{B}_{sc}$ 。

速度比光速慢得多的粒子的空间电荷场的影响比速度接近光速的粒子的空间电荷场重要很多, 特别是针对高电流 LINAC。因此空间电荷场只有在 LINAC 的低能量传输和加速部分中才会被考虑。另外, 因为粒子的速度低, 带电粒子衍生的电磁场接近一个静电场。因此, 空间电荷场能够通过求解泊松方程得到:

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0}, E_{sc} = -\nabla \phi, \quad (2)$$

$$\nabla^2 \mathbf{A} = -\mu_0 \mathbf{J}, \quad \mathbf{B}_{sc} = \nabla \times \mathbf{A}, \quad (3)$$

其中 ρ 电荷密度, $\mathbf{J}$ 是电流密度, ϕ 是标量势, $\mathbf{A}$ 是矢量势, ϵ_0 是真空介电常数, μ_0 是真空磁导率. 值得注意的是, 公式(3)和 $\mathbf{B}_{sc}$ 在程序 NEWBEAM 没有用到, 因为 $\mathbf{B}_{sc}$ 比 $\mathbf{E}_{sc}$ 小很多以致可以忽略不计.

下图 1 是束流软件模拟流程图.

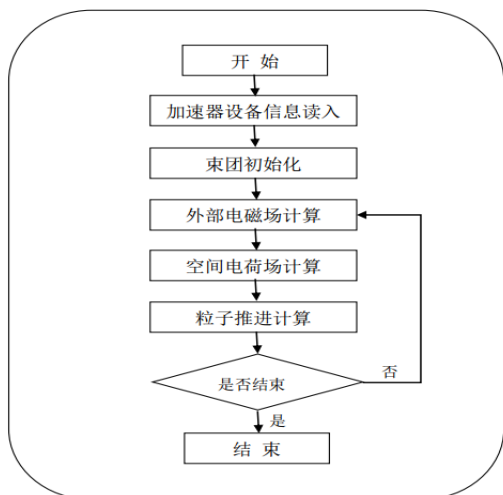


图1 束流软件流程图

2.2 数值方法

粒子的运动轨迹由公式(1)沿时间进行积分获得, 我们采用了文献[3]中提出的时间分裂格式. 这种格式能够保证粒子在相空间中的体积守恒, 使得粒子的运动符合实际的物理规律.

在公式(2)中, 所有的物理量是空间连续的. 但是粒子电荷的分布是空间离散的. 在 PIC 方法中, 粒子束团中电荷需要分布到三维的结构网格上. 然后在这个网格上采用快速傅里叶变换(FFT)进行求解.

公式(2)在步骤 2 中计算. 电荷密度、标量势和空间电荷场三者的谱关系可以从公式(2)中获得.

$$\hat{\phi} = \frac{\hat{\rho}}{\omega_x^2 + \omega_y^2 + \omega_z^2}, \quad (4)$$

$$\hat{E}_{sc,x} = -j\omega_x \hat{\phi}, \quad \hat{E}_{sc,y} = -j\omega_y \hat{\phi}, \quad \hat{E}_{sc,z} = -j\omega_z \hat{\phi}, \quad (5)$$

空间电荷场通过 FFT 的逆变换来获得. 值得注意的是, FFT 只需要网格节点上的样本数据而不是整个空间域. 电荷密度样本通过步骤 1 中的空间电荷的插值获得, 粒子所在位置的空间电荷场通过步骤 3 中的网格节点的空间电荷样本插值来获得.

NEWBEAM-MIC 的每个基本时间步的流程图如下图所示. 步骤 1 到步骤 3 是 PIC 方法的核心, 用来计算每个粒子所在位置的外电场. 步骤 4 通过外电场网格的插值或者直接通过计算解析公式来完成.

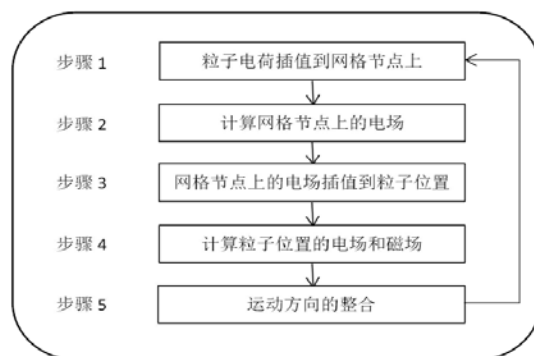


图2 基本时间积分步的示意图

3 并行算法

束流模拟 MIC 异构并行软件 NEWBEAM-MIC 是基于 NEWBEAM 的 CPU 版本改进的使用 CPU 和 MIC 的粒子动力学模拟异构并行软件. 它基于 MIC 编程架构、MPI 类库、OPENMP 技术和面向对象编程语言 C++ 开发. 目标是有效地利用现代超级异构(MIC)计算机的性能, 例如天河二号超级计算机. 正如图 1 所示, 程序主要分为空间电磁场的计算、泊松方程的计算及推进粒子计算这三个模块. 在推进粒子的过程中, 所有的粒子是被平均分布到各个 MIC 卡上的, 每一块 MIC 卡保存一部分粒子的完整信息, 每一步推进中, MIC 卡将少量粒子信息传输到 CPU 端进行间接 MIC 间通信获得所有粒子的平均值, 再将这个平均值传回到各 MIC 卡中进行下一步运算.

进行 CPU+MIC 异构并程序的开发, 需要首先了解和选择 CPU 和 MIC 间的应用模式. MIC 编程方式极其灵活, MIC 卡既可以作为一个协处理器存在, 也可以被看到独立的节点. CPU 与 MIC 的工作配合模式可以简单划分为三类, 即以 CPU 为中心、以 MIC 为中心以及 CPU 与 MIC 对等 3 种模式. 考虑到程序的通用性, 软件采用以 CPU 为中心, MIC 协同计算的应用模式. 程序在 CPU 端启动, 在高并行计算部分通过 offload 语句交给 MIC 计算, 并行计算部分主要运行于 MIC 卡上. 其中 offload 语句用于 CPU 与 MIC 的主从模式, 这种方式在实际使用中也覆盖到了 90% 以上的 MIC. 由于 MIC 资源比较紧张且稀缺, 为了使 CPU 与

MIC 间负载均衡并将每个节点上的 MIC 卡充分利用, 软件总是选择用一个 CPU 进程控制一块 MIC 卡, 开多个 CPU 进程, 并在每块 MIC 卡中再通过 OPENMP 技术开启多线程来达到加速目的^[8-11].

虽然在理论上 MIC 技术提供 MIC 间直接通信, 但在实际中这种技术使用仍然有限, 本算法中使用 CPU 间用 MPI 进行直接通信, MIC 卡之间通过控制它们的 CPU 实现间接通信, 这种并行模型如图 3.

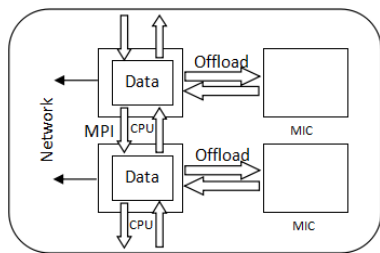


图 3 并行模型

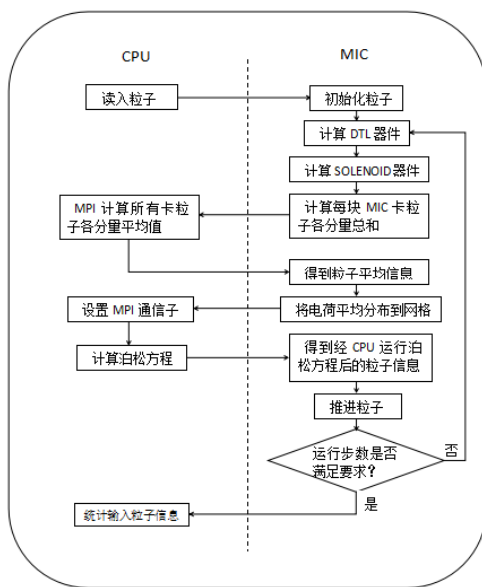


图 4 完整程序流程图

首先, 程序由 CPU 端启动, 初始化 MPI 进程, 各处理器分别读入粒子信息文件, 将读入的信息通过 offload 语句传输到 MIC 端, MIC 端利用这些信息初始化粒子, 然后开始迭代计算. 粒子分别经过计算 DTL 器件以及 SOLENOID 器件, 每块 MIC 卡各自负责的粒子信息都有所改变, 此时将每块 MIC 卡中粒子的每一种分量进行加和, 连同粒子数目, 一起传回 CPU 端, 在 CPU 端通过 MPI 计算所有卡所有粒子每个分量的平均

值, 计算后每一个 CPU 进程将此平均值传输给它所控制的 MIC 卡. MIC 卡得到此信息进行下一步的计算. 其次, MIC 卡将电荷平均分布到网格上, 并把信息传输到 CPU 端, 由 CPU 端进行泊松方程的计算, 并将计算后的结果传回到 MIC 端. 最后, MIC 端进行推进粒子, 如果粒子推进步数达到事先设定好的上限, 则交给 CPU 端统计并输出粒子信息, 否则, 进行下一步迭代运算.

程序完整的运算流程, 如图 4.

4 验证与并行测试

软件的开发和测试是在广州超算中心的天河二号超级计算机上进行的. 天河二号在硬件配置方面能够提供良好的条件, 它的计算阵列包含 16,000 个计算结点. 每个计算结点包含 2 块 12 核的 CPU 卡和 3 块 57 核 MIC 卡, 每个结点有 64GB 主存, 每个 MIC 卡有 8GB 内存, 每个结点共计 88GB 的容量.

为了测试在使用单 CPU 进程单 MIC 卡的单多线程、多 CPU 多 MIC 卡的单多线程等几种不同情况下程序的加速效果, 测试对不同粒子规模, 采用了不同 CPU 和 MIC 卡数, 每块卡中开不同数量的线程的多种组合模式. 以下展示了最具代表性的 3 个测试, 包括单卡多线程测试、多卡单线程测试及多卡多线程测试. 为了使负载均衡, 以下测试均采用每 1 核 CPU 控制 1 块 MIC 卡的数量映射.

文章引入理论加速比、实际加速比及有效加速比率等概念, 用来直观的比较加速效果, 其中:

理论加速比(TAR) = 使用的进程数(或线程数)/1 ;
实际加速比(RAR) = 使用一个进程(线程)运行所花时间 / 使用多个进程(线程)运行所花时间; 有效加速比(EAR) = 实际加速比(RAR)/理论加速比(TAR).

单卡多线程测试. 对计算外电场部分中的不同器件: DTL 器件和 SOLENOID 器件及推进粒子部分进行了单核 CPU+单块 MIC 卡的测试, 网格划分采用 $32 \times 32 \times 32$ 规模, 粒子数目为 10^6 的规模. 下面是用单核 CPU 控制单卡 MIC, MIC 卡分别开 1 个线程和 224(单卡 MIC 支持的最大线程数)进行的测试. 结果证明网格数不变, 粒子数不变, 单卡情况下, 线程数开得越大, 并行效率越高, 程序运行 10 步的每一步平均运行时间及在 MIC 卡中开启 224 个线程的实际加速比 (RAR)测试结果显示如下:

表1 10^6 粒子用 1CPU+1MIC 处理器在 $32 \times 32 \times 32$ 网格上的并行效率

器件/T	1	224	RAR
DTL	28.1	0.23	122.2
SOLENOID	3.39	0.04	83.82
推进粒子	10.6	0.11	97.46

从表 1 可以看到, 当网格数和粒子规模数不变时, 使用单卡时开 224 线程最多可达到 122 倍的加速效果.

其次, 多卡多线程测试. 对计算外电场部分中的不同器件: DTL 器件和 SOLENOID 器件、泊松方程及推进粒子部分, 进行了多核 CPU+多块 MIC 卡的测试, 每块 MIC 卡采用单线程, 网格划分采用 $32 \times 32 \times 32$ 规模, 粒子数目为 10^7 的规模. 测试比较在相同粒子规模和网格数规模的情况下不同数目 CPU 和 MIC 卡的加速效果. 下面即是使用 $32 \times 32 \times 32$ 规模的网格, 10^7 规模的粒子, 分别用 1CPU+1MIC, 2CPU+2MIC, 4CPU+4MIC 进行测试的结果. 程序推进 10 步, 每一步计算分别的平均时间及加速效果统计, 如下表:

表2 10^7 粒子用 nCPU+nMIC 处理器在 $32 \times 32 \times 32$ 网格上的并行效率

nCPU+nMIC	n=1	n=2	n=4
外电场	57.45	29.36	14.72
泊松方程	78.43	40.29	20.83
推进粒子	1.389	0.568	0.284
RAR	1	1.96	3.90
EAR	1	98%	99.5%

从表 2 可以看到, 当粒子数和网格大小不变, 在小范围内采用不同数量处理器时, 处理器数量越多, 实际加速比越大. 其中有效加速比率高达 98%及以上, 并且它并不会随着处理器个数的增加而下降, 加速效果非常理想.

最后, 多卡多线程测试. 对计算外电场部分中的不同器件: DTL 器件和 SOLENOID 器件及推进粒子部分, 进行了多核 CPU+多块 MIC 卡的测试, 每块 MIC 卡采用不同线程, 网格划分采用 $32 \times 32 \times 32$ 规模, 粒子数目为 10^9 的规模. 测试比较在相同粒子规模和网格数规模的情况下, 不同数目 CPU 和 MIC 卡的加速效果. 下表为使用 $32 \times 32 \times 32$ 规模的网格, 用 100CPU+100MIC 处理器, 计算 10^9 规模粒子所测试的加速效果:

表3 10^9 粒子用 100CPU+100MIC 处理器在 $32 \times 32 \times 32$ 网格上的并行效率

thread 数	1	4	16	64	224
DTL	503	166	41.4	12.5	4.71
SOLENOID	44.2	12.7	3.17	1.00	0.53
计算外电场	667	190	57.5	26.3	17.80
推进粒子	133	37.4	9.68	3.10	1.522
RAR	1	3.57	13.9	44.4	106.7
EAR(%)	100	89	87	69	47

从表 3 可以看到, 当粒子数和网格大小不变, 采用多处理器, 每块 MIC 卡开不同数目的线程时, 线程数目越多, 加速倍数越高. 当线程数开到单块 MIC 卡能支持的最大线程数 224 时, 程序的实际加速效果可以达到 106 倍. 与使用多进程加速不同, 在每块 MIC 卡中使用多线程加速, 有效加速比随着线程数量增加而减少.

结合表 1 和表 3 可以得出结论, 同是 10^9 粒子, 用 $32 \times 32 \times 32$ 的网格, 对比使用单核 CPU+单卡 MIC(单线程)的情况, 使用 100 核 CPU+100 块 MIC(224 线程), 将总计达到 10000 倍以上的加速效果.

图 5-图 8 分别给出了程序计算外电场中 DTL 器件、SOLENOID 器件、整个外电场及计算推进粒子部分的不同线程数花费时间的对比图. 其中在各图中, 红色线代表理论加速比的值, 绿色线代表实际加速比的值, 横坐标表示每块 MIC 卡所开线程数, 测试中取 5 个点, 分别为 1、4、16、64、224 个线程不等, 纵坐标代表相应模块运行所花费的时间(s).

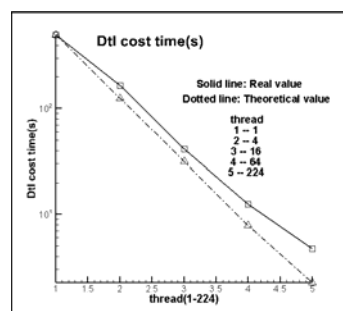


图5 DTL 器件运行时间随线程增加而下降

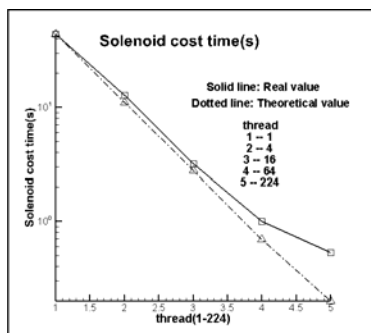


图 6 SOLENOID 器件运行时间随线程增加而下降

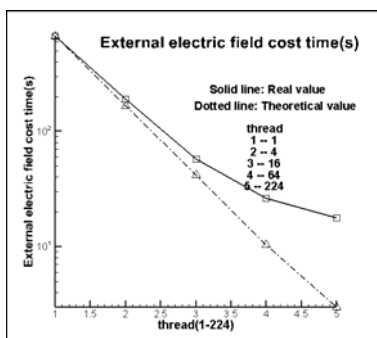


图 7 计算外电场时间随线程增加而下降

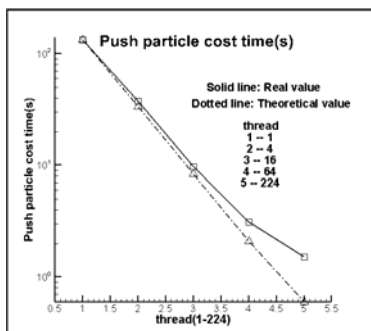


图 8 推进粒子时间随线程增加而下降

5 DTL模拟

利用 NEWBEAM-MIC, 本文模拟了一个用于低能段粒子加速的漂移管加速器(DTL), 它由157个缝隙组成, 束流电流 0.25 mA, 相位差是-30.0 度, 初始能量 150keV, 出口能量 3MeV, 长度近 2.29m. 在粒子初始化阶段, NEWBEAM 使用的参数包括: KV 分布, 束流包络大小为 0.15227 cm, 束流包络沿运动方向的斜率 0.0.

图 9 显示了在 DTL 中沿 z 方向粒子束团的能量变化, 可以看到束团的能量增加, 说明粒子在 DTL 中不断被加速, 且能量达到了接近 3.0 MeV. 这与设计相符合, 图 10 中的发射度变化曲线说明在入口处粒子分布参数设置正确的情况下, 该 DTL 能较好地保持粒子发

射度在一定范围内, 没有显著增大. 这说明 DTL 具有较好的加速结构, 参数是合适的.

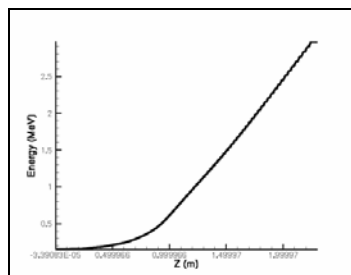


图 9 粒子能量沿 z 方向的变化

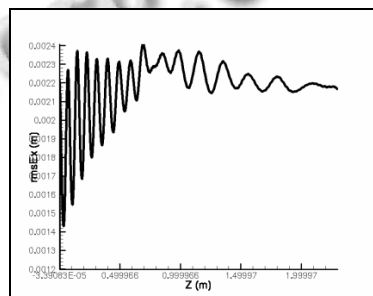


图 10 沿 z 方向(x,x')发射度变化

6 结论

这篇文章介绍了近期开发的基于 MIC 的异构直线加速器并行束流模拟软件(NEWBEAM-MIC)的工作进展, 它是为了能够有效地利用现有的异构 MIC 超级计算机, 实现程序加速效果的有效提升. NEWBEAM-MIC 是基于 PIC 方法并且通过求解泊松方程来描述空间电荷效应的软件. 首先进行粒子初始化, 粒子被均匀地分配到所有的 MIC 处理器, 经过器件运算后, 传输到 CPU 端, 在 CPU 端进行并行求解泊松方程, 求解泊松方程后再将数据传输到 MIC 端进行推进粒子运算. 文中给出了异构环境下的运算 DTL 器件和 SOLENOID 器件、泊松方程和程序总体并行效率的测试. 测试分别给出了单卡多线程、多卡单线程、多卡多线程等几种不同模式下的并行效率测试结果. 这些结果说明 NEWBEAM 具有很好的加速潜力, 可以有效地利用异构超级计算机的计算能力.

致谢

本文是在中山大学国家超级计算广州中心的天河二号计算机上调试开发的, 得到了中心系统管理员钟英、刘庆等的大力协助, 在此表示感谢. 希望能进一步同中心合作, 取得更好的研究成果.

参 考 文 献

- 1 Zhang Y, Kang X, Li M, Wu J, Jia H, Zhu G. Design and test status of beam position monitors for ADS injector II proton linac. Proc. of the 4th International Particle Accelerator Conference. Geneva: JACoW. 2013. 574–576.
- 2 Aseev VN, Ostroumov PN, Lessner ES, Mustapha B. TRACK: the new beam dynamics code. C. Horak. Proc. of the 2005 Particle Accelerator Conference. Piscataway, NJ. IEEE. 2005. 2053–2055.
- 3 Batygin YK. Particle-in-cell code BEAMPATH for beam dynamics simulations in linear accelerators and beamlines. Nucl. Instr. and Meth. Phys. Res. A, 2005, 539(3): 455–489.
- 4 Qiang J, Ryne RD, Habib S, Decyk V. An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators. J. Comput. Phys., 2000, 163(2) 434–451
- 5 Plimpton SJ, Seidel DB, Pasik MF, et al. A load-balancing algorithm for a parallel electromagnetic particle-in-cell code. Comp. Phys. Comm., 2003, 152: 227–241.
- 6 Zhao R, Xu J, Li Ch, Qi X, Yang L, He Y. A new scalable software package for large scale beam dynamic simulations. Z. Dai et al. Proc. of the 4th International Particle Accelerator Conference. Geneva: JACoW, 2013, 912–914.
- 7 Xu J, Mustapha B, Aseev VN, Ostroumov PN. Parallelization of a Beam Dynamics Code and First Large Scale RFQ Simulations, Phys. Rev. ST Accel. Beams 10, 014201 (2007).
- 8 Sanders J, Kandrot E. CUDA BY EXAMPLE An Introduction to General-Purpose GPU Programming. Addison-Wesley, 2010.
- 9 王恩东,等.MIC 高性能编程指南.北京:中国水利水电出版社,2012.
- 10 Intel Xeon Phi 协处理器高性能编程指南.北京:人民邮电出版社,2013.
- 11 罗秋明,等著.OpenMP 编译原理及实现技术.北京:清华大学出版社,2012.