МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА»

КАФЕДРА МНОГОПРОЦЕССОРНЫХ СИСТЕМ И СЕТЕЙ

Реферат

СОВРЕМЕННЫЕ ГРАФИЧЕСКИЕ ПРОЦЕССОРЫ

Петрова Андрея Александрович
студента 2 курса, группа 14
специальность «Прикладная информатика»

Научный руководитель:
доктор технических наук,
профессор М.К. Буза

Минск, 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Видеокарта — электронное устройство, преобразующее графический образ, хранящийся, как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора. Первые мониторы, построенные на электронно-лучевых трубках, работали по телевизионному принципу сканирования экрана электронным лучом, и для отображения требовался видеосигнал, генерируемый видеокартой.

В настоящее время эта базовая функция утратила основное значение, и, в первую очередь, под графическим адаптером понимают *устройство с графическим процессором* — графический ускоритель, который и занимается формированием самого графического образа. Современные видеокарты не ограничиваются простым выводом изображения, они имеют встроенный графический процессор, который может производить дополнительную обработку, снимая эту задачу с центрального процессора компьютера. Например, все современные видеокарты Nvidia и AMD (ATi) осуществляют рендеринг графического конвейера OpenGL и DirectX на аппаратном уровне. В последнее время также имеет место тенденция использовать вычислительные возможности графического процессора для решения неграфических задач.

Графический процессор — отдельное устройство персонального компьютера или игровой приставки, выполняющее графический рендеринг. Современные графические процессоры очень эффективно обрабатывают и отображают компьютерную графику. Благодаря специализированной конвейерной архитектуре они намного эффективнее в обработке графической информации, чем типичный центральный процессор. Графический процессор в современных видеоадаптерах применяется в качестве ускорителя трёхмерной графики.

ГЛАВА 1
ФУНКЦИИ И НАЗНАЧЕНИЕ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

Графические процессоры (graphics processing unit, GPU) — яркий пример того, как технология, спроектированная для задач графической обработки, распространилась на несвязанную область высокопроизводительных вычислений. Современные GPU являются сердцем множества сложнейших проектов в сфере машинного обучения и анализа данных.

Графические процессоры за последние двадцать лет сильно изменились. Помимо колоссального прироста производительности, произошло разделение устройств по типу использования. Так, в отдельное направление выделяются видеокарты для домашних игровых систем и установок виртуальной реальности. Появляются мощные узкоспециализированные устройства: для серверных систем одним из ведущих ускорителей является Nvidia Tesla P100, разработанный именно для промышленного использования в дата-центрах.

Graphics Processing Unit (далее — GPU) — это графический процессор, широко используемый в настольных и серверных системах, отличительной особенностью которого является ориентированность на массовые параллельные вычисления. В отличие от графических процессоров архитектура вычислительного модуля Central Processor Unit (далее — CPU) предназначена для последовательной обработки данных. Если количество ядер в обычном CPU измеряется десятками, то в GPU их счет идет на тысячи, что накладывает ограничения на типы выполняемых команд, однако обеспечивает высокую вычислительную производительность в задачах, включающих параллелизм [11].

Развитие графических процессоров на ранних этапах было тесно связано с нарастающей потребностью в отдельном вычислительном устройстве для обработки двух и трехмерной графики. До появления отдельных схем видеоконтроллеров в 70-х годах вывод изображения осуществлялся через использование дискретной логики, что сказывалось на увеличенном энергопотреблении и больших размерах печатных плат. Специализированные микросхемы позволили выделить разработку устройств, предназначенных для работы с графикой, в отдельное направление.

Изначально графические процессорные устройства не предназначались для высокопроизводительных вычислений. Задачей GPU первого поколения, получивших благодаря компьютерным играм широкое распространение в середине 90-х годов, было построение в реальном времени изображений по описанию трехмерных сцен. Для этого требовалась быстрая однородная обработка большого количества элементов (вершины, треугольники, пикселы),

которая достигалась, прежде всего, за счет аппаратной реализации основных алгоритмов.

Первые возможности программирования появились в 2001 году в GPU второго поколения. Программы, выполняемые на GPU, стали называться **«шейдерами»** (от английского shade — «закрашивать»), и основным их назначением было вычисление цвета пиксела на экране. Тогда же стали предприниматься первые попытки использования GPU для общих вычислений, однако из-за низкой точности вычислений, составлявшей лишь 16 бит в формате с фиксированной запятой, и ограниченных возможностей программирования — длина шейдера не превышала 20 команд — это не получило распространения. Ситуация изменилась с появлением в 2003 году серии NVIDIA GeForce FX, предоставившей поддержку полноценной 32-разрядной вещественной арифметики. Тогда же был предложен термин «общие вычисления на GPU» (General Purpose GPU, GPGPU). Графические процессоры начали активно применяться для решения задач математической физики, обработки изображений и матричных вычислений [3].

GPU первого поколения — GeForce 6, GeForce 7, ATI Radeon X1K и более поздние — еще больше расширили возможности программирования за счет введения дополнительных команд управления, в частности поддержки полноценных циклов.

В 2006 году NVIDIA объявила о выпуске линейки продуктов GeForce 8 series, которая положила начало новому классу устройств, предназначенных для общих вычислений на графических процессорах. В ходе разработки NVIDIA пришла к пониманию, что большее число ядер, работающих на меньшей частоте, более эффективны для параллельных нагрузок, чем малое число более производительных ядер. Видеопроцессоры нового поколения обеспечили поддержку параллельных вычислений не только для обработки видеопотоков, но также для проблем, связанных с машинным обучением, линейной алгеброй, статистикой и другими научными или коммерческими задачами.

Все это время основным средством программирования GPU оставались интерфейсы трехмерной графики DirectX и OpenGL, а также шейдерные языки HLSL, GLSL и Cg. Последние представляют собой модифицированный язык Си, из которого выброшены массивы и указатели, добавлены специфические для GPU типы данных и встроенные переменные, а также введены некоторые ограничения, в частности запрет на рекурсию. И хотя сам шейдер пишется на них довольно просто, программа для GPU состоит не только из него. Для решения на GPU реальной задачи обрабатываемые данные необходимо представить в виде двухмерных массивов — текстур, загрузить и скомпилировать сам шейдер, установить его параметры и выходные буферы, что делается на основном процессоре, и только после этого исполнить шейдер, «нарисовав квадрат» по

размеру выходного буфера. Подобные действия не вызывают вопросов у программиста трехмерной графики, но совершенно непривычны для прикладного программиста, что и сдерживало развитие GPGPU. Требовались средства программирования более высокого уровня, и, конечно же, они были созданы, но прежде посмотрим на скрытые в архитектуре возможности современных GPU, позволяющие говорить о сходстве графических процессоров и суперкомпьютеров.

ГЛАВА 2
АРХИТЕКТУРА ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

Конструктивно графический процессор представляет собой отдельную вычислительную микросхему, работающую параллельно с центральным процессором. На персональных компьютерах GPU обычно входят в состав графических ускорителей (или видеокарт) – дополнительных устройств, изначально предназначенных для визуализации двух- и трехмерной графики. Графические ускорители, помимо GPU, включают в себя микросхемы видеопамяти (см. ниже) и собственную систему охлаждения. Они подключаются к системной плате через шины данных PCI-Express либо AGP, которые обеспечивают центральному процессору доступ к видеопамяти и GPU.

Современные GPU, например третьего поколения, такие как NVIDIA GeForce 8, NVIDIA GeForce 9, AMD HD 2K и AMD HD 3K, содержат набор одинаковых вычислительных устройств — «потоковых процессоров» (далее — ПП), работающих с общей памятью GPU (видеопамять). Число ПП меняется от четырех до 128, размер видеопамяти может достигать 1 Гбайт. Все ПП синхронно исполняют один и тот же шейдер, что позволяет отнести GPU к классу SIMD (Single Instruction, Multiple Data). За один проход, являющийся этапом вычислений на GPU, шейдер исполняется для всех точек двухмерного массива. Система команд ПП включает в себя арифметические команды для вещественных и целочисленных вычислений с 32-разрядной точностью, команды управления (ветвления и циклы), а также команды обращения к памяти. GPU выполняют операции только с данными на регистрах, число которых может достигать 128. Из-за высоких задержек (до 500 тактов) команды доступа к оперативной памяти выполняются асинхронно. С целью скрытия задержек в очереди выполнения GPU может одновременно находиться до 512 потоков, и, если текущий поток блокируется по доступу к памяти, на исполнение ставится следующий. Поскольку контекст потока полностью хранится на регистрах GPU, переключение осуществляется за один такт — за эту операцию отвечает диспетчер потоков, который не является программируемым [3].

Тактовые частоты GPU ниже, чем у обычных процессоров, и лежат в диапазоне от 0,5 до 1,5 ГГц, однако благодаря большому количеству потоковых процессоров производительность GPU весьма значительна. Современные GPU верхнего ценового сегмента имеют пиковую производительность 200-500 GFLOPS, что в сочетании с возможностью установки в одну машину двух графических карт позволяет получить пиковую производительность в 1 TFLOPS на одном персональном компьютере. Более того, на некоторых реальных задачах достигается до 70% пиковой производительности. Одновременно с этим, в

сравнении с классическими кластерными системами, GPU обладают значительно лучшими характеристиками как по цене (менее 1 долл. на GFLOPS), так и по энергопотреблению (менее 1 Вт на GFLOPS).

2.1 Графический процессор NVIDIA GeForce 8800

Одно из центральных мест в архитектуре графического процессора NVIDIA GeForce 8800 занимает унифицированный процессор, что позволяет избежать главного недостатка классической архитектуры — невозможности достижения сбалансированной нагрузки вершинных и пиксельных шейдеров.

Перед рассмотрением архитектуры ядра графического процессора разберемся, в чем заключается сущность унифицированного процессора.

Предположим, что в воображаемом графическом процессоре с классической архитектурой присутствуют четыре вершинных и восемь пиксельных процессоров (рисунок 1.1). Если, к примеру, в игре используются преимущественно вершинные шейдеры (трехмерные модели с насыщенной геометрией), то может возникнуть ситуация, что будут заняты все четыре вершинных процессора и только один пиксельный процессор, а оставшиеся семь пиксельных процессоров будут бездействовать. В этом случае производительность всего графического процессора определяется производительностью и количеством вершинных процессоров. В случае если в игре используются преимущественно пиксельные шейдеры (трехмерные модели с насыщенными пиксельными эффектами), то может возникнуть обратная ситуация, когда будут заняты только один вершинный процессор и все семь пиксельных процессоров. В этом случае производительность всего графического процессора определяется производительностью и количеством пиксельных процессоров [12].

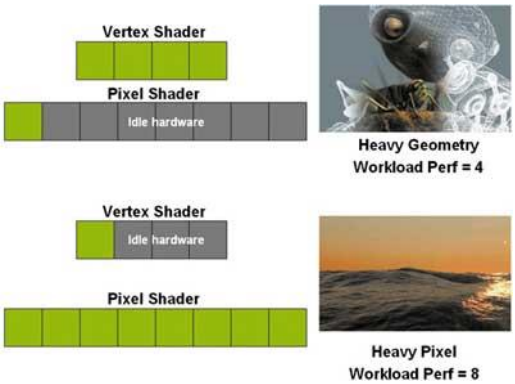

Рисунок 1.1 – Проблема сбалансированной нагрузки при использовании вершинных и пиксельных конвейеров

Данной проблемы можно избежать, если вместо четырех вершинных и восьми пиксельных процессоров (в сумме 12) использовать 12 унифицированных процессоров, которые могли бы выполнять как вершинные, так и пиксельные шейдеры (рисунок 1.2).

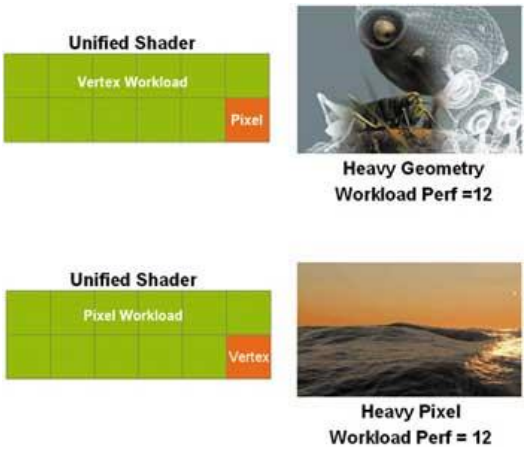

Рисунок 1.2 – Решение проблемы сбалансированной нагрузки при использовании унифицированных процессоров

Впрочем, говорить, что главная особенность унифицированных процессоров NVIDIA заключается только в том, что они способны выполнять как вершинные, так и пиксельные шейдеры, было бы не вполне корректно. Унифицированные процессоры способны выполнять геометрические (Geometry) и физические (Physics) расчеты, чего вообще не было предусмотрено в графических процессорах предыдущих поколений.

Унифицированные процессоры NVIDIA называются унифицированными потоковыми процессорами (Unified Streaming Processors, SP) и представляют собой скалярные процессоры общего назначения для обработки данных с плавающей запятой. Традиционно в процессорах существует два типа математики: векторная и скалярная. В случае векторной математики данные (операнды) представляются в виде n-мерных векторов, при этом над большим массивом данных проводится всего одна операция. Самый простой пример — задание цвета пиксела в виде четырехмерного вектора с координатами R, G, B, A, где первые три координаты (R, G, B) задают цвет пиксела, а последняя — его прозрачность. В качестве простого примера векторной операции можно рассмотреть сложение цвета двух пикселов. При этом одна операция осуществляется одновременно над восемью операндами (двумя 4-мерными векторами). В скалярной математике операции осуществляются над парой чисел. Понятно, что векторная обработка увеличивает скорость и эффективность обработки за счет того, что обработка целого набора (вектора) данных выполняется одной командой [11].

В графическом процессоре NVIDIA GeForce 8800 применяются 128 потоковых унифицированных процессоров, каждый из которых работает на тактовой частоте 1,35 ГГц. Структурная схема GPU NVIDIA GeForce 8800 представлена на рисунке 2.1

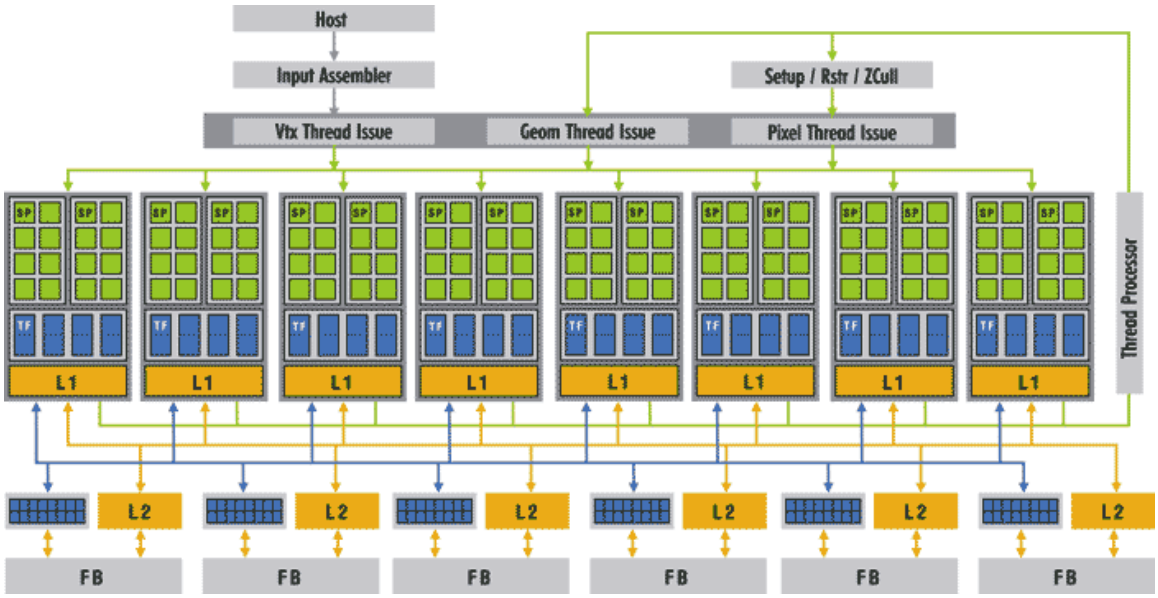

Рисунок 2.1 – Структурная схема графического процессора NVIDIA GeForce 8800

Потоковые процессоры сгруппированы в восемь блоков по 16 штук, каждый из которых оснащен четырьмя текстурными модулями и общим L1-кэшем. Каждый блок представляет собой два шейдерных процессора (состоящих из восьми потоковых процессоров каждый), при этом все восемь блоков имеют доступ к любому из шести L2-кэшей и к любому из шести массивов регистров общего назначения. Таким образом, обработанные одним шейдерным процессором данные могут быть использованы другим шейдерным процессором. На каждые четыре потоковых процессора приходится один текстурный блок, включающий один блок адресации текстур (Texture Address Unit, TA) и два блока фильтрации текстур (Texture Filtering Unit, TF). При этом текстурные блоки и кэш работают на частоте 575 МГц [6].

В архитектуре GeForce 8800 входящие данные (input stream) поступают на вход одного унифицированного процессора, обрабатываются им, по выходе (output stream) записываются в регистры, а затем вновь подаются на вход другого процессора для исполнения следующей операции обработки.

Возможность такой циклической потоковой обработки данных одновременно с унифицированными процессорами позволяет решить проблему их повторной обработки, довольно часто встречающуюся в современных играх. К примеру, унифицированный процессор под управлением вершинного шейдера

создает куб, который передается для дальнейшей обработки (подвергается текстурированию), одновременно данные поступают на вход другого унифицированного процессора, который создает из этого куба пирамиду или другую трехмерную фигуру. В случае традиционной линейной архитектуры процессора при такой операции пришлось бы дождаться полного формирования изображения и повторно считывать данные из кадрового буфера.

Данный графический процессор поддерживает множество новых для своего времени функций и технологий, среди которых [12]:

- новые режимы антиалиасинга и анизотропной фильтрации;
- поддержка геометрических шейдеров, реализованных в DirectX 10;
- режим HDR;
- технология расчета физических эффектов NVIDIA Quantum Effects;
- поддержка режима Extreme High Definition Gaming;
- технология PureVideo и PureVideo HD.

Архитектура GeForce 8 достаточно популярна в сообществе GPGPU, и причина этого — интерфейс программирования CUDA (C Unified Driver Architecture). В NVIDIA первыми обеспечили возможность написания полных программ на диалекте языка Си. Это означает, что на этом языке можно одновременно писать код, исполняемый на графическом процессоре, и код, исполняемый на центральном процессоре, и все это в рамках одного проекта. Язык Си был расширен дополнительными квалификаторами памяти и функций для представления статической памяти и шейдеров соответственно. И хотя остался ряд ограничений, например отсутствие рекурсии на GPU, и программирование перемещения данных по-прежнему лежит на разработчике, язык впервые позволил программировать на GPU в терминах, понятных обычным программистам.

Безусловно, такое решение нельзя считать идеальным. Во-первых, CUDA-программы исполняются только на графических процессорах от NVIDIA и не могут быть перенесены на другие архитектуры. Во-вторых, CUDA, как и Си, — это низкоуровневое средство программирования, и для написания эффективных программ необходимо хорошо понимать архитектуру GPU NVIDIA.

2.2 Архитектура AMD FireStream

Современные GPU компании AMD, представленные сериями HD 2K и HD 3K, больше похожи на традиционные GPU, чем продукты NVIDIA. Общая схема их организации представлена на рисунке 3.1.

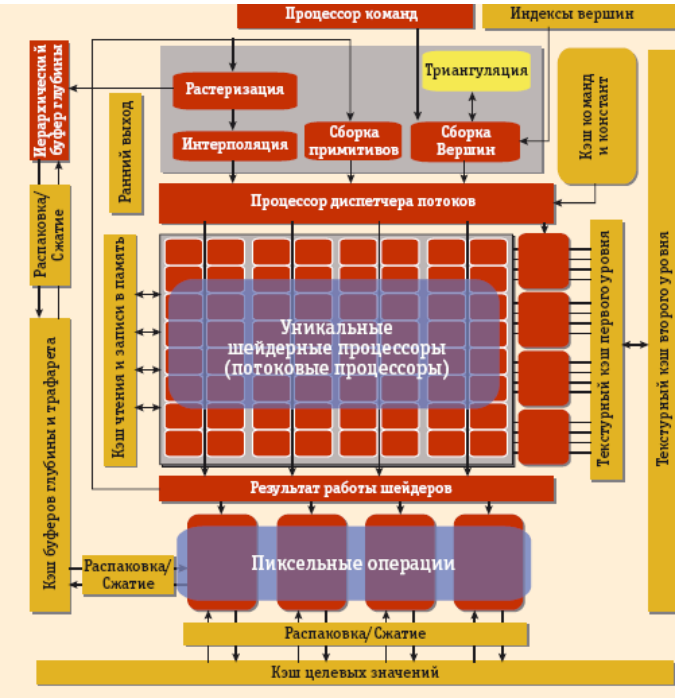

Рисунок 3.1 – Архитектура GPU серий AMD Radeon 2K и 3K

В отличие от NVIDIA, потоковый процессор (далее — ПП) AMD имеет архитектуру с длинным командным словом. Один ПП содержит шесть функциональных устройств (далее — ФУ), из которых одно отвечает за команды перехода, четыре — за обычные арифметические команды и еще одно — за вычисление элементарных функций (log, sin и т.д.). Арифметические ФУ способны выполнять до двух команд за такт, а ФУ, вычисляющее элементарные функции, — одну. Таким образом, один такой ПП способен выполнять до восьми команд за такт, что больше, чем у NVIDIA.

При рассмотрении данной архитектуры можно в первом приближении ограничиться только командами, задающими одну и ту же операцию для четырех арифметических ФУ. Это тем более логично, что обмен данными с памятью идет в терминах 128-битовых единиц, а это равно четырем 32-разрядным словам. С этой точки зрения ПП от AMD похож на один элемент SPE (Synergistic Processing Element) процессора Cell или ядро x86-процессора с расширениями SSE2.

Доступ к памяти на чтение возможен через один из 32 сегментных регистров, каждый из которых адресует двухмерный участок памяти, содержащий до 8192*8192 элементов (один элемент = 128 бит). В отличие от NVIDIA, GPU «в исполнении» AMD предоставляют два режима записи. В режиме регулярной записи шейдер может записать по одному элементу в каждый из выходных буферов, при этом индекс элемента в буфере жестко задан. В режиме произвольной записи доступен только один буфер, зато запись может

производиться по произвольному адресу. Средства синхронизации отсутствуют [11].

Серия GPU от AMD, ориентированная на высокопроизводительные вычисления, носит название FireStream. В отличие от NVIDIA, в AMD ориентируются только на производство графических карт, оставляя дизайн решений партнерам. Заметим, что в FireStream 9170 впервые появилась поддержка вычислений с двойной точностью, при этом пиковая производительность составляет 100 GFLOPS.

AMD первой обратила внимание на GPGPU, выпустив инструментарий низкоуровневого доступа, библиотеку DPVM (Data-Parallel Virtual Machine). В отличие от CUDA подобные средства скорее являются промежуточным звеном во взаимодействии средств более высокого уровня с GPU.

DPVM представляла GPU на двух уровнях— уровне отдельных шейдеров (они описывались на шейдерном языке PS 3.0 или на ассемблере AMD) и уровне последовательности команд, который отвечал за установку параметров и адресов, работу с кэшем и исполнение шейдера. Сама DPVM предоставляла только два библиотечных вызова: запуск последовательности команд и ожидание его завершения. Средств управления памятью в библиотеке не было. При этом была доступна вся видеопамять, а также память порта PCI-Express. Несмотря на отсутствие развитого языка для написания шейдеров, DPVM давала простой и понятный интерфейс доступа, который позволил ускорить ряд операций, например, умножение матриц, на порядок. По непонятным причинам, от него было решено отказаться. Современная версия библиотеки, CAL (Compute Abstraction Layer), поддерживает несколько GPU и возможность быстрого обмена, однако более сложна в применении и по-прежнему позволяет использовать только шейдерные языки, что, вероятно, и объясняет ее невысокую популярность.

2.3 Новейшая архитектура NVIDIA Turing

В семействе Turing можно выделить несколько ключевых изменений. Это абсолютно новая архитектура GPU, появление новых вычислительных блоков — тензорных и RT ядер, ускоренная обработка шейдеров, а также функция marquee. Она предназначена для гибридного рендеринга, который сочетает в себе трассировку лучей с традиционной растеризацией.

Новым является специализированный вычислительный блок RT Core. Его функция — поддержка трассировки лучей. Эти процессорные блоки ускоряют

как проверку пересечения лучей и треугольников, так и манипуляции с BVH (иерархии ограничивающих объемов).

Самые быстрые компоненты Turing могут обсчитывать 10 миллиардов (Гига) лучей в секунду, что по сравнению с неускоренным Pascal является 25-кратным улучшением характеристик трассировки лучей. Архитектура Turing включает тензорные ядра Volta, которые были усилены. Тензорные ядра являются важным аспектом нескольких инициатив NVIDIA. Наряду с ускорением трассировки лучей, важным инструментом NVIDIA является уменьшение количества лучей, требуемых в сцене с помощью шумоподавления AI, чтобы очистить изображение, здесь тензорные ядра справляются лучше всех. Конечно, это не единственная область, где они хороши – все нейронные сети и AI империи NVIDIA построены на них [1, 2].

Важные изменения произошли на уровне мультипроцессорных блоков SM, которые имеют стандартную структуру во всех вариантах GPU Turing. Новая архитектура наследует возможности вычислительной архитектуры Volta и игровой архитектуры Pascal. Все вычислительные блоки внутри SM сгруппированы в четыре массива обработки данных со своей управляющей логикой (данные регистров, планировщик). В одном SM насчитывается 64 потоковых процессора. И эти вычислительные блоки теперь умеют одновременно выполнять целочисленные операции (INT32) и операции с плавающей запятой (FP32).

Тензорные ядра Turing являются улучшенными ядрами Volta. Они нужны для выполнения задач с применением искусственного интеллекта. Эти блоки поддерживают расчеты в режимах INT8, INT4 и FP16 при работе с массивами матричных данных для глубокого обучения в реальном времени. Каждое тензорное ядро выполняет до 64 операций с плавающей запятой, используя входные данные формата FP16. То есть один SM с восемью ядрами обрабатывает 512 операций FP16 за такт. Вычисления INT8 проходят на удвоенной скорости 1024 операций, а для INT4 выполняется 2048 операций за такт. И топовый GPU TU102 способен обеспечить пиковую тензорную производительность до 130,5 TFLOPS (Quadro RTX 6000) [5].

Между тем, чтобы лучше использовать тензорные ядра вне задач трассировки лучей и узконаправленных задач глубокого обучения, NVIDIA будет развертывать SDK, NVIDIA NGX, что позволит интегрировать нейронные сети в обработку изображений. NVIDIA предполагает использование нейронных сетей и тензорных ядер для дополнительной обработки изображений и видео, включая такие методы, как предстоящее Deep-Anti-Aliasing (DLAA).

Все процессоры производятся по технологии 12-нм FinFET. Они сохраняют кластерную структуру, когда GPU состоит из нескольких GPC, и,

меняя количество таких кластеров, масштабируется производительность каждого конкретного чипа.

Графический процессор архитектуры Turing TU102 состоит из 18,6 миллиардов транзисторов при площади кристалла 754 кв.мм. Если сравнить его с GP102 (GeForce GTX 1080 Ti), то площадь нового чипа и количество транзисторов выросло на 55–60%. У TU102 всего шесть кластеров GPC, каждый содержит по шесть текстурно-процессорных кластеров TPC, объединяющих мультипроцессорные блоки SM. Каждый SM-блок насчитывает 64 основных вычислительных блока (CUDA-cores). При 72 SM всего получается 4608 потоковых процессоров. Однако GPU GeForce RTX 2080 Ti немного урезан. У топовой видеокарты отключены два SM, в итоге общее количество потоковых процессоров равно 4352. Также у данного решения имеется 544 новых тензорных ядра и 68 RT-ядер, 272 текстурных блока и 88 блоков растеризации ROP [5].

Обновленная унифицированная структура кэша L1 позволяет конвейеру TPC эффективнее работать с ним. При сохранении общего объема кэша L1 на уровне 96 КБ меньше латентность, а общая пропускная способность может вырасти до двух раз. Также во всех процессорах увеличен объем общего кэша L2. К примеру, в GPU TU102 это 6 МБ вместо 3 МБ у старого GP102 (рисунок 4.1)

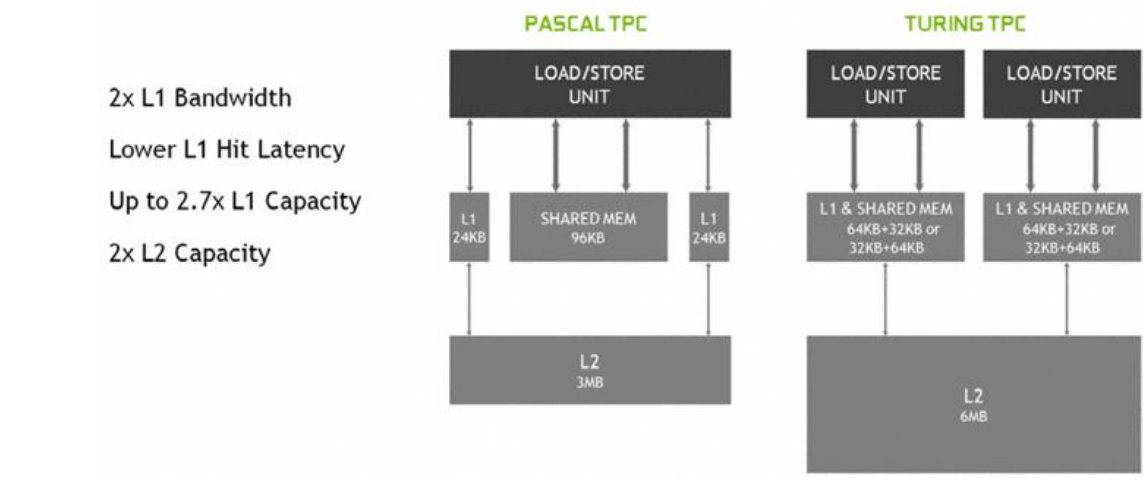

Рисунок 4.1 – Сравнение структур кэша архитектур Pascal и Turing

ГЛАВА 3
ВЫЧИСЛЕНИЯ С ПРИМЕНЕНИЕМ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

Для проведения различных манипуляций с такими объемами данных сегодня применяют многоядерные и многопроцессорные системы, суперкомпьютеры и компьютерные сети. Более того, для хранения таких объемов информации уже недостаточно «местных» серверов и с помощью облачных технологий используются удаленные серверы.

Все это обусловливает создание параллельных алгоритмов и использование систем параллельного действия для их обработки. В связи с этим актуальной становится задача разработки общей технологии проектирования параллельных программ и использования ее при решении ряда прикладных задач на различных вычислительных установках.

Графический процессор – это устройство, предназначенное для эффективной обработки компьютерной графики. В современных видеоадаптерах он успешно используется как ускоритель трехмерной графики благодаря специальной организации конвейера в его архитектуре, изначально нацеленной на параллельные вычисления. Графический конвейер представляет собой набор программно-аппаратных средств, преобразующих описанные виртуальные объекты в матрицу ячеек видеопамяти растрового дисплея. Для сбалансированной нагрузки вершинных и пиксельных процессоров, как упоминалось ранее, начали использовать унифицированные процессоры [8].

Рост частот универсальных процессоров упёрся в физические ограничения и высокое энергопотребление, и увеличение их производительности всё чаще происходит за счёт размещения нескольких ядер в одном чипе. Продаваемые сейчас процессоры содержат лишь до четырёх, и они предназначены для обычных приложений, используют MIMD — множественный поток команд и данных. Каждое ядро работает отдельно от остальных, исполняя разные инструкции для разных процессов.

Специализированные векторные возможности (SSE2 и SSE3) для четырехкомпонентных (одинарная точность вычислений с плавающей точкой) и двухкомпонентных (двойная точность) векторов появились в универсальных процессорах из-за возросших требований графических приложений, в первую очередь. Именно поэтому для определённых задач применение GPU выгоднее, ведь они изначально сделаны для них [9].

Например, 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.

Итак, перечислим основные различия между архитектурами CPU и GPU.

Во-первых, ядра 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.

Во-вторых, разработчики CPU стараются добиться выполнения как можно большего числа инструкций параллельно, для увеличения производительности. Для этого, начиная с процессоров Intel Pentium, появилось суперскалярное выполнение, обеспечивающее выполнение двух инструкций за такт, а Pentium Pro отличился внеочередным выполнением инструкций. Но у параллельного выполнения последовательного потока инструкций есть определённые базовые ограничения и увеличением количества исполнительных блоков кратного увеличения скорости не добиться.

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). Так, архитектура Tesla в некоторых условиях запускает на исполнение операции MAD+MUL или MAD+SFU одновременно [7].

В-третьих, GPU отличается от CPU ещё и по принципам доступа к памяти. В GPU он связанный и легко предсказуемый — если из памяти читается тексель текстуры, то через некоторое время придёт время и для соседних текселей. Да и при записи то же — пиксель записывается во фреймбуфер, и через несколько

тактов будет записываться расположенный рядом с ним. Поэтому организация памяти отличается от той, что используется в CPU. И видеочипу, в отличие от универсальных процессоров, просто не нужна кэш-память большого размера, а для текстур требуются лишь несколько (до 128-256 в нынешних GPU) килобайт.

Да и сама по себе работа с памятью у GPU и CPU несколько отличается. 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 [4].

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.

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.

Кроме того, центральные процессоры используют SIMD (одна инструкция выполняется над многочисленными данными) блоки для векторных вычислений, а видеочипы применяют SIMT (одна инструкция и несколько потоков) для скалярной обработки потоков. SIMT не требует, чтобы разработчик преобразовывал данные в векторы, и допускает произвольные ветвления в потоках [10].

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.

В итоге, основой для эффективного использования мощи GPU в научных и иных неграфических расчётах является распараллеливание алгоритмов на

сотни исполнительных блоков, имеющихся в видеочипах. К примеру, множество приложений по молекулярному моделированию отлично приспособлено для расчётов на видеочипах, они требуют больших вычислительных мощностей и поэтому удобны для параллельных вычислений. А использование нескольких GPU даёт ещё больше вычислительных мощностей для решения подобных задач.

Выполнение расчётов на GPU показывает отличные результаты в алгоритмах, использующих параллельную обработку данных. То есть, когда одну и ту же последовательность математических операций применяют к большому объёму данных. При этом лучшие результаты достигаются, если отношение числа арифметических инструкций к числу обращений к памяти достаточно велико. Это предъявляет меньшие требования к управлению исполнением (flow control), а высокая плотность математики и большой объём данных отменяет необходимость в больших кэшах, как на CPU.

ЗАКЛЮЧЕНИЕ

Представленная компанией NVIDIA программно-аппаратная архитектура для расчётов на видеочипах CUDA хорошо подходит для решения широкого круга задач с высоким параллелизмом. CUDA работает на большом количестве видеочипов NVIDIA, и улучшает модель программирования GPU, значительно упрощая её и добавляя большое количество возможностей, таких как разделяемая память, возможность синхронизации потоков, вычисления с двойной точностью и целочисленные операции.

CUDA — это доступная каждому разработчику ПО технология, её может использовать любой программист, знающий язык Си. Придётся только привыкнуть к иной парадигме программирования, присущей параллельным вычислениям. Но если алгоритм в принципе хорошо распараллеливается, то изучение и затраты времени на программирование на CUDA вернутся в многократном размере.

Сегодня имеется хорошая перспектива использования графических процессоров для высокопроизводительных вычислений. Для рассмотренных архитектур GPU ясно видны как разрыв между высокой пиковой производительностью и возможностью ее реального достижения для широкого класса задач, так и отсутствие адекватных средств программирования. Определенные шаги в этом направлении уже выполнены в системах типа RapidMind или C$, однако здесь еще многое нужно сделать.

Параллелизм в современных архитектурах не ограничивается суперкомпьютерами — он идет «в массы» в виде многоядерных процессоров, GPU, ПЛИС и других архитектур. Поэтому, с одной стороны, стоит ожидать появления стандартов, которые предоставят единый способ организации программ в неоднородных вычислительных средах, а с другой— высокоуровневых языков программирования, которые будут способны эффективно отображать параллельные программы на различные целевые архитектуры.

Отходя от вопроса вычислений, NVIDIA Turing — передовая графическая архитектура, которая расширяет возможности привычного рендеринга, добавляя трассировку лучей в реальном времени и возможность использовать нейронные сети для вспомогательных функций. Новые аппаратные возможности обеспечивают поддержку совершенно новых технологий и графических эффектов. Появление Turing стало знаковым событием, которое обозначает старт новой эры и постепенную интеграцию трассировки в игровую индустрию.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. NVIDIA. Раскрывая тайны архитектуры GPU Turing следующего поколения: удвоенный Ray Tracing, GDDR6 и многое другое // Хабр [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/ua-hosting/blog/422773/. – Дата доступа: 15.12.2020

2. Smith R., NVIDIA Reveals Next-Gen Turing GPU Architecture: NVIDIA Doubles-Down on Ray Tracing, GDDR6, & More // Информационно-аналитический портал AnandaTech [Электронный ресурс]. – Режим доступа: https://www.anandtech.com/show/13214/nvidia-reveals-next-gen-turing-gpu-architecture. – Дата доступа: 15.12.2020

3. Адинец А., Воеводин В., Графический вызов суперкомпьютерам // Издательство “Открытые системы” [Электронный ресурс]. – Режим доступа: https://www.osp.ru/os/2008/04/5114497. – Дата доступа: 15.12.2020

4. Архитектура CUDA следующего поколения, кодовое название Fermi. Сердце суперкомпьютера в теле GPU // Сайт компании NVIDIA [Электронный ресурс]. – Режим доступа: http://www.nvidia.ru/object/fermi_architecture_ru.html. – Дата доступа: 15.12.2020

5. Архитектура Turing и особенности новых видеокарт GeForce RTX // Информационно-аналитический портал Overclockers.ua [Электронный ресурс]. – Режим доступа: https://www.overclockers.ua/video/nvidia-turing-geforce-rtx/all/. – Дата доступа: 15.12.2020

6. Архитектура современных графических процессоров // Информационно-технический ресурс Radeon.ru [Электронный ресурс]. – Режим доступа: https://radeon.ru/articles/technology/chiparch/. – Дата доступа: 15.12.2020

7. Беррило А., Nvidia CUDA? Неграфические вычисления на графических процессорах // Информационно-аналитический ресурс IXBT.com [Электронный ресурс]. – Режим доступа: https://www.ixbt.com/video3/cuda-1.shtml. – Дата доступа: 15.12.2020

8. Буза М.К., Высокоэффективные вычисления с применением графических процессоров // Электронная библиотека БГУ [Электронный ресурс]. – Режим доступа: https://elib.bsu.by/handle/123456789/134541. – Дата доступа: 15.12.2020

9. Валях Е., Параллельно-последовательные вычисления. М., 1985.

10. Вычисления на графических процессорах // Информационный портал «Клуб DNS» [Электронный ресурс]. – Режим доступа: https://club.dns-shop.ru/digest/7843-vyichisleniya-na-graficheskih-protsessorah/. – Дата доступа: 15.12.2020

11. Завьялов Н., Графические процессоры в решении современных IT-задач // Официальный блог компании Selectel [Электронный ресурс]. – Режим доступа: https://selectel.ru/blog/gpu-for-business/. – Дата доступа: 15.12.2020

12. Пахомов С., Революция в мире графических процессоров // КомпьютерПресс [Электронный ресурс]. – Режим доступа: http://www.compress.ru/Article.aspx?d=16963. – Дата доступа: 15.12.2020