Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Национальный исследовательский университет «МЭИ»

Реферат на тему
“Особенности применения тензорных ядер в видеокартах NVIDIA в матричных операциях”
Курс: “Системная интеграция”

Группа: А-07м-23

Выполнил: Балашов С.А.

Проверил: Рыбинцев В.О.

Москва, 2023

Содержание

Введение

В настоящее время сложные вычисления производятся практически во всех сферах жизни. Для ускорения этих расчётов постоянно разрабатываются новые технологии и аппаратные средства. Одной из таких технологий являются тензорные ядра[1], которые впервые были представлены компанией NVIDIA в графических процессорах поколения Volta. Сейчас выпущено уже четвертое поколение тензорных ядер[1] и они повсеместно используются в машинном обучении и других задачах, требующих работы с матрицами.

1. Введение в видеокарты

Для рассмотрения особенностей применения тензорных ядер[1] от NVIDIA следует сначала дать краткое описание аппаратной части, на которой выполняются вычисления.

Видеокарта - устройство, преобразующее данные изображения, хранящиеся в памяти компьютера или самой карты, в форму, пригодную для отображения на экране монитора. Видеокарта состоит из графического процессора, памяти и разных сопроцессоров, назначение которых зависит от исполняемых на устройстве задач.

Графический процессор (GPU) - это процессор в видеокарте, специализирующийся на параллельных вычислениях. Он изначально спроектирован для обработки изображений, и визуализации трёхмерных сцен. Он содержит гораздо больше транзисторов, чем центральный процессор, специализирующихся на вычислениях с плавающей точкой. С течением времени GPU развился в высокопараллельный, мультипоточный процессор с

исключительной производительностью.

Основное отличие между GPU и CPU состоит в том, что CPU содержит от одного до нескольких ядер, каждое из которых может выполнять свой поток команд (MIMD архитектура), а GPU - содержит множество потоковых мультипроцессоров, каждый из которых содержит некоторое количество ядер. Все ядра одного потокового мультипроцессора выполняют одни и те же инструкции (архитектура SIMD). Устройство графического процессора NVIDIA поколения Turing[2] представлено на рисунке 1.

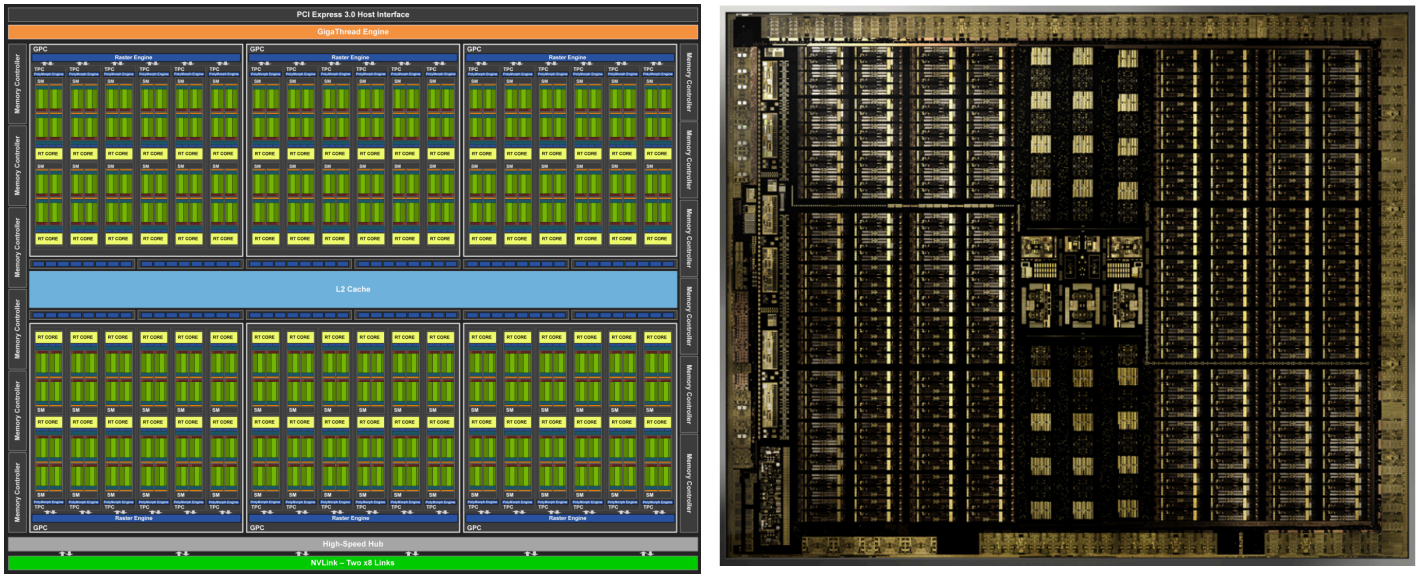


Рис. 1. Устройство графического процессора TU102

Процессор представляет из себя набор кластеров, состоящих из мультипроцессоров, у каждого из которых есть свой кэш 1-го уровня и общий для всех кэш 2-го уровнях. Каждый мультипроцессор в графическом процессоре NVIDIA из набора исполнительных блоков, в каждом из которых есть ядра вычислений с плавающей точкой, целочисленных вычислений и тензорные ядра[1]. Архитектура мультипроцессора поколения Turing[2] представлена на рисунке 2.

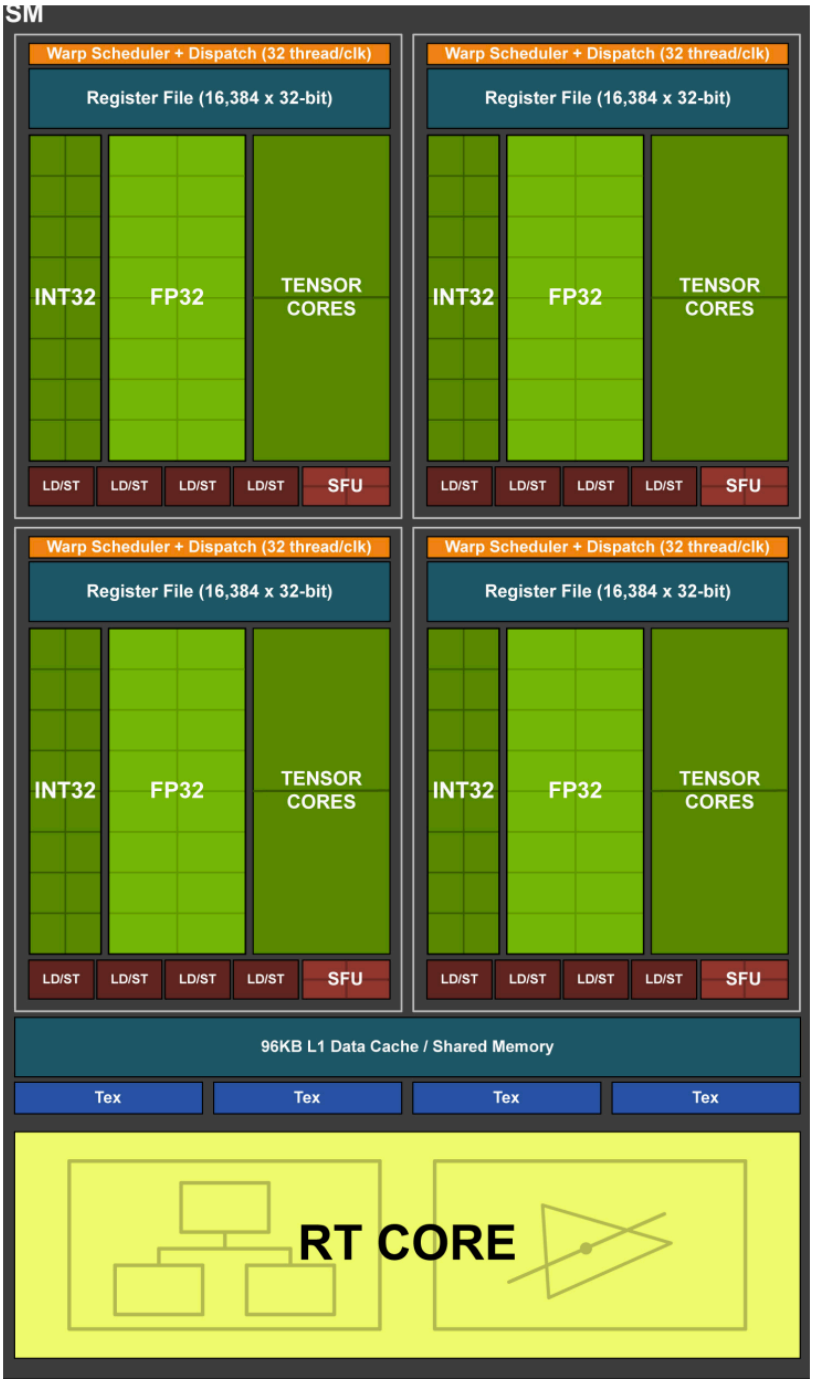


Рис. 2. Архитектура мультипроцессора поколения Turing

2. Введение в CUDA

Перед описанием архитектуры и возможностей тензорных ядер[1] сначала необходимо рассмотреть ядра CUDA[3]. CUDA (Compute Unified Device Architecture) - это собственная платформа параллельной обработки NVIDIA и API для графических процессоров, а ядра CUDA[3] - это стандартный модуль с плавающей точкой в видеокарте NVIDIA. Они присутствуют в каждом графическом процессоре NVIDIA, выпущенном за последнее десятилетие, и являются определяющей особенностью микроархитектуры графических процессоров NVIDIA.

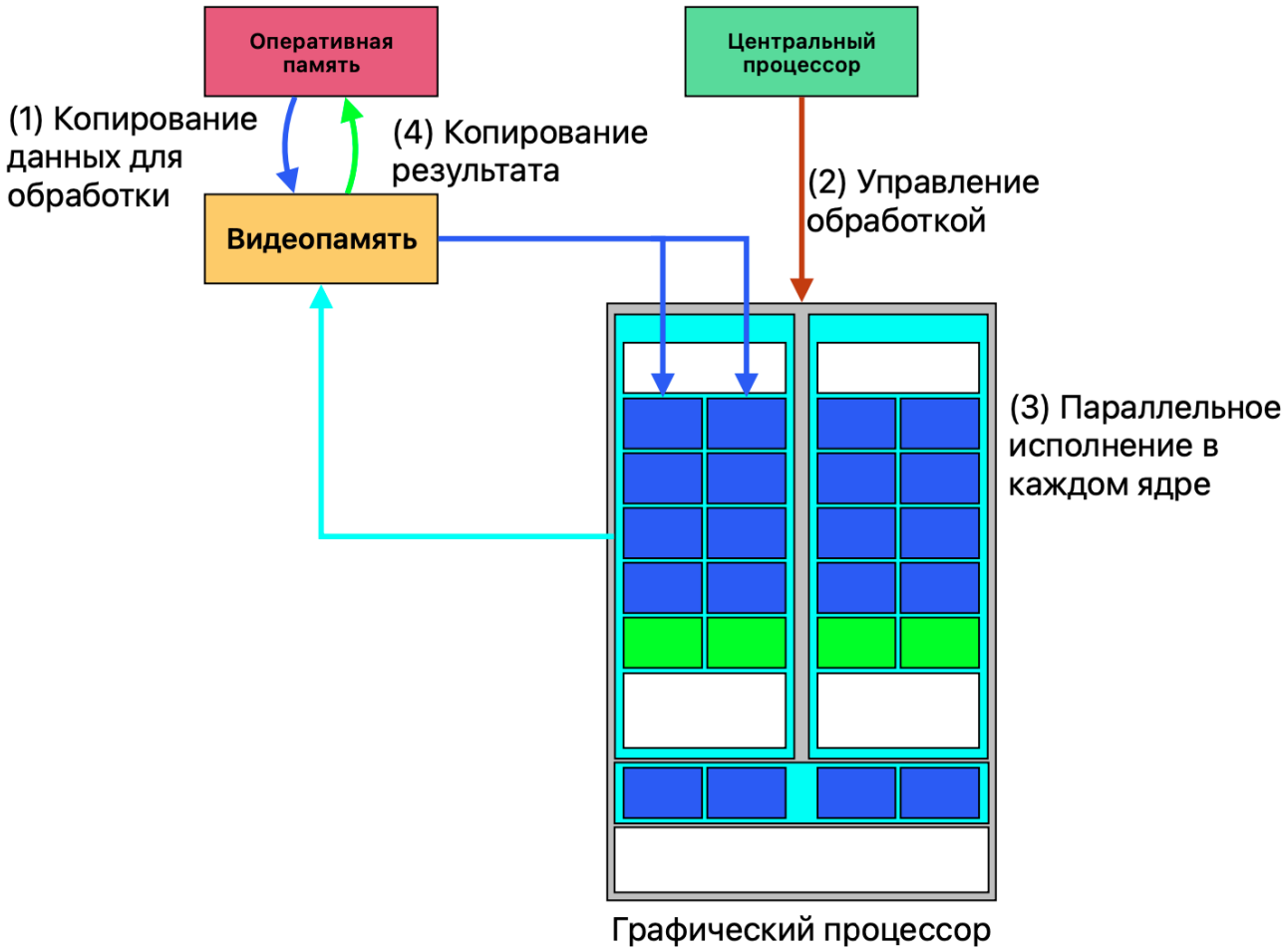


Рис. 3. Ход вычислений с CUDA

Каждое ядро CUDA[3] может выполнять одну операцию за такт. Хотя одно ядро CUDA[3] много медленнее ядра CPU, при совместном использовании большого их числа для расчётов, они могут ускорять вычисления за счет параллельного выполнения процессов.

До выпуска тензорных ядер[1] ядра CUDA[3] были определяющим оборудованием для ускорения глубокого обучения. Поскольку они могут выполнять только одно вычисление за такт, графические процессоры, ограниченные производительностью ядер CUDA[3], также ограничены количеством доступных ядер CUDA[3] и тактовой частотой каждого ядра. Чтобы преодолеть это ограничение, NVIDIA разработала тензорные ядра[1].

3. Введение в тензоры

Тензорные ядра[1] - специализированные вычислительные блоки, спроектированные для осуществления тензорных операций, которые являются основой для нейросетевого обучения.

По сути, тензорные ядра[1] - это процессоры, ускоряющие процесс умножения матриц. Это технология, разработанная Nvidia для своих высокопроизводительных потребительских и профессиональных графических процессоров. В настоящее время он доступен на ограниченных графических процессорах, например, на процессорах семейства Geforce RTX, Quadro RTX и Titan. Он может обеспечить повышенную производительность в области искусственного интеллекта, игр и создания контента.

Тензор - это математический объект, описывающий соотношения между другими математическими объектами, связанными друг с другом. Наиболее распространенные в математических вычислениях тензоры - скаляр, вектор и матрица.

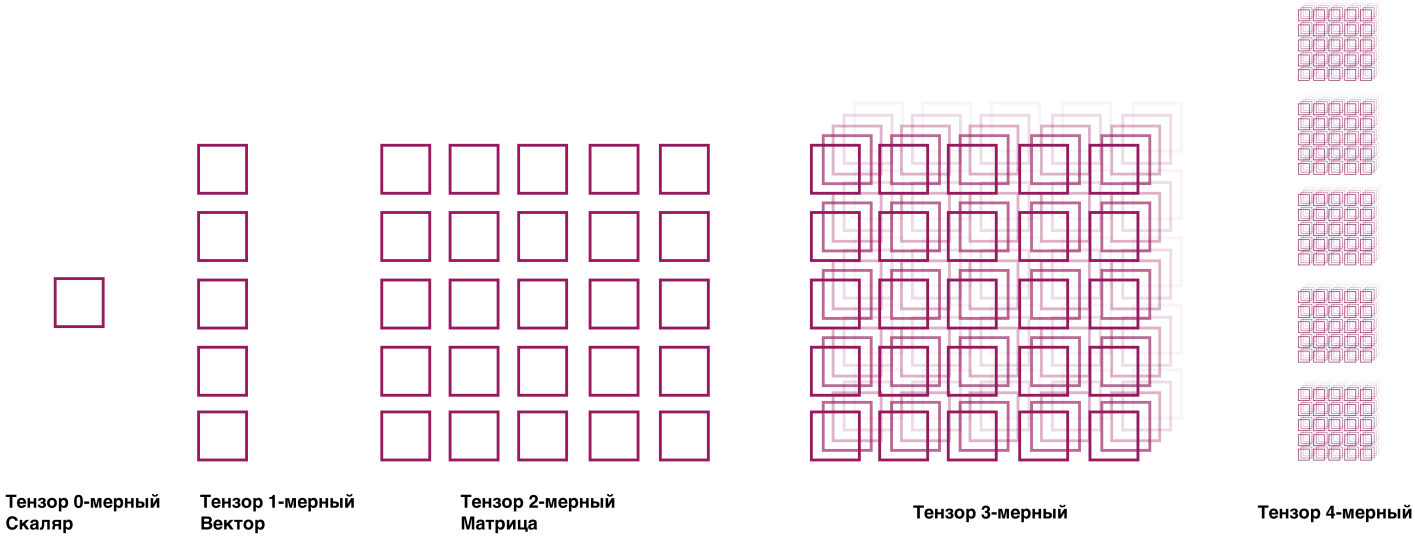


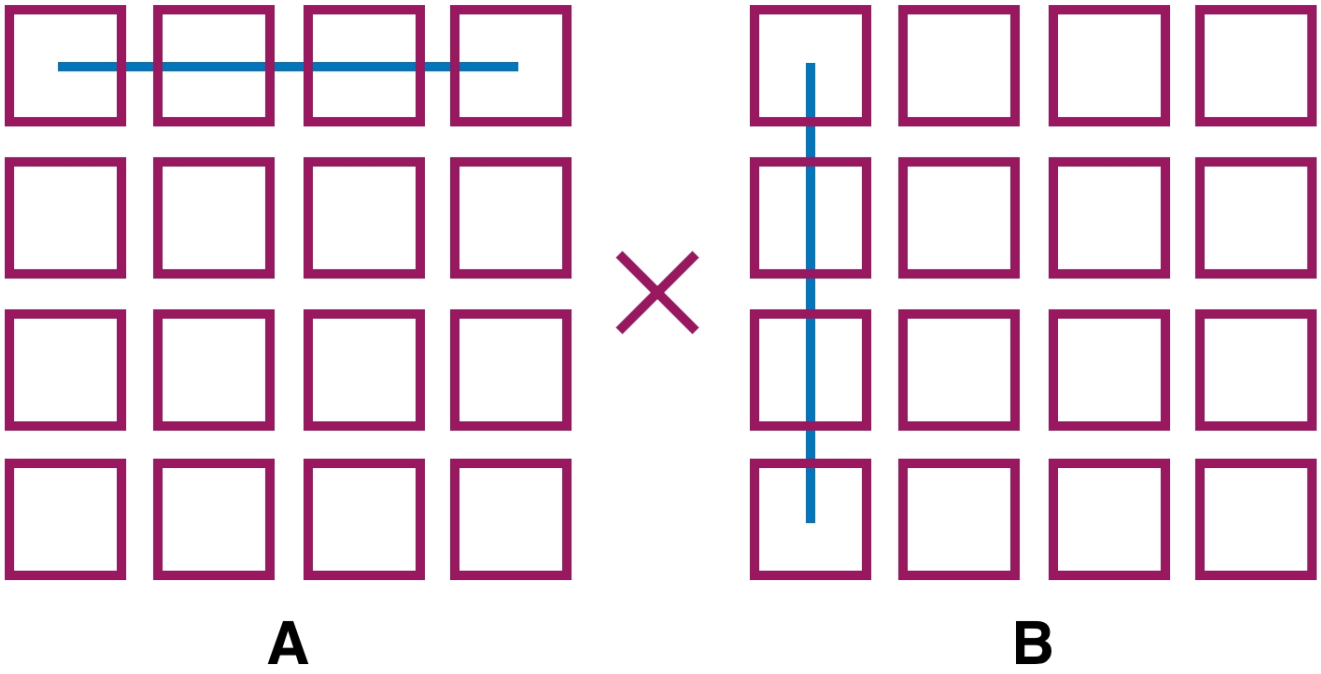
Рис. 4. Тензоры разных измерений от 0 до 4

Тензорные ядра[1] в основном используются при решении математических задач и глубоком обучении для повышения разрешения и качества изображений и графики в компьютерных играх. Технология DLSS[5] использует именно тензорные ядра[1] для вычислений. Ядра позволяют производить умножение матриц быстрее стандартных ядер графического или центрального процессоров.

Конкретно в графическом процессоре NVIDIA TU102[2] содержится 576 тензорных ядер[1]: по 8 на каждый мультипроцессор или по 2 на каждый исполнительный блок. Каждое ядро может выполнять до 64 операций умножения/сложения с плавающей точкой за такт с 16 битными значениями с плавающей точкой на входе. Итого суммарно все тензорные ядра могут выполнять 1024 операции с плавающей точкой или 2048 целочисленных операций за такт.

4. Принцип работы

Все микропроцессоры предназначены для выполнения арифметических и логических операций. Одной из арифметических операций, имеющих большое значение, является умножение матриц. Умножение двух матриц 4×4 включает 64 умножения и 48 сложений.



$$c_{ij}=\sum_{k} a_{ik}\cdot b_{kj}$$

Рис. 4. Умножение матриц

Тензорные ядра[1] особенно полезны при свёртке и умножении. Сложность вычислений многократно возрастает с увеличением размера и размерности тензора. Машинное обучение, глубокое обучение, трассировка лучей[4] - это задачи, требующие чрезмерного количества умножений.

Ядра CUDA[3] присутствуют в каждом графическом процессоре, разработанном Nvidia за последнее десятилетие, а тензорные ядра[1] были представлены недавно. Они могут выполнять вычисления намного быстрее, чем ядра CUDA[3]. В отличие от

последних, они могут выполнять несколько операций за такт. При этом из-за сильного ускорения вычислений страдает их точность. По сути, всё, что делают тензорные ядра[1], - это ускорение умножения матриц. Их применение практически не ограничено. Ниже описаны некоторые из вариантов.

4.1. Машинное обучение

Глубокое обучение предполагает обработку огромного количества данных. Набор исходных данных пропускают через множество свёрточных слоёв перед тем как получить сохранить результат. Этот процесс включает в себя огромное количество перемножения матриц.

В настоящее время большинство суперкомпьютеров оснащены графическими процессорами Nvidia, и это помогает компьютерным инженерам использовать тензорные ядра[1] для своей работы.

Университетам и научным работникам, работающим над алгоритмами искусственного интеллекта и машинного обучения, необходимо тренировать свои модели ИИ. Наличие платформы, способной ускорять тренировку, значительно облегчает задачу.

4.2. Беспилотные электромобили

Графические процессоры Nvidia являются идеальным выбором для моделирования преобразователей электрической энергии и обучения алгоритмов автономного управления.

4.3. Медиа и сфера развлечений

Высокопроизводительные компьютеры могут оказаться очень полезными при создании медиа высокого качества. Создание графики и видео высокого разрешения требует серьезных вычислительных мощностей, которые могут предоставить GPU.

4.4. Игры

Графические процессоры Nvidia серии RTX поддерживают технологию под названием DLSS[5] (Deep Learning Super Sampling). DLSS[5] использует алгоритмы глубокого обучения для рендеринга графики с низким разрешением и повышения ее качества за счет алгоритма супер-разрешения[6]. Если включен DLSS[5], ваш компьютер может рендерить игры в разрешении 540p и повышать их разрешение до 1080p.

С анонсом технологии трассировки лучей[4] в реальном времени второго поколения от Nvidia поддержание разрешения 4K и 60 кадров в секунду стало непростой

задачей. Только лучшие графические процессоры могут удовлетворить этим быстрорастущим требованиям.

Трассировка лучей[4] - это высоконагрузочный процесс. Чтобы обеспечить достаточный уровень кадров в секунду при включенной трассировке лучей[4], разработчикам игр приходится приложить огромную работу по оптимизации игр. Добавление алгоритмов супер-разрешения[6] увеличивает нагрузку. Тензорные ядра [1]совместно с ядрами RT (трассировки лучей[4]) могут ускорить вычисления.

Хотя большинство этих процессов по-прежнему выполняются на ядрах CUDA[3], ядра трассировки лучей[4] и тензорные ядра[1] вскоре будут играть важную роль в процессорах с точки зрения как стоимости, так и скорости вычислений.

5. Поколения

	Hopper	Ampere	Turing	Volta
Supported Tensor Core precisions	FP64, TF32, bfloat16, FP16, FP8, INT8	FP64, TF32, bfloat16, FP16, INT8, INT4, INT1	FP16, INT8, INT4, INT1	FP16
Supported CUDA® Core precisions	FP64, FP32, FP16, bfloat16, INT8	FP64, FP32, FP16, bfloat16, INT8	FP64, FP32, FP16, INT8	FP64, FP32, FP16, INT8

Рис. 5. Сравнение возможностей поколений тензорных ядер

Первое поколение тензорных ядер[1] имело микроархитектуру графического процессора Volta. Эти ядра обеспечивали обучение со смешанной точностью с числовым форматом FP16 (плавающая точка с длиной мантиссы 16). Это увеличило потенциальную пропускную способность этих графических процессоров почти в 12 раз в пересчете на терафлопс. По сравнению с графическими процессорами Pascal предыдущего поколения, 640 ядер флагманского V100 обеспечивают пятикратное увеличение производительности.

Второе поколение тензорных ядер[1] появилось с выпуском графических процессоров Turing[2]. Поддерживаемые точности тензорный ядер[1] были расширены и теперь включают Int8, Int4 и Int1. Это позволило при выполнении операций обучения смешанной точности повысить производительность графического процессора почти в 32 раза по сравнению с графическими процессорами Pascal.

Помимо графических процессоров второго поколения, графические процессоры Turing[2] также содержат ядра трассировки лучей[4], которые используются для расчета свойств графической визуализации, таких как свет и звук, в трехмерных средах.

Линейка графических процессоров Ampere представила третье поколение тензорных ядер[1], самое мощное на данный момент. Архитектура графического процессора Ampere основана на предыдущих инновациях микроархитектур Volta и Turing[2], расширяя вычислительные возможности до точности FP64, TF32 и bfloat16. Эти дополнительные форматы точности еще больше ускоряют глубокое обучение и задачи свёртки. Например, формат TF32 работает аналогично FP32, одновременно обеспечивая ускорение до 20 раз без изменения какого-либо кода. Таким образом,

реализация автоматической смешанной точности еще больше ускорит обучение с помощью всего лишь нескольких строк кода. Кроме того, микроархитектура Ampere имеет дополнительные функции, такие как специализация на разреженной матричной математике, NVLink третьего поколения, обеспечивающий быстрое взаимодействие нескольких графических процессоров, и ядра трассировки лучей третьего поколения.

Четвертое поколение тензорных ядер[1] с микроархитектурой Hopper оснащено тензорными ядрами[1] четвертого поколения, которые имеют расширенные возможности обработки форматов точности FP8 и которые ускоряют большие языковые модели в 30 раз по сравнению с Ampere.

Благодаря этим достижениям графические процессоры Hopper, в частности H100 для центров обработки данных, в настоящее время являются самыми мощными графическими процессорами, доступными на рынке.

В дополнение к этому, NVIDIA утверждает, что их новая технология NVLink позволяет подключать до 256 графических процессоров H100. Это позволяет увеличить масштаб вычислений для обработки данных.

Заключение

Тензорные ядра[1] - технология с изначально узким профилем задач применяется в целом ряде разных направлений и помогает ускорять вычисления. В этом они похожи на графические ускорители нескольк десятилетий назад, когда их начали использовать не только для графики, то и для параллельных вычислений. Сейчас, графические процессоры с тензорными ядрами[1] стоят во всех вычислительных центрах и обеспечивают высокую производительность в ряде расчётов. Их развитие позволяет сосредоточиться на решении сложных задач и не тратить время на попытки оптимизации матричных операций.

Список литературы

1. NVIDIA. NVIDIA DLSS 2.0: A Big Leap In AI Rendering [Электронный ресурс]: https://www.nvidia.com – Электронные данные. Режим доступа: URL.: https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/, свободный. − Загл. с экрана.

2. NVIDIA. NVIDIA TURING GPU ARCHITECTURE [Электронный ресурс]: https://images.nvidia.com. - Электронные данные. Режим доступа: URL.: https://images.nvidia.com/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf, свободный. − Загл. с экрана.

3. Сандерс, Дж. Технология CUDA в примерах: введение в программирование графических процессоров: Пер. с англ. Слинкина А.А., научный редактор Боресков А.В. / Сандерс Дж., Кэндрот Э. - М.:ДМК Пресс, 2018 - 232 с.: ил.

4. NVIDIA. NVIDIA RTX Platform [Электронный ресурс]: https://developer.nvidia.com – Электронные данные. Режим доступа: URL.:https://developer.nvidia.com/rtx, свободный. − Загл. с экрана.

5. NVIDIA. NVIDIA DLSS 2.0: A Big Leap In AI Rendering [Электронный ресурс]: https://www.nvidia.com – Электронные данные. Режим доступа: URL.: https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/, свободный. − Загл. с экрана.

6. Github. Learning a Single Convolutional Super-Resolution Network for Multiple Degradations [Электронный ресурс]: https://github.com – Электронные данные. Режим доступа: URL.:https://github.com/cszn/SRMD, свободный. − Загл. с экрана.