Электронное научное издание «Устойчивое инновационное развитие: проектирование и управление»
www.rypravlenie.ru **том 4 (2009), ст. 2**

УДК 65.012.26

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ CUDA ФИРМЫ NVIDIA ДЛЯ САПР НЕЙРОННЫХ СЕТЕЙ

Денис Владимирович Калитин, кандидат технических наук, доцент, МГГУ

Аннотация

Развитие вычислительной техники во всём мире идёт различными путями. Существующая реальность требует увеличения вычислительной мощности, однако зачастую, повышение производительности на порядок или несколько, требует значительных финансовых вложений. В статье рассматривается новая технология распараллеливания обработки данных, которая может дать существенный выигрыш в скорости работы, но при этом не требует огромных финансовых вложений. САПР нейронных сетей, по сути своей, объект с высокой долей задач, в основе которых лежат параллельные вычисления. Очевидно что, используя рассматриваемую технологию можно на несколько порядков ускорить работу такой САПР.

КЛЮЧЕВЫЕ СЛОВА: нейронные сети, САПР, параллельная обработка, программирование, CUDA, GPGPU.

USING CUDA TECHNOLOGY COMPANY NVIDIA FOR CAD NEURAL NETWORKS

Denis Kalitin. the Candidate of Technical Science, Associate Professor, MSMU

Abstract

The development of computer technology throughout the world are in various ways. The current reality requires an increase in computing power, but often, increasing productivity by an order or more, require significant financial investment. This article describes a new parallelization of data processing, which can give a significant gain in performance, but it does not require huge financial investments. CAD neural networks, in essence, an object with a high proportion of tasks, which are based on parallel computing. It is obvious that, using the technologies can be several orders of magnitude speed up a CAD system.

KEYWORDS: neural networks, CAD, parallel processing, programming, CUDA, GPGPU.

Искусственные нейронные сети — попытка моделирования работы биологического мозга. Как и биологический аналог, нейронные сети состоят из нейронов — кирпичиков вычислительной сети. Главное преимущество нейронных сетей это параллелизм работы. В многослойном персептроне нейроны каждого слоя работают параллельно. За счёт этого достигается значительная скорость обработки данных. Однако большинство нейронных сетей моделируются на компьютерах с фон-неймановской архитектурой. В них, как правило, весь параллелизм сходит на нет. Один процессор, за небольшой промежуток времени, может «обсчитать» выход только одного нейрона. Даже с появлением многоядерных процессоров, построить компьютер, который мог бы реально параллельно просчитывать работу нейронной сети, остаётся слишком сложно и дорого. Однако потребность в нейронных сетях всё возрастает.

Зададимся вопросом, какие же вычисления необходимо произвести для расчёта выхода одного нейрона? Если мы ограничимся простым персептроном, то ответ очевиден. Нам необходимо будет вычислить взвешенную сумму входов нейрона и значение пороговой

функции. Для вычисления взвешенной суммы необходимо произвести число умножений и сложений в соответствии с числом входных каналов. После этого вычислить значение пороговой функции, а это, в зависимости от сложности самой функции, от одной операции сравнения, до нескольких математических операций. Запишем программу вычисления выхода нейронной сети на псевдокоде:

```
For i=0 to количество_слоёв_в_сети
For j=0 to количество_нейронов_в_слое [ i ]
Sum=0
For k=0 to количество_входов_в_нейрон [ j ]
Sum += веса_нейрона_j [ k ] * вход_нейрона_j [ k ]
Next k
Выход_нейрона_j = Function (Sum)
Next j
Next i
```

При этом все операции в цикле по переменной j могут быть выполнены одновременно для разных нейронов находящихся в одном слое. Очевидный параллелизм работы. Какие же вычислительные системы в настоящее время могут реализовать эту параллельность? Это классически многопроцессорные системы и многомашинные вычислительные комплексы. Но как уже было сказано, построение таких вычислительных систем требует немалых финансовых затрат. Другой путь решения состоит в использовании других многопроцессорных систем, которые уже присутствуют на рынке и требуют финансовых вливаний на несколько порядков меньших. Это видеокарты различных производителей, например фирм Nvidia и AMD.

В 2007 году фирма Nvidia представила новую технологию CUDA для своих видеокарт.

CUDA (англ. Compute Unified Device Architecture) – технология GPGPU (англ. General-Purpose computing on Graphics Processing Units), позволяющая программистам реализовывать на языке программирования Си алгоритмы, выполнимые на графических процессорах ускорителей GeForce восьмого поколения и старше, Nvidia Quadro и Tesla компании Nvidia.

GPGPU (англ. General-purpose graphics processing units – «GPU общего назначения») – техника использования графического процессора видеокарты для общих вычислений, которые обычно проводит центральный процессор.

Так же аналогичную технологию предложила фирма AMD для своих видеокард - AMD FireStream.

На сегодняшний день, при небольших финансовых вложениях, мы можем построить вычислительную систему обладающую следующими вычислительными возможностями:

- количество GPU: 4;
- максимальная частота каждого GPU: 1500 MHz;
- количество потоковых процессоров: 960;
- максимальная частота каждого потокового процессора: 1500 MHz;
- размер памяти: 16384 Mb;
- полоса пропускания: 408 GB/s;
- частота памяти: 800 MHz;
- приблизительная производительность: 4000 GigaFLOPS.

Данные показатели приведены при использовании только одной видеокарты Nvidia Tesla S1070. Однако при конфигурировании компьютера, мы можем использовать несколько таких видеокарт, что очевидно повысит общую производительность системы.

Технология CUDA — это среда разработки на Си, которая позволяет программистам и разработчикам писать программное обеспечение для решения сложных вычислительных задач за меньшее время благодаря многоядерной вычислительной мощности графических процессоров. Проще говоря, графическая подсистема компьютера с поддержкой CUDA может быть использована, как вычислительная.

CUDA даёт разработчику возможность по своему усмотрению организовывать доступ к набору инструкций графического ускорителя и управлять его памятью, организовывать на нём сложные параллельные вычисления. Графический ускоритель с поддержкой CUDA становится мощной программируемой открытой архитектурой подобно сегодняшним центральным процессорам.

Всё это предоставляет в распоряжение разработчика низкоуровневый, распределяемый и высокоскоростной доступ к оборудованию, делая CUDA необходимой основой при построении серьёзных высокоуровневых инструментов, таких как компиляторы, отладчики, математические библиотеки, программные платформы.

В технологии CUDA применяется grid-модель памяти, кластерное моделирование потоков и SIMD инструкции.

Первоначальная версия CUDA SDK была представлена 15 февраля 2007 года. В основе CUDA API лежит расширенный язык Си. Для успешной трансляции кода на этом языке, в состав CUDA SDK входит собственный Си-компилятор командной строки nvcc компании Nvidia. Компилятор nvcc создан на основе открытого компилятора Open64 и

предназначен для трансляции host-кода (главного, управляющего кода) и device-кода (аппаратного кода) (файлов с расширением .cu) в объектные файлы, пригодные в процессе сборки конечной программы или библиотеки в любой среде программирования, например в Microsoft Visual Studio.

Рассмотренный выше алгоритм на тестовой системе, без применения технологии CUDA, выполнялся порядка 2 миллисекунд, с применением технологии CUDA 0,002 миллисекунды. В процессе обучения нейронной сети, например, алгоритмом обратного распространения ошибки, будет использован алгоритм такой же структуры. В итоге мы можем получить прирост производительности в 1000 раз. Конечно, этот показатель будет немного ниже при использовании нейронных сетей для реальных задач, особенно тех задач САПР, в которых математический аппарат нейронных сетей не является основным, а лишь вспомогательным. Однако перевод других алгоритмов на параллельную концепцию обработки данных, например обработка мографов, проектирование функций автоматов, построение диаграмм Хассе и др., позволяет значительно ускорить получение проектных решений.

Литература

1. Круглов, В.В., Борисов, В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия-Телеком, 2001. – 382 с.
2. Электронный ресурс NVIDIA CUDA Programming Guide, режим доступа www.nvidia.com, свободный.
3. Электронный ресурс Wikipedia, режим доступа www.wikipedia.com, свободный.