История развития видеокарт

Изначально видеокарта была устройством предназначенным для преобразования графического образа из памяти компьютера в вид, пригодный для отображения на экране монитора. Со временем к преобразованию графического образа на видеокарту была возложена задача обработки и формирования графического образа. Так возник ``графический ускоритель".

В 1981 году был выпущен один из самых ранних графических адаптеров в истории вычислительной техники --- MDA (Monochrome Display Adapter) для компьютеров фирмы IBM PC. Данный адаптер поддерживал только текстовый режим с разрешением 80x25 символов, помимо просто текста поддерживались текстовые атрибуты: обычный, яркий, инверсный, подчёркнутый и мигающий. Какой либо графической или цветовой информации данный адаптер обрабатывать не мог, под цветностью тогда понималось лишь свечение люминофора электронно-лучевой трубки. Последующим развитием адаптера MDA стал видеоадаптер HGC (Hercules Graphics Controller) созданный в 1982 году фирмой Hercules. Адаптер HGC поддерживал графическое разрешение 720x348 точек и две графические страницы. Поддержки цветов всё ещё не было.

Пионером в цветном изображении стала видеокарта CGA (Color Graphics Adapter) от фирмы IBM. В текстовом режиме существовало два разрешения: 40x25символов и 80x25 символов (на каждый символ приходилась матрица 8x8 точек) с 256 символами. На каждое знакоместо приходилось 16 цветов и 16 цветов фона (либо 8 цветов фона и атрибут мигания). В графическом режиме также было два разрешения: 320x200 точек (цветность: четыре палитры по четыре цвета каждая) и 640x200 точек (данный режим был монохромным). Развитием этого адаптера стал адаптер EGA (Enhanced Graphics Adapter) с палитрой в 64 цвета.

В 1987 году IBM создала компонентный видеоинтерфейс VGA (Video Graphics Array). Добавлены: текстовое разрешение 720x400 и графический режим 640x480.

В 1991 году появилось SVGA (Super VGA) --- расширение VGA с более высокими режимами и дополнительными возможностями, например, задание произвольной частоты кадров. Число цветов стало равно 65536 (High Color, 16 bit) и 16777216 (True Color, 24 bit).

Устройство видеокарты

1. Графический процессор

Графический процессор (Graphics processing unit (GPU) — графическое процессорное устройство) занимается расчётами выводимого изображения, освобождая от этой обязанности центральный процессор, производит расчёты для обработки команд трёхмерной графики. Является основой графической платы, именно от него зависят быстродействие и возможности всего устройства. Современные графические процессоры по сложности мало чем уступают центральному процессору компьютера, и зачастую превосходят его как по числу транзисторов, так и по вычислительной мощности, благодаря большому числу универсальных вычислительных блоков.

2. Видеоконтроллер

Видеоконтроллер отвечает за формирование изображения в видеопамяти и осуществляет обработку запросов центрального процессора. Современные графические адаптеры (AMD,

Nvidia) обычно имеют не менее двух видеоконтроллеров, работающих независимо друг от друга и управляющих одновременно одним или несколькими дисплеями каждый.

3. Видео-ПЗУ

Видео-ПЗУ (Video ROM) — постоянное запоминающее устройство (ПЗУ), в которое записаны BIOS видеокарты, экранные шрифты, служебные таблицы и т. п. ПЗУ не используется видеоконтроллером напрямую — к нему обращается только центральный процессор.

4. Видео-ОЗУ

Видеопамять выполняет функцию кадрового буфера, в котором хранится изображение, генерируемое и постоянно изменяемое графическим процессором и выводимое на экран монитора (или нескольких мониторов). В видеопамяти хранятся также промежуточные невидимые на экране элементы изображения и другие данные. Видеопамять бывает нескольких типов, различающихся по скорости доступа и рабочей частоте.

Разница между CPU и GPU в параллельных расчётах

В видеочипах Nvidia основной блок — это мультипроцессор с восемью-десятью ядрами и сотнями ALU в целом, несколькими тысячами регистров и небольшим количеством разделяемой общей памяти. Кроме того, видеокарта содержит быструю глобальную память с доступом к ней всех мультипроцессоров, локальную память в каждом мультипроцессоре, а также специальную память для констант.

Самое главное — эти несколько ядер мультипроцессора в GPU являются SIMD (одиночный поток команд, множество потоков данных) ядрами. И эти ядра исполняют одни и те же инструкции одновременно, такой стиль программирования является обычным для графических алгоритмов и многих научных задач, но требует специфического программирования. Зато такой подход позволяет увеличить количество исполнительных блоков за счёт их упрощения.

CPU созданы для исполнения одного потока последовательных инструкций с максимальной производительностью, а GPU проектируются для быстрого исполнения большого числа параллельно выполняемых потоков инструкций. Универсальные процессоры оптимизированы для достижения высокой производительности единственного потока команд, обрабатывающего и целые числа и числа с плавающей точкой. При этом доступ к памяти случайный.

У видеочипов работа простая и распараллеленная изначально. Видеочип принимает на входе группу полигонов, проводит все необходимые операции, и на выходе выдаёт пиксели. Обработка полигонов и пикселей независима, их можно обрабатывать параллельно, отдельно друг от друга. Поэтому, из-за изначально параллельной организации работы в GPU используется большое количество исполнительных блоков, которые легко загрузить, в отличие от последовательного потока инструкций для CPU. Кроме того, современные GPU также могут исполнять больше одной инструкции за такт (dual issue).

Как Nvidia, так и ATI поддерживают реализацию быстрого вычисления основных математических функций за один такт. К основным математическим функциям относятся: квадратный корень, экспонента, логарифм, синус, косинус и ряд других функций.

На видеокартах применяется более быстрая память, и в результате видеочипам доступна в разы большая пропускная способность памяти, что также весьма важно для параллельных расчётов, оперирующих с огромными потоками данных.

Возможность работы с тысячами потоков накладывает свои ограничения: из-за большого количества потоков становится невозможным дать ядрам большую локальную память, ведь в этом случае вся временная память видеокарты будет уходить на обслуживание ядер. Поэтому у ядер маленькое стековое пространство, а следовательно, функции, исполняемые ядрами, не могут использовать рекурсию. Эмуляция стека за счёт памяти видеокарты возможна, но ограничена небольшим количеством итераций и является нетипичной для GPU.

Вкратце можно сказать, что в отличие от современных универсальных CPU, видеочипы предназначены для параллельных вычислений с большим количеством арифметических операций. И значительно большее число транзисторов GPU работает по прямому назначению — обработке массивов данных, а не управляет исполнением (flow control) немногочисленных последовательных вычислительных потоков.

Параллельные вычисления на GPU начали активно развиваться с появлением шейдеров --- специальных программ предназначенных для работы на GPU. Тогда же появился компилятор языка Brook --- BrookGPU. BrookGPU облегчал программистам работу с шейдерами. Компилятор обрабатывал файлы с расширением .br и C++ программами давая на выходе скомпилированную программу работающую через DirectX или OpenGL.

Компании Nvidia и ATI увидели возможный потенциал BrookGPU и начали разрабатывать свои аналогичные проекты. Таким образом у Nvidia появился проект CUDA (Compute Unified Device Architecture), а у ATI --- CTM (Close-to-the-Metal) который был началом для AMD FireStream.

Nvidia CUDA

В основе программного интерфейса CUDA лежит расширенный язык Си. Для трансляции текста программы в исполняемые файлы используется компилятор nvcc, созданный на основе открытого компилятора Open64.

Обычная процедура работы с GPU выглядит следующим образом: блок геометрии вычисляет треугольники, блок растеризации вычисляет пиксели которые в дальнейшем будут отображены на экране

Поэтому использование GPGPU являлось достаточно трудоёмким процессом. Ранние методы работы с графическим процессором были нетривиальными приёмами вследствии чего были крайне неудобными. Данные представлялись изображениями (текстурами), а алгоритмы --- процессами растеризации.

CUDA представляла ряд удобств, вместо непосредственной работы с GPU:

- интерфейс программирования приложений CUDA основан на стандартном языке программирования Си с расширениями, что упрощает процесс изучения и внедрения архитектуры CUDA;
- более эффективная передача данных между системной и видеопамятью;
- отсутствие необходимости в графических API с избыточностью и накладными расходами;
- линейная адресация памяти, возможность записи по произвольным адресам;
- аппаратная поддержка целочисленных и битовых операций.

Основные недостатки CUDA:

- отсутствие поддержки рекурсии для выполняемых функций;
- минимальная ширина блока в 32 потока;
- закрытая архитектура CUDA, принадлежащая Nvidia.

AMD FireStream

Хотя цели видеокарт что у ATI, что у Nvidia одни и те же, но подходы к внутренней архитектуре всё же различаются. В первую очередь это касается основной вычислительной единицы: у Nvidia блок вычисления называется ``warp'' и состоит из 32-х нитей, у ATI блок называется ``wave front'' и состоит из 64-х нитей. Но данное различие не принципиально, практически любую программу для вычисления на GPU можно переписать для использования другого количества нитей.

Более важным отличием AMD является применение технологии ``VLIW'' --- Very Long Instruction Word. В графических процессорах Nvidia используются простые скалярные инструкции для работы со скалярными регистрами. В архитектуре ATI задействованы 128 битные векторные регистры.

Производительность операций на видеокартах ATI при работе над числами одинарной точности достигает нескольких терафлопов благодаря векторным инструкциям.

Ещё одним отличием подхода ATI от Nvidia является использование особого формата расположения инструкций в двоичной коде программы. У ATI инструкции расположены не традиционно (по тексту исходного кода программы), а секционно.

Прежде всего идёт секция с набором инструкций условных переходов, в них содержатся ссылки на секции арифметических инструкций не содержащих переходов. В секциях с арифметическими операциями (VLIW bundles --- связки VLIW-инструкций) содержатся только арифметические инструкции над данными из регистров и/или локальной памяти. Таким образом становится проще управлять потоком инструкций и доставлять их к устройствам-исполнителям. Кроме того имеются секции для инструкций обращения к памяти.

Применение параллельных вычислений

Медицина
-В Австралийском национальном университете проводятся исследования развития болезни Паркинсона с помощью методов машинного обучения.
-В стартапе Zebra Medical Vision накопленные данные в клинических исследованиях используются для оценки рисков развития болезней, их предупреждения и помощи в организации и проведении профилактических лечений.
-В нью-йоркской Школе медицины Икана при больнице Маунт-Синай глубокое обучение используется для анализа медицинских карт и определения пациентов, с высоким риском заболевания опасными болезнями в течении года.

Энергетика

-Компания-стартап PowerScout из Калифорнии применяет GPU для прогнозирования, какие домохозяйства могут с большой долей вероятности приобрести солнечные панели.

Разработки PowerScout также позволяют определить объём энергии, который можно получить с крыши одного дома. При этом необязательно самостоятельно проводить подсчёты. Необходимые данные извлекаются из коммерческих баз, спутниковых снимков. Также учитываются возможные затенения, например деревья рядом с домами, отбрасывающие тень на крыши.

Астрономия
В Университетском колледже Лондона вычисления на GPU используются для определения планет, на которых возможно поддерживать жизнь. Название программы --- RobERt (Robotic Exoplanet Recognition, ``роботизированное распознавание экзопланет").

Агрономия
Немецкая компания PEAT применяет глубокое обучение для создания инструмента для диагностики и лечения болезней растений. Пользователь фотографирует больные растения и загружает полученные изображения в программу PEAT ``Plantix", после чего получает рекомендации по лечению.

Майнинг
Завод находится на юго-востоке Москвы, на Волгоградском проспекте. Он объединяет фермы мощностью 20 МВт, которые принадлежат клиентам компании Мариничева Дмитрия Николаевича — они фактически арендуют пространство и электричество под майнинг.