

# 面向数千处理器的三维等离子体粒子模拟程序研制<sup>\*</sup>

曹小林<sup>\*\*</sup> 郑春阳 张爱清 朱少平 莫则尧 肖 丽

北京应用物理与计算数学研究所, 北京 100088

**摘要** 等离子体粒子模拟是研究激光聚变中激光与等离子体相互作用的有效手段. 有些应用需要模拟包含上百亿粒子的模型, 其中粒子分布不均匀且动态运动. 为满足需求, 研制了可扩展到数千处理器的三维等离子体粒子模拟程序, 其关键技术包括基于网格片的负载平衡策略、粒子数据结构 and 数据通信的封装、大规模数据的可视化等. 在某并行机的 2500 个处理器上, 该程序成功模拟了短脉冲激光与锥壁等离子的相互作用, 其中粒子数达到 12 G, 模拟 5000 时间步花费 15 h, 共输出 50 GB 的三维数据. 这些数据可直接在并行可视化软件上显示, 有助于物理学家深入理解模拟结果.

**关键词** 等离子体粒子模拟 数千处理器 网格片 负载平衡

受控核聚变是解决人类能源问题的有效途径. 为了获得高增益聚变, 物理学家提出了激光聚变的新途径——“快点火”方案. 该方案将压缩与点火两个过程分开进行研究, 大大降低了对激光辐照对称性和驱动能量的要求. 该方案关键之一是深入理解超强激光在低密度等离子体中的传播特征以及复杂的激光等离子体相互作用过程. 等离子体粒子模拟是研究这些过程的有效手段.

等离子体粒子模拟方法是指, 通过追踪大量的在自洽和外加电磁场作用下的带电粒子的运动, 研究等离子体集体性质的动力学模拟方法. 它可以描述强激光与等离子体相互作用中的非线性过程, 主要用于预测、设计、解释实验结果. 在模拟与真实物理过程比较接近的物理模型时, 该方法面临的并行计算问题是:

(1) 计算规模大. 模型边长为几十个激光波长. 离散后的计算区域约包含  $1000 \times 1000 \times 1000$  个单元, 粒子数超过 10 G, 内存需求达到 1 T 字节. 这需要在并行机的数千个处理器上进行高效的模拟,

而且还需要对输出的大规模数据进行可视化分析.

(2) 负载不平衡. 在有些实际模型中, 粒子分布不均匀且动态运动通常导致处理器间负载不平衡. 我们将具有这些特征的模型称之为非均匀负载模型. 对于非均匀负载模型, 需要根据粒子的分布和运动, 在处理器间动态均衡地分配粒子.

针对以上问题, 国外的研究团体提出了一些并行算法, 并研制了可扩展到几千个处理器的等离子体粒子模拟并行程序, 如 VORPAL<sup>[1]</sup>, QuickPIC<sup>[2]</sup>, VPIC<sup>[3]</sup>, GTC<sup>[4]</sup> 等. VORPAL 通过自适应地重分计算区域实现负载平衡. QuickPIC 采用了一维动态负载平衡方法, 可在几千个处理器上模拟包含 20 G 个粒子的模型. GTC 采用区域分解和粒子分解相结合的方法, 可在并行机 Cray XT3 的 4800 个处理器上包含模拟 28 G 个粒子的模型. 这些方法能较好地处理单方向的负载不平衡问题, 但难以适应更复杂的负载不平衡需求. 国内的研究团体主要采用基于区域分解的并行算法<sup>[5]</sup>, 但对负载不平衡考虑较少. 因此, 需要研究新的负载平衡策略, 而

2008-06-18 收稿, 2008-07-28 收修改稿

<sup>\*</sup> 国家自然科学基金(批准号: 60603050, 90718029)和国家重点基础研究发展计划(批准号: 2005CB321702, 2007CB814802)资助项目

<sup>\*\*</sup> E-mail: xiaolincao@iapcm.ac.cn

且设计与之匹配的高效数据结构, 以提高并行程序的可扩展性能。

几年前, 在 MPI 环境下我们开发了基于区域分解的二维等离子体粒子模拟并行程序<sup>[6]</sup>, 后来又发展为三维并行程序 LAPRED-P. 该程序在模拟均匀负载模型时, 具有较好的并行性能. 但是, 在模拟非均匀负载模型时, 难以扩展到几千个处理器. 因此, 针对以上问题, 在前人基础上, 我们提出新的负载平衡策略和并行算法, 重新研制三维等离子体粒子模拟并行程序, 以满足应用需求。

## 1 串行算法

等离子体粒子模拟的基本方程包括 3 部分: 描述电磁场随时间变化的 Maxwell 方程组(简称方程 1), 描述带电粒子运动的相对论 Newton-Lorentz 方程组(简称方程 2), 描述以上两个方程组耦合的粒子云方程组(简称方程 3). 这些方程组所涉及的公式及其求解方法请参考文献[5, 6], 以下简要阐述求解过程及其中的数据依赖关系, 包括:

(1) 离散化. 如图 1 所示, 将计算区域离散为均匀矩形结构网格, 其中单元边长在各方向相等. 为便于描述, 假设电场、磁场等物理量均定义在网格结点(大圆点). 在实际模拟中, 它们还可能定义在网格中心、面心或棱心上. 网格单元内包含多个粒子(电子或离子), 单元间的粒子数可能不一致, 而且粒子会在单元之间运动。

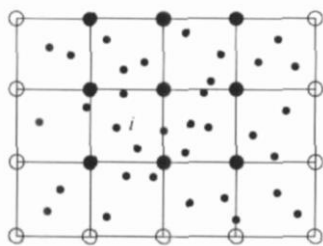


图 1 等离子体粒子模拟示意图

(2) 求解方程 1. 基于 Yee 格式, 采用具有二阶精度的中心差分法求解. 在三维计算中, 单个电场分量的更新依赖于其周围环绕的 4 个磁场分量和同一位置的电流密度; 单个磁场分量的更新依赖于其周围环绕的 4 个电场分量。

(3) 求解方程 2. 首先, 通过二阶 Taylor 展开, 由网格上的电磁场近似计算得到粒子所在位置

处的电磁场; 然后, 依据该电磁场计算粒子所受的 Lorentz 力; 最后, 用时间上的二阶中心差分方法求解 Newton 运动方程得到粒子的新位置. 如图 2 所示, 在二维时, 计算粒子  $i$  处的电磁场 ( $E_i, B_i$ ) 需要使用周围 9 个网格结点(图中的大黑点)上的电磁场; 在三维计算时, 需要周围的 27 个. 依据 ( $E_i, B_i$ ) 和粒子自身信息就可计算粒子  $i$  所受的 Lorentz 力. 在最后一步中, 更新粒子位置仅依赖于粒子自身信息。

(4) 求解方程 3. 通过加权平均计算粒子对周围网格上电流密度和电荷密度的贡献. 如图 2 所示, 在二维时, 粒子  $i$  只对周围 9 个网格结点(图中的大黑点)有贡献; 在三维计算时, 对周围的 27 个有贡献。

由以上分析可知, 只有相邻网格之间存在数据依赖关系, 因此可基于区域分解方法实现并行计算. 当粒子分布较均匀且运动幅度小时, 该方法的并行效率比较好. 但是, 当粒子分布不均匀且动态运动时, 该方法会产生负载不平衡. 而且, 随着粒子规模增大和所用处理器数增多, 负载不平衡越来越严重, 从而成为大规模并行计算的瓶颈。

为解决负载不平衡问题和提高程序的质量, 我们基于 JASMIN 框架重新研制等离子体粒子模拟并行程序. JASMIN 框架由北京应用物理与计算数学研究所研制, 其全称为并行自适应结构网格应用支撑软件框架<sup>[8]</sup>. 它围绕科学计算中普遍采用的结构网格, 屏蔽高性能计算和自适应计算的实现, 支撑创新计算方法和高性能算法的研究, 支撑可有效使用数千台处理器的高效并行应用程序的研制. 它基于层次化、模块化和面向对象的软件构建, 是一个不断发展和完善的支撑软件框架. 围绕等离子体粒子模拟程序研制所作的工作包括并行算法的设计, 并行程序的研制等。

## 2 并行算法

为解决负载不平衡问题, 首先介绍网格片<sup>[7, 8]</sup>概念; 然后, 描述网格片分配算法; 最后, 阐述基于网格片的负载平衡策略和并行算法。

### 2.1 网格片

文献[8]详细介绍了网格片及其相关的概念. 现

结合等离子粒子模拟, 简要描述本文中用到的几个概念: 网格层、box 和网格片. 由第 1 节可知, 等离子粒子模拟采用了均匀矩形结构网格, 其网格单元可被相应维数的 Cartesian 整数下标唯一地索引, 称该下标为网格单元的逻辑索引; 网格单元之间的相邻关系可以由逻辑索引唯一地确定. 对于这类结构网格, JASMIN 框架通过网格层来管理整个计算区域. 该网格层的索引范围定义为计算区域中所有网格单元的索引集合. 网格层的索引范围可以是一个 box, 也可以是多个 box 的集合. 其中, box 定义为:

$$[(i_0, \dots, i_{D-1}), (j_0, \dots, j_{D-1})] = \{(k_0, \dots, k_{D-1}) : i_d \leq k_d \leq j_d, d = 0, 1, \dots, D-1\} \quad (1)$$

这里  $D$  为网格维数, 方括号中的两部分分别为 box

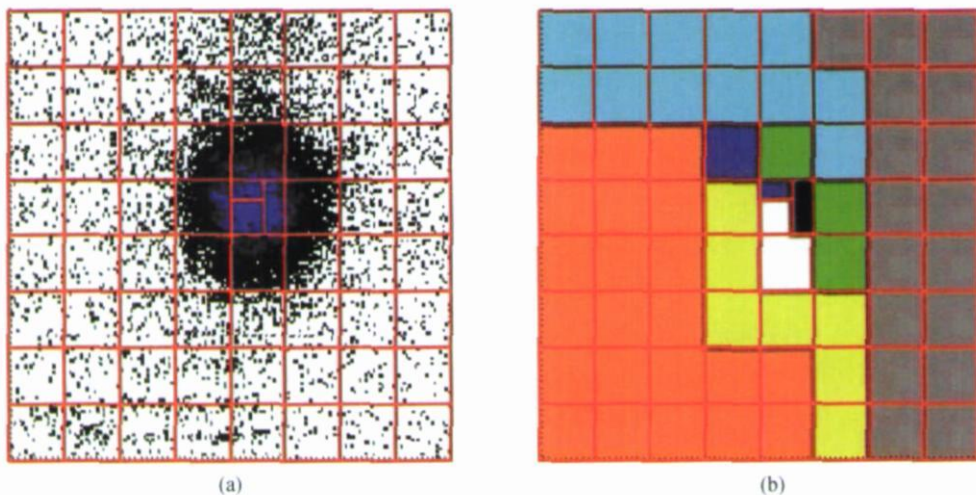


图 2 负载平衡示意图

(a) 将计算区域切割成多个网格片; (b) 分配网格片到 8 个处理器, 其中同颜色的被分配给同一处理器

网格片是一个独立的计算子区域. 它包含计算子区域上的所有网格单元和单元上的物理量值. 但是, 为了完成数值计算通常需要: (i) 基于数据片, 管理定义在网格片上的物理量值; (ii) 基于影像区, 获得定义在相邻网格片的物理量值. 有关数据片和影像区的详细介绍请参看文献[8]. 在 JASMIN 框架中, 物理量的值分布存储在数据片中. 对任意的物理量(如电场、磁场等), 只要为其声明在网格单元的几何位置、类型等, JASMIN 框架就在网格片上为其设置一个数据片, 该数据片存储该物理量定义在网格片网格

的索引下界和索引上界.

在 JASMIN 框架中, 网格层的索引范围被拆分成互不重叠的多个 box, 每个 box 索引一个逻辑矩形的计算区域, 称这些子区域上网格单元及网格单元上所有物理量值的集合为网格片. 网格层可以表示为多个互不重叠的网格片的集合. 每个网格片拥有一个独立的索引 box, 它描述网格片的索引范围. 如图 2 所示, 对于包含  $160 \times 160$  个网格单元的计算区域, 假设其左下角单元的逻辑索引是  $(0, 0)$ , 右上角单元的逻辑索引是  $(159, 159)$ , 那么其网格层的索引范围是  $[(0, 0), (159, 159)]$ . 该网格层被拆分成 66 个 box(图中红色方框), 每个 box 界定的计算子区域是一个网格片, 除了中间的小网格片外, 其他网格片均包含了  $20 \times 20$  个网格单元. 左下角网格片的索引范围为  $[(0, 0), (19, 19)]$ .

单元几何位置上的值. 在数值计算过程中, 网格片通过影像区引用相邻网格片上的物理量值. 为简化这种数据引用, JASMIN 框架为数据片设置影像区, 存放相邻网格片上物理量值. 在某阶段, 当网格片获取所需要的相邻网格片上物理量值后, 那么该网格片上的数据就可以被独立地处理.

## 2.2 网格片分配算法

在 JASMIN 框架中, 网格片是负载平衡的基本单位. 因此, 应用问题中的负载平衡就被转化为如

何将网格层中的所有网格片均衡地分配给处理器, 以达到处理器间负载均衡, 并最小化处理器间的数据通信. 一个处理器可以拥有单个或多个网格片. JASMIN 框架完成并管理这些网格片的分布存储. JASMIN 框架现中的网格片分配算法如下:

**算法 1 网格片分配算法**

输入: 存储网格层索引范围的 box 数组、约束条件(如网格片的最小和最大尺寸等)、网格单元负载、处理器数  $P$ 、剖分方法名.

输出: 存储网格片索引范围的 box 数组、存储网格片所属进程号的整型数组.

(1) 计算 box 的负载阈值  $\eta$ .

$$\eta = \frac{\sum_{i=1}^N w_i}{P} \times \alpha \quad (2)$$

这里  $N$  为网格单元总数,  $w_i$  为第  $i$  个网格单元的负载,  $\alpha$  为最大负载因子.

(2) 递归地切割输入的 box, 确保新产生的 box 上的负载小于负载阈值  $\eta$ .

(3) 重新调整新产生的 box, 使其满足各种约束条件, 最终得到输出的 box 数组.

(4) 计算输出 box 上的负载.

(5) 调用指定的剖分方法, 分配输出 box 对应的网格片到各处理器, 输出网格片所属的进程号.

在分配网格片时, 可以调用比较成熟的剖分方法. 目前, JASMIN 框架提供两类剖分方法, 包括基于空间填充曲线的剖分方法和图剖分方法. 前者包括 Morton 空间填充曲线方法和 Hilbert 空间填充曲线方法; 后者包括 Metis 软件库中的多水平方法和图排序方法. 今后, 我们将集成更多的剖分方法到框架, 以供用户选用.

以上算法对于处理局部子区域上负载过大的问题非常有用. 如图 2(b) 所示, 在计算区域, 包含很多粒子的圆形体在运动中与激光相互作用. 由于圆心处(蓝色部分)粒子密度非常大, 而计算量和粒子密度成正比, 因此圆心处 3 个小网格片的负载之和占总负载的 40%. 如果不切割覆盖 3 个小网格片的原单网格片, 那么使用任何剖分方法得到的负载均衡效率不会超过 31%. 在进行第(2)步处理后, 该

处的网格片被拆分成 3 个小网格片, 并被分配给不同的处理器(如图 2(a)图所示), 分配后的负载均衡效率可以达到 70% 以上

**2.3 基于网格片的负载均衡策略**

在数千处理器上执行大规模粒子模拟时, 需要在初始时刻作静态负载均衡, 其原因是局部子区域上粒子数过多可能导致内存需求超过单机内存, 而导致初始化失败. 因此, 基于网格片分配算法, 设计了初始时刻的静态负载均衡策略, 具体步骤为:

(1) 创建临时网格层. 假设单元负载相同, 基于初始输入数据, 调用网格片分配算法将计算区域切割为多个网格片, 并分配给各处理器, 形成临时网格层.

(2) 预估单元负载. 在临时网格层上, 首先只为存储单元负载的变量分配内存; 然后, 基于物理模型中的初始粒子分布, 各处理器并行地预估本地网格片上单元的负载.

(3) 创建新网格层. 基于预估的单元负载, 调用网格片分配算法重新将计算区域切割为多个网格片, 并分配给各处理器, 形成新网格层.

(4) 初始化新网格层. 在新网格层上, 首先, 为所有物理变量分配内存; 然后, 根据物理模型, 各处理器并行地初始化本地网格片上的物理变量值.

相应地, 设计了模拟过程中的动态负载均衡策略, 具体步骤为:

(1) 计算负载均衡效率. 监控各处理器的计算时间, 计算得到负载均衡效率, 如果其值小于用户给定的阈值, 则执行以下三步.

(2) 计算单元上的负载. 在原网格层上, 根据当前粒子分布和实测信息, 各处理器并行计算本地网格片单元上的负载.

(3) 分配网格片. 基于单元负载, 调用网格片分配算法重新将计算区域切割为多个网格片, 并分配给各处理器, 形成新网格层.

(4) 迁移数据. 根据前后两次的分配信息, 将数据从原网格层传输到新网格层, 随后, 删除原网格层. 在实际传输中, 仅传输网格片大小或所属进程号发生变化的网格片上的数据.

根据以上两种算法, 再结合第 1 节中的串行算法, 设计了如下的并行算法:

### 算法 2 总体并行算法

- (1) 执行静态负载平衡, 初始化网格上的数值解.
- (2) Do  $it=1, NT // NT$  为最大时间步数
  - (a) 并行求解 Maxwell 方程组;
  - (b) 并行求解相对论 Newton-Lorentz 方程组;
  - (c) 并行求解粒子云方程组;
  - (d) if(负载不平衡)执行动态负载平衡;

Enddo

在以上算法中, 并行求解单个方程组的过程包括通信和计算两步. 在通信步, 处理器之间相互通信, 从相邻网格片获取计算阶段所依赖的数据, 并填充在对应数据片的影像区; 在计算步, 每个处理器遍历本地网格片, 独立地更新网格片上的数值解.

## 3 程序设计

JASMIN 框架已集成了以上的算法 1 和两个负载平衡策略, 从而支撑应用程序实现负载平衡. 为了封装与粒子相关的数据结构和数据通信, 我们在 JASMIN 框架内部增添粒子模拟支撑类. 最后, 基于算法 2, 在 JASMIN 框架上研制了等离子体粒子模拟并行程序.

### 3.1 粒子模拟支撑类

粒子模拟支撑类包括粒子量数据片类、粒子量通信构件类. 粒子量数据片类管理网格片上的粒子, 详细介绍请参看文献[9]. 它的特征包括:

(1) 按单元存储和索引粒子. 采用单元紧凑格式存储粒子, 即将同一单元中的粒子存储在一起, 而且粒子之间没有存储空位. 采用该格式有利于提高 Cache 命中率. 而且, 由单元索引号就可访问到指定单元中的所有粒子.

(2) 随粒子数变化, 动态调整内存. 当网格片上的粒子数逐步减少时, 缩减内存空间以节约内存; 当粒子数逐步增多时, 增大内存空间以存储新迁入的粒子.

粒子量通信构件类封装了粒子量通信. 粒子量通信包括迁移粒子和填充影像区两种. 迁移粒子是指, 粒子位置更新后, 将粒子从原来所属的单元迁移到当前所属的单元. 当这两个单元不在同一个网格片时, 需要通过通信将该粒子传输到相邻网格片. 填充影像区是指从相邻网格片获取影像区的数据. 通过调用该类中的函数, 上层应用程序就可完成粒子量通信.

### 3.2 并行程序研制

在实现以上类后, JASMIN 框架就可有效地支撑粒子模拟, 加速该类应用程序的研制. 基于框架提供的数据结构、数据通信和负载平衡, 我们在 JASMIN 框架上研制成功了等离子体粒子模拟并行程序 LARED-P. 以下从用户角度来阐述在 JASMIN 框架上实现此类应用程序的关键之处, 即深入了解 3 个积分类以及它们之间的关系(见图 3), 具体为:

(1) 时间积分算法类. 该类是框架内部类, 用户只需了解其算法和使用方法. 它封装了负载平衡, 而与应用相关的操作通过抽象接口调度网格层时间积分类中的函数完成.

(2) 网格层时间积分类. 该类由用户实现. 它抽象描述网格层上的整体操作. 它包括: 初始化网格层、向前积分一个时间步、计算网格层上所有单元的负载. 对于 LARED-P 程序, 在向前积分中, 按通信需求分解为 3 个阶段, 分别对应算法 2 中的 (a), (b), (c) 三步. 每个阶段包含通信和计算两步. 通信由框架内部完成. 计算是指每个处理器独立地遍历本地网格片, 调用网格片时间积分类函数更新网格片上的数值解.

(3) 网格片时间积分类. 该类由用户实现, 它按串行方式更新单网格片上的数值解. 其中, 与物理模型、计算方法、边界条件相关的计算代码由 F77 语言编写, 形成串行计算子程序, 以便于重用计算代码和领域专家修改. 在计算负载时, 根据单元中的粒子数和物理量数进行估计. 在初始化时, 根据初值条件设置数值解.

在编写完网格层时间积分类和网格片时间积分类后, 再编写一个主函数 main(), 就形成了新的等离子体粒子模拟并行程序 LARED-P. 它包括 5 部分: JASMIN 框架类库、主函数、网格层时间积分类、网

格片时间积分类、串行子程序，其中主函数包含 400 行 C++ 代码，两个时间积分类包含 6000 行 C++ 代码，串行计算子程序包含 4000 行 F77 代码。

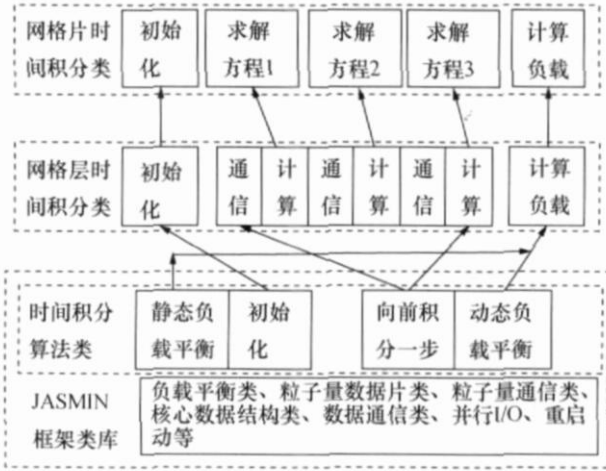


图 3 JASMIN 框架上，三维等离子体粒子模拟程序结构框图

## 4 性能测试

在包含数千个处理器的某并行机上进行了两项测试。一项是在均匀负载模型上，比较原程序(MPI 环境下开发的 LARED-P 程序)和新程序(JASMIN 框架上开发的 LARED-P 程序)的性能，并测试其可扩展性；另一项是在非均匀负载模型上，对新程序进行可扩展测试。

### 4.1 均匀负载模型

由于原程序难以模拟非均匀负载模型，因此设计了一个均匀负载模型比较两者性能。在该模型中，粒子均匀地分布，而且粒子运动幅度小。该模型的网格单元数是  $320 \times 160 \times 160$ ，粒子总数为 0.48 G，在 128 各处理器上共模拟 200 步。在原程序中，计算区域被剖分成 128 个子区域，其大小为  $40 \times 40 \times 40$ ；在新程序中，网格片的大小平均为  $10 \times 10 \times 10$ 。由表 1 可看出，在新程序中，由 JASMIN 框架类引入了 3% 的管理开销；新程序的通信开销比原程序大 14 s，其原因是处理器上拥有多个网格片，所需填充的影像单元总数更多。但是，由于粒子量数据片类封装的数据结构更适用于当前的微处理器体系结构，因此新程序数值计算的性能比原程序高。从总体上看，新程序的性能比原程序稍高一些。

表 1 均匀负载模型，两个程序的性能比较

	总时间/s	数值计算/s	数据通信/s	管理开销/s
新并行程序	635.6	574.3	40.9	20.4
原并行程序	671.1	644.3	26.7	

对于以上均匀负载模型，固定单机上的计算规模，处理器个数从 1—2048 变化，共模拟 1000 个时间步。在 2048 个处理器上，粒子达到 15 G。从表 2 可看出，新程序在 256 个处理器以下，性能均很好。在 2048 个处理器上，由于负载不平衡导致程序的并行性能下降。该负载不平衡主要是由于边界处理、Cache 效应、通信不平衡、剖分质量等原因造成。因此，需要更细致地研究几千个处理器上程序的执行行为，监控各处理器的执行状态，并将这些信息融入到负载平衡方法中，而不是仅从计算方法上粗略地估算单元上的负载。

表 2 均匀负载模型，程序的可扩展性能测试结果

处理器数	粒子数	网格数	时间/s	负载平衡效率/%	并行效率/%
1	$7.3 \times 10^6$	$3.7 \times 10^5$	5782	100	100
4	$2.9 \times 10^7$	$1.5 \times 10^6$	6196	99	93
32	$2.4 \times 10^8$	$1.2 \times 10^7$	6121	97	94
256	$1.9 \times 10^9$	$9.6 \times 10^7$	6458	96	89
2048	$1.5 \times 10^{10}$	$7.7 \times 10^8$	7883	79	73

### 4.2 非均匀负载模型

非均匀负载模型是短脉冲激光在锥体中的传播及其与锥壁等离子相互作用。如图 4 所示，在该模型中，粒子按开口锥体形状非均匀的分布在计算区域，激光在传播过程中与锥壁等离子体相互作用。固定单机上的计算规模，处理器个数从 1—2048 变化，共模拟 1000 个时间步。在 2048 个处理器上，粒子达到 12 G。从表 2 可看出，新程序在 2048 个处理器上，负载平衡效率超过 70%，加速 1000 多倍。当处理器超过 256 时，并行效率较低的原因除了上节中提到的没有考虑实测信息外，还有一个重要原因是：图 3 中的 3 个阶段负载分布不一致。在求解方程二和方程三时，单元上的负载与粒子数成正比；在求解方程一时，单元上的负载相同。因此，难以同时平衡各阶段的负载，需要进一步研究考虑多阶段平衡的剖分方法。

表3 非均匀负载模型, 程序的可扩展性能测试结果

处理器数	粒子数	网格数	时间/s	负载平衡效率/%	并行效率/%
1	$5.6 \times 10^6$	$3.7 \times 10^5$	5706	100	100
4	$2.2 \times 10^7$	$1.5 \times 10^6$	6852	96	83
32	$1.8 \times 10^8$	$1.2 \times 10^7$	8588	84	66
256	$1.4 \times 10^9$	$9.6 \times 10^7$	10429	74	54
2048	$1.2 \times 10^{10}$	$7.7 \times 10^8$	11136	72	51

## 5 实际应用

该程序现已应用于模拟“快点火”方案中的激光等离子体相互作用过程. 在某并行机的2500个处理器上, 模拟了短脉冲激光在锥体中的传播及其与锥壁等离子相互作用. 该模型的计算区域边长是 $140 \times \lambda / 2\pi$ , 其中激光波长 $\lambda = 1.06 \mu\text{m}$ . 它包含 $1000 \times 1000 \times 1000$ 个网格单元, 粒子稠密区上单元平均粒子数100个, 总粒子数达到12G, 模拟5000步约需要15h. 模拟中共输出了50GB的数据. 这些数据可直接在并行可视化软件JaVis上进行可视化显示, 从而有助于物理学家快速分析模拟结果. 图4显示的是5000步上的模拟结果, 从中可清晰地观察到锥对光束的聚焦作用, 其中黄色部分是等离子体密度, 锥壁中间是电场强度. 由图可见, 锥壁等离子体电子在激光作用下的向外运动.

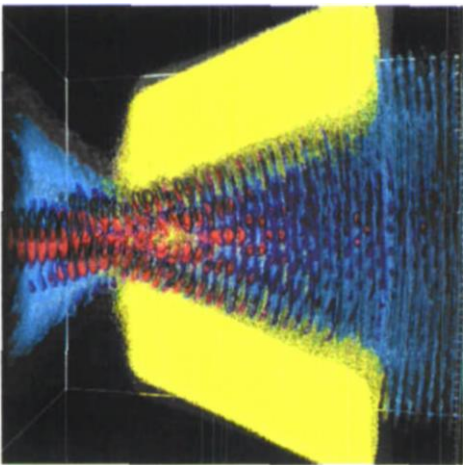


图4 激光与锥壁等离子体相互作用

## 6 结论

本文解决了等离子体粒子模拟中的动态负载不平衡问题, 使得并行程序可在几千个处理器上更高效地模拟与实际物理过程较接近的物理模型. 这些成果已集成到JASMIN框架. 基于JASMIN框架提供的数据结构、数据通信和负载平衡模块, 领域专家只需编写与物理模型、计算方法和边界条件相关的串行子程序, 无须关注并行计算细节, 就可快速研制类似的粒子模拟程序. 下一步, 我们将持续优化改进JASMIN框架底层的负载平衡、数据通信等模块, 支撑上层粒子模拟程序更高效地使用高性能并行机.

## 参考文献

- Nieter C, Cary JR. VORPAL: A versatile plasma simulation code. *Journal of Computational Physics*, 2004, 196(2): 448—473
- Huanga C, Decyk VK, Rena C. QUICKPIC: A highly efficient particle-in-cell code for modeling wakefield acceleration in plasmas. *Journal of Computational Physics*, 2006, 217(2): 658—679
- Bowers KJ, Albright BJ, Yin L, et al. Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation. *Physics of Plasmas*, 2008, 15(5): 1—7
- Ethier S, Tang WM, Walkup R, et al. Large scale gyrokinetic particle simulation of micro-turbulence in magnetically confined plasmas. *IBM Journal of Research and Development*, 2008, 52(1): 105—117
- 马燕云, 常文蔚, 银燕, 等. 三维面向对象的并行粒子模拟程序 PLASIM3D. *计算物理*, 2004, 21(3): 305—311
- 莫则尧, 许林宝, 张宝琳. 二维等离子体模拟粒子云网格方法的并行计算与性能分析. *计算物理*, 1999, 16(5): 496—504
- 莫则尧, 张爱清, 刘青凯, 等. JASMIN 框架用户指南. 北京应用物理与计算数学研究所内部技术文档, 2008, 29—39
- Hornung RD, Kohn SR. Managing application complexity in the SAMRAI object-oriented framework. *Concurrency and Computation: Practice and Experience*, 2002, 14(5): 347—368
- 曹小林, 张爱清, 莫则尧. 基于面向对象的粒子类模拟并行计算研究. *计算机研究与发展*, 2007, 44(10): 1647—1651