Параллельные алгоритмы для решения обратных задач динамики небесных тел[1]

И.Н. Чувашов

НИИ прикладной математики и механики Томского государственного университета, Томск

Рассматривается возможность использования параллельных вычислений для решения обратных задач динамики небесных тел. Особенности и трудоемкость этой задачи позволяют на всем этапе численного моделирования применять параллельные алгоритмы, что приводит к значительному увеличению скорости работы приложения и дает возможность использовать разрядную сетку высокого порядка (128 бит) без существенных временных затрат.

Формулировка прямой и обратной задач динамики ИСЗ

Прямая задача динамики ИСЗ определяется уравнениями движения спутника с заданными начальными условиями. Как известно, движение искусственного спутника Земли можно представить как движение материальной частицы бесконечно малой массы в поле тяготения центрального тела с массой M под действием сил, определенных потенциальными функциями V и R, а также совокупности сил P, не имеющих потенциала, и записать в виде

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}}, \quad \frac{d\dot{\mathbf{x}}}{dt} = Q\frac{\partial V}{\partial \mathbf{x}} + \frac{\partial R}{\partial \mathbf{x}} + \mathbf{P} \tag{1}$$

с начальными условиями

$$\mathbf{x}_0 = \mathbf{x}(t_0), \quad \dot{\mathbf{x}}_0 = \dot{\mathbf{x}}(t_0), \tag{2}$$

причем Q – матрица перехода из вращающейся системы координат в инерциальную систему [1].

---

[1] Работа выполнена по заданию Министерства образования и науки РФ № 8.4859.2011 и при частичной финансовой поддержке РФФИ, грант 11-02-00918-а

Обратная задача динамики ИСЗ и алгоритм ее решения могут быть сформулированы следующим образом. Пусть $\rho_j = \rho_j\left(q_i\right)$– измеренные величины, а q_i – определяемые параметры, связь между которыми через решение уравнений (1) задается нелинейным соотношением

$$\rho_j = F_j\left(q_i\right),\ j = 1, 2, ...N,\ i = 1, 2....m,$$

причем $j \gg i$. Следовательно, мы имеем избыточную нелинейную систему уравнений для определения неизвестных параметров. Решение этой системы возможно при условии существования минимума функции

$$\Phi(q) = \sum_{i=1}^{N}\left[\Delta\rho_i(q)\right]^2 = \min\ , \tag{3}$$

где $\Delta\rho_i = \rho_o - \rho_c$ – невязки, представляющие собой разность наблюденных и вычисленных значений измеряемых величин. Задачу минимизации (3) принято называть задачей наименьших квадратов (НК). Решение нелинейной задачи НК производится методом Гаусса–Ньютона, путем нахождения дифференциальных поправок Δq в определяемые элементы.

Итерационный процесс считается завершенным при выполнении следующих условий:

$$\left|q_i^{n+1} - q_i^{n}\right| \prec \varepsilon, \tag{4}$$

где ε – заданная точность вычислений.

Вычисление обратной матрицы осуществляется методом сингулярного анализа [2].

Особенности программной реализации алгоритма

При решении задач динамики небесных тел в среде параллельных вычислений используются два способа реализации: функциональная декомпозиция задачи и декомпозиция системы объектов. В первом случае параллельно вычисляются модели сил, действующих на объект. При таком подходе распараллеливания неизбежно возникает проблема балансировки, т. е. обеспечения равномерного распределения вычислительной нагрузки между параллельными процессами. Кроме того, адаптация программного комплекса для суперкомпьютера может занять значительное время.

Во втором подходе система объектов разделяется на подсистемы, число которых равно числу процессов, что позволяет без существенных трудозатрат на модификацию программного кода применить программный комплекс для решения прямых и обратных задач.

Нами было сделано сравнение обоих этих способов решения обратных задач по быстродействию и эффективности.

Параллельное решение обратной задачи с помощью функциональной декомпозиция, реализованное в разработанном программном комплексе, позволяет задействовать не более четырех ядер (рис. 1).

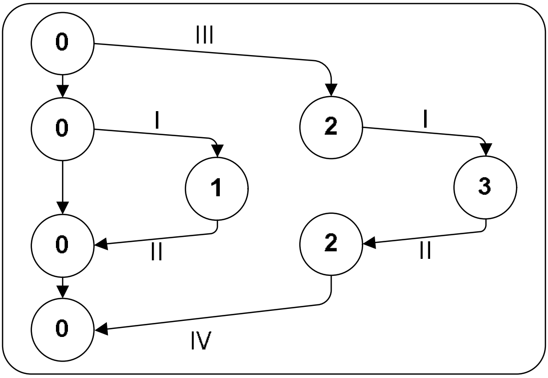

Рис. 1. Блок-схема программного комплекса. Этап I, II – обмен данными в функции правых частей; III, IV – обмен входными и выходными данными

Нами были рассмотрены промежуточные варианты реализации задачи для одного, двух и трех ядер. Использование одного потока для решения обратной задачи приведено в качестве сравнения с работой на персональном компьютере. Распараллеливание задачи на два потока позволило интегрировать уравнения (1) с прямым и обратным шагом. Решение задачи с использованием трех ядер позволило выделить отдельный поток для вычисления модели сил для первого и второго ядер, что приводит к сильной разбалансировке. Эта проблема решается с выделением еще одного ядра, и, таким образом, количество потоков увеличивается до четырех.

Результаты этих исследований приведены в табл. 1. В качестве примера было рассмотрено время решения обратной задачи для ИСЗ Лагеос на одной итерации.

Таблица 1.Зависимость времени работы от количества используемых ядер

Количество ядер	Время решения обратной задачи, сек
1	33,65
2	16,88
3	16,11
4	12,11

Видно, что время выполнения программного комплекса уменьшается и минимальное время достигается при решении задачи на четырех ядрах. Применение трех ядер для решения этой же задачи является нецелесообразным, так как на третье ядро ложится значительная вычислительная нагрузка, что вызывает ожидания на первых двух потоках. Это приводит к разбалансировке в решаемой задаче, а использование четырех ядер устраняет эту проблему. Дальнейшее увеличение количества ядер в рассматриваемой задаче приведет к большой разбалансировке и создаст дополнительные сложности в реализации программного комплекса. К тому же явным преимуществом решения задачи на четырех ядрах по сравнению с решением на большем количестве потоков является выполнение задачи на одном четырехъядерном узле без использования протокола пересылки данных между узлами, что тоже сильно влияет на быстродействие. Иногда в таких реализациях можно получить значительное ускорение, связанное с пересылкой информации через кэш-память процессора, не используя даже оперативную память узла.

Нами для полноты исследования параллельной реализации обратной задачи в небесной механике была внедрена в программный комплекс декомпозиция системы объектов. Так как в программном комплексе изохронные производные находятся численно варьированием соответствующей компоненты вектора состояния, система состоит из восьми траекторий: одна изначальная траектория объекта, шесть варьированных траекторий по координатам и скоростям и одна траектория, которая используется для определения ускорения вдоль орбиты объекта. Используя эту информацию, программный комплекс сначала был разделен на восемь ядер (рис. 2), а потом, чтобы корректно сравнить с предыдущим исследованием, на четыре ядра – по две траектории на поток.

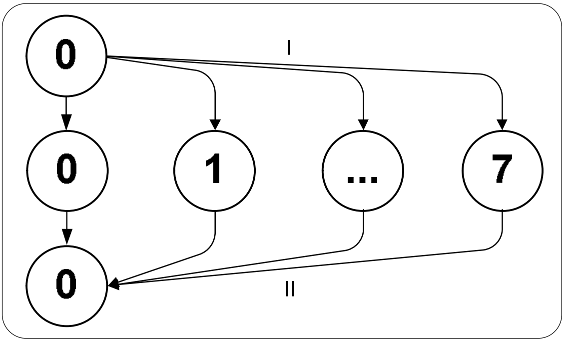

Рис 2. Информационный граф программного комплекса. I, II – обмен входными и выходными данными

Структурно информационные графы для четырех и восьми ядер не отличаются друг от друга, поэтому мы приводим только информационный граф для восьми ядер. На начальном этапе инициализации внутренних переменных программного комплекса происходит пересылка данных для нужного ядра (I), где выполняется независимое интегрирование уравнений движений (1). Когда моменты наблюдений совпадают с шагом интегрирования, происходит сборка информации по изохронным производным на нулевом ядре (II), где в дальнейшем, после интегрирования определяется новый вектор состояния.

После реализации двух совершенно разных подходов для решения обратных задач было проведено сравнение быстродействия (табл. 2).

Таблица 2. Сравнение быстродействия двух реализаций

Номер алгоритма	Время выполнения, сек	
	На 64-битной разрядной сетке	На 128-битной разрядной сетке
1	12,11	231,4
2	43,78	875,6
3	34,10	682,1

В табл. 2 номера алгоритмов означают: 1 – параллельная реализация функциональной декомпозиции на четырех потоках; 2 –

распределение системы объектов на четыре потока; 3 – распределение системы объектов на восемь потоков. Время в табл. 2 измерено для одной итерации. Видим, что программный комплекс, использующий параллельное вычисление модели сил, показывает быстродействие в три-четыре раза выше, чем другие реализации. Такое быстродействие можно объяснить тем, что информация между потоками обменивается через кэш-память процессора.

Результаты исследования показывают, что не всегда простая и быстрая параллельная реализации задач может принести эффективную работу приложения. Иногда нужно учитывать особенности программной реализации программного комплекса и аппаратной конфигурации кластера.

Литература

1. *Бордовицына Т.В., Авдюшев В.А.* Теория движения ИСЗ. Аналитические и численные методы. Томск: Изд-во Том. ун-та, 2007. 105 с.
2. *Форсайт Дж., Малькольм М., Моулер К.* Машинные методы математических вычислений. М.: Мир, 1980. 279 с.
3. *Немнюгин С.А., Стесик О.Л.* Параллельное программирование для многопроцессорных вычислительных систем. СПб.: БХВ-Петербург, 2002.