

DOI: <https://doi.org/10.23670/IRJ.2024.139.74>

ГИБРИДНОЕ ПРОГРАММИРОВАНИЕ MPI+T++ ДЛЯ T-СИСТЕМЫ С ОТКРЫТОЙ АРХИТЕКТУРОЙ

Научная статья

Осипов В.И.¹, Матвеев Г.А.^{2,*}, Роганов В.А.³¹ORCID : 0000-0002-8983-3791;²ORCID : 0000-0002-2988-1791;³ORCID : 0000-0002-5500-0804;^{1,2,3}Институт программных систем им. А.К. Айламазяна РАН, Переславль-Залесский, Российская Федерация³НИИ Механики МГУ им. М.В. Ломоносова, Москва, Российская Федерация

* Корреспондирующий автор (gera[at]prime.botik.ru)

Аннотация

В настоящее время существует большое количество высокопроизводительных вычислительных систем, состоящих из узлов с многоядерными процессорами и ускорителями от разных производителей оборудования. Удобной моделью программирования в таких системах является модель MPI+X. Гибридные программы MPI+X часто имеют более высокую производительность по сравнению с программами на «чистом» MPI. Научные результаты, полученные при разработке таких программ, используются для создания новых версий стандарта MPI (Message Passing Interface). Гибридный метод параллельного программирования состоит в использовании модели общей памяти внутри узла и модели передачи сообщений MPI для обмена данными между узлами. Примеры гибридных моделей: MPI+OpenMP, MPI+Posix Threads, MPI+CUDA, MPI+T++ и другие. Программирование в смешанном режиме даёт разработчику возможность добиться более высокой производительности и масштабируемости приложения. В некоторых случаях смешанный код может выполняться медленнее, чем код, написанный на «чистом» MPI. В статье описываются некоторые аспекты, связанные с гибридной моделью программирования MPI+T++ в среде OpenTS. T-система (OpenTS) – система для параллельного программирования, поддерживающая динамически загружаемые библиотеки для коммуникационного уровня. В системе OpenTS реализован язык для параллельных вычислений T++, который является расширением языка программирования C++. Синтаксис языка T++ отличается от синтаксиса языка C++ добавлением в него нескольких ключевых слов. В статье приведены результаты испытаний производительности приложений MPI+T++ из пакета Mantevo (CoMD, HPCCG, MiniAero, phdMesh) и примеров из пакета LAMMPS, реализованных в среде OpenTS. В пакет Mantevo входят несколько параллельных приложений, в которых реализованы алгоритмы для решения некоторых дифференциальных уравнений с частными производными. Дается оценка эффективности такой реализации в сравнении с оригинальными MPI-версиями приложений. В статье приведен простой пример гибридного MPI+T++ приложения на языке T++.

Ключевые слова: параллельный алгоритм, язык программирования T++, OpenTS, T-система, T-приложение, гибридные модели программирования.

MPI+T++ HYBRID PROGRAMMING FOR T-SYSTEM WITH OPEN ARCHITECTURE

Research article

Osipov V.I.¹, Matveev G.A.^{2,*}, Roganov V.A.³¹ORCID : 0000-0002-8983-3791;²ORCID : 0000-0002-2988-1791;³ORCID : 0000-0002-5500-0804;^{1,2,3}Ailamazyan Program Systems Institute of the Russian Academy of Sciences, Pereslavl-Zalessky, Russian Federation³Research Institute of Mechanics, Lomonosov Moscow State University, Moscow, Russian Federation

* Corresponding author (gera[at]prime.botik.ru)

Abstract

Nowadays, there are many high-performance computing systems consisting of nodes with multicore processors and accelerators from different hardware manufacturers. A convenient programming model in such systems is the MPI+X model. Hybrid MPI+X programmes often have higher performance compared to programmes on "pure" MPI. Scientific results obtained while developing such programs are used for creating new versions of MPI (Message Passing Interface) standard. The hybrid method of parallel programming involves using the shared memory model inside a node and MPI message passing model for data exchange between nodes. Examples of hybrid models: MPI+OpenMP, MPI+Posix Threads, MPI+CUDA, MPI+T++ and others. Programming in mixed mode gives a developer an opportunity to achieve higher performance and scalability of an application. In some cases, mixed code can be executed slower than code written in "pure" MPI. This article describes some aspects associated with the hybrid MPI+T++ programming model in the OpenTS environment. T-system (OpenTS) is a system for parallel programming supporting dynamically loaded libraries for communication layer. The OpenTS system implements C++ language for parallel calculations, which is an extension of C++ programming language. C++ syntax differs from C++ syntax by adding several keywords to it. The work presents the results of performance tests of MPI+T++ applications from Mantevo package (CoMD, HPCCG, MiniAero, phdMesh) and examples from LAMMPS package developed in OpenTS environment. The Mantevo package includes several parallel applications in which algorithms for solving some

partial differential equations are implemented. An estimation of efficiency of such implementation in comparison with original MPI-versions of applications is presented. The article gives a simple example of a hybrid MPI+T++ application in T++ language.

Keywords: parallel algorithm, C++ programming language, OpenTS, T-system, T-application, hybrid programming models.

Введение

Интерфейс передачи сообщений MPI (Message Passing Interface) является стандартом для межузловой связи. Он определяет семантику и синтаксис коммуникационных функций. Он был разработан научным сообществом и промышленностью для стандартизации программирования передачи сообщений. На сегодняшний день есть две основные реализации MPI: OpenMPI [1] и MPICH [2]. Существует также несколько других реализаций MPI [3].

Архитектуры узлов HPC (High-Performance Computing) имеют тенденцию к увеличению количества ядер в одном процессоре, а также используют ускорители, например, графические процессоры. Чтобы лучше использовать общие ресурсы внутри узла и программировать ускорители, пользователи обратились к гибриднему программированию, которое сочетает в себе MPI с моделями параллельного программирования на уровне узла.

Параллельные многопоточные программы становятся все более распространенными. Параллельные вычисления становятся все более популярными, и создание эффективных моделей параллельного программирования является приоритетной задачей. Модели с общей памятью и распределенной памятью – это две хорошо известные классификации модели параллельной аппаратной архитектуры. Использование ресурсов всех ядер и снижение временных затрат является основной целью модели многоядерного программирования [4].

Гибридное параллельное программирование в модели MPI+X стало нормой в высокопроизводительных вычислениях [5]. Этот подход к параллельному программированию отражает иерархию параллелизма в современных высокопроизводительных системах, в которых высокоскоростное соединение объединяет множество узлов.

Последние тенденции в области высокопроизводительных вычислений (HPC) – использовать ускорители (графические процессоры), программируемые пользователем вентильные матрицы (FPGA), сопроцессоры, -- привели к значительной неоднородности в вычислительных подсистемах и подсистемах памяти.

Разработчики приложений обычно используют модель интерфейса передачи сообщений MPI на вычислительных узлах кластера, и внутриузловую модель, такую как OpenMP или библиотеку для ускорителя (например: CUDA, OpenACC).

Гибридный метод параллельного программирования состоит в использовании модели общей памяти внутри узла и модели передачи сообщений для обмена данными между узлами [6]. Программирование в смешанном режиме даёт разработчику возможность добиться более высокой производительности и масштабируемости приложения.

T-система (OpenTS) является средой программирования для разработки высокопроизводительных и хорошо масштабируемых приложений [7]. Она поддерживает язык T++, который дополняет стандартный язык C++ включением в него нескольких ключевых слов.

Ключевые слова необходимы для определения T-переменных и T-функций в теле T-программы, которая будет выполняться параллельно на вычислительной установке или суперкомпьютере.

Динамическая библиотека OpenTS DMPI является неотъемлемой частью T-системы.

T-система является самодостаточной системой в том смысле, что программисту нет необходимости использовать в своей программе MPI-вызовы. Однако, если есть необходимость использовать вызовы MPI, например, для достижения более высокой производительности программы, то это можно делать.

Гибридное программирование MPI+T+ в среде OpenTS предполагает, что основная часть кода реализуется на языке T++, при этом в теле T-программы используется библиотека MPI.

Цель статьи – познакомить читателя с приложениями MPI+T++, которые занимают свое место среди гибридных приложений. В статье рассматривается простой пример гибридного MPI+T++ приложения на языке T++. Приводятся результаты тестирования четырех MPI+T++ реализаций приложений из пакета Mantevo и четырех примеров из пакета LAMMPS в сравнении с их оригинальными MPI-версиями.

Испытание происходило в несколько этапов, на каждом этапе определялось среднее значение производительности приложения.

Простой пример гибридного MPI+T++ приложения на языке программирования T++

Рассмотрим простой пример гибридного MPI+T++ приложения на языке T++.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#define USRTAG 1024
#define MSGSIZE 128
```

```
static int root_rank = 0;
static int worker_rank = 1;
```

```
tfun int tworker() {
    printf("I am node %d of %d.\n", ts::myRank, ts::realsuperSize);
    tval int tx;
    char msg[MSGSIZE];
```

```

MPI_Status stat;
memset(msg, '\0', sizeof(msg));
MPI_Recv(msg, sizeof(msg) - 1, MPI_BYTE, root_rank, USRTAG, MPI_COMM_WORLD, &stat);
printf("node=%d, msg=\"%s\"\n", ts::myRank, msg);
tx = 1;
return (int) tx;
}

```

```

tfun int main(int argc, char *argv[]) {
    static char *MSG = (char *) "This is a message from the root T-process!";
    if (ts::realsuperSize != 2) {
        printf("\n\nSorry, this program works correctly only for the two ranks!\n\n");
        exit(0);
    }
    printf("I am node %d of %d.\n", ts::myRank, ts::realsuperSize);
    tval int tv;
    tct(atRank(worker_rank % ts::realsuperSize));
    tv = tworker();
    MPI_Send(MSG, strlen(MSG), MPI_BYTE, worker_rank, USRTAG, MPI_COMM_WORLD);
    (int) tv;
    return 0;
}

```

В приведенном примере создаются два процесса на двух узлах (рангах) вычислительной установки: первый – основной процесс – создается на нулевом узле, а второй – рабочий процесс – на первом узле. Основной процесс представлен в виде Т-функции main(), а рабочий процесс – в виде Т-функции tworker(). Т-функция main() создает рабочий процесс на первом ранге вычислительной установки с помощью операторов:

```

tval int tv;
tct(atRank(worker_rank % ts::realsuperSize));
tv = tworker();

```

В теле Т-программы определяется Т-переменная tv. Выход Т-функции tworker() присваивается Т-переменной tv. Далее, основной процесс посылает данные -- сообщение: "This is a message from the root T-process!"-- рабочему процессу с помощью вызова MPI-функции MPI_Send(). После этого основной процесс ждет завершения работы рабочего процесса (готовности Т-переменной tv) с помощью оператора:

```
(int) tv;
```

В теле Т-функции tworker() выполняется вызов MPI-функции MPI_Recv() для приема сообщения от основного процесса. После этого рабочий процесс производит печать полученного сообщения и завершает свою работу.

Приложение CoMD

CoMD – простое приложение молекулярной динамики проекта Mantevo. Имитирует реализацию SPaSM (Scalable Parallel Short-Range Molecular Dynamics).

Реализация приложения CoMD (и других приложений) на MPI+T++ происходила следующим образом. Функцию C++ main пакета CoMD мы переименовали в worker. В пакет мы добавили файл на языке программирования T++, который параллельно запускает столько задач, сколько в текущей вычислительной системе вычислительных узлов, по одной задаче на каждом вычислительном узле. Каждая из этих задач вызывает функцию worker. Обеспечивается передача параметров командной строки в каждый из экземпляров функции worker. Все вхождения файла <mpi.h> заменяются на заголовочный файл, необходимый для корректной работы Т-системы. Происходит сборка исполняемого файла с использованием компилятора T++.

Были проведены сравнительные испытания приложения CoMD с использованием OpenMPI и CoMD с использованием MPI+T++. Результаты испытания представлены на рисунке 1.

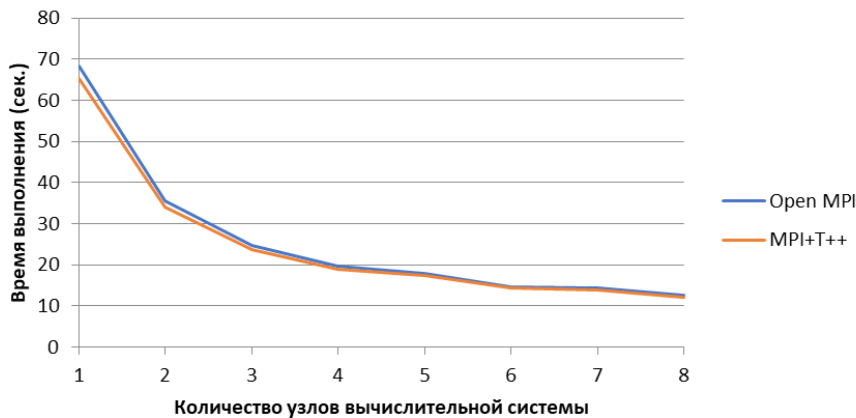


Рисунок 1 - На рисунке представлены два графика: среднее время выполнения CoMD с использованием OpenMPI и среднее время выполнения CoMD с использованием MPI+T++
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.1>

Отклонение среднего времени работы приложения составило -3.5%, то есть производительность приложения CoMD MPI+T++ приблизительно совпадает с производительностью приложения CoMD OpenMPI.

Приложение HPCCG

Приложение HPCCG из проекта Mantevo является решателем линейной системы уравнений с частными производными на области, которая имеет форму пучка лучей (beam shaped). Для решения системы уравнений используется метод сопряжённых градиентов с предобуславливанием.

Были проведены сравнительные испытания оригинального MPI-приложения HPCCG и приложения HPCCG, реализованного на платформе MPI+T++. Для проведения испытаний использовался OpenMPI. Результаты испытания представлены на рисунке 2.

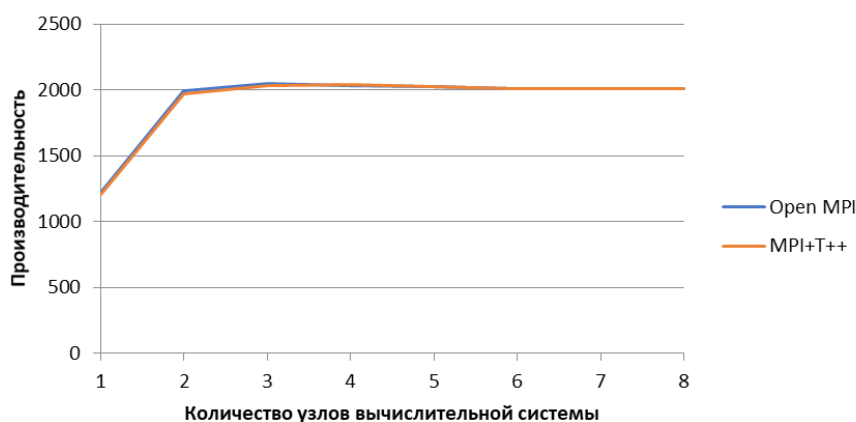


Рисунок 2 - На рисунке представлены два графика: средняя производительность HPCCG с использованием OpenMPI и средняя производительность HPCCG с использованием MPI+T++
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.2>

Отклонение среднего значения производительности приложения составило 0.3%, то есть производительность приложения HPCCG MPI+T++ приблизительно совпадает с производительностью приложения HPCCG OpenMPI. Производительность приложения HPCCG определялась исходя из статистики, выдаваемой этим приложением

Приложение MiniAero

В приложении MiniAero из проекта Mantevo строится решение уравнения Навье-Стокса для сжимаемой жидкости методом Рунге-Кутты четвертого порядка.

Было проведено сравнительное тестирование приложений MiniAero OpenMPI и MiniAero MPI+T++. Испытывался один из тестов miniAero FlatPlate_Parallel на 8 процессорах. Проводилось 10 испытаний. Отклонение среднего значения производительности приложения составило 0.1%, то есть производительность приложения miniAero MPI+T++ приблизительно совпадает с производительностью приложения miniAero OpenMPI.

Приложение phdMesh

Пакет phdMesh проекта Mantevo поддерживает параллельные гетерогенные вычисления с использованием динамической неструктурированной сетки.

Было проведено сравнительное тестирование приложений phdMesh OpenMPI и phdMeshMPI+T++ на 8 процессорах. Отклонение среднего значения времени работы приложения составило 1%, то есть производительность приложения phdMeshMPI+T++ приблизительно совпадает с производительностью приложения phdMesh OpenMPI.

Модель погруженного иона

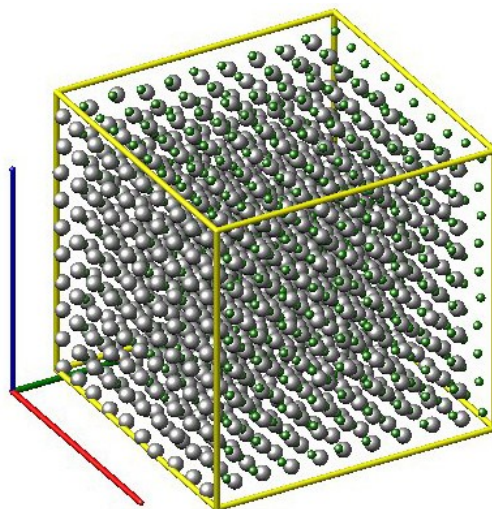


Рисунок 3 - Пример eim LAMMPS. Модель погруженного иона для хлористого натрия
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.3>

Для метода погруженного иона (Embedded Ion Method) [8] энергия E системы атомов задается формулой

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=i_1}^{i_N} \phi_{ij}(r_{ij}) + \sum_{i=1}^N E_i(q_i, \sigma_i),$$

где N – количество атомов, ϕ_{ij} – функция парного потенциала, r_{ij} – расстояние между i -м и j -м атомами, i_1, \dots, i_N – соседние атомы атома i , E_i – энергия погружения атома i , q_i – заряд, σ_i – электрический потенциал.

$$E_i(q_i, \sigma_i) = \frac{1}{2} \cdot q_i \cdot \sigma_i,$$

$$q_i = \sum_{j=i_1}^{i_N} \eta_{ji}(r_{ij}),$$

$$\sigma_i = \sum_{j=i_1}^{i_N} q_j \cdot \psi_{ij}(r_{ij}),$$

Где η_{ji} – парная функция, описывающая поток электронов от атома i к атому j ,

ψ_{ij} – парная функция, связанная с кулоновским взаимодействием.

В пакете LAMMPS [9] модель применялась к системе из 1000 молекул хлористого натрия (рисунок 3).

Было проведено сравнительное тестирование примера eim пакета LAMMPS на платформах OpenMPI и MPI+T++. Отклонение среднего значения времени работы приложения составило 0.2%, то есть производительность приложения eim LAMMPS MPI+T++ приблизительно совпадает с производительностью приложения eim LAMMPS OpenMPI.

Связанные наностержни (tethered nanorods)

В пакете LAMMPS [9] представлена симуляция модели наностержней со связями (tethered nanorods [10]). В модели описывается композит длинноцепочного полимера и наностержней. Длинноцепочный полимер в композите играет роль лиганда. Длинноцепочный полимер моделируется с использованием потенциала FENE (Finitely Extensible Nonlinear Elastic model). FENE означает конечно растяжимую нелинейную упругую модель длинноцепочечного полимера. Он упрощает цепочку мономеров, соединяя последовательность бусинок с нелинейными пружинами. На рисунке 4 представлен снимок одного из моментов симуляции модели.

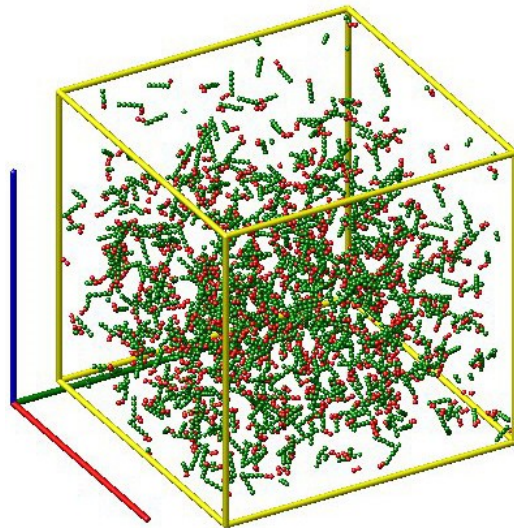


Рисунок 4 - Пример rigid LAMMPS. Наностержни со связями (tethered nanorods)
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.4>

Было проведено сравнительное тестирование примера rigid пакета LAMMPS на платформах OpenMPI и MPI+T++. Отклонение среднего значения времени работы приложения составило 3.7%, то есть производительность приложения rigid LAMMPS MPI+T++ приблизительно совпадает с производительностью приложения rigid LAMMPS OpenMPI.

Симуляция деформации сдвига металла

На рисунке 5 отображается симуляция деформации металла в пакете LAMMPS. Нижняя часть куска металла неподвижна, верхняя часть сдвигается вправо. На рисунке 6 иллюстрируется симуляция деформации куска металла, имеющего внутри пустоту (пузырь). При сдвиге нижняя часть образца остается неподвижной, верхняя часть сдвигается вправо.

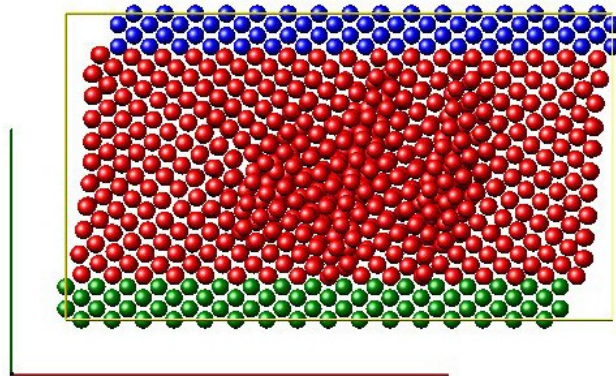


Рисунок 5 - Пример shear LAMMPS. Симуляция деформации сдвига куска металла
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.5>

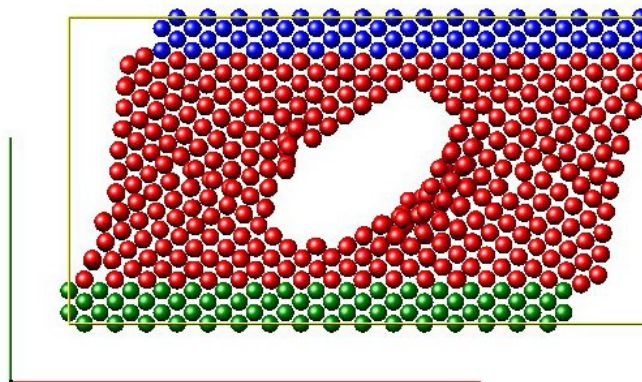


Рисунок 6 - Пример shear LAMMPS. Симуляция деформации сдвига куска металла с пузырем внутри
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.6>

Было проведено сравнительное тестирование примера shear пакета LAMMPS на платформах OpenMPI и MPI+T++. Отклонение среднего значения времени работы приложения составило 1.4%, то есть производительность приложения shear LAMMPS MPI+T++ приблизительно совпадает с производительностью приложения shear LAMMPS OpenMPI.

Динамика многочастичных взаимодействий

Динамика многочастичных взаимодействий (Multi-particle collision dynamics), также известная как стохастическая динамика вращения (Stochastic rotation dynamics), представляет собой метод мезомасштабного моделирования сложных жидкостей на основе частиц, который полностью включает в себя тепловые флуктуации и гидродинамические взаимодействия. Связь внедренных частиц с крупнозернистым растворителем достигается с помощью молекулярной динамики. Модель может использоваться для изучения поведения микроэмульсий или растворенных гибкоцепных полимеров.

В пакете LAMMPS [9] растворитель моделируется в виде набора N частиц массы m с непрерывными координатами r_i и скоростями v_i . Алгоритм симуляции состоит из шагов моделирования движения и столкновения частиц. На шаге моделирования движения координаты r_i всех частиц растворителя в момент времени t одновременно обновляются в соответствии с

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t \cdot$$

При моделировании взаимодействий частицы группируются по ячейкам, и они взаимодействуют только с частицами из той же ячейки, обычно для этого строится решетка. Как правило, количество частиц в каждой ячейке – от 3 до 20. Скорости частиц в каждой ячейке обновляются в соответствии с правилом столкновения

$$v_i(t + \Delta t) = u(t) + R(v_i(t) - u) \cdot$$

Где u – средняя скорость всех частиц в ячейке, R – матрица вращения. В трехмерном пространстве вращение осуществляется на угол α вокруг случайной оси вращения. Одно и то же вращение применяется ко всем частицам в пределах данной ячейки. Рисунок 7 иллюстрирует модель многочастичных взаимодействий.

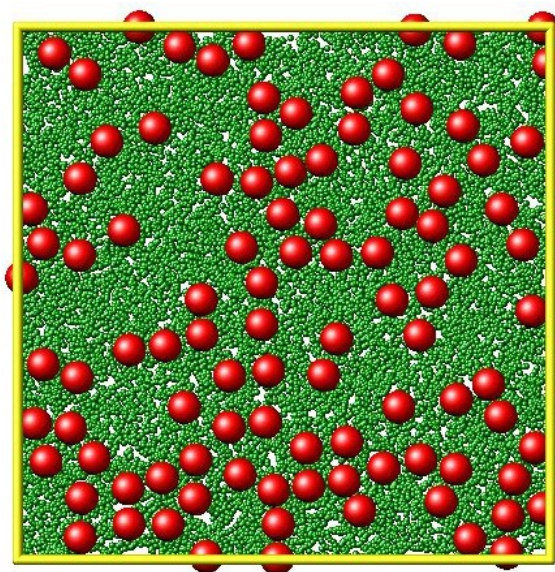


Рисунок 7 - Пример LAMMPS srd. Большие частицы находятся в крупнозернистом растворителе
DOI: <https://doi.org/10.23670/IRJ.2024.139.74.7>

Примечание: модель динамики многочастичных взаимодействий

Было проведено сравнительное тестирование примера `srd` пакета LAMMPS на платформах OpenMPI и MPI+T++. Отклонение среднего значения времени работы приложения составило 2.2%, то есть производительность приложения `srd` LAMMPS MPI+T++ приблизительно совпадает с производительностью приложения `srd` LAMMPS OpenMPI.

Заключение

Приложения MPI+X занимают важное место среди приложений для высокопроизводительных вычислений. Приложения MPI+T++ – это приложения, написанные на языке для параллельных вычислений T++ с добавлением вызовов функций MPI. В статье приводится пример простого приложения MPI+T++ на языке программирования T++ и описываются результаты тестирования производительности восьми приложений MPI+T++. По результатам тестирования можно сделать вывод, что производительность MPI+T++ реализаций приложений из пакета Mantevo незначительно отличается от производительности соответствующих OpenMPI реализаций.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Open MPI: Open Source High Performance Computing. — URL: <https://www.open-mpi.org/> (accessed: 11.09.2023)
2. MPICH | High-Performance Portable MPI. — URL: <https://www.mpich.org/> (accessed: 11.09.2023)
3. Haensel D. A C++ Based MPI-enabled Tasking Framework to Efficiently Parallelize Fast Multipole Methods for Molecular Dynamics / D. Haensel // Dresden University of Technology. — 2018. — P. 1-121.
4. hyPACK-2013. — URL: <https://www.tezu.ernet.in/dcompsc/facility/HPCC/hypack/index.html> (accessed: 11.09.2023)
5. Wright G. A Primer on Parallel Programming / G. Wright. — 2021.— URL: <https://researchcomputing.princeton.edu/sites/g/files/toruqf311/files/documents/A%20Parallel%20Programming%20Primer.pdf> (accessed: 11.09.2023)
6. Zambre R. Lessons Learned on MPI+Threads Communication / R. Zambre, A. Chandramowlishwaran // ACM/IEEE International Conference for High-Performance Computing. — 2022.
7. Абрамов С.М. Динамическое распараллеливание программ на базе параллельной редукции графов. Архитектура программного обеспечения новой версии T-системы / С.М. Абрамов, В.А. Васенин, Е.Е. Мамчиц [и др.] // Научная сессия МИФИ-2001. — Москва, 2001. — Т. 2. — С. 234.
8. Zhou X.W. Embedded-ion Method: An Analytical Energy-conserving Charge-transfer Interatomic Potential and Its Application to the La-Br System / X.W. Zhou, F.P. Doty // Phys. Rev. — 2008. — № 78. — 224307.
9. LAMMPS Molecular Dynamics Simulator. — URL: <https://www.lammps.org/> (accessed: 11.09.2023)
10. Liu K. Self-assembly of Inorganic Nanorods / K. Liu, N. Zhao, E. Kumacheva // Chem. Soc. Rev. — 2011. — № 40. — P. 656-671.

Список литературы на английском языке / References in English

1. Open MPI: Open Source High Performance Computing. — URL: <https://www.open-mpi.org/> (accessed: 11.09.2023)
2. MPICH | High-Performance Portable MPI. — URL: <https://www.mpich.org/> (accessed: 11.09.2023)
3. Haensel D. A C++ Based MPI-enabled Tasking Framework to Efficiently Parallelize Fast Multipole Methods for Molecular Dynamics / D. Haensel // Dresden University of Technology. — 2018. — P. 1-121.
4. hyPACK-2013. — URL: <https://www.tezu.ernet.in/dcompsc/facility/HPCC/hypack/index.html> (accessed: 11.09.2023)
5. Wright G. A Primer on Parallel Programming / G. Wright. — 2021.— URL: <https://researchcomputing.princeton.edu/sites/g/files/toruqf311/files/documents/A%20Parallel%20Programming%20Primer.pdf> (accessed: 11.09.2023)
6. Zambre R. Lessons Learned on MPI+Threads Communication / R. Zambre, A. Chandramowlishwaran // ACM/IEEE International Conference for High-Performance Computing. — 2022.
7. Abramov S.M. Dinamicheskoe rasparallelivanie programm na baze parallel'noj redukcii grafov. Arhitektura programnogo obespechenija novej versii T-sistemy [Dynamic Parallelization of Programs Based on Parallel Graph Reduction. Software Architecture of the New Version of the T System] / S.M. Abramov, V.A. Vasenin, E.E. Mamchic [et al.] // Nauchnaja sessija MIFI-2001 [Scientific Session MEPhi-2001]. — Moscow, 2001. — Vol. 2. — P. 234. [in Russian]
8. Zhou X.W. Embedded-ion Method: An Analytical Energy-conserving Charge-transfer Interatomic Potential and Its Application to the La-Br System / X.W. Zhou, F.P. Doty // Phys. Rev. — 2008. — № 78. — 224307.
9. LAMMPS Molecular Dynamics Simulator. — URL: <https://www.lammps.org/> (accessed: 11.09.2023)
10. Liu K. Self-assembly of Inorganic Nanorods / K. Liu, N. Zhao, E. Kumacheva // Chem. Soc. Rev. — 2011. — № 40. — P. 656-671.