Реализация решения задачи N тел
и анализ эффективности разработанных решений

Каримова Р.Ф.[1], Вахлаева К.П.[2]

[1]karimova.renata1@yandex.ru,[2]vax01@yandex.ru

Саратовский государственный университет имени Н.Г. Чернышевского

В данной работе рассмотрена реализация трех различных алгоритмов решения гравитационной задачи N тел на языке Java с использованием технологии OpenMP для выполнения параллельных вычислений.

Ключевые слова: задача N тел, параллельное программирование, Java, OpenMP.

Введение

Java является современным, простым в изучении объектно-ориентированным языком программирования, который применяется для реализации многих программных решений, существующих на сегодняшний день. Данный язык также подходит для преподавания компьютерных наук, в связи с наличием обширной документации и возможностью освоения приёмов программирования на примерах реализации математических задач. Java поддерживает различные библиотеки для многопоточного программирования, в том числе технологию распараллеливания OpenMP, которая востребована для решения целого ряда вычислительно трудоёмких задач. К таким задачам, например, относится задача моделирования динамики системы N точечных масс, которая представляет собой расчёт поведения N тел, взаимодействующих по гравитационному закону.

1. Задача N тел

Задача N тел является задачей расчета эволюции поведения системы из N тел, которые взаимодействуют при помощи дальнодействующих сил. Если количество частиц N, то количество взаимодействий между ними, которые необходимо учитывать, растёт как $O(N^2)$. Это обуславливает высокую вычислительную сложность задачи. Помимо тривиального алгоритма взаимодействия, существуют различные приближенные схемы. В среднем они позволяют снизить сложность с $O(N^2)$ до $O(N\log N)$ или даже $O(N)$, однако имеют худшие показатели точности [1].

В задаче N тел взаимодействие вычисляется отдельно между парами частиц, а сила, действующая на каждую частицу (i-частицу) является суммой вкладов отдельных частиц (j-частиц), действующих на неё. Пусть даны N тел, их позиции x_i и скорости v_i, $1 \le i \le N$, вектор силы f_{ij} действует на тело i со стороны тела j, определяется на основании ньютоновских законов движения и закона тяготения по следующей формуле:

$$f_{ij} = G\frac{m_i m_j r_{ij}}{\|r_{ij}\|^2 \|r_{ij}\|} \quad (1)$$

где m_i, m_j массы тел i, j соответственно, $r_{ij} = x_j - x_i$ вектор, направленный от тела i к телу j, G – гравитационная постоянная, которая равняется $6{,}67 \cdot 10^{-11}$ Нм^2/кг^2 [2].

В результате некоторых преобразований имеем следующую систему дифференциальных уравнений:

$$
\begin{cases} \dfrac{dr_i}{dt} = v_i \\ \dfrac{dv_i}{dt} = G\sum_{j\neq i} m_j \dfrac{r_{ij}}{\|r_{ij}\|^3} \end{cases} \quad (2)
$$

Воспользуемся подходом, основанном на расчёте взаимодействия между всеми парами тел, для моделирования данной системы. Единственная неточность этого метода заключается в том, что приходится в течение каждого небольшого промежутка времени (шага расчёта) считать ускорение при вычислениях неизменным, в то время как оно переменно. Непосредственные вычисления по формуле (2) являются достаточно ресурсоемкими: для N частиц объем вычислений растёт как $O(N^2)$.

2. Метод Leapfrog-Verlet

Метод Leapfrog-Verlet применим к задаче моделирования гравитационного взаимодействия, так как ускорение тела зависит только от расположения других тел и не зависит от скоростей.

В результате некоторых преобразований с использованием метода Эйлера, алгоритм решения (2) имеет вид:

$$
\begin{cases} x_i^{(n+1)} = x_i^{(n)} + hv_i^{(n+1/2)} \\ v_i^{(n+3/2)} = v_i^{(n+1/2)} + hG\sum_{j\neq i} m_j \dfrac{r_{ij}^{(n+1)}}{\left\|r_{ij}^{(n+1)}\right\|^3} \\ v_i^{(1/2)} = v_i^{(0)} + \dfrac{h}{2} G\sum_{j\neq i} m_j \dfrac{r_{ij}}{\|r_{ij}\|^3} \end{cases} \quad (3)
$$

Значения r_{ij} известны, локальная погрешность имеет порядок $O(h^3)$.

Алгоритм Leapfrog-Verlet является алгоритмом второго порядка, таким образом, он имеет более высокую точность по сравнению с методом Эйлера [3].

3. Метод всех пар

В случае, когда взаимодействующие тела располагаются близко друг от друга, описанный простейший метод расчёта оказывается неустойчивым в силу увеличения погрешностей вычисления. Для того чтобы описанный выше простейший вариант алгоритма не приводил к нестабильному поведению системы близко расположенных тел, можно наложить ограничение на максимальное значение силы, действующей на тело со стороны другого (близко расположенного) тела.

Метод всех пар для задачи N тел является алгоритмом полного перебора, который вычисляет все попарные взаимодействия между N телами. Данный метод не используется при моделировании больших систем из-за его вычислительной сложности $O(N^2)$ [1].

Рассматриваемый алгоритм основан на вычислении всех попарных взаимодействий $f_{ij}, i, j = \overline{1, N}$, образующих матрицу взаимодействий f_{nm} размера $N \times N$.

Результирующая сила F_i , действующая на i тело получается в результате суммирования всех взаимодействий i строки матрицы взаимодействий f_{nm}. Вычисления значений результирующих сил могут быть выполнены независимо друг от друга.

4. Метод Барнса-Хата

Основная идея алгоритма Барнса-Хата заключается рекурсивном разделении пространства на подпространства до того момента, пока не выполнится какой-то критерий, например пока в каждом подпространстве будет не больше k тел. Иерархическое разбиение пространства записывается в дерево квадрантов. При этом группировка пространств происходит с использованием центра масс. Центр масс группы тел – это среднее положение тела в этой группе, взвешенное по массе. Алгоритм Барнса-Хата – это алгоритм для группировки достаточно близких тел. Он рекурсивно делит множество тел на группы, сохраняя их в дереве. Дерево квадрантов похоже на двоичное дерево, за исключением того, что каждый узел имеет 4 дочерних элемента (некоторые из которых могут быть пустыми). Каждый узел представляет область двухмерного пространства [4].

Данный алгоритм уменьшает вычислительную сложность с $O(N^2)$ до $O(N \log N)$, что позволяет решать более сложные вычислительные задачи [5].

5. Результат работы программы

Данные алгоритмы были реализованы на языке Java, при этом распараллеливание алгоритмов было произведено с использованием технологии OpenMP [6-8]. Java предоставляет широкие возможности для расспаралеливания программ. Данная задача была реализована с использованием технологий omp4j, которая является простой в использовании и позволяет разрабатывать переносимые, хорошо масштабируемые параллельные приложениия на Java.

В результате работы программы получились следующие результаты, представленные в таблице, при этом шаг разбиения равнялся 0.1.

Таблица 1 – Результаты работы программы

Алгоритм	Количество частиц	Последовательный алгоритм, мсек	Параллельный алгоритм, мсек
Барнса-Хата	10	446	958
	100	4694	2771
	1000	7453	6303
Всех пар	10	1063	1234
	100	10109	10076
	1000	107692	100975
Leapfrog-Verlet	10	569	609
	100	5582	3445
	1000	11378	6503

Алгоритм Барнса-Хата показал лучшее время работы в последовательном и параллельных вариантах программы при количестве точек 100 и 1000. При этом алгоритм Leapfrog-Verlet при количестве точек равных 10 в последовательной версии работает быстрее. Алгоритм всех пар показал худшее время работы и не рекомендуется при большом количестве взаимодействий, при этом сравнительно небольшая разница результатов работы в параллельной и последовательной версиях обусловлена особенностью алгоритма.

Заключение

В данной работе было приведено описание математической постановки задачи взаимодействия N тел по гравитационному закону для расчёта эволюции системы N тел. Произведен анализ существующих методов численного решения задачи, а именно: метода всех пар расчёта взаимодействия по принципу «каждый-с-каждым», методов Барнса-Хата и Leapfrog-Verlet. Была произведена реализация программ решения задачи N тел в последовательной и параллельной версиях и сделан анализ времени выполнения расчётов. Алгоритм Барнса-Хата показал лучшее время работы на процессоре Intel Core i5-4300U по сравнению с остальными.

Список литературы

[1] *Рой А.* Движение по орбитам. – Москва: Мир, 1981 – 544 с.

[2] *Алексеев В.М.* Лекции по небесной механике. – Ижевск: Ред. Журнала «Регулярная и хаотическая механика», 1999 – 160 с.

[3] *Крашенинников К.Г., Морозов А.Ю.* Численное моделирование гравитационной задачи N-тел на GPU с использованием технологии CUDA. Москва: Мир, 2016 – 16 с.

[4] *Адинец А.В.* Анализ эффективности решения задачи N тел на различных вычислительных архитектурах [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/analiz-effektivnosti-resheniya-zadachi-n-tel-na-razlichnyh-vychislitelnyh-arhitekturah (дата обращения 11.04.2019)

[5] Алгоритм Барнса-Хата [Электронный ресурс]. URL: http://arborjs.org/docs/barnes-hut (дата обращения 11.04.2019).

[6] Численное решение задачи N тел [Электронный ресурс]. URL: https://evatutin.livejournal.com/47718.html (дата обращения 11.04.2019)

[7] Технология OpenMP [Электронный ресурс]. URL: http://ccfit.nsu.ru/arom/data/openmp.pdf (дата обращения 11.04.2019)

[8] Omp4j [Электронный ресурс]. URL: http://www.omp4j.org/ (дата обращения 11.04.2019)