Высокоуровневые средства программирования для GPU

Ю. Сердюк
Институт программных систем РАН, г.Переславль-Залесский
Yury@serdyuk.botik.ru

Представлен обзор систем программирования для GPU на основе .NET и языка C#: язык программирования MC#, проект Accelerator V2 компании Microsoft и система GPU.NET компании TidePowerd. Рассмотрен синтаксис средств языка MC# для программирования GPU и их реализация. Приведены результаты тестирования приложений, разработанных на языке MC#, на машинах с GPU Nvidia Tesla C1060и Tesla M2050, включая машины с несколькими GPU и кластерные системы.

1. Введение

В настоящий момент, для достижения более высокой производительности вычислительных устройств, всё более часто применяются их гибридные варианты, использующие различного типа ускорители: графические процессоры (GPU), процессоры Cell, программируемые логические матрицы (FPGA).

Использование такого рода гибридных параллельных архитектур дополнительно усложнило процесс разработки приложений для них, поскольку потребовало применения целой группы технологий/библиотек (таких как MPI, Pthreads, CUDA и т.п.) для создания одного приложения. Частично проблему унификации разнообразных программных средств для программирования такого рода архитектур решает библиотека OpenCL, но она достаточно сложна в изучении и использовании и имеет свои ограничения.

В данной работе представлено расширение языка MC# конструкциями и средствами для программирования графических процессоров [1].

2. Средства языка MC# для программирования графических процессоров

Язык программирования MC# является расширением языка C# и основан на платформе .NET. Он предназначен для разработки параллельных, распределенных приложений, предназначенных для исполнения на всех типах параллельных архитектур, включая гибридные системы и кластеры.

Базовая идея адаптации системы программирования MC# к вычислительным системам на базе GPU состоит во введении дополнительного модификатора *gpu*, который должен быть использован в определении метода объектно-ориентированной программы, предназначенного для исполнения на графическом процессоре.

Общая идеология программирования GPU на языке MC# совпадает с базовыми принципами технологии программирования CUDA. В частности, в системе программирования MC#, параметры конфигурации графического процессора (номер устройства, размеры решетки и размеры блока) устанавливаются путем задания значений для объекта GpuConfig, относительно которого *gpu*-методы вызываются как extension-методы.

Составной частью системы программирования MC#, поддерживающей графические процессоры, является библиотека GPU.NET, реализованная на языке C# и включающая в себя JIT-компилятор для GPU и набор функций, соответствующих функциям базовой библиотеки CUDA.

При программировании на языке MC# GPU, программист освобожден от необходимости явно программировать передачу данных из основной памяти в память графического устройства и обратно. Эта задача решается компилятором языка MC#, который генерирует вызовы соответствующих функций библиотеки GPU.NET, реализующих копирование данных. Также, в *gpu*-методах возможно использование

“разделяемой памяти” (shared memory) графического процессора. В языке MC#, массивы, размещаемые в разделяемой памяти, объявляются как параметризованные (generic) массивы.

3. Тестовые результаты

Тестирование программ на языке MC# для GPU проводилось в Межведомственном суперкомпьютерном центре (МСЦ) РАН, г.Москва, на 2-ух узловом кластере под управлением ОС Linux, где каждый узел был оснащен двумя GPU типа Nvidia Tesla C1060, и также на сервере с четырьмя устройствами Nvidia Tesla M2050.

В качестве тестовых приложений использовались программы перемножения матриц, решение задачи Дирихле для уравнения Пуассона, программа N-Queens, вычисляющая количество всех возможных различных расстановок ферзей на шахматной доске размером N x N, где ферзи попарно не атакуют друг друга.

Построены графики зависимости времени исполнения указанных программ как в зависимости от размера исходных данных, так и от количества использованных GPU (при фиксированном размере входных данных). Для задачи перемножения матриц, тестировались варианты как с элементами матриц типа float, так и с элементами типа double. Приведены сравнительные графики времени исполнения одних и тех же задач на GPU и на обычных (многоядерных) процессорах. Показано, что во многих случаях, программы, написанные полностью на MC# и исполняемые на GPU, превосходят по производительности программы, использующие функции библиотеки Intel MKL и исполняемые на узле с обычными многоядерными процессорами (а именно, на машине с двумя 6-ядерными процессорами Intel Xeon X5660).

4. Смежные работы и выводы.

Ключевая идея системы “Accelerator V2 for GPGPU and SIMD x86 multicore target” компании Microsoft состоит во введении специальных объектов − “параллельных массивов” и операций над ними, которые можно использовать в программах на языках C# и F#. Данные операции транслируются в код, исполняемый на графическом устройстве. При данном подходе, программист не имеет возможности написать собственную функцию, использующую произвольные операции и предназначенную для исполнения на GPU. А потому круг задач, решаемых на основе данного подхода, является достаточно узким.

Система GPU.NET компании TidePowerd по своей идеологии очень близка к подходу, принятому в языке MC#. Основное ее отличие от MC# состоит в том, что компоненты программы, относящиеся к выполнению на GPU, помечаются с помощью атрибутов языка C#. В частности, методы, предназначенные для исполнения на GPU, помечаются атрибутом *Kernel*, а массивы, размещаемые в разделяемой памяти, определяются с помощью атрибута *SharedMemory*. К недостаткам текущей реализации системы GPU.NET компании TidePowerd следует отнести невозможность работы с несколькими GPU в одной программе, отсутствие версии для ОС Linux, отсутствие библиотеки базовых математических функций типа sin, cos и др.

В данной работе продемонстрированы возможности использования высокоуровневых средств программирования для современных GPU. Приведены тестовые результаты для приложений, написанных на языке MC#.

Текущую версию системы MC# можно найти на сайте www.mcsharp.net.

Литература.

1. Ю. Сердюк. “Программирование графических процессоров на языке MC#”. - Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: Труды Международной суперкомпьютерной конференции (20-25 сентября 2010 г., г. Новороссийск). - М.: Изд-во МГУ, 2010.