

材料微观结构演化大规模分子动力学软件比较*

聂宁明¹⁺, 胡长军², 张云泉³, 贺新福⁴, 张博尧¹, 李士刚³

1. 中国科学院计算机网络信息中心, 北京 100190
2. 北京科技大学计算机与通信工程学院, 北京 100083
3. 中国科学院计算技术研究所, 北京 100190
4. 中国原子能科学院研究院, 北京 102413

Comparison of Large-scale Molecular Dynamics Software for Materials Computing*

NIE Ningming¹⁺, HU Changjun², ZHANG Yuquan³, HE Xinfu⁴, ZHANG Boyao¹, LI Shigang³

1. Computer Network Information Center, Chinese Academy of Science, Beijing, 100190, China
 2. School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing 100083, China
 3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
 4. China Institute of Atomic Energy, Beijing 102413, China
- + Corresponding author: E-mail: nienm@sccas.cn

NIE Ningming, HU Changjun, ZHANG Yuquan, et al. Comparison of large-scale molecular dynamics software for materials computing. Journal of Frontiers of Computer Science and Technology, 2000, 0(0): 1-000.

Abstract: Molecular dynamics simulation is an important tool for studying materials microstructure evolution under radiation effects. Four kinds of mainstream large-scale molecular dynamics open source software for materials computing, LAMMPS, Ls1-MarDyn, IMD and CoMD, are introduced in detail in this paper, with data structures, computational methods, parallel decomposition methods and atomic storage, etc. are compared and analyzed. By the design examples, computational efficiency, parallel performance and memory usage have been tested. A new idea of less memory-used data structure design for the single crystal is prepared for the simulation characteristic of metal materials evolution under irradiation defects. It provides the basis for subsequent research to realize large temporal and spatial scale molecular dynamics simulation.

Key words: Molecular Dynamics; Large-scale; Data Structure; Material Irradiation Damage

摘 要: 分子动力学模拟是研究材料辐照效应下微观结构演化的重要工具。本文对四款主流材料计算大规模分子动力学开源软件 LAMMPS, Ls1-MarDyn, IMD 和 CoMD 进行了详细的介绍, 从数据结构、计算方法、并行分解方式、原子存储等多个方面进行横向分析比较。通过设计算例测试了各软件的计算效率, 并行性能和内存使用情况。后续将针对压力容器关键材料辐照缺陷演化计算特点, 提出设计节省存储的面向单晶的新型数据结构的设想, 为后续实现大时空尺度分子动力学模拟研究提供了研究基础。

关键词: 分子动力学模拟; 大规模; 数据结构; 材料辐照损伤

文献标志码: A **中图分类号:** TP391

*The National Natural Science Foundation of China under Grant No. 61303050, 国家自然科学基金; the Hi-Tech Research and Development Program (863) of China under Grant No. 2015AA01A303, 国家高技术研究发展计划 863; the Youth Innovation Promotion Association, CAS under Grant No. 2015375, 中国科学院青年创新促进会; Prospective Science and Technology Project of State Grid Corporation of China (2013) under Grant No. XXH2503-02, 中国科学院信息化专项“面向云服务的超级计算环境建设与应用”。

1 引言

材料辐照损伤研究涉及从原子尺度到宏观尺度的 9 个数量级的跨越, 以及皮秒量级原子键断裂过程到几十年工程结构失效和破坏的跨时间尺度的非线性过程。这一演化和发展过程在实验很难观测和发现。因此, 目前为止, 材料工作者对材料辐照损伤过程中内部微观过程和演化机理尚缺乏深入的了解。高性能计算机技术的发展使人们从微观层次深入理解材料成为可能, 成为当今国际材料界的重要研究领域。分子动力学模拟作为原子尺度模拟的经典计算方法, 在材料微观结构演化规律研究中处于重要地位。

分子动力学模拟(moleculardynamics simulation, MD)是通过利用计算机求解体系内所有粒子的运动方程来模拟粒子的运动轨迹, 从而获得系统的温度、体积、压力、应力等宏观量和微观过程量。从上世纪五十年代发展以来[1,2], 已经在包括物理, 计算化学, 计算生物, 材料科学, 药物设计等多个领域有了十分丰富的应用。分子动力学模拟作为一种非常有效的材料计算技术, 已成为与实验同等重要的科学研究的方法。

由于计算能力的限制, 目前分子动力学通常模拟的是几万到几千万个原子条件下材料的内部演化过程, 而通常材料的微观结构及缺陷尺寸远远大于现有计算能力所实现的计算规模, 计算模拟结果与实际材料的微观演化过程及宏观性能等还有很大的差异。因此分子动力学模拟的发展的关键在于空间规模的扩大和时间尺度的延长。只有在原子个数无限多, 计算的时间足够长时, 才能真实地反映材料性质的宏观行为。

近些年, 随着高性能计算机技术的飞速发展, 分子动力学计算技术被注入了新的活力。2015 年 7 月, 我国研制的“天河二号”超级计算机再度荣登全球超级计算机 500 强排行榜榜首, 连续五次称冠世界[3]。高性能计算机的发展为大规模分子动力学模拟提供了发展的肥沃土壤。制约实际应用的瓶颈在于大规模分子动力学软件的研发。开展大规模分子动力学模拟软件的研究成为国际上的研究热点。

科学家们基于混合架构高性能计算机开发了新的分子动力学程序, 例如 LLNL 与 IBM 的团队开发了面向高性能计算机的经典分子动力学代码--ddcMD 代码。2005 年, 他们首次从原子层次原子尺度研究了金属钽和铀在超高温高压下的凝固过程, 在 IBM BlueGene/L 的 131072 个 CPU 上, 达到 101.7 万亿次/秒的持续计算性能, 获得了 Gordon Bell 奖, 该研究对评估核武器储备的可靠性具有重要意义[4]。2007 年, 他们又实现了第一个基于分子动力学的微米尺度原子级 K-H 不稳定性数值模拟, 在 IBM BlueGene/L 全系统 212,992 个 CPU 上, 获得了 54.3 万亿次/秒的持续计算性能, 再次获得了 Gordon Bell 奖[5]。近年来美国启动了名为“Spatio-Temporal Frontiers of Atomistic Simulations in the Petaflop Computational World” [6]的科研项目, 旨在基于高性能计算机开发可扩展的分子动力学程序, 研究晶界与位错的相互作用以及材料中空洞的形核与长大过程, 在此基础上采用可扩展并行短程 MD 程序 SPaSM 在 Roadrunner 上对 10^6 - 10^{12} 原子进行了大规模的仿真模拟, 达到了 369 万亿次/秒持续计算性能[7]。除此之外, 针对不同领域应用的通用分子动力学软件也有许多, 如 LAMMPS[8], NAMD[9], GROMACS[10]等等。它们各有各的特点, 在不同领域获得了广泛的应用。本文关注的是在材料计算领域广泛使用, 可进行大规模并行计算的开源分子动力学软件。我们选取了 LAMMPS, Ls1-MarDyn, IMD 和 CoMD 四款开源分子动力学软件进行详细的比较说明, 为我们的后续研发工作提供研究基础与支持。

本文的安排如下: 首先是分子动力学软件的介绍, 将对选取的 LAMMPS, Ls1-MarDyn, IMD 和 CoMD 四款大规模并行计算的分子动力学软件进行详细介绍。接下来, 对这四款软件在并行性能, 数据结构, 原子结构存储等几个方面进行横向比较, 并通过数值算例进行验证和分析说明。最后是全文总结, 通过总结前文对四款 MD 软件在大规模并行计算上特点的分析, 提出针对辐照效应下金属材料计算特点的数据结构设计思想, 为下一步研究工作指出方向。

2. 大规模并行材料计算分子动力学软件介绍

2.1. LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) 软件

LAMMPS 软件[8]是由美国 Sandia 国家实验室开发的一款通用的大规模分子动力学并行计算软件，是世界上使用者最广泛的几款分子动力学软件之一。从 2004 年到 2011 年之间，就达到了 8 万 9 千多次的下载量。

LAMMPS 主要用于分子动力学相关的一些计算和模拟工作，可在包括气态，液态或者固态相形态、各种系综下，采用不同的力场和边界条件来模拟全原子，聚合物，生物，金属，粒状和粗粒化体系。可实现从几个粒子到上亿粒子体系的分子动力学计算。LAMMPS 可提供并支持多种势函数计算，包括对势，如 Lennard-Jones(L-J)势（多用于气体，液体分子间作用力计算）、Morse 势等，多体势，如 Embedded Atom Model(EAM)势（多用于单一金属或合金体系计算）、modified EAM (MEAM)势等，应用十分广泛。

LAMMPS 软件为开源软件，以 C++编写，支持用户的自行修改和扩展。LAMMPS 支持串行与并行计算。并行 LAMMPS 采用 MPI 并行，针对并行计算特点设计数据结构与并行策略，具有良好的并行扩展性。其软件结构可由如图 1 所示的类结构关系图来展示。

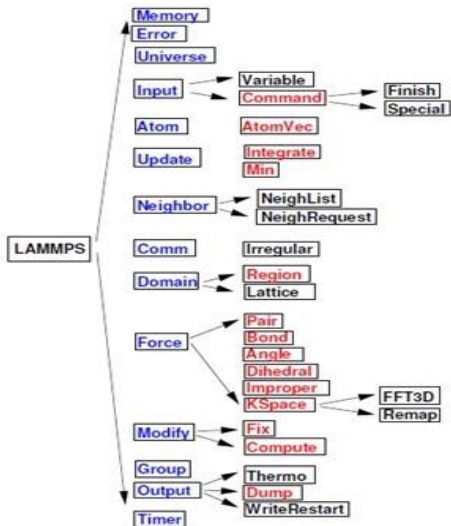


图 1 LAMMPS 软件结构示意图[8]

Fig. 1 Software Structure Chart of LAMMPS[8]

在计算粒子间相互作用时,LAMMPS 通过 cutoff 来设定每个粒子需计算的相互作用粒子范围大小，采用 neighbor list，也即近邻列表的数据结构来记录每个粒子的邻居粒子信息，如图 2 所示。粒子信息存储在数组中，每个粒子的邻近表数组中存储粒子的邻居粒子索引。在计算时，通过 neighbor list 来追踪所有与该粒子有相互作用的粒子间作用力等信息。

LAMMPS 采用空间分解方法来并行分配模拟区域。将整个模拟区域分解成若干子区域分配到各个处理器上。在每个处理器的计算子区域上设置 ghost 区域来存储子区域边界原子信息，以便并行计算时各个处理器间的相互通信与计算。

LAMMPS 提供了两种方法来实现负载均衡，Shift 方法和 RCB 方法，分别如图 3a 和图 3b 所示。Shift 方法通过改变处理器间的分割线，来调整分给每个处理器的空间。最后划分的子区域还是网格的形式。RCB 方法是一种类似“砌瓷砖”的方法，它不是把模拟区域划成网格状，而是大小不等的四边形区域，来让每个区域有相等数目的原子。

此外，LAMMPS 已有支持其他架构计算的版本，包括 GPU(支持 CUDA 和 OpenCL),和 Intel Xeon Phi 等架构，以及 OPENMP 版本的代码。

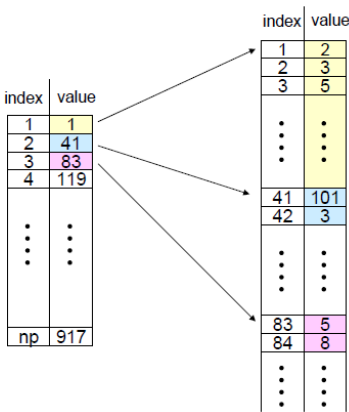
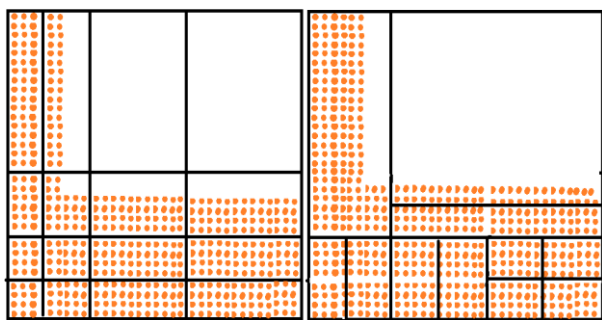


图 2 LAMMPS neighbor list 数据结构展示

Fig. 2 Data Structure of Neighbor List for LAMMPS



a) shift方法实现负载均衡 b) RCB方法实现负载均衡
a) shift method b) RCB method

图3 LAMMPS的负载均衡

Fig. 3 Load Balancing Method of LAMMPS

2.2. IMD(ITAP Molecular Dynamics)软件

IMD 是一款由德国 University of Stuttgart 于 1997 年开始研发的用于经典分子动力学模拟的开源软件包[11], 支持包括对势, 适用于金属材料的 EAM 势, 适用于共价系统的 Stillinger-Weber 和 Tersoff 势, 以及适用于液体的 Gay-Berne 势等多种势函数的分子动力学计算。1999 年, IMD 在德国 Julich 超算中心的超级计算机 T3E-1200-262 上用 512 核实现了 5×10^9 个粒子的分子模拟, 取得了当时的世界纪录[12]。

IMD 采用的模块化设计, C 语言编写, 使用标准的 MPI 库来实现并行消息传递。其软件结构如图 4 所示。IMD 采用 link cell 的数据结构来存储数据。先将整个 box 分成若干 cell, 每个 cell 是个结构体, 每一项存储指向原子相关信息数组的指针, 具体如图 5 所示。此外, IMD 还提供可选项, 对于 3D 的 EAM 势计算算例可提供与 LAMMPS 类似的 neighbor list 的数据结构进行数据存储[13]。

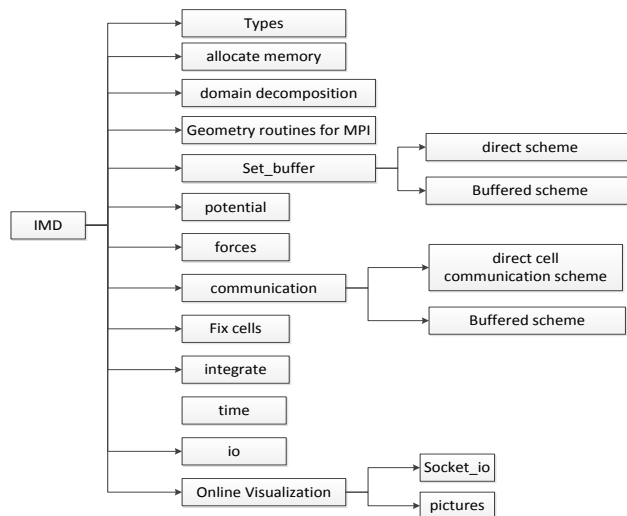


图4 IMD 软件结构示意图

Fig.4 Software Structure Chart of IMD

并行分解时, IMD 将处理器排列成三维的笛卡尔网格, 然后按区域将 box 里 cells 数组细分成相同大小的块, 然后将它们分配给不同的处理器。各处理器设置 ghost 区域用作接收与发送粒子数据的缓冲区。IMD 不提供负载均衡的策略。

IMD 除支持 MPI 的版本之外, 还有 OPENMP、MPI+OPENMP 和 GPU 版本的实现。

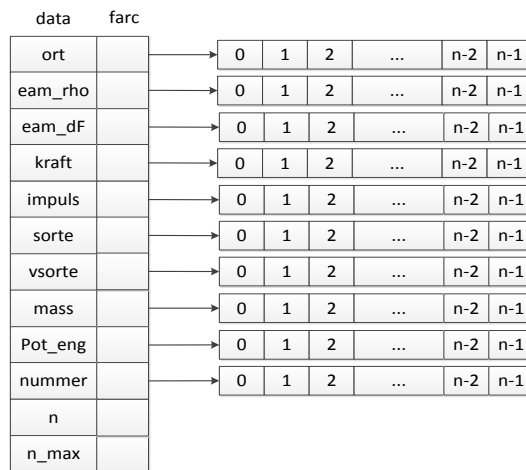


图5 IMD 中 cell 的数据结构展示

Fig. 5 Data sStructure of Cell in IMD

2.3. Ls1-MarDyn(Large systems 1: molecular dynamics)软件

以 IMD 的开发团队为基础, 由德国 University of Stuttgart, University of Kaiserslautern, Technische Universität München 和 University of Paderborn 几家

大学联合开发的旨在挑战万亿级别的分子模拟的 MD 代码——“Ls1-MarDyn” [14]。Ls1-MarDyn 保持了目前最大规模的分子动力学模拟记录，在德国莱布尼兹超级计算中心的超级计算机 SuperMUC 上用 14 万核实现了 4×10^{12} 个粒子的分子模拟 [15]。Ls1-MarDyn 只支持刚性粒子和恒体积系宗下的 MD 计算，适用于短程力计算，因此 ls1 不适用于离子计算。

Ls1-MarDyn 是以 C++编写的开源代码，基于 MPI 实现并行。Ls1-MarDyn 采用模块化设计，将代码分为物理模型计算模块和并行算法模块两类，其软件结构如图 6 所示。目前，Ls1-MarDyn 只支持 Lennard-Jones(L-J)势， Tersoff 势等对势计算，无法实现 EAM 等多体势计算。

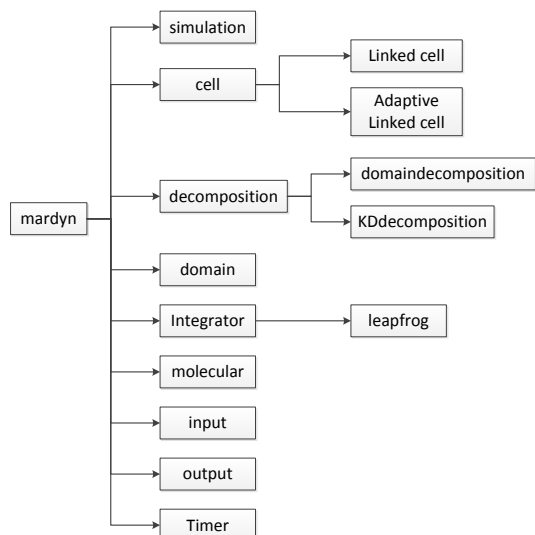
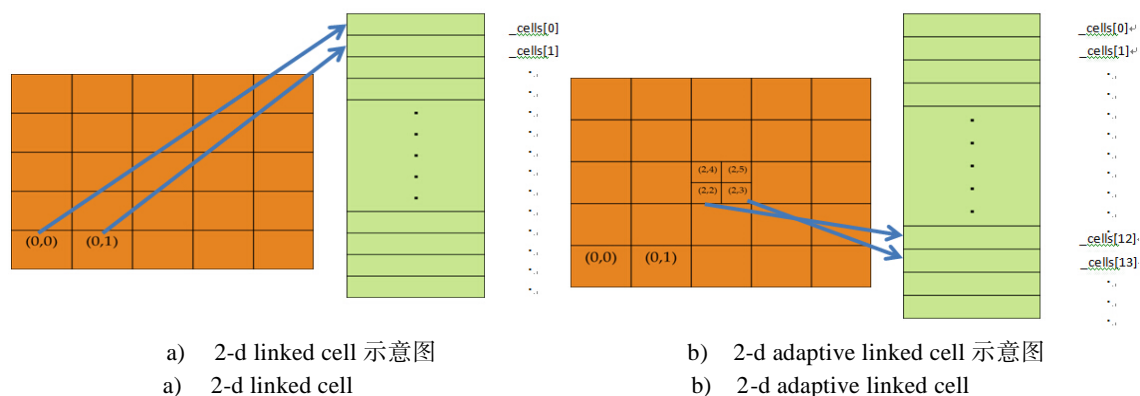


图 6 Ls1-MarDyn 软件结构示意图

Fig. 6 Software Structure Chart of Ls1-MarDyn

针对其适用于气体流体之类分子运动变化频繁的计算的特点，Ls1-MarDyn 代码设计 linked cell 数据结构来进行数据存储，将模拟区域以网格划分成小的单元格 cell，单元格长为截断半径，依据分子位置信息，将每个分子对应到相应的 cell 中，如图 7a 所示。针对不均匀的分子分布，Ls1-MarDyn 对 linked cell 数据结构做出改进，对包含粒子数目较多的单元格进行 2 次分割，设计了自适应 linked cell 的数据结构，如图 7b 所示。

Ls1-MarDyn 采用了两种并行区域分解的方式 [16]。一种是按进程数等分模拟区域，即将 n 个进程以尽量分配给每个维度相同进程数的原则建立笛卡尔坐标，将模拟区域等分为 n 个等体积的小区域分到每个进程。这种区域分解方法简单易操作，只需初始化时建立一次即可。但是当计算到后期粒子变化移动较多时，会造成计算负载的不均衡。另一种是基于 KD 树的区域分解方法，也即用垂直于坐标轴的线将模拟区域递归地划分为两个负载相近的子区域，直到分割的子区域被分配到处理器上（负载近似于 $1/p$ ），其中负载的计算以 cell 为基本单位进行估算。Ls1-MarDyn 暂时还未有支持 GPU, openmp, intel MIC 等异构计算版本代码。


 a) 2-d linked cell 示意图
a) 2-d linked cell

 b) 2-d adaptive linked cell 示意图
b) 2-d adaptive linked cell

图 7 Linked cell 数据结构示意图

Fig.7 Data Structure of Linked Cell for Ls1-MarDyn

2.4. COMD: Classical Molecular Dynamics Proxy Application 软件

COMD[17]是由美国能源部 ASCR 资助的 Exascale Co-Design Center for Materials in Extreme Environment (ExMatEx)中心设计开发的 MD 代码,由 LANL 的 Jamal Mohd-Yusof, Chris Sewell 和 Sriram Swaminarayan 基于 SPaSM 的分子动力学代码来进行开发维护。ExMatEx 中心旨在极端环境下材料特性的 E 级超大规模计算模拟研究,该项目强调多尺度模型,计算机科学,以及通过协同设计实现百亿亿次的超大

规模材料模拟。因此 COMD 的研发是作为协同设计的工具,方便整个 ExMatEx 根据需要对新的体系结构、编程模型等进行多尺度建模计算、功能扩展和重新实现,以及性能测试等而开发实现的。COMD 是开源代码,目前发布的是 1.1 版本[18],只能计算 LJ 势和 EAM 势,EAM 势的计算只支持 fcc 晶格结构,只能建立规则的长方体 box。COMD 是以 C 语言编写的基础 MD 程序,功能不多,主要结构见图 8 所示。

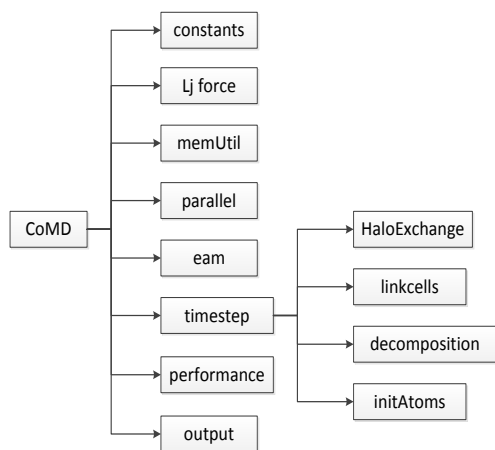


图 8 CoMD 软件结构示意图

Fig. 8 Software structure chart of CoMD

COMD 基于 MPI 实现并行,采用 link cell 的数据结构存储数据,空间分解方法进行任务划分。COMD 根据 cutoff 大小划分 cell,按坐标 XYZ 方向进行处理器核数的划分,再将每个方向上的 cell 均分到每个处理器上,将整个计算区域进行空间的分解。COMD 暂时还未考虑负载平衡策略的实现。COMD 除 MPI 版本之外,还实现了 OPENMP 并行。

对于不同计算架构,COMD 还实现了针对各计算架构的版本,如 AMD 下的 OpenCL 版,基于 NVIDIA 的 GPU 下的 CUDA 版,正在调试 Intel MIC (aka Xeon Phi)和 Intel's FF candidate architecture emulator 架构下的众核计算版本[19]。

3. 大规模材料计算分子动力学软件分析与比较

3.1 大规模并行分子动力学软件比较

通过上一节的介绍,对 LAMMPS, IMD, Ls1-MarDyn 以及 COMD 这四种 MD 软件的大致情况有了初步的了解,本节将对这四种软件从数据结构、计算方法、与并行分解方式等角度进行横向的详细比较。详见表 1。

四种软件都采用空间分解的方式进行并行分解任务。只有 LAMMPS 和 Ls1-MarDyn 两种提供了负载平衡技术。相比较而言,基于自适应 link cell 结构的 Ls1-MarDyn 在负载均衡的调节上会比 LAMMPS 更加灵活。

由于我们考虑的都是设置有 cutoff 的短程力计算,对于此类型计算,通常有两类数据结构用于粒子信息的存储,分别为 neighbor list 和 link cell。neighbor list 结构是每个粒子都根据 cutoff 来确定并存储自己的邻居粒子,每次迭代时只遍历自己的邻居粒子。这种存储方式的好处是计算时很容易找到需计算的对象,并且不会多出额外粒子的计算,计算效率很高。但是存储代价高,每个粒子都要维护自己的 list,在运行一定步数之后还需更新自己的 list。因此这类数据结构更适用于邻居粒子变化缓慢的应用,如固体材料等的计算。LAMMPS 采用的便是 neighbor list 的数据结构存储原子,IMD 也提供此类数据结构的存储选项。Link cell 是将计算区域划分成若干 cell,按 cell 来存储粒子信息。计算时,按照 cutoff 来确定每个 cell 的邻居 cell 信息,按 cell 遍历粒子进行计算。这种方式计算效率低,每次迭代计算时都需遍历本身以及周围邻近 cell,会多出非邻居粒子的计算,计算量大。但是好处是存储量较少,且易于做负载平衡。因此此类结构更适于类似

气体，流体材料等邻居粒子频繁变化应用的计算。 据结构存储粒子信息。
IMD, Ls1-MarDyn 和 COMD 都采用了 link cell 的数

表 1 四款软件比较总结
table 1 Summary of Comparison for Four kinds Software

| 软件名称 | LAMMPS | IMD | Ls1-MarDyn | COMD |
|----------|--|---------------------------------|----------------|--|
| 软件类型 | 通用型 | 通用型 | 专用型（气体流体计算） | 通用型（软件模块） |
| 编程语言 | C++ | C | C++ | C |
| 支持势函数类型 | 对势（LJ 等）,多体势（EAM 等）等多种 | 对势（LJ 等）,多体势（EAM 等）等多种 | 仅支持对势（LJ 势） | 对势（LJ）,多体势（EAM） |
| 势函数计算方式 | 查表插值 | 查表插值 | 参数化计算 | 查表插值 |
| 数据结构存储类型 | Neighbor list | Link cell (Neighbor list 可选) | Link cell | Link cell |
| 数据划分方法 | 空间分解 | 空间分解 | 空间分解 | 空间分解 |
| 是否实现负载均衡 | 提供 Shift 方法和 RCB 方法两种方法 | 不提供 | 基于 KD 树的区域分解方法 | 暂时未提供 |
| 不同计算架构版本 | OPENMP 版, GPU（CUDA 和 OpenCL）版, Intel MIC 版 | OPENMP 版, MPI+OPENMP 版 GPU 版 | 无 | OPENMP 版, GPU（CUDA）版, OpenCL 版, Intel MIC 版等 |

MD 计算中对于粒子间作用力计算的关键在于势函数的计算。对于 EAM 多体势的计算，由于需要用到电子云密度，因此基本上都是采用的查表插值的方法来计算 EAM 势，LAMMPS, IMD 和 COMD 都采用的这种方式。Ls1-MarDyn 只能计算对势（LJ 势），用的则是参数化的方式计算势。已经有其他软件，如 ddcMD，尝试过绕开查表的方法使用参数化方式来实现计算 EAM 势，也取得不错的结果[5]。

随着各种异构计算系统的出现，针对异构计算设计优化的软件版本也日益增多。不论是成熟全面的 LAMMPS, IMD, 还是作为面向 E 级计算的 ExMatEx 项目中的基础程序模块的 CoMD 软件，在异构计算和节点内优化方面都做了些尝试，在不同的计算机体系架构（如 AMD, NVIDIA GPU, aka Xeon Phi 等）上开发了不同的程序版本（OpenACC, OpenCL, GPU, MIC 等等），并做了优化。

3.2 MD 计算模拟数值实验

本节我们将对以上四款 MD 软件进行算例测试。所有的测试算例都是在中科院超级计算中心的超级计算集群“元”上完成。其计算节点配置为 2 颗 Intel E5-2680 V2（Ivy Bridge | 10C | 2.8GHz）处理器，每

个处理器上有 10 个 cpu 核，共 64 GB DDR3 ECC 1866MHz 内存。采用的 MPI 版本为 OPENMPI1.6.5。采用的编译器为 Intel composer_xe_2013_sp1.0.080。

我们设计了铜熔化的算例来测试各 MD 软件性能，对铜原子系统在 600K 的温度下的分子动力学过程进行了计算模拟，cutoff 为 5 埃，时间步为 0.005 皮秒。对于 LAMMPS, IMD 和 COMD 软件，都选取了单核内存占用量最大的原子规模进行计算，也即，LAMMPS 为 7.4×10^8 原子，IMD 为 8.6×10^8 原子，COMD 为 3.4×10^8 原子。由于 Ls1-MarDyn 软件无法计算 EAM 势，因此采用的是其自带的 4 万原子的 LJ 势计算算例。

四款 MD 软件的测试结果见表 2 所示。其中，参数原子更新频率表征的是 MD 软件计算效率的高低，是由“每步计算时间 x CPU 核数/原子数”计算得到。从算例测试结果来看，计算效率最高的是 LAMMPS，原子更新频率达到了 1.6469。其次是 IMD，原子更新频率在 7.7 左右。COMD 软件的计算效率，比 LAMMPS 低了一个数量级，但也能有 12 左右。以上三款 MD 软件的并行性能与可扩展性都非常好。Ls1-MarDyn 软件串行计算时原子更新频

率为 66.9658。虽然 Ls1-MarDyn 计算的是 LJ 势,但由于其原子密度大,link cell 中邻居多,遍历开销大,导致其原子更新频率差。

我们对四款 MD 软件在计算时原子存储所占空间也进行了测试。对于 LAMMPS,IMD 和 COMD 软件的测试,我们选取的是使得程序在每 CPU 核上所能运行的内存达到最大时的原子规模的算例。而 Ls1-MarDyn 软件测试的是其自带的小规模算例,其测试结果不具比较意义。测试结果见表 2 所示。从测试结果来看,每原子所占空间最少的是 IMD, LAMMPS 次之,最多的是 COMD。分析三款软件的原子存储构成,大致都包括原子变量存储空间,通信 buffer 空间和 neighbor list 或 link cell 的存储空间三类,其理论分析结果与测试结果基本一致。

从以上的测试结果来看,若根据现有有限的计算资源情况估计,也即,每个计算节点有 20 个 CPU 核,共 64GB 内存,每 CPU 核计算内存不能超过 2.5

G,那么理论上对应 10 万核时的原子计算规模, LAMMPS 为 6.6×10^{11} , COMD 为 3.4×10^{11} , IMD 为 8.6×10^{11} 。Ls1-MarDyn 软件因为没有测试其内存使用的最大规模而无法准确估计,但从上面的分析来看, Ls1-MarDyn 在 10 万核上计算的原子规模量级也与前三款软件差不多。

因此从存储角度上来说,要在现有内存条件计算节点上,实现 10 万核上核材料辐照缺陷演化的 10^{12} 的原子规模的计算,我们还需对现有 MD 软件进行存储优化。我们下一步将考虑从核材料辐照缺陷演化模拟计算的特点出发,针对该材料的特殊性设计更加节省存储空间的数据结构来减少分子动力学计算中对于邻居存储的需要,从而达到扩大可计算的原子模拟规模的目的。

表 2 软件并行性能与原子存储测试结果

Table 2 Results of Software Parallel Performance Test

| 软件名称 | 势函数类型 | 原子计算规模 (fcc 边长) | Cpu 核数 | 内存使用 | 平均每原子内存使用 (B/atom) | 每时间步计算 时间(s) | 原子更新频率 ($\mu s \times proc / atoms$) |
|------------|-------|-------------------------|----------|----------|-----------------------|-----------------|---|
| LAMMPS | EAM | 7.4×10^8 (570) | 100 | 2.1(G/核) | 315.55 | 12.81 | 1.7293 |
| | | 7.4×10^8 (570) | 100 0 | \ | \ | 1.22 | 1.6469 |
| IMD | EAM | 8.6×10^9 (600) | 100 | 2.3(G/核) | 266.20 | 63.5835 | 7.3592 |
| | | 8.6×10^9 (600) | 100 0 | \ | \ | 6.8928 | 7.9778 |
| Ls1-MarDyn | LJ | 40000 | 1 | 44(M/核) | 456.00 | 2.67863 | 66.9658 |
| | | 40000 | 16 | \ | \ | 0.83931 666 | 335.726 |
| CoMD | EAM | 3.4×10^8 (440) | 100 | 2.3(G/核) | 675.01 | 40.06 | 11.7578 |
| | | 3.4×10^8 (440) | 100 0 | \ | \ | 4.34 | 12.7410 |

4. 总结

在本文中,我们对四款主流材料计算大规模分子动力学开源软件 LAMMPS, LS1--MarDyn, IMD 和 CoMD 进行了详细的介绍,从数据结构、计算方法、并行分解方式、原子存储等多个方面进行横向分析比较。在数据结构方面,MD 模拟中常见两类数据结构 neighbor list 和 link cell 分别被 LAMMPS 和 IMD,LS1--MarDyn 以及 CoMD 所采用。LAMMPS 采用了易于查找邻居粒子信息的 neighbor list 结构,

在金属材料铜熔化算例测试中获得了最好的计算效率。采用 link cell 结构的 IMD, CoMD 等软件由于 cell 结构特征增加了多余的计算量,计算效率稍逊于 LAMMPS。但由于 cell 结构节省的存储量,也使得 IMD 软件在同等内存限制条件下能计算更大规模的分子动力学模拟。算例测试展示了几款软件不俗的并行性能,想要得到更有区分度的测试结果,还需要更大规模(几千甚至上万核)的算例实验。通过对各软件内存使用情况的测试和分析结果,我们将在下一步的研究工作中,针对材料辐照缺陷演化计

算特点，提出了节省原子存储量的新型数据结构，希望在 10 万 CPU 核上实现 10^{12} 原子规模的分子动力学模拟。

References

- [1] B. J. Alder and T. E. Wainwright, Phase Transition for a Hard Sphere System. [J] J. Chem. Phys. 1957, 27 (5): 1208.
- [2] Alder B. J., Wainwright T. E. Studies in Molecular Dynamics. I. General Method, [J] J. Chem. Phys., 1959, 31(2): 459-466.
- [3] TOP500 Supercomputer Site: <http://www.top500.org/>
- [4] F. H. Streitz, J. N. Glosli, M. V. Patel, et al., 100+ TFlop Solidification Simulations on BlueGene/L, In: Proceeding IEEE/ACM SC05, Seattle, WA (Nov 2005).
- [5] James N. Glosli, David F. Richards, K. J. Caspersen, R. E. Rudd, John A. Gunnels, and Frederick H. Streitz, Extending stability beyond CPU millennium: a micron-scale atomistic simulation of Kelvin-Helmholtz instability, In: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, SC 2007, pp. 58:1–58:11. ACM, New York (2007)
- [6] Spatio-Temporal Frontiers of Atomistic Simulations in the Petaflop Computational World, lanl.gov/orgs/adts/sc10/Heterogwebfiles/Germann.pdf
- [7] Germann, T. C., Kadau, K., and Swaninarayan, S., 369 Tflo/s Molecular Dynamics Simulations on the Petaflop Hybrid Supercomputer ‘Roadrunner’, Concurr. Comp.-Pract. E. 21, 2143 (2009).
- [8] LAMMPS manual: <http://lammps.sandia.gov/>
- [9] NAMD: <http://www.ks.uiuc.edu/Research/namd/>
- [10] Gromacs: <http://www.gromacs.org/>
- [11] J. STADLER_, R. MIKULLA and H.-R. TREBIN, IMD: A SOFTWARE PACKAGE FOR MOLECULAR DYNAMICS STUDIES ON PARALLEL COMPUTERS, International Journal of Modern Physics C, 1997, 8(5), 1131-1140
- [12] J. ROTH, F. GAHLER, and H.-R. TREBIN, A MOLECULAR DYNAMICS RUN WITH 5 180 116 000 PARTICLES, International Journal of Modern Physics C, 2000, 11(2), 317-322
- [13] IMD, The ITAP Molecular Dynamics Program: <http://imd.itap.physik.uni-stuttgart.de/>
- [14] Large systems 1: molecular dynamics; <http://www.lsl-mardyn.de/>, accessed August 19, 2014.
- [15] Eckhardt, W.; Heinecke, A.; Bader, R.; Brehm, M.; Hammer, N.; Huber, H.; Kleinhenz, H.G.; Vrabec, J.; Hasse, H.; Horsch, M.; Bernreuther, M.; Glass, C. W.; Niethammer, C.; Bode, A.; Bungartz, J. In Supercomputing – Proceedings of the XXVIII. International Supercomputing Conference (ISC); Kunkel, J. M., Ludwig, T., Meuer, H. W., Eds.; Lecture Notes in Computer Science 7905; Springer: Heidelberg, 2013; pp 1–12.
- [16] W. Eckhardt, A. Heinecke, An efficient vectorization of linked-cell particle simulations. In ACM International Conference on Computing Frontiers 2012, 241-243, 2012.
- [17] CoMD: <http://www.exmatex.org/comd.html>
- [18] “CoMD: Classical Molecular Dynamics Proxy Application, Version 1.1” (February 2013) [open source at GitHub: <https://github.com/exmatex/CoMD>]
- [19] J. Mohd-Yusof, CoDesign Molecular Dynamics (CoMD) Proxy App Deep Dive, Exascale Research Conference, Oct. 1-3, 2012 Arlington, Virginia (LA-UR-12-25066)



NIE Ningming was born in 1983. She received the Ph.D degree in Computational Mathematics from Academy of Mathematics and Systems Science (AMSS) in the Chinese Academy of Sciences in 2010. She is an Associate Professor at Computer Network Information Center (CNIC) of the Chinese Academy of Sciences. Her main research interest is parallel algorithm.

聂宁明(1983—), 女, 湖南衡阳人, 2010 年于中国科学院数学与系统科学研究院获博士学位, 目前任中国科学院计算机网络信息中心副研究员, 主要研究领域为高性能计算。



HU Changjun, full Professor and PhD supervisor. His research interests lie in the area of High Performance Computing.

胡长军, 男, 北京科技大学计算机与通信工程学院教授, 博士生导师, CCF 会员。主要研究领域为高性能计算等。



ZHANG Yunquan, full Professor and PhD supervisor. His research interests include heterogeneous computing, largescale parallel software optimization, parallel programming model and performance model, etc.

张云泉, 男, 中国科学院计算技术研究所研究员, 博士生导师, CCF 会员。主要研究领域为异构计算, 大规模并行软件优化等。



HE Xinfu was born in 1981, PhD, Associate Professor. His main research interest is multiscale modeling of radiation damage in nuclear materials.

贺新福(1981—), 男, 中国原子能科学院研究院副研究员, 主要研究领域为核材料辐照损伤的多尺度建模



ZHANG Boyao was born in 1987. Master of Engineering. His main research interest is parallel computing

张博尧(1987—), 男, 中国科学院计算机网络信息中心助理工程师, 主要研究领域为并行计算



LI Shigang, PhD, Assistant Professor. His research interests include large-scale parallel computing based on MPI, heterogeneous computing and scalable deep learning algorithms on multi-/many-core clusters.

李士刚, 男, 中国科学院计算技术研究所助理研究员, CCF 会员。主要研究领域包括基于 MPI 的大规模并行计算, 多核众核集群上的异构计算和可扩展深度学习算法研究。