

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ ЯДЕР OPENCL С ПОМОЩЬЮ БИБЛИОТЕКИ VEXCL

Д.Е. Демидов

Применение видеокарт для научных расчетов в последнее время приобретает все большую популярность. Основными программными платформами здесь являются NVIDIA CUDA и OpenCL. CUDA — это проприетарная программно-аппаратная архитектура, разрабатываемая компанией NVIDIA и поддерживающая вычисления общего назначения на видеокартах NVIDIA. OpenCL — это открытый стандарт для разработки кросс-платформенных приложений, использующих современные многоядерные процессоры и видеокарты. Обе платформы обладают как достоинствами, так и недостатками. CUDA предлагает более полный набор библиотек для научных расчетов, но завязана на аппаратное обеспечение NVIDIA. OpenCL поддерживает гораздо более широкий набор оборудования, но низкоуровневый программный интерфейс OpenCL требует больших усилий со стороны разработчика.

Еще одно важное отличие CUDA от OpenCL — это принцип работы с вычислительными ядрами. Платформа компании NVIDIA компилирует вычислительные ядра в код PTX вместе с основной программой. PTX — это псевдоассемблер, который во время выполнения преобразуется в машинный код для конкретного устройства, на котором выполняется вычислительное ядро. Так как PTX — это низкоуровневый язык, то окончательный этап компиляции в машинный код занимает незначительное время. В стандарте OpenCL вычислительные ядра компилируются во время выполнения программы из высокоуровневого C-подобного языка. Это ведет к значительному увеличению времени инициализации программ OpenCL. В то же время, данная особенность позволяет генерировать во время выполнения программы вычислительные ядра, оптимизированные под конкретную вычислительную задачу. Эта возможность рассмотрена в данной работе с применением VexCL (<https://github.com/ddemidov/vexcl>) — библиотеки шаблонов векторных выражений с открытым исходным кодом [1]. Библиотека VexCL разработана для упрощения разработки C++ приложений, использующих технологию OpenCL. Библиотека предоставляет удобную и интуитивную нотацию для векторной арифметики, поддерживает одновременное использование нескольких вычислительных устройств.

Рассматривается модельная задача решения системы обыкновенных дифференциальных уравнений (ОДУ). В решении используется Boost.odeint — современная C++ библиотека для численного решения ОДУ [2]. Абстрактный дизайн Boost.odeint позволяет легко использовать существующие CUDA или OpenCL библиотеки для ускорения вычислений. Конкретный пример, рассмотренный здесь, это параметрическое исследование системы Лоренца — системы из трех связанных ОДУ, имеющую хаотическое решение в широком диапазоне параметров [3]. Данная система часто используется в качестве контрольной задачи в нелинейной динамике.

В простейшей реализации задачи в качестве переменных состояния используются вектора VexCL. Поддержка библиотекой VexCL арифметических выражений позволяет прозрачно использовать алгоритмы Boost.odeint для одновременного решения большого числа систем Лоренца с различными параметрами. При использовании ускорителя NVIDIA Tesla C2070 получено примерно десятикратное ускорение по сравнению с CPU Intel Core i7 930. Однако, простейшая реализация имеет определенные недостатки. В частности, схема Рунге-Кутты четвертого порядка выделяет четыре временных вектора, которые располагаются в глобальной памяти ГПУ. Более того, каждая итерация метода Рунге-Кутты ведет к запуску нескольких вычислительных ядер. Показано, что с помощью механизма генерации вычислительных ядер OpenCL, предоставляемым библиотекой VexCL, можно автоматически получить монолитное ядро, выполняющее одну итерацию метода Рунге-Кутты. Это позволяет ускорить решение задачи еще на один порядок.

Таким образом, библиотека VexCL позволяет автоматически преобразовывать существующие шаблоны алгоритмов C++ в эффективные вычислительные ядра OpenCL во время выполнения программы. Существуют очевидные ограничения данного подхода: поддерживаются только задачи с явным параллелизмом (*embarrassingly parallel*), в коде не допускаются ветвления и циклы с переменным числом итераций. Однако, несмотря на ограничения, данная методика может привести к существенной экономии как времени разработчика, так и машинного времени.

ЛИТЕРАТУРА:

1. D. Demidov, K. Ahnert, K. Rupp, P. Gottschling "Programming CUDA and OpenCL: A Case Study Using Modern C++ Libraries" // SIAM Journal of Scientific Computing, accepted for publication.
2. K. Ahnert, M. Mulansky "Odeint - Solving Ordinary Differential Equations in C++" // IP Conf. Proc., vol. 1389, 2011, pp. 1586–1589.
3. E. N. Lorenz, "Deterministic Nonperiodic Flow" // Journal of Atmospheric Sciences, 20 (1963), pp. 130–141.