УДК 004.42

ИСПОЛЬЗОВАНИЕ GPGPU ДЛЯ РЕШЕНИЯ ЗАДАЧИ ДИРИХЛЕ ДЛЯ УРАВНЕНИЯ ПУАССОНА

Ефремов А.А.
Научный руководитель – к.ф.-м.н., доцент Карепова Е.Д.
Сибирский федеральный университет

В работе на примере численного решения задачи Дирихле для уравнения Пуассона методом Якоби на графических процессорах общего назначения (GPGPU) рассмотрены основные проблемы распараллеливания достаточно широкого круга задач и описаны возможные пути повышения эффективности использования гибридных параллельных систем. Приведены результаты замеров времени поиска решения при использовании CPU и GPGPU.

GPGPU (англ. General-purpose graphics processing units — «графический процессор общего назначения») — техника использования графического процессора видеокарты для общих вычислений.

В качестве модельной задачи для изучения особенностей программирования численных методов на GPGPU выбрана задача Дирихле для уравнения Пуассона, а в качестве численного метода ее решения – метод Якоби. Выбор модельной задачи, несмотря на кажущуюся ее простоту и неэффективность использованного метода ее решения, продиктован следующими соображениями.

1. Целью исследования является определение «узких мест» при программировании на гибридной архитектуре CPU-GPGPU основных численных алгоритмов, а также выработку стратегии эффективного распараллеливания. В этом контексте, чем проще модельная задача, тем успешнее всестороннее исследование возможностей сложных параллельных вычислительных систем.
2. Метод Якоби обладает хорошим потенциалом для распараллеливания, в частности, широко известны эффективные его параллельные реализации для вычислительных систем с общей (технология OpenMP) и распределенной памятью (технология MPI). Поскольку технология CUDA является технологией для массивно-параллельных вычислений, то простой перенос приемов распараллеливания с предыдущих технологий не представляется возможным и, соответственно, для эффективной реализации метода для GPGPU требуются дополнительные исследования.
3. Основные идеи выбранного *итерационного* алгоритма решения уравнения Пуассона лежат в основе широкого класса эллиптических, а также нестационарных задач (поскольку итерационный процесс может быть интерпретирован, как шаги по времени). Например, рассматриваемый алгоритм является аналогом широкого круга численных методов в части решения проблем, связанных с декомпозицией области, появлением теневых граней, шаблонами доступа к различным видам памяти и т.д.

Исследование проводилось с использованием наиболее популярной технологии в научных вычислениях – Nvidia CUDA.

CUDA (англ. Compute Unified Device Architecture) — программно-аппаратная архитектура, позволяющая производить вычисления с использованием графических процессоров Nvidia, поддерживающих технологию GPGPU.

В работе рассмотрены узкие для параллельной реализации места алгоритма, для них разработаны решения в рамках технологии CUDA.

Поиск максимального элемента в массиве при параллельной обработке. Для вычислении критерия останова итерационного процесса был запрограммирован метод параллельной редукции поиска максимума..

Декомпозиция области. Одной из основных проблем переноса метода Якоби на параллельную архитектуру CUDA является получение подходящего разбиения вычислительной области по блокам нитей. Первый из возможных подходов предполагает 2D-разбиение с теневыми гранями, которые вносят существенную неоднородность в вычислениях в ядре CUDA, а также затрудняет эффективное применение шаблонов доступа к памяти видеокарты (невозможно бесконфликтное использование разделяемой памяти при обработке одного узла сетки). Второй подход – 1D-декомпозиция области, при которой избегается повторное копирование из глобальной памяти на границах подобластей.

Шаблоны доступа к памяти видеокарты. Самым узким местом при создании эффективного параллельного алгоритма в технологии CUDA является работа с различными типами памяти видеокарты. Первый подход для хранения входных данных итерации предполагает использование глобальной (global) памяти. Поскольку глобальная память видеокарты обладает большой латентностью, данный способ приводит к значительному увеличению времени расчетов. Второй подход предполагает использование более быстрой разделяемой (shared) памяти. Однако, без дополнительных усилий при подготовке входных данных итерации, элементы массива ложатся в память таким образом, что бесконфликтный доступ к ним становится невозможным. Для реализации бесконфликтного доступа требуется в четыре раза больше памяти для хранения данных, появляются нежелательные ветвления в ядре CUDA, усложняются расчеты индексов элементов массива.

Использование текстурной памяти. Наиболее перспективным способом доступа к памяти видеокарты при реализации метода Якоби предполагается использование текстурной памяти. Отличительной особенностью текстурной памяти видеокарты является наличие текстурного кэша. Вычислительный шаблон метода Якоби предполагает многократное использование по чтению одних и тех же элементов данных, поэтому кэширование должно значительно уменьшить время расчетов. Однако, текстурную память можно использовать только на чтение, поэтому насчитанные значения придется записывать в глобальную память с большой латентностью доступа.

Численные эксперименты проводились на высокопроизводительном вычислительном сервере ИВМ СО РАН Flagman RX240T8.2 с 8-ю вычислителями Tesla. Ниже приведены их основные характеристики.

- Пиковая производительность операций с плавающей запятой одинарной точности – 8.03 TFlops.
- Общее число потоковых ядер CUDA – 3584.
- Конфигурация сервера:
  - процессоры: 2 шт. 2.93-3.33GHz Intel® Xeon® X5670 Westmere-EP SixCore w/HyperThreading 6.4GT/s QPI, 12MB Smart cache;
  - установленная память – 48GB DDR-III PC3-10600 ECC Registered;
  - установленные GPU: 8 шт. nVidia® Tesla® C2050 PCI-Express ×16 3072 MB GDDR5;
  - характеристика GPU: число ядер CUDA 448, частота работы ядер CUDA 1.15 GHz, производительность операций с плавающей запятой двойной точности (пиковая) 515 ГФлоп, производительность операций с плавающей запятой одинарной точности (пиковая) 1.03 Тфлоп, полный объем специальной памяти 3 ГБ GDDR5, частота памяти 1.5 GHz, интерфейс памяти 384-bit, пропускная способность памяти 144 Гб/с, макс. потребление энергии 238 Вт, системный интерфейс PCIe x16 Gen2, активный вентилятор, двухканальный DVI-I, максимальное разрешение дисплея 2560x1600;
- Операционные системы: Windows Server 2008 R2 7 x64 / Ubuntu Linux (альтернативная загрузка).

- Среды программирования и библиотеки: CUDA, Visual Studio Professional 2010, PGI Accelerator Fortran Workstation Windows, IMSL Fortran Library 64-bit & 32-bit Windows.