

**МЕЖДУНАРОДНАЯ НАУЧНАЯ
КОНФЕРЕНЦИЯ**

**«АКТУАЛЬНЫЕ ПРОБЛЕМЫ МАТЕМАТИКИ
И МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ»**

ТЕЗИСЫ ДОКЛАДОВ

Алматы-2015

..... 268
 269
 опрессовке
 272
 авнения с
 274
 ости для
 вающихся
 275
 tov A.A.
 ase change
 materials 275
 l program-
 276
 all param-
 277
 ls 277
 278
 lectric arc 278
 cal bridge 280
 at closure
 283
 y 286

Nurmanov E.A., Beketauov B.
 Mathematical modelling for unsteady groundwater flow 286

Sarıyölekova M., Tuimebayeva A.
 On the reduction of boundary value problem for a loaded equation to
 an integral equation 287

Yermakunova B.N., Zhexembay L. and Karjanto N.
 Boundary value problems in Black-Scholes equation 288

5 Информационные технологии и вычислительная математика 289

Атыш А.Ш. Метод расщепления для нелинейного уравнения Каца .. 289

Башинев Г.Б., Дильман Т.В. Об одной задаче интегральной геометрии
 для некоторого семейства кривых 290

Башинев Г.Б., Турмаганбет К.А. Необходимое и достаточное условие
 существования решения одномерной дискретной обратной задачи . 291

Башинев Р.Г., Нысанбаева С.Е., Бегимбаева Е.Е. Модификация схемы
 цифровой подписи DSA 293

*Башинев Р. Г., Нысанбаева С. Е., Капалова Н.А., Хакимов
 Р.А.* Моделирование программной реализации непозиционного
 алгоритма шифрования 295

Дарбибаев Г. Численное решение обратной задачи для уравнения
 Беттагольда 297

Дарбибаев Б.С., Азмед-Заки Д.Ж., Иманкулов Т.С.
 Высокпроизводительные вычисления на мобильных платформах . 298

Евандибаев К.С., Биярова А.О., Махсут Г. Обучение нейронных сетей
 при решении модельного уравнения в частных производных 301

Кабанов С.И., Шишенин М.А., Шолпанбаев Б.Б. Численное
 решение задачи продолжения при наличии локализованных
 объектов 302

Климов В.С., Климов А.С. Применение нейронных сетей для
 диагностики качества контактной сварки 304

Лавинин Л. П. Графы, вершины которых составляют одну орбиту .. 306

Мамурова М.Е., Алимжанов Е.С., Кашкымбай И.Ж. Применение
 технологии MapReduce Hadoop для сжатия RDF данных 308

Диэлектрическая проницаемость в антеннах принимает следующие значения

$$\varepsilon = \begin{cases} \varepsilon_1, & \text{если } (x, y) \in G_3, \\ \varepsilon_2, & \text{если } (x, y) \in G_2 \setminus G_3, \\ \varepsilon_3, & \text{если } (x, y) \in G_1 \cup G_4. \end{cases}$$

На решение задачи (1)-(4) наложим условия склейки

$$\begin{aligned} \varepsilon_1 u_x(a-0, y) &= \varepsilon_2 u_x(a+0, y), & y \in [a, c] \cup [-c, a], \\ \varepsilon_2 u_x(-a-0, y) &= \varepsilon_1 u_x(-a+0, y), & y \in [a, c] \cup [-c, a], \\ \varepsilon_1 u_x(x, c-0) &= \varepsilon_2 u_x(x, c+0), & x \in [-a, a], \\ \varepsilon_2 u_x(x, -c-0) &= \varepsilon_1 u_x(x, -c+0), & x \in [-a, a], \\ \varepsilon_3 u_x(-d-0, y) &= \varepsilon_2 u_x(-d+0, y), & y \in [-b, b], \\ \varepsilon_2 u_x(d-0, y) &= \varepsilon_3 u_x(d+0, y), & y \in [-b, b], \end{aligned} \quad (5)$$

Литература

1. *Larry J. Segerlind* Applied finite element analysis. - New York: United States Copyright 1984. - 411 p.
2. *Кабиталин С.И.* Обратные и некорректные задачи. - Новосибирск: Сибирское научное издательство, 2009. - 457 с.

УДК 025.4.03; 002.6:004.65

Дарибаев Б.С., Ахмед-Заки Д.Ж., Иманкулов Т.С.

Казахский национальный университет имени аль-Фараби, Казахстан, Алматы
e-mail: beimbet.daribaev@gmail.com

Высокопроизводительные вычисления на мобильных платформах

В последнее время развитие мобильных технологий является одним из самых распространенных сегментов мировой промышленности и, как следствие, эти технологии развиваются с огромной скоростью. Мобильные процессоры представляют собой полноценные вычислительные единицы, которые могут быть использованы как в качестве дополнительных потоков для вычислительных кластеров, так и в одном блоке каждая может работать только с разделяемой памятью, известно, что у разделяемой памяти очень высокая пропускная способность, и здесь не надо каждый раз обращаться к глобальной памяти. Это дает ощутимую производительность. В оптимизированном параллельном алгоритме для вычисления выходного массива функция ядра объявляет временной массив в разделяемой памяти, и устанавливает его значение со значением входного массива и, в дополнение к этому, устанавливает граничные значения из соседних блоков. И тогда при вычисления уравнения каждый раз вызывает входной массив не из глобальной памяти, а из разделяемой памяти. За счет высокой пропускной способности разделяемой памяти ускоряется время расчета программы. При получении граничных значений из соседних блоков, нужно правильно подобрать индексы для того, чтобы работать с правильными данными, для самостоятельного расчета задач, требующих относительно малых ресурсов. Часто в производстве появляется необходимость для запуска задачи в оперативном режиме для внесения изменений в рабочей системе. Такая необходимость может возникнуть в различных областях. Данная статья описывает идею использования графического процессора (GPU) мобильного устройства для вычислительных целей.

иные значения

$$\begin{aligned} & a], \\ & -c, a], \end{aligned} \quad (5)$$

United States Copyright,

рек: Сибирское научное

ТОВ Т.С.

захстан, Алматы

МОБИЛЬНЫХ

гся одним из самых
цствие, эти технологии
представляют собой
заваны как в качестве
юм блоке каждая нить
еляемой памяти очень
ащаться к глобальной
ванном параллельном
няет временной массив
входного массива и, в
с блоков. И тогда при
з глобальной памяти,
разделяемой памяти,
значений из соседних
отать с правильными
льно малых ресурсов.
и оперативном режиме
ь может возникнуть
ования графического
й.

Использование мобильных технологий для расчетов обусловлено увеличением вычислительных возможностей и простоты использования для малых задач в отсутствие доступа к традиционным вычислительным мощностям. На стадии разработки была использована задача несмешивающегося вытеснения нефти (модель Баклея-Леверетта). Для распараллеливания на видеокарте была использована технология CUDA (Compute Unified Device Architecture) разработанная компанией NVIDIA [1]. Данная технология обеспечивает платформу для параллельных вычислений и модель программирования для распараллеливания с использованием любой современной видеокарты NVIDIA. Взаимодействие устройства и программного обеспечения, разработанных одним производителем обеспечивает высокую производительность распараллеливания на GPU. До прошлого года на мобильных устройствах было невозможно запускать приложения с применением технологии CUDA. Именно в 2014 году был представлен Nvidia Tegra K1 - первый мобильный процессор, поддерживающий технологию CUDA. Представленная программа была разработана и апробирована на Xiaomi MiPad - первом мобильном устройстве, работающем на данном процессоре. Для сравнения результатов на персональном компьютере используется видеокарта NVIDIA GeForce GTX 550Ti, обеспечивающей скорость вычислений 691 GFLOPS.

При параллельном алгоритме с использованием технологии CUDA сетка разбивается на блоки, где сами блоки имеют несколько нитей, и каждая нить вычисляет один узел, то каждый блок может копировать данные в разделяемую память, после чего каждый узел отдельного блока выполняет расчет и сохраняет рассчитанные данные в глобальную память. Для расчета в каждой подобласти, нужно использовать данные из соседней подобласти, т.е. нужно скопировать граничные данные из глобальной памяти [2].

До появления процессора NVIDIA Tegra K1 на мобильных устройствах невозможно было запустить программу на больших расчетных сетках. Идея создания параллельных программ на мобильных устройствах - показать гидродинамический расчет всегда и везде, при этом за малое количество времени. Эту идею можно развивать до распределения задач на нескольких мобильных устройствах по беспроводной или иной сети. Во время тестирования поставленной задачи, произведены сравнения результатов расчетов последовательной и параллельной программы на мобильном устройстве и персональном компьютере. Результаты тестирования показали, что, для сетки с размером 128x128x128 время вычисления параллельного алгоритма на персональном компьютере равна 21.07 секунд, а на мобильном устройстве 42.845 секунд (Таблица 1).

Table 1: Время расчета (сек.) вычисления давления в модели Баклея-Леверетта на разных сетках и устройствах (размер блока 8x8x8)

Протестированные алгоритмы	16x16x16	32x32x32	64x64x64	128x128x128
Параллельный алгоритм (персональный компьютер)	0,010	0,150	2,229	21,070
Параллельный алгоритм (мобильное устройство)	0,177	0,802	5,625	42,845
Последовательный алгоритм (персональный компьютер)	0,042	1,125	19,198	185,611

Сравнивая соотношение времени расчета на мобильных устройствах и персональном компьютере, при увеличении размера сетки, их соотношение уменьшается (Рисунок 1).

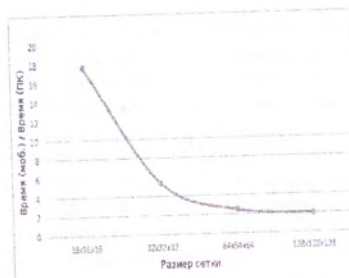


Figure 1: Соотношение времени расчета на мобильном устройстве и персональном компьютере

В статье приведены результаты тестирования параллельного численного решения задачи трехмерного случая модели Баклея-Левретта с использованием технологии CUDA на мобильных устройствах. С помощью этих тестов можно сделать множество выводов. Прежде всего, как уже было сказано ранее, мобильные устройства превращаются в значительные вычислительные мощности. Несмотря на то, что представленная работа направлена на демонстрацию одного отдельного приложения ничего не мешает представить аналогичное приложение, работающее на нескольких устройствах и использующее сеть для обмена данными. Во введении было упомянуто существование подобных систем и представленные результаты позволяют сделать, по крайней мере, первичные выводы об эффективности такой системы. А именно эффективность распараллеливания на GPU по сравнению с одним процессором дает хорошие предпосылки на высокую производительность гетерогенного распараллеливания с использованием графических процессоров мобильных устройств. Здесь под гетерогенным распараллеливанием понимается распараллеливание, использующее принципы и CPU (распределения данных между устройствами) и GPU (обработка данных на одной) распараллеливания [3].

В некоторых случаях увеличение скорости сравнимо с производительностью аналогичной программы на видеокарте персонального компьютера, что, в свою очередь, доказывает, что мобильные технологии в настоящее время являются самостоятельными вычислительными мощностями. Эта работа является первым шагом на пути к реализации функциональных возможностей, которые дадут специалистам возможности решить целый ряд задач в любых условиях на мобильных платформах.

Литература

1. Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. – 2010. – 312 p.
2. Иноземцева Н.Г., Перепёлкин Е.Е., Садовников Б.И. Оптимизация алгоритмов задач математической физики для графических процессоров. – 2012. – 240 с.
3. Wilt N. CUDA Handbook: A Comprehensive Guide to GPU Programming // The Paperback. – 2013. – 528 p.