

А. В. Медведев¹, В. М. Свешников², И. Ю. Турчановский¹

¹Институт вычислительных технологий СО РАН
пр. Акад. Лаврентьева, 6, Новосибирск, 630090, Россия

²Институт вычислительной математики и математической геофизики СО РАН
пр. Акад. Лаврентьева, 6, Новосибирск, 630090, Россия

E-mail: andertsk@gmail.com; victor@lapasrv.sccc.ru; tur@hcei.tsc.ru

РАСПАРАЛЛЕЛИВАНИЕ РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ НА КВАЗИСТРУКТУРИРОВАННЫХ СЕТКАХ С ИСПОЛЬЗОВАНИЕМ ГИБРИДНЫХ ВЫЧИСЛЕНИЙ CPU + GPU *

Дается описание технологий распараллеливания решения двумерных краевых задач на квазиструктурированных сетках специальным методом декомпозиции в гибридной вычислительной среде: центральный процессор (CPU) + графические ускорители (GPU). Приведены результаты численных экспериментов на модельной задаче, показывающие эффективность предлагаемого подхода.

Ключевые слова: квазиструктурированные сетки, распараллеливание алгоритмов, краевая задача, гибридные вычисления, CUDA, OpenMP.

Введение

Квазиструктурированные сетки, рассматриваемые в настоящей работе, формируются из прямоугольных структурированных сеток и строятся в два этапа. На первом из них расчетная область покрывается структурированной прямоугольной макросеткой, т. е., по сути дела, проводится декомпозиция на подобласти, являющиеся макроэлементами. На втором этапе в каждой подобласти строится своя структурированная микросетка. Для адаптации проводится локальная модификация сетки вблизи криволинейных границ [1]. Решение краевых задач на таких сетках ищется вариантом метода декомпозиции, основанном на прямой аппроксимации уравнения Пуанкаре – Стеклова на границе сопряжения подобластей (интерфейсе) [2]. В работе [3] рассматривается эффективность данного подхода к решению краевых задач в системе MPI на центральном процессоре.

В настоящей работе для решения краевых задач на квазиструктурированных сетках применяются графические ускорители (GPU) и вычисления проводятся в гибридной среде CPU + GPU. Суть метода декомпозиции заключается в проведении итераций по подобластям, причем на каждом шаге необходимо решать краевые подзадачи. Для решения системы линейных алгебраических уравнений, аппроксимирующих уравнение Пуанкаре – Стеклова, исполь-

* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 12-01-00076) и СО РАН (интеграционные проекты № 104, 126, 130).

Медведев А. В., Свешников В. М., Турчановский И. Ю. Распараллеливание решения краевых задач на квазиструктурированных сетках с использованием гибридных вычислений CPU + GPU // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2014. Т. 12, вып. 1. С. 50–54.

зуется итерационный метод сопряженных градиентов¹. Решение подзадач в подобластях осуществляется методом последовательной верхней релаксации [4], который обладает внутренним параллелизмом, легко отображаемым на GPU. Более подробно на эту тему см. работу [5], где экспериментально исследована эффективность применения GPU только для решения подзадач в подобластях.

Сравнение быстродействия в настоящей работе проводилось между тремя версиями кода: 1) последовательный код на CPU; 2) параллельный код на CPU; 3) гибридный код – параллельный код на CPU + код GPU. Были проведены численные эксперименты на модельной задаче, которые показали, что быстродействие варианта 3 по отношению к варианту 1 увеличивается в 25,2 раза, а по отношению к варианту 2 – в 8,3 раза.

Постановка краевой задачи

В замкнутой двумерной области $\bar{G} = G \cup \Gamma$ с границей Γ требуется решить краевую задачу:

$$\Delta u = g_1, lu|_{\Gamma} = g_2,$$

где $u = u(T)$ – искомая функция; $g_1 = g_1(T)$, $g_2 = g_2(T)$ – заданные функции ($T = (x, y)$ – текущая точка); x, y – декартовы или цилиндрические координаты); Δ – оператор Лапласа; l – оператор граничных условий. Решение данной задачи ищется на квазиструктурированной сетке методом декомпозиции.

Построение квазиструктурированных сеток

Вокруг расчетной области \bar{G} описывается прямоугольник $R = \{0 \leq x \leq D_x, 0 \leq y \leq D_y\}$, где D_x, D_y заданы. В R строится равномерная *макрорешетка*:

$$\Omega_H = \left\{ X_I = IH_x; Y_J = JH_y; I = \overline{0, N_x}; J = \overline{0, N_y}; H_x = \frac{D_x}{N_x}; H_y = \frac{D_y}{N_y} \right\},$$

где N_x, N_y – заданные числа, H_x, H_y – шаги, которые намного превосходят шаги сетки, на которой аппроксимируется исходная задача.

Таким образом, исходная область фактически подвергается декомпозиции на непересекающиеся подобласти. Все подобласти делятся на три типа: внутренние, содержащие только точки G , граничные, содержащие точки G и Γ , а также внешние, не содержащие точек G . В дальнейшем внешние подобласти исключаются из расчетов.

Точки пересечения координатных линий образуют *макроузлы*. Координатные линии макросетки являются границами сопряжения подобластей или *интерфейсом*.

После того как построена макросетка, внутри каждой подобласти строится своя сетка (*микросетка*) $\bar{\Omega}_{h,k} = \{x_{i_k} = X_I + i_k h_{x,k}, y_{j_k} = X_J + j_k h_{y,k}, i_k = \overline{0, n_{x,k}}, j_k = \overline{0, n_{y,k}}\}$, где $n_{x,k}, n_{y,k}$ –

заданные целые числа, с шагами $h_{x,k} = \frac{X_{I+1} - X_I}{n_{x,k}}$; $h_{y,k} = \frac{X_{J+1} - X_J}{n_{y,k}}$. Подчеркнем, что величи-

ны $n_{x,k}, n_{y,k}$, которые определяют плотность узлов в подсетках, задаются в зависимости от особенностей решаемой задачи, т. е. подсетки могут быть несогласованными.

Для адаптации сетки к внешней границе проводится сдвиг узлов, отстоящих от нее на расстояние меньше половины шага, на данную границу, т. е. локальная модификация сетки. Более подробно процедура модификации описана в работе [1].

На интерфейсе строится своя сетка ω_h , которая в нашем случае состоит из узлов подсеток $\bar{\Omega}_{h,k}$.

¹ Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. URL: <http://www.siam.org/books>.

Технология проведения расчетов

Для нахождения решения исходной краевой задачи на квазиструктурированной сетке применяется метод декомпозиции расчетной области на непересекающиеся подобласти. Сам метод декомпозиции и технология его применения к квазиструктурированным сеткам изложены в работах [2; 5]. Суть рассматриваемого подхода заключается в проведении итераций по подобластям или макроэлементам квазиструктурированной сетки. На каждой такой итерации необходимо решать краевые задачи Дирихле во внутренних и граничных подобластях. Аппроксимация этих задач проводится по пятиточечным или девятиточечным (вблизи криволинейной границы) схемам. Система линейных алгебраических уравнений, аппроксимирующая уравнение Пуанкаре – Стеклова, решается методом сопряженных градиентов, для чего необходимо вычислять суммы внутренних нормальных производных на интерфейсе. Подзадачи в подобластях решаются методом последовательной верхней релаксации (ПВР), который обладает внутренним параллелизмом и хорошо отображается на архитектуру GPU [5]. Распараллеливание самого итерационного процесса по подобластям проводится на CPU в рамках системы MPI [6]. Распараллеливание решения краевых задач осуществляется на GPU в рамках системы CUDA². В целом технология проведения расчетов выглядит следующим образом.

1. На CPU на каждом шаге итерационного процесса по подобластям вычисляются значения искомой функции в узлах сетки ω_n на интерфейсе.
2. Эти значения копируются в память GPU.
3. На GPU методом ПВР решаются краевые задачи в подобластях и на сторонах вычисляются внутренние нормальные производные.
4. Полученные значения на сторонах подобластей копируются в память CPU.
5. Пересчитываются значения на интерфейсе и описанный процесс повторяется, начиная с шага 2, пока он не сойдется с заданной точностью.

Численные эксперименты

Численные эксперименты проводились на следующей тестовой задаче:

$$\Delta u = 0, \quad u|_{\Gamma} = g$$

в квадрате $\bar{G} = \{1 \leq x \leq 1,5, 0 \leq y \leq 0,5\}$. Данная задача представляла собой фрагмент более общей задачи о решении уравнения Лапласа в области, ограниченной двумя концентрическими окружностями с радиусами $R_1 = 1$, $R_2 = 2$, на которых задаются значения искомой функции \bar{u} , равные соответственно 0, 1. Последняя задача имеет точное решение:

$$\bar{u} = \ln \frac{r}{R_1} / \ln \frac{R_2}{R_1},$$

где r – радиус-вектор. Функция g выбиралась, как $g = \bar{u}|_{\Gamma}$, и тогда точное решение исходной задачи есть $u = \bar{u}$ в G .

Цель проводимых экспериментов – установить ускорение вычислений с использованием гибридной схемы по отношению к последовательным и параллельным расчетам на CPU. Для этого была написана программа на языке программирования CUDA C. Расчетные эксперименты проводились на оборудовании Intel Core i5, NVIDIA GPU Tesla C2070 в среде linux CentOS 6.2.

Квазиструктурированная сетка состояла из подсеток, каждая из которых содержала 32×32 узла, что обусловлено тем, что на каждом мультипроцессоре именно 32 дискретных вычислителя, а потоки (нити) исполняются группами по 32. Макросетка имела параметры $N_x = N_y = N$, причем N изменялось в пределах 3, ..., 256.

² NVIDIA. CUDA C Best Practices Guide. 2012.

Приведем два графика зависимости ускорения от N – числа подобластей в одном направлении (рис. 1 – ускорение программы с использованием GPU по отношению к однопоточной программе, а рис. 2 – по отношению к многопоточной – 4 потока). Критерием останова метода сопряженных градиентов в итерациях по подобластям было уменьшение нормы начальной невязки до величины $\varepsilon = 1,0e - 14$ (использовались данные типа *double*).

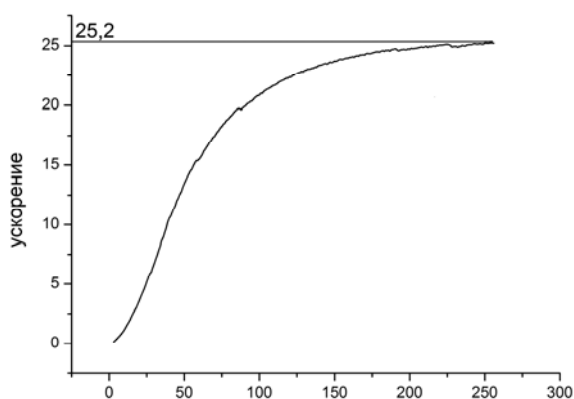


Рис. 1. График ускорения вычислений по отношению к однопоточному коду

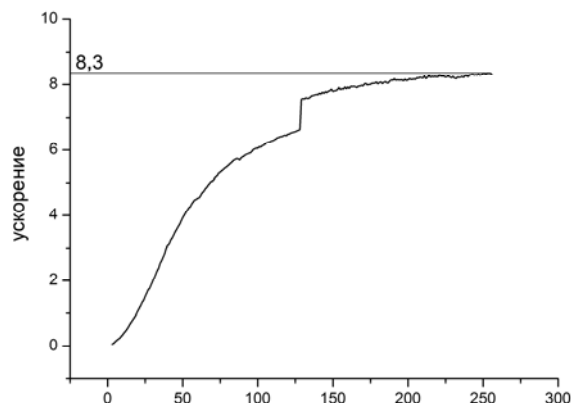


Рис. 2. График ускорения вычислений по отношению к многопоточному коду

Поведение графиков обусловлено тем, что, начиная с некоторого N , на GPU становятся задействованными практически все процессорные ресурсы, т.е. каждый мультипроцессор загружен на $\approx 100\%$, после чего значение ускорения стабилизируется. Если сравнивать графики между собой, то видно, что значения ускорения разнятся примерно в 2 раза. Это объясняется тем, что операций над типами *double* работают с уменьшенной в 2 раза тактовой частотой. Объяснение резкому скачку между $N = 128$ и $N = 129$ на рис. 2 пока не найдено.

Время расчетов с использованием GPU для $N = 256$ и $n = 32$ (количество узлов квазиструктурированной сетки равно $256 \cdot 256 \cdot 32 \cdot 32 \approx 67$ млн) составило 51,38 с.

Заключение

В настоящей работе получены следующие основные результаты.

1. Разработаны технологии распараллеливания решения двумерных краевых задач на квазиструктурированных сетках методом декомпозиции, основанном на прямой аппроксимации уравнения Пуанкаре – Стеклова на интерфейсе в гибридной вычислительной среде CPU + GPU.

2. Получено, что быстродействие программы с применением GPU в рамках текущей реализации по отношению к аналогичной программе на CPU увеличивается в 25,2 раза при использовании последовательного кода на CPU, и в 8,3 раза – при параллельном коде на CPU (4 ядра).

Список литературы

1. Беляев Д. О., Свешников В. М. Построение квазиструктурированных локально-модифицированных сеток для решения задач сильноточной электроники // Вестн. Южно-Урал. гос. ун-та. 2012. № 40 (299). С. 118–128.
2. Свешников В. М. Построение прямых и итерационных методов декомпозиции // Сиб. журн. индустр. матем. 2009. Т. 12, № 3 (39). С. 99–109.
3. Свешников В. М., Рыбдылов Б. Д. О распараллеливании решения краевых задач на квазиструктурированных сетках // Вестн. Южно-Урал. гос. ун-та. Серия ВМИ. 2013. Т. 2, № 3. С. 63–72
4. Ильин В. П. Методы конечных разностей и конечных объемов для эллиптических уравнений. Новосибирск: Изд. Ин-та математики, 2000. 345 с.

5. *Медведев А. В., Свешников В. М., Турчановский И. Ю.* Распараллеливание решения сеточных уравнений с использованием графических ускорителей // XIV Рос. конф. с участием иностранных ученых «Распределенные информационные и вычислительные ресурсы» – DICR – 2012 (Новосибирск, Россия, 26–30 ноября 2012 г.): Материалы конф. Новосибирск: ИВТ СО РАН, 2012. URL: <http://conf.nsc.ru/dicr2012/ru/reportview/140603>

6. *Корнеев В. Д.* Параллельное программирование в MPI. Новосибирск: ИВМиМГ СО РАН, 2002.

Материал поступил в редколлегию 10.04.2014

A. V. Medvedev, V. M. Sveshnikov, I. Yu. Turchanovskiy

**PARALLELIZATION OF SOLUTION OF BOUNDARY PROBLEMS ON QUASISTRUCTURED MESHES
USING HYBRID COMPUTATIONS CPU + GPU**

In the article describes parallelization technology of boundary value problems on quasistructured meshes with special decomposition method in a hybrid computing environment: the central processor (CPU) + graphics accelerators (GPU). Contains results of numerical experiments with a model problem, showing the effectiveness of the proposed approach.

Keywords: quasistructured meshes, algorithms parallelization, boundary problem, hybrid computations, CUDA, OpenMP.