

Д. С. Братчиков¹✉, Д. М. Рыжков¹, К. Г. Гадьлышин¹

Повышение производительности полноволнового моделирования на GPU за счет использования MIG и асинхронных вычислений

¹Институт математики им. С. Л. Соболева СО РАН, г. Новосибирск, Российская Федерация
e-mail: d.s.bratchikov@math.nsc.ru

Аннотация. Численное моделирование сейсмических волновых полей является вычислительно сложной задачей, требующей расчета тысяч независимых источников. Современные графические ускорители обеспечивают высокое ускорение конечно-разностных алгоритмов, однако моделирование одного источника часто не позволяет полностью загрузить вычислительные ресурсы видеокарт класса A100/H100. В настоящей работе предлагается стратегия пакетной обработки источников, при которой на одной GPU одновременно рассчитывается группа независимых источников. Рассмотрены две реализации: асинхронная с использованием CUDA Streams и с расширением сетки потоков фиктивной четвертой размерностью. Показано, что пакетная обработка позволяет добиться ускорения и повышает загрузженность графического ускорителя. Выполнено сравнение с технологией NVIDIA MIG. Оптимальный размер пакета составляет 4-8 источников в зависимости от размера сетки и доступного объема видеопамати. Полученные результаты позволяют сформулировать практические рекомендации по выбору стратегии распараллеливания при кластерном моделировании в разведочной геофизике.

Ключевые слова: сейсмическое моделирование, конечно-разностная схемы, графические ускорители, пакетная обработка, MIG, высокопроизводительные вычисления

D. S. Bratchikov¹✉, D. V. Ryzhkov¹, K. G. Gadylyshin¹

Improving the performance of wave modeling on GPUs using MIG and asynchronous computing

¹Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russian Federation
e-mail: d.s.bratchikov@math.nsc.ru

Abstract. Numerical modeling of seismic wave fields is a computationally complex problem requiring the calculation of thousands of independent sources. Modern GPU provide high acceleration of finite-difference algorithms; however, modeling a single source often does not fully utilize the computational resources of A100/H100 class video cards. In this paper, we propose a strategy for batch processing of sources, in which a group of independent sources is simultaneously calculated on a single GPU. Two implementations are considered: an asynchronous one using CUDA Streams and one with an expansion of the flow grid by a fictitious fourth dimension. It is shown that batch processing allows for acceleration and increases the utilization of the graphics accelerator. A comparison with NVIDIA MIG technology is performed. The optimal batch size is 4-8 sources, depending on the grid size and available video memory. The obtained results allow us to formulate practical recommendations for choosing a parallelization strategy for cluster modeling in exploration geophysics.

Keywords: seismic modeling, FDTD, GPU, batching, MIG, HPC

Введение

Численное моделирование распространения сейсмических волн является фундаментальным инструментом в задачах разведочной и промысловой геофизики. Оно лежит в основе современных методов сейсмической визуализации, таких как RTM (Reverse Time Migration) [1], FWI (Full Waveform Inversion) [2], AVO/AVA анализ [3], а также в задачах микросейсмического мониторинга и контроля разработки месторождений. Вычислительная сложность этих задач обусловлена необходимостью многократного решения трехмерного волнового уравнения на высокоразрешающих сетках для тысяч источников возбуждения, что требует значительных вычислительных ресурсов и оптимизированных алгоритмических подходов.

С появлением графических ускорителей (GPU) проблема производительности конечно-разностного (FDTD) и конечно-элементного моделирования была существенно смягчена. Архитектура GPU, характеризующаяся высокой пропускной способностью памяти (до 3-4 ТБ/с) и тысячами вычислительных потоков, позволяет достичь ускорения в 10-50 раз по сравнению с многопроцессорными CPU-системами, особенно для трехмерных задач моделирования [4, 5]. Тем не менее, эффективность использования GPU сильно зависит от стратегии распараллеливания и конфигурации геометрии сбора данных.

Типичная поверхностная система наблюдения сбора 3D-данных сейсморазведки может включать от нескольких тысяч до десятков тысяч источников. Поскольку моделирование каждого источника представляет собой независимую задачу во временной области, естественным подходом является распределение источников по вычислительным узлам с использованием интерфейса передачи сообщений (MPI). Однако физическая область моделирования для отдельного источника ограничена максимальными выносами и глубиной целевых горизонтов. В результате локальная вычислительная сетка, достаточная для моделирования одного источника, зачастую оказывается недостаточно большой для полной загрузки современных высокопроизводительных GPU (например, NVIDIA A100, H100). При такой конфигурации коэффициент загрузки вычислительных блоков (occupancy) остается низким, а накладные расходы на запуск ядер и синхронизацию перестают быть пренебрежимыми.

В данной работе предлагается стратегия пакетной обработки источников, при которой на одной видеокарте одновременно моделируется группа источников. Данный подход требует выделения большего объема памяти для хранения волновых полей всех источников в пакете, однако позволяет существенно повысить загрузку потоковых процессоров GPU, снизить частоту передач данных по шине PCIe за счет уменьшения числа копирований и разделить вспомогательные буферы данных (параметры скорости, коэффициенты поглощающих слоев). Выполнено сравнение программной пакетной обработки с аппаратной технологией виртуализации NVIDIA Multi-Instance GPU (MIG).

Методы и материалы

1. Конечно-разностный FDTD-алгоритм моделирования на GPU одного источника

Рассмотрим простейшее скалярное акустическое волновое уравнение в изотропной среде, описывающее процесс распространения сейсмических волн

$$\frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = \Delta_{\vec{x}} u + S(t) \delta(\vec{x} - \vec{x}_s), \quad (1)$$

где $u(\vec{x}, t)$ – волновое поле; $v(\vec{x}, t)$ – скорость распространения волн; $S(t)$ – импульс зондирующего сигнала; $\vec{x}_s$ – координата источника.

Для численного решения уравнения (1) применяется явная конечно-разностная схема 2-го порядка по времени и 8-го по пространству

$$u_{ijk}^{n+1} = 2u_{ijk}^n - u_{ijk}^{n-1} + v^2 \Delta t^2 D_h u_{ijk}^n \quad (2)$$

где n – номер шага по времени; Δt – шаг по времени; h – шаг по пространству; D_h – дискретный оператор Лапласа.

Подавление нефизичных отражений от искусственных границ расчетной области выполнено с помощью поглощающих краевых условия CPML (Convolutional Perfectly Matched Layer) [6]. Высокопроизводительная реализация алгоритма выполнена с использованием среды программирования NVIDIA CUDA. Трехмерная вычислительная область отображается на трехмерную сетку потоков, где каждый поток обновляет одну ячейку сетки.

2. Пакетная обработка источников на GPU

При пакетной обработке на одной видеокарте одновременно рассчитывается N_s источников, что обеспечивает повышение коэффициента загрузки потоковых процессоров (осцирапсу) за счет увеличения параллельных вычислительных потоков. Основной вычислительный граф при этом не изменяется, однако архитектура управления памятью и запуска ядер претерпевает некоторые модификации. В частности, внедряются общие буферы для параметров среды и краевых условий.

Ключевым аспектом пакетной обработки является формирование объединенной вычислительной области. Поскольку каждый источник имеет собственный максимальный вынос, локальные сетки перекрываются. Для пакета формируется единая прямоугольная область, являющаяся объединением всех локальных сеток. Это приводит к росту потребляемой памяти и времени вычисления по сравнению с суммой изолированных расчетов. Поэтому необходимо группировать близко расположенные источники, чтобы увеличение области моделирования. Выделяют два подхода к реализации пакетной обработки:

- Стратегия с использованием CUDA Streams:

В данной реализации каждый источник в пакете выполняется в отдельном асинхронном потоке (CUDA Stream). Потоки планируются аппаратным планировщиком GPU, что позволяет перекрывать вычисления. Для каждого источника

выделяются независимые буферы в глобальной памяти с использованием SoA (Structure-of-Arrays), содержащие волновые поля, а также вспомогательные переменные CPML. Параметр скорости и коэффициенты поглощающих слоев являются общими, так как они не зависят от номера источника. Данный подход не требует модификации кода самого CUDA-ядра, поскольку изменения затрагивают исключительно логику его асинхронного запуска с использованием механизма CUDA Streams.

- Стратегия с добавлением фиктивной четвертой размерности:

В данном случае вводится дополнительная индексация источников $s \in [0, N_s - 1]$, и трехмерная сетка потоков расширяется до четырехмерной: (i_x, i_y, i_z, i_s) . На уровне CUDA это реализуется путем отображения четвертой размерности на `blockIdx.z`. Все источники в пакете обрабатываются в рамках одного запуска ядра, что минимизирует задержки планировщика и накладные расходы на запуск ядра. Общие переменные загружаются один раз и используются всеми источниками пакета, что позволяет их эффективно кэшировать. Память организуется в виде многомерных массивов $u[N_x][N_y][N_z][N_s]$. Реализация этой стратегии предполагает корректировку кода CUDA-ядер.

На практике оптимальный размер пакета N_s определяется из баланса между загрузкой GPU и доступным объемом видеопамати. Для современных карт класса A100 (80 ГБ) типичные значения составляют от 2 до 32 в зависимости от размера сетки (от 128^3 до 1024^3 узлов).

Так же стоит упомянуть технологию NVIDIA Multi-Instance GPU (MIG), доступную в архитектуре Ampere (A100) и Hopper (H100). Каждый MIG-инстанс обладает выделенными вычислительными ресурсами, включая потоковые мультипроцессоры, кэш-память и подсистему памяти, что обеспечивает аппаратную изоляцию вычислений и предсказуемость производительности. Использование MIG представляет практический интерес для задач, характеризующихся неполной загрузкой вычислительных ресурсов GPU. В настоящей работе рассматривается задача трехмерного конечно-разностного моделирования волновых полей, используемого при численном расчете большого числа независимых сейсмических источников. Для подобных вычислительных схем характерно преобладание операций обмена с глобальной памятью над арифметическими вычислениями, вследствие чего производительность алгоритма в значительной степени определяется пропускной способностью подсистемы памяти графического ускорителя. Выполненные вычислительные эксперименты показывают, что для рассматриваемых размеров расчетной области используемый алгоритм обладает выраженным memory-bound поведением, в результате чего использование полной вычислительной мощности GPU в рамках одного расчета оказывается недостаточно эффективным.

Результаты

Для сравнения стратегий пакетной обработки была проведена серия численных экспериментов. Тестирование проводилось на высокопроизводительной вычислительной системе "СЕРГЕЙ ГОДУНОВ" (Институт математики им. С.Л.

Соболева СО РАН), оснащенной узлом с графическими ускорителями NVIDIA Tesla A100 (4 x 80 ГБ GDDR6X, 2 x Intel Xeon Gold 6342, 1 ТБ DDR4).

Параметры моделирования выбраны в соответствии со значениями, типичными для реальных геофизических приложений. Размер области моделирования $10 \times 10 \times 1.4$ км. Область моделирования дискретизирована грубой $G_1 = 50 \times 50 \times 40$ м и мелкой $G_2 = 25 \times 25 \times 20$ м и сеткой. Скоростная модель однородная со значением 3000 м/с. Геометрия наблюдения включает 1024 источников, равномерно распределенных на поверхности. Максимальные выносы приемников для каждого источника 2 км. Число временных шагов 1001 с шагом 1 мс. Ширина поглощающих слоев 20 узлов сетки.

Тестирование выполнено для двух стратегий пакетной обработки (4D-сетка потоков и асинхронные CUDA Streams) при размере пакета $N_s = 1 \dots 32$ на одном графическом ускорителе. Результаты замеров приведены в табл. 1 – табл. 4.

Таблица 1. Полное время моделирования всех источников, с

Стратегия	Сетка	$N_s = 1$	$N_s = 2$	$N_s = 4$	$N_s = 8$	$N_s = 16$	$N_s = 32$
4D-сетка	G_1	76	65	59	56	54	53
	G_2	222	216	211	202	204	198
Stream	G_1	70	59	57	56	56	59
	G_2	208	199	195	195	198	201

Таблица 2. Ускорение полного времени расчета

Стратегия	Сетка	$N_s = 2$	$N_s = 4$	$N_s = 8$	$N_s = 16$	$N_s = 32$
4D-сетка	G_1	1.17	1.29	1.36	1.41	1.43
	G_2	1.03	1.05	1.1	1.09	1.12
Stream	G_1	1.19	1.23	1.25	1.25	1.19
	G_2	1.05	1.07	1.07	1.05	1.03

Таблица 3. Среднее время расчета одного пакета / N_s , мс

Стратегия	Сетка	$N_s = 1$	$N_s = 2$	$N_s = 4$	$N_s = 8$	$N_s = 16$	$N_s = 32$
4D-сетка	G_1	66	60	55	52	50	49
	G_2	194	183	176	172	173	171
Stream	G_1	62	52	51	50	51	53
	G_2	182	168	164	165	166	170

Таблица 4. Ускорение времени расчета одного источника в пакете

Стратегия	Сетка	$N_s = 2$	$N_s = 4$	$N_s = 8$	$N_s = 16$	$N_s = 32$
4D-сетка	G_1	1.1	1.2	1.27	1.32	1.35
	G_2	1.06	1.1	1.13	1.12	1.13
Stream	G_1	1.19	1.21	1.24	1.21	1.17
	G_2	1.08	1.11	1.1	1.09	1.07

Пакетная обработка обеспечивает сокращение общего времени моделирования для обеих стратегий. На грубой сетке G_1 ускорение достигает 20-30 %, на мелкой G_2 5-10 %. Это указывает на то, что при высокой вычислительной нагрузке (мелкая сетка) ресурсы GPU используются более полно и потенциал для дополнительного ускорения за счет пакетирования ограничен.

Стратегия с использованием CUDA Streams демонстрирует максимальное ускорение при малых размерах пакета ($N_s = 4, 8$), после чего наблюдается стабилизация или незначительная деградация производительности. Вероятной причиной является рост накладных расходов на управление потоками и синхронизацию, которые начинают компенсировать выгоду от асинхронного выполнения. Стратегия с 4D-сеткой потоков показывает монотонный рост ускорения с увеличением N_s . При $N_s = 32$ достигается максимальное сокращение времени расчета одного источника. Абсолютные значения времени (табл. 1, 3) свидетельствуют, что при $N_s \leq 8$ стратегия Streams обеспечивает несколько меньшее время выполнения для сетки G_1 . При больших размерах пакета преимущество переходит к 4D-подходу. Для сетки G_2 разница между стратегиями менее выражена, что, по-видимому, связано с доминированием вычислительной нагрузки над накладными расходами пакетной обработки.

Приведем сравнение расчета на 4-х графических ускорителях NVIDIA Tesla A100 на мелкой сетке G_2 без технологии MIG, и с применением технологии MIG. Был произведен расчет плотной системы наблюдения, состоящей из 100000 источников, максимальный вынос 2км, время записи 1с. Время расчета составило 10.8 часа. Во втором эксперименте был произведен такой же расчет 100000 источников, когда на каждом ускорителе была активирована технология MIG с разделением GPU на два независимых логических устройства конфигураций 4g.40gb и 3g.40gb. Таким образом, суммарно вычислительный узел содержал восемь независимых MIG-инстансов, предназначенных для одновременного выполнения вычислений. Время счета составило 14.7 часа. Таким образом ускорение расчета после включения MIG составило 1.36.

Заключение

В настоящей работе исследованы подходы к повышению производительности трехмерного конечно-разностного моделирования сейсмических волновых полей на графических ускорителях за счет пакетной обработки независимых источников. Рассмотрены две стратегии реализации пакетных вычислений: асинхронное выполнение с использованием CUDA Streams и подход с расширением сетки потоков фиктивной четвертой размерностью. Проведенные вычислительные эксперименты показали, что пакетная обработка позволяет существенно повысить эффективность использования вычислительных ресурсов GPU и сократить общее время моделирования. Для задач с относительно небольшими размерами расчетной области достигнуто ускорение порядка 20–30 %, тогда как для более крупных сеток эффект снижается вследствие более полной загрузки вычислительных ресурсов графического ускорителя.

Дополнительно выполнено исследование технологии NVIDIA MIG применительно к задачам массового расчета независимых сейсмических источников. Полученные результаты показывают, что для рассматриваемого класса задач, характеризующегося выраженным memory-bound поведением, разделение GPU на несколько независимых логических устройств позволяет повысить эффективность загрузки вычислительных ресурсов. Для эксперимента с расчетом 100000 источников на четырех графических ускорителях NVIDIA A100 использование MIG обеспечило ускорение вычислений примерно в 1.36 раза по сравнению с режимом без разделения GPU. Полученные результаты подтверждают перспективность применения технологий пакетной обработки, асинхронных вычислений и аппаратного разделения GPU для задач крупномасштабного сейсмического моделирования.

Благодарности

Работа выполнена в рамках государственного задания Минобрнауки России для ИМ СО РАН (тема №FWNF-2026-0018).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Baysal E., Kosloff D. D., Sherwood J. W. C. Reverse time migration // *Geophysics*. – 1983. – Vol. 48, No. 11. – P. 1514–1524.
2. Virieux J., Operto S. An overview of full-waveform inversion in exploration geophysics // *Geophysics*. – 2009. – Vol. 74, No. 6. – P. WCC1–WCC26.
3. Ostrander W. J. Plane-wave reflection coefficients for gas sands at nonnormal angles of incidence // *Geophysics*. – 1984. – Vol. 49, No. 10. – P. 1637–1648.
4. Micikevicius P. 3D finite difference computation on GPUs using CUDA // *Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units (GPGPU-2)*. – New York: Association for Computing Machinery, 2009. – P. 79–84.
5. Michéa D., Komatitsch D. Accelerating 3D seismic wave propagation using GPU computing // *SEG Technical Program Expanded Abstracts*. – 2010. – Vol. 29, No. 1. – P. 2898–2902.
6. Komatitsch D., Martin R. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation // *Geophysics*. – 2007. – Vol. 72, No. 5. – P. SM155–SM167.

© Д. С. Братчиков, Д. В. Рыжков, К. Г. Гадильшин, 2026