УДК 519.6
Г.В. Заручевская, И.В. Костин

ПРИМЕНЕНИЕ МЕЛКОЗЕРНИСТОГО ЛОКАЛЬНО-ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ ДЛЯ ВЫЧИСЛЕНИЯ КРАТНЫХ ИНТЕГРАЛОВ МЕТОДОМ ТРАПЕЦИИ В ТРЁХМЕРНОМ ПРОСТРАНСТВЕ

Аннотация. В статье рассматриваются возможности использования стиля мелкозернистого локально-параллельного программирования для приближенного вычисления кратных интегралов. Приведен МЛПП – алгоритма приближенного вычисления кратных интегралов с использованием метода трапеции.

Ключевые слова: мелкозернистое локально-параллельное программирование, методы численного интегрирования, метод трапеции.

Господствующим способом распараллеливания задач до сих пор является крупноблочное. Очевидно, что с ростом числа процессоров блоки измельчаются, и вычисления в подавляющем большинстве случаев будут идти медленнее: параллелизм вырождается. Избежать вырождения можно только при условии, что обмены происходят и одновременно, и локально, т.е. физическое расстояние между взаимодействующими процессорами мало и не зависит от размера задачи.

При этом задача должна быть разбита на множество небольших однотипных подзадач, которые будут исполняться параллельно на отдельных вычислительных машинах (ВМ). Данные максимально распределены по системе, а программы в каждой ВМ используют минимально возможные наборы данных. При мелкозернистом программировании решающее значение имеет организация обменов данными между ВМ. В общем случае число обменов имеет тот же порядок, что и число вычислительных операций. Таким образом, мелкозернистость, или массовое распараллеливание означает, что в каждом вычислительном процессе в каждый момент времени содержится минимальное число команд (тело внутреннего цикла) и данных (элементы массивов, необходимые для вычисления одного витка цикла). Такой подход к распараллеливанию алгоритмов носит название мелкозернистого локально-параллельного программирования (МЛПП) [1, С. 55].

При описании алгоритма параллельных вычислений на ЭВМ класса MIMD предполагается, что MIMD –машина состоит из p одинаковых процессоров, каждый из которых обладает определенным объемом своей локальной памяти (одинаковым для всех параллельных процессоров) и способен осуществлять численную обработку информации в автономном и управляемом режимах.

Отметим существенные особенности архитектуры MIMD – машины, при которых стиль МЛП – программирования был наиболее эффективен.

1. Попарное соединение процессоров осуществляется за очень короткий промежуток времени и поэтому оно не учитывается.

2. Все возможные в данный момент обмены машинными словами совершаются параллельно и одновременно с процессом счёта за время, сравнимое со временем выполнения одной арифметической операции (из-за близости связанных процессоров в физическом пространстве).

3. Имеется возможность программировать структуру межпроцессорных связей.

Разработка и исследование МЛПП для задач математической физики – одно из актуальных направлений современного параллельного программирования. В [2, с. 17] для этих целей применяются модели клеточных автоматов.

Обычно MIMD-машины имеют следующие структуры межпроцессорных связей: линейка, кольцо, кольцо с хордами, сетка (решетка), гиперкуб, дерево. Однако, в теории однородных вычислительных структур, помимо вышеназванных типов структур связей между процессорами, рассматривается тор. Важно отметить, что данная евклидова структура сходна с сеткой процессоров, в которой ВМ на противоположных сторонах решетки связаны регулярным каналом. При этом тор позволяет легко масштабировать произвольную сеточную область. Покажем способ вложения произвольной сеточной области в тор $C \times D \times L$. Согласно [1, с. 51], такую тороидальную евклидову структуру будем обозначать $E_3\{C, D, L\}$. Каждому процессору тора (x, y, z) ставится соответствующая ячейка сетки (x+Ck, y+Dn, z+Lp), $k, n \in N$. Данное распределение данных называется циклическим. При данном распределении сеточных узлов в тороидальной структуре параллельный алгоритм соответствует МЛП-стилю.

Для дальнейшего использования вводится понятие процессорного тороидально связанного куба $C \times D \times L$. Каждый процессор обозначается Pi, j, m, где i, j, m – координаты узла, в котором располагается процессор. Дополним процессорный куб следующими связями: каждый процессор $P_{1,j,m}$ соединим регулярным каналом с процессором $P_{C,j,m}$, j = 1..D, m = 1..L; $P_{i,1,m}$ соединим регулярным каналом с процессором $P_{i,D,m}$, i = 1..C, m = 1..L; $P_{i,j1}$ соединим регулярным каналом с процессором $P_{i,j,L}$, i = 1..C, j = 1..D. Получим новую структуру – тороидально связанный куб C×D×L [3, с. 92].

Целью настоящей работы является разработка МЛПП-алгоритма приближенного вычисления кратных интегралов с использованием метода трапеции.

В качестве модельной задачи рассмотрим задачу приближенного вычисления трехмерного интеграла с конечными пределами методом трапеции:

$$\int_a^b \int_c^d \int_e^r f(x, y)\mathrm{d}x\mathrm{d}y\mathrm{d}z \approx 0{,}125 \sum_{m=0}^{k-1} \sum_{j=0}^{n-1} \sum_{i=0}^{p-1} h_i^x \cdot h_j^y \cdot h_m^z \cdot f(x_i, y_j, z_m) +$$

$$+ f(x_{i+1}, y_j, z_m) + f(x_i, y_{j+1}, z_m) + f(x_{i+1}, y_{j+1}, z_m) + f(x_i, y_j, z_{m+1}) + f(x_{i+1}, y_j, z_{m+1}) +$$
$$+ f(x_i, y_{j+1}, z_{m+1}) + f(x_{i+1}, y_{j+1}, z_{m+1}), \qquad (1)$$

где $f(x, y, z)$ – непрерывная в области $\overline{\Omega}$ функция трех переменных x, y, z,

$\overline{\Omega} = \{(x, y, z):\ a \leqslant x \leqslant b,\ c \leqslant y \leqslant d,\ e \leqslant z \leqslant r\}$.

В области $\overline{\Omega}$ введем равномерную по обеим переменным сетку

$$\overline{\omega} = \{(x_i, y_j, z_m) \in \overline{\Omega}, x_i = a + ih_1, i = 0,1, \dots, N_1, y_i = c + jh_2;$$
$$j = 0,1, \dots, N_2, zm = mh_z, m = 0,1, \dots, N_3, N_1 = \frac{b-a}{h_1},$$
$$N_2 = \frac{d-c}{h_2}, N_3 = \frac{r-e}{h_3}\}.$$

Рассмотрим выполнение алгоритма параллельных вычислений, основанного на формуле трапеции в тороидальной структуре процессоров $E_3\{C, D, L\}$ машины третьей модели.

Каждому процессору тороидального куба $E_3\{C, D, L\}$ с координатами (x, y, z) сопоставляется ячейка сетки (x+Ck, y+Dn, z+Lp), k, n, p∈N, т.е. сетка произвольной размерности $N_1 \times N_2 \times N_3$ вкладывается в тороидальный куб C×D×L. Итак, область ϖ разбивается на $([N_1/C]+1)\cdot$ $\cdot([N_2/D]+1)\cdot([N_3/L]+1)$ подобластей $\omega^{k,n,p}$, где $k = 1, \dots, ([N_1/C]+1)$, $n = 1, \dots, ([N_2/D]+1)$, $p = 1, \dots, ([N_3/L]+1)$, размерности C×D×L прямыми, параллельными обеим координатным осям.

Шаг 1. Распределяются значения $f(x_i, y_j, z_m)$, h^x_i, h^y_j, h^z_m в соответствующий процессор (i- [i/C]*C; j- [j/D]*D; m-[m/L]*L), вводится переменная для подсчёта суммы $S_{xyz} = 0$, (i, j, m) – номер процессора.

Шаг 2. Все процессоры по регулярному каналу совершают 1 сдвиг полученных значений данных вперёд.

Шаг 3. Все процессоры по регулярному каналу совершают 1 сдвиг влево.

Шаг 4. Все процессоры по регулярному каналу совершают 1 сдвиг вниз. Собранные в процессорах значения суммируются по формуле:

$$S_{ijm} = S_{ijm} + 0{,}125\cdot h^x_i \cdot h^y_j \cdot h^z_m \cdot f(x_i, y_j, z_m)+ f(x_{i+1}, y_j, z_m)+f(x_i, y_{j+1}, z_m)+$$
$$+f(x_{i+1}, y_{j+1}, z_m)+f(x_i, y_j, z_{m+1})+f(x_{i+1}, y_j, z_{m+1})+f(x_i, y_{j+1}, z_{m+1})+f(x_{i+1}, y_{j+1}, z_{m+1}). \qquad (2)$$

2, 3, 4 этап повторяются до тех пор, пока не будет рассчитана вся область.

Шаг 7. Все процессоры по регулярному каналу совершают D-1 сдвигов влево и передают полученные суммы S_{ijm} в процессоры (1, j, z), переданные значения суммируются.

Шаг 8. Все процессоры плоскости СОЕ по регулярному каналу совершают С-1 сдвигов значений S_{1jm} в линейку процессоров (1, 1, z).

Шаг 9. Все процессоры в линейке (1, 1, z) совершают L-1 сдвигов в процессор (1, 1, 1). Собранные в процессоре значения суммируются и передаются на хост-машину.

Можно заменить одной глобальной операцией 7, 8, 9 шаги – сбор полученных сумм S_{ij} и расчёт приближённого интеграла $S = \sum S_{ij}$.

Количество узлов сетки ω Q = (N1-1)×(N2-1)×(N3-1); время для реализации алгоритма на одном процессоре T_1 = MQt, где M – количество арифметических операций, необходимое для вычисления (t+1)-го приближения к решению в одном узле сетки, t – среднее время выполнения одной арифметической операции. $T_p = M([N_1/C]+1)\cdot([N_2/D]+1)\cdot([N_3/L]+1)\cdot(t+t_0)$, где t_0 – время обмена одним машинным словом между двумя ПП, следовательно, коэффициент ускорения $k_у = T_1/T_p \approx CDE$ и коэффициент эффективности $k_э \approx 1$, т.е. параллелизм максимален.

В [3, с. 94] автором показано, что вышеописанные евклидовы тороидальные структуры столь же эффективно применимы для мелкозернистого локально-параллельного программирования сеточных задач.

В заключение необходимо отметить то, что в настоящее время ведутся разработки по реализации МЛП программирования. Например, в [4, с. 20] представлена реализация мелкозернистого параллелизма на чиповом мультипроцессоре как на программном, так и на аппаратном уровнях. Исследование МЛПП-алгоритмов способствует развитию инженерного мышления у современной молодежи.

Библиографический список

1. Воробьев В.А. Об эффективности параллельных вычислений // Автометрия – 2000. – № 1. С. 50–58.

2. Бандман О.Л. Мелкозернистый параллелизм в математической физике // Программирование – 2001. – № 4. С. 12–25.

3. Заручевская Г.В. Реализация решения разностной схемы расщепления трехмерного уравнения теплопроводности в мелкозернистом локально-параллельном стиле программирования //-Системы управления и информационные технологии, 2007, № 4 (30), 104 с., С. 91–94.
4. Lee S.H. Exploration of Fine-Grained Helper Computing Parallelism on a Chip Multiprocessor [Electronic resource]: дис… Ph. D. / Lee Sang Hoon. – North Carolina State University, 2012–88 p. – Electronic text data – Mode of access: https://search.proquest.com/docview/1034721752/8F6002038B214B-22PQ/1? accountid=141903, free access (21.10.18). – Title from screen.

УДК 101.1, 141.112, 165.62, 168 **А.С. Скутин**

ОБЪЕКТИВНО-ЛОГИЧЕСКОЕ АПРИОРИ НАУЧНО-КОНСТРУКТИВНОГО ПОЗНАНИЯ: ВОЗМОЖНОСТИ И ОГРАНИЧЕНИЯ

Аннотация. В статье исследуется априорная платформа научно-конструктивного познания, его возможности и ограничения. Применяется генеративно-телеологический подход к раскрытию исторического генезиса рационалистической традиции. Осуществляется сущностная дескрипция объективно-логического априори и сопутствующей ему дуалистической онтологии, частично продемонстрирована интенциональная динамика соответствующей им научно-конструктивной установки сознания.

Ключевые слова: телеологически-исторической самоосмысление, объективно-логическое априори, научно-конструктивное познание, идеализированная предметность, дуализм, жизненный мир.

Обращение к теме научно-конструктивного познания объясняется рядом трудностей уже ставшими само собой разумеющимися для большинства не только исследователей ученых, но и для любого, кто так или иначе согласует свою профессиональную практику с общепризнанными научно-теоретическими установлениями. Последствия этих трудностей, однако, оказывают непосредственное влияние на нашу повседневную жизнедеятельность. Но научное познание не безосновно, его невидимое основание задается объективно-логическим априори, которое являет себя в своего рода сущностной форме признаваемой каждым ученым бытийственно-