Федеральное государственное бюджетное образовательное учреждение высшего образования Московский технологический университет

Институт комплексной безопасности и специального приборостроения

Кафедра: КБ-5 «Аппаратного, программного и математического обеспечения вычислительных систем»

Реферат по дисциплине «Компьютерная графика»

Архитектура параллельных вычислений на GPU (Nvidia Cuda, AMD FireStream)

Студент(ка): Аметов И.И.
Группа: ТМБО-01-15, 3 курс
Форма обучения: очная

Доля авторского текста (оригинальности)
в результате автоматизированной проверки составила ____ %
Работа защищена на оценку _____ «__» _____ 2017 г.

Преподаватель: Ладынин А.И.

Москва, 2017

Содержание

Введение

Целью данной курсовой работы является получение знаний в области компьютерной графики, изучение основных принципов и приёмов построения современной графики.

В качестве темы курсовой работы была выбрана архитектура параллельных вычислений на GPU (англ. *graphics processing unit, GPU*). В качестве примера в данной работе рассматриваются две наиболее распространённые технологии: NVIDIA CUDA и AMD FireStream.

Компьютерная техника в целом и компьютерная графика в частности продолжают быстро и бурно развиваться. Практически каждый год возникают новые и отмирают старые технологии. Если раньше компьютеры в основном могли выполнять вычисления лишь в однопоточном режиме, то теперь с появлением архитектур и машин с поддержкой многопоточных вычислений возникают новые горизонты возможностей в плане быстродействия и решения новых задач. Одной из таких возможностей стала реализация эффективных многопоточных вычислений на видеокартах. Изначально параллельные вычисления в видеокартах предназначались для обсчёта высококачественной графики. Но со временем исследователи обнаружили что с помощью видеокарт можно не только обрабатывать графику, но и решать задачи в таких областях как флуоресцентная микроскопия, молекулярная динамика, финансовая аналитика, ультразвуковые исследования и диагностика рака и многие другие.

1. 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 современные видеокарты встроен графический процессор, который может выполнять дополнительную обработку данных снимая тем самым нагрузку на центральный процессор. В наши дни все современные видеокарты Nvidia и AMD (Ati) выполняют обработку графических данных на аппаратном уровне.

Рис. 1: “Зелёный” монохромный монитор, используемый с видеоадаптером MDA

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 80×25 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 720×348 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: 40×25 символов и 80×25 символов (на каждый символ приходилась матрица 8×8 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: 320×200 точек (цветность: четыре палитры по четыре цвета каждая) и 640×200 точек (данный режим был монохромным). Развитием этого адаптера стал адаптер EGA (Enhanced Graphics Adapter) с палитрой в 64 цвета. Разрешение было увеличено до 640×350. Для режима 80×25 использовалась большая матрица — 8×14, одновременно можно было использовать 16 цветов, цветовая палитра была расширена до 64 цветов. Графический режим также позволял использовать при разрешении 640×350 16 цветов из палитры в 64 цвета.

В ранних моделях компьютеров от IBM PS/2 использован новый графический адаптер MCGA (Multicolor Graphics Adapter). Текстовое разрешение было поднято до 640×400, что позволило использовать режим 80×50 при матрице 8×8 точек, а для режима 80×25 использовать матрицу 8×16. Количество цветов увеличено до 262144 (64 уровня яркости по каждому цвету).

В 1987 году IBM создала компонентный видеоинтерфейс VGA (Video Graphics Array),

улучшение графического адаптера MCGA. Добавлены: текстовое разрешение 720×400 для эмуляции MDA и графический режим 640×480.

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).

Рис. 2: Видеокарта NVIDIA GTX 550 TI. 2011 год.

1.1. Устройство видеокарты

1.1.1. 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.

Рис. 3: Графический процессор GeForce 6600GT (NV43)

1.1.2. 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.

1.1.3. 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.

1.1.4. 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.

При написании истории использовался материал [1].

2. Вычисления на графических процессорах

В немалой степени развитию параллельных вычислений на графических процессорах способствовали компьютерные игры. Устройства для параллельных векторных вычислений, которые часто применяются в 3D графике, достигают очень высокой производительности, недоступной для универсальных процессоров. Вслед за первыми видеокартами с поддержкой параллельных вычислений возникли технологии неграфических расчётов общего назначения GPGPU (General Purpose computation on GPUs). Современные видеочипы содержат сотни математических исполнительных блоков, и эта мощь может использоваться для значительного ускорения множества вычислительно-интенсивных приложений. Разработчики задумали сделать так, чтобы GPU рассчитывали не только изображение в 3D приложениях, но и применялись в других параллельных расчётах.

В дальнейшем, два основных производителя видеочипов, Nvidia и AMD, разработали и анонсировали соответствующие платформы под названием CUDA (Compute Unified Device Architecture) и AMD FireStream соответственно. В отличие от предыдущих моделей программирования GPU, эти были выполнены с учётом прямого доступа к аппаратным возможностям видеокарт. Платформы не совместимы между собой. Зато обе платформы ликвидировали некоторые из важных ограничений предыдущих моделей GPGPU, использующих традиционный графический конвейер и соответствующие Direct3D или OpenGL интерфейсы.

2.1. Разница между CPU и GPU в параллельных расчётах

Быстрый рост повышения частоты и производительности универсальных процессоров упёрся в физические ограничения и высокое энергопотребление, и увеличение их производительности всё чаще происходит за счёт размещения нескольких ядер в одном чипе. В универсальных процессорах каждое ядро работает отдельно от остальных, исполняя различные инструкции для различных процессов.

В видеочипах Nvidia основной блок — это мультипроцессор с восемью-десятью ядрами и сотнями ALU в целом, несколькими тысячами регистров и небольшим количе-

ством 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.

Подытожим основные различия между строением CPU и GPU. 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.

Есть различия в работе с памятью у GPU и CPU. Так, не все центральные процессоры имеют встроенные контроллеры памяти, а у всех GPU обычно есть по несколько контроллеров, вплоть до восьми 64-битных каналов в чипе Nvidia GT200. Кроме того, 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.

В универсальных процессорах большие количества транзисторов и площадь чипа идут на буферы команд, аппаратное предсказание ветвления и огромные объёмы начиповой кэш-памяти. Все эти аппаратные блоки нужны для ускорения исполнения немногочисленных потоков команд. Видеочипы тратят транзисторы на массивы исполнительных блоков, управляющие потоками блоки, разделяемую память небольшого объёма и контроллеры памяти на несколько каналов. Вышеперечисленное не ускоряет выполнение отдельных потоков, оно позволяет чипу обрабатывать нескольких тысяч потоков, одновременно исполняющихся чипом и требующих высокой пропускной способности памяти. 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.

Есть множество различий и в поддержке многопоточности. CPU исполняет 1-2 потока вычислений на одно процессорное ядро, а видеочипы могут поддерживать до 1024 потоков на каждый мультипроцессор, которых в чипе несколько штук. И если переключение с одного потока на другой для CPU стоит сотни тактов, то GPU переключает несколько потоков за один такт.

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.

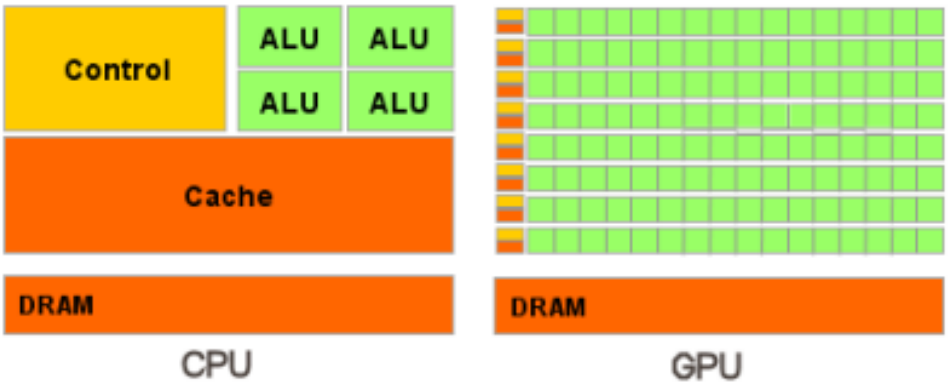

Рис. 4: Различие в устройстве CPU и GPU. DRAM — Оперативная память, Cache — память процессора для часто используемых операций, Control — управляющий блок процессора, ALU — Арифметико-логическое устройство.

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.

2.2. 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:

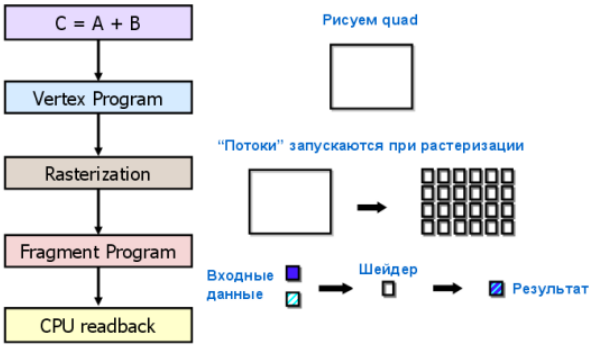

Рис. 5: Обычная процедура обработки фигуры в видеокарте

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, и gather и scatter, 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.

При написании использовался материал из [2].

2.3. 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. Конечно, может наблюдаться либо падение, либо увеличение производительности, но программа всё же будет работать.

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. Условно назовём компоненты регистра A как a_1, a_2, a_3 и a_4; а у регистра B как b_1, b_2, b_3 и b_4. Тогда за один такт оказывается возможным вычислить число $a_1 \times b_1 + a_2 \times b_2 + a_3 \times b_3 + a_4 \times b_4$ или двумерный вектор $(a_1 \times b_1 + a_2 \times b_2, a_3 \times b_3 + a_4 \times b_4)$.

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

Также один векторный регистр можно использовать для хранения одного числа двойной точности — double. В этом случае можно сложить два double числа или умножить два числа, или умножить два числа и сложить с третьим за одну инструкцию. Отсюда при работе над double скорость падает в пять раз чем по сравнению с float.

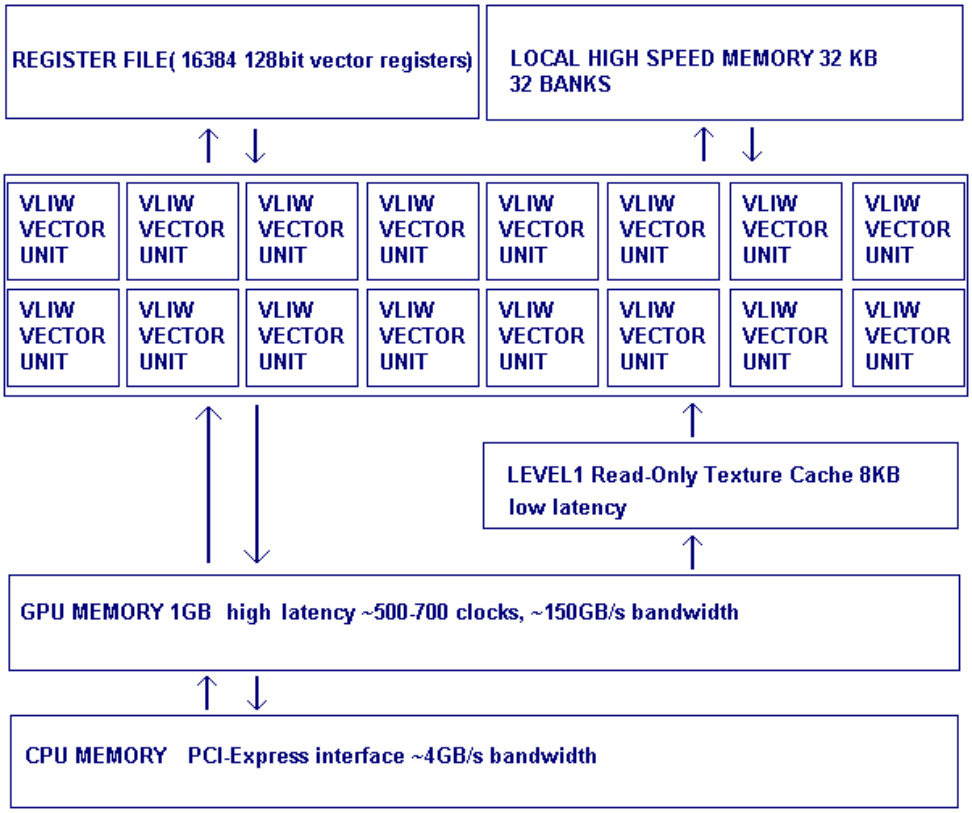

Рис. 6: Условная схема работы видеокарт ATI. На рисунке только один минипроцессор из нескольких параллельно работающих.

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 со-

Секции инструкций условных переходов				
Секция 0	Ветвление 0	Ссылка на секцию 3 непрерывных арифметических инструкций		
Секция 1	Ветвление 1	Ссылка на секцию 4		
Секция 2	Ветвление 2	Ссылка на секцию 5		
Секции непрерывных арифметических инструкций				
Секция 3	VLIW 0	VLIW 1	VLIW 2	VLIW 3
Секция 4	VLIW 4	VLIW 5		
Секция 5	VLIW 6	VLIW 7	VLIW 8	VLIW 9

Таблица 1: Схема разбивки программы в архитектуре видеокарт ATI

держатся 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.

Также существует различие в кэше L1-L2 для видеокарт Nvidia и ATI. Размеры кэша в среднем одинаковы, но существенно различаются время доступа к данным. Задержка доступа к данным у Nvidia несколько больше, и текстурные кэши позволяют сократить загрузку шины памяти, но не ускоряют доступ. У ATI задержка текстурного кэша меньше, но выше задержка локальной памяти минипроцессоров. Для быстрого перемножения матриц у Nvidia лучше использовать локальную память и загружать матрицу поблочно, в случае с AMD лучше использовать быстрый текстурный кэш и читать элементы матрицы по мере необходимости.

Для тех видов задач, которые подходят под идеологию GPU, в частности задачи векторной природы (например, умножение матриц), видеокарты AMD показывают близкую к теоретической производительность. Особенно на задачах с одинарной точностью.

Поэтому архитектура AMD более подходит для научных, профессиональных задач. Для полностью векторных задач, например в задаче параллельного подбора ключей видеокарты AMD в несколько раз опережают Nvidia и в несколько десятков раз быстрее CPU.

В целом идея AMD в том, что графический процессор должен дополнять центральный процессор компьютера выступая в роли своеобразного параллельного сопроцессора для векторных задач.

При написании раздела использовался материал из [3].

3. Краткое сравнение видеокарт AMD и Nvidia

В этом сравнении будут представлены некоторые ключевые особенности графических процессоро AMD и Nvidia. Следует, однако, учитывать что это сравнение носит примерный характер. Между фирмами AMD и Nvidia всегда идёт борьба за рынок поэтому нет ничего удивительного что внутренняя архитектура может меняться от модели к модели.

AMD	Nvidia
Технология FireStream	CUDA
64 нити в блоке	32 нити в блоке
Использование секций с данными	Традиционное построение программ
Меньше возможности для рекурсии	Больше возможностей для рекурсии
Теоретически более быстрая производительность	Меньшая производительность

Таблица 2: Краткое сравнение графических процессоров AMD и Nvidia

При написании использовался материал из [2] и [3].

4. Применение параллельных вычислений

Сегодня параллельные вычисления с применением графических процессоров используются во многих сферах начиная с медицинской науки и заканичивая, но не ограничиваясь прогнозированием погоды. Ниже представлены некоторые области применения графических ускорителей.

- Медицина
  - В Австралийском национальном университете проводятся исследования развития болезни Паркинсона с помощью методов машинного обучения[4].
  - 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[4].
  - 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[5].
- 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[4].
- 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”)[5].

- 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[6].

- Химия

  - Abalone (`http://www.biomolecular-modeling.com/Abalone/`) — программа молекулярного моделирования, предназначенная для изучения молекулярной динамики биополимеров[7].
  - CP2K (`https://www.cp2k.org/`) — программа для атомарной и молекулярной симуляции твёрдых тел, жидкостей, молекулярных и биологических систем[7].

5. Литературные источники

Список литературы

[1] История развития видеокарт для настольных ПК. Часть 1: Эволюция двухмерной графики. [Электронный ресурс] Дата публикации: 18.03.2014. URL: http://www.compbegin.ru/articles/view/_120 (дата обращения: 13.11.2017)

[2] Алексей Берилло. Nvidia CUDA — неграфические вычисления на графических процессорах. [Электронный ресурс] Дата публикации: 23.09.2008. URL: http://www.ixbt.com/video3/cuda-1.shtml (дата обращения: 13.11.2017)

[3] Лев Дымченко. Вычисления на GPU. Особенности архитектуры AMD/ATI Radeon. [Электронный ресурс] Дата публикации: 20.09.2010. URL: http://www.ixbt.com/video3/rad.shtml

[4] CUDA® АЛЬМАНАХ ОКТЯБРЬ 2016 [Электронный ресурс]. URL: http://www.nvidia.ru/content/EMEAI/images/tesla/almanac/CUDA_almanac_October16.pdf (дата обращения: 12.11.2017)

[5] CUDA® АЛЬМАНАХ СЕНТЯБРЬ 2016 [Электронный ресурс]. URL: http://www.nvidia.ru/content/EMEAI/images/tesla/almanac/CUDA_almanac_September16.pdf (дата обращения: 12.11.2017)

[6] CUDA® АЛЬМАНАХ АВГУСТ 2016 [Электронный ресурс]. URL: http://www.nvidia.ru/content/EMEAI/images/tesla/almanac/CUDA_almanac_August16.pdf (дата обращения: 12.11.2017)

[7] Macs in Chemistry [Электронный ресурс]. Cambridge MedChem Consulting. 2011. URL: https://www.macinchem.org/applications/gpuScience.php (дата обращения: 12.11.2017)