УДК 538.1+51-72

***© В.А. Гой., А.В. Молочков, А.М. Бегун, 2015***

**СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОДА НА CPU И GPU НА ПРИМЕРЕ АЛГОРИТМА ГЕНЕРАЦИИ КОНФИГУРАЦИЙ В МОДЕЛИ ИЗИНГА**

*Молочков А.В.* - д.ф.-м.н., кафедра “Теоретическая и экспериментальная физика” (ДВФУ), e-mail: molochkov.alexander@gmail.com; *Гой В.А*. - аспирант кафедры “Теоретическая и экспериментальная физика” (ДВФУ), e-mail: vovagoy@gmail.com; *Бегун А.М.* – студент кафедры “Общая физика” (ДВФУ), e-mail: beg.alex93@gmail.com

В последнее время, с приходом технологии CUDA Nvidia, многие вычисления переносятся с центрального процессора (CPU) на графический процессор (GPU). На рынке CPU сейчас доступны образцы с 6-ю ядрами, и возможностью выполнения 12 потоков одновременно, с пиковой производительностью около 100 GFLOPs. Топовые модели видеокарт имеют 1536 потоковых ядер, работающих на частоте 1 GHz с максимальной производительностью АЛУ 3.09 TFLOPs, что соответственно в 30 раз больше, чем производительность у CPU и в 60 раз дешевле при пересчете FLOPs/руб. Однако ускорение в 30 раз нужно еще получить, написав код для GPU. В данной работе проводится исследование алгоритма генерации конфигурации для двумерной модели Изинга. Для генерации конфигурации используется алгоритм Монте Карло.

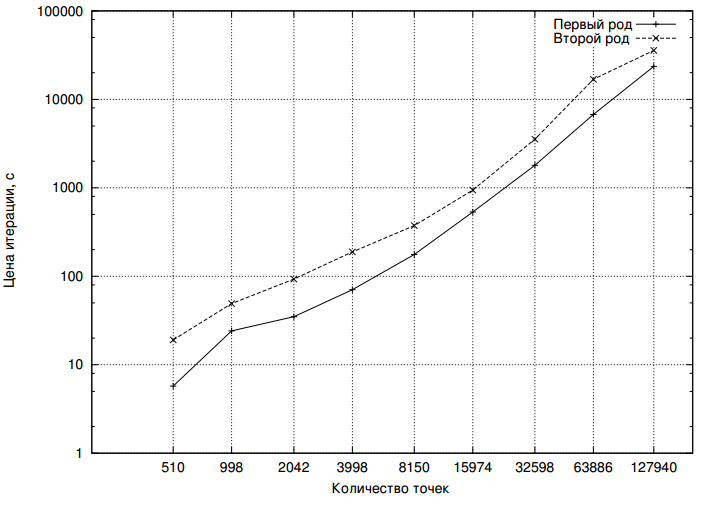
Recently, with the advent of technology CUDA Nvidia, many calculations moved from the CPU (CPU) to the graphics processor (GPU). CPU with 6 cores are now available on the market, and the possibility of 12 threads runned at a time, with a peak performance of about 100 GFLOPs. High-end graphics cards have 1536 stream cores clocked at 1 GHz with a maximum output of ALU 3.09 TFLOPs, which accordingly is 30 times greater than the performance of the CPU and is 60 times more cheaply in terms FLOPs/rub. However, to obtain 30 times the acceleration it requires writing code for GPU. In this paper, we investigate the algorithm for generating a two-dimensional configuration of the Ising model. To generate a configuration was used Monte Carlo algorithm.

*Ключевые слова:* Nvidia CUDA, модель Изинга, расчеты на видеокартах, решеточные расчеты.

Статистическая сумма для двумерной модели Изинга определяется как

,

где  – проекция спина на ось Z в узле решетки с координатами ,  – параметр модели, а сумма берется по всем состояниям спина на решетке. Для данной статистической суммы был написан алгоритм генерации решеточной конфигурации для запуска на CPU и GPU. Было выбрано два разных метода упаковки решеточной информации о спинах в один одномерный массив. Использовалась обычная адресация и шахматная адресация, в которой решетка была поделена на две подрешетки, которые хранились последовательно в одномерном массиве. На подрешетке использовалась обычная адресация.



*Рис. 1* Сравнение числа итераций

При работе с глобальной памятью на GPU важно соблюдать правило коалесинга (coalescing). Основная идея заключается в том, что нити должны обращаться к последовательным ячейкам памяти. Шахматная адресация введена именно для этого. Как видно из таблицы, приведенной ниже, было получено ускорение в 21 раз для решетки  и 1000 проходов алгоритма Монте Карло. Результаты получены с использованием процессора Intel Core i3-3225 3.3GHz и видеокарты ASUS GTX 650 Ti.

Табл. 1.

|  |  |  |
| --- | --- | --- |
|  | Обычная адресация узла решетки () | Шахматная адресация узла решетки |
| Время работы на CPU | 36.40 сек. | 22.85 сек. |
| Время работы на GPU | 1.54 сек. | 1.08 сек. |
| Эффективность кода на GPU | быстрее в14.84 раза | быстрее в 21.16 раза |

**Библиографические ссылки**

1. Costabel M., Dauge M. Weighted regularization of Maxwell equations in polyhedral domains// Numer. Math. 2002. Vol. 93, No 2. P. 239–277.
2. А. Боровский. Qt 4.7+ Практическое программирование на C++. — СПб.: «БХВ-Петербург», 2012. — С. 496. — ISBN 978-5-9775-0757-8