**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2010 Т. 2 № 3 С. 309–317**

Ки&М

МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ

УДК: 519.63

Построение высокопроизводительного вычислительного комплекса для моделирования задач газовой динамики

О. В. Геллер[a], М. О. Васильев[b], Я. А. Холодов[c]

Московский физико-технический институт,
Россия, 141700, Долгопрудный, пер. Институтский, 9

E-mail: [a] oleg@geller.su, [b] mick_vav@mail.ru, [c] kholodov@crec.mipt.ru

Получено 15 сентября 2010 г.

Целью исследований является разработка программного комплекса для решения задач газовой динамики в многосвязных областях правильной геометрии на высокопроизводительной вычислительной системе. Сравниваются различные технологии реализации параллельных вычислений. Программный комплекс реализован на многопоточных параллельных системах, использующих для организации расчета как многоядерную архитектуру, так и массивно-параллельную. Проведено сравнение численных результатов на основе программного комплекса с известными решениями модельных задач. Проведено исследование производительности различных вычислительных платформ.

Ключевые слова: метод С. К. Годунова, графическая карта, CUDA, OpenCL, OpenMP, GP GPU

Building a high-performance computing system for simulation of gas dynamics

O. V. Geller, M. O. Vasilev, Ya. A. Kholodov

Moscow Institute of Physics and Technology, 9 Institutskii per, Dolgoprudny, 141700, Russia

Abstract. — The aim of research is to develop software system for solving gas dynamic problem in multiply connected integration domains of regular shape by high-performance computing system. Comparison of the various technologies of parallel computing has been done. The program complex is implemented using multithreaded parallel systems to organize both multi-core and massively parallel calculation. The comparison of numerical results with known model problems solutions has been done. Research of performance of different computing platforms has been done.

Keywords: S. K. Godunov method, GPU, CUDA, OpenCL, OpenMP, GP GPU

Citation: *Computer Research and Modeling*, 2010, vol. 2, no. 3, pp. 309–317 (Russian).

Работа проведена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы и поддержана грантом РФФИ № 08-07-00429-а.

© 2010 Олег Владимирович Геллер, Михаил Олегович Васильев, Ярослав Александрович Холодов

Введение

Интерес к аэрозолям возник в конце XIX – начале XX века в связи с известными работами Р. Милликена по определению величины единичного электрического заряда и Дж. Вильсона по созданию ионизационной камеры, а также ученого-химика Дж. Гиббса (1896 г.). Большим толчком к развитию исследований физико-химических свойств аэрозолей послужило их использование в военных целях (в виде маскирующих дымов и отравляющих веществ). Физика аэрозолей начала развиваться, когда стало ясно, что исследования оптических явлений в атмосфере и облачных процессов не могут продолжаться дальше без понимания физической картины образования и трансформации аэрозолей.

Современное физическое исследование невозможно без численного моделирования исследуемой системы. Задача моделирования динамики аэрозольных примесей требует значительных вычислительных ресурсов, поэтому крайне важно наиболее полно использовать имеющиеся возможности вычислительной системы.

Для описания течения многокомпонентного газа [Нигматулин, 1987; Уоллис, 1972] использовалась модель односкоростной однотемпературной газовой динамики. Нахождение скорости и давления несущей фазы в рамках данной модели эквивалентно нахождению скорости и давления при трехмерном течении идеального газа. При этом считается, что движение частиц примесей происходит под действием давления несущей газовой фазы (т. е. парциальные давления фракций примеси совпадают с парциальным давлением несущей фазы). Кроме того, при расчете концентрации частиц примесей учитывается диффузия. Взаимодействием частиц примеси друг с другом в модели пренебрегается. Такое приближение справедливо лишь при малых объемных концентрациях примесей.

Необходимая скорость выполнения программ при численных расчетах достигалась за счет использования многопоточных параллельных вычислений. Выбор оптимальных с точки зрения эффективности методов и алгоритмов является одной из основных целей работы.

Математическая модель распространения примесей

Математическая модель

При построении модели в случае односкоростного, однотемпературного течения газа считается, что движение примесей происходит под действием давления несущей газовой фазы.

Решалась трехмерная задача газовой динамики с примесями в односкоростной однотемпературной постановке с диффузией:

$$\frac{\partial \rho_s}{\partial t} + \frac{\partial}{\partial x_i}(\rho_s v_i) = f_s^0,$$

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_i}\left(\delta_{ij} p + \rho v_i v_{ij}\right) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{3} \frac{\rho v_i^2}{2} + \rho\varepsilon\right) + \frac{\partial}{\partial x_i}\left(\rho v_i \left(\sum_{i=1}^{3} \frac{\rho v_i^2}{2} + \varepsilon + \frac{p}{\rho}\right)\right) = 0.$$

Система уравнений газовой динамики замыкалась с использованием следующих алгебраических соотношений:

$$\rho = \sum_{s=1}^{S} \rho_s,$$

$$p = (\gamma - 1)\,\rho_1 \varepsilon_1, \tag{2}$$

$$\varepsilon_1 = c_{V_1} T,$$

$$\varepsilon = \frac{T}{\rho} \sum_{s=1}^{S} \rho_s c_{V_s}.$$

Правая часть уравнений учитывает диффузионный пренос частиц примесей

$$f_1^0 = 0, \quad f_s^0 = \sum_{s=2}^{S} D_s \Delta \rho_s, \quad s = 2, \ldots,$$

здесь D_s — коэффициент диффузии s-ой компоненты примеси, $s \equiv 1$ для несущей компоненты. Заметим, что гиперболическая часть системы уравнений может быть сведена к системе уравнений газовой динамики с переменным показателем адиабаты. Действительно,

$$\frac{\partial \rho_s}{\partial t} + \frac{\partial}{\partial x_i}(\rho_s v_i) = 0 \Rightarrow \begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} \rho v_i = 0, \\ \frac{\partial (\rho_s/\rho)}{\partial t} + \frac{\partial}{\partial x_i}(\rho_s/\rho) = 0, \end{cases}$$

$$\rho = \sum_{s=1}^{S} \rho_s, \tag{3}$$

$$p = (\gamma - 1)\, \rho_1 \varepsilon_1 = (\gamma^* - 1)\, \rho \varepsilon,$$

$$\gamma = 1 - (\gamma - 1) \Bigg/ \left(1 + \frac{\sum_{s \neq 1}^{S} \rho_s c_{V_s}}{\rho_1 c_{V_1}} \right).$$

После этих подстановок гиперболическая часть системы уравнений сводится к системе уравнений обычной газовой динамики с переменным показателем адиабаты, дополненной независимыми уравнениями переноса компонент примесей. В результате получается, что решение задачи Римана о распаде произвольного разрыва для такой системы уравнений эквивалентно решению задачи Римана для среды с переменным показателем адиабаты. Это позволяет численно решать задачи данного класса при наличии сильных разрывов [Годунов, 1965].

Расщепление по физическим процессам

Расщепим рассматриваемую систему на две задачи. Первая — задача о течении газа — имеет гиперболический тип:

$$\frac{\partial \rho_s}{\partial t} + \frac{\partial}{\partial x_i}(\rho_s v_i) = 0,$$

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_i}\left(\delta_{ij} p + \rho v_i v_j \right) = 0, \tag{4}$$

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{3} \frac{\rho v_i^2}{2} + \rho \varepsilon \right) + \frac{\partial}{\partial x_i}\left(\rho v_i \left(\sum_{i=1}^{3} \frac{\rho v_i^2}{2} + \varepsilon + \frac{p}{\rho} \right) \right) = 0.$$

Вторая — учитывающая диссипативные процессы — параболическая:

$$\frac{\partial \rho_s}{\partial t} = \frac{\partial}{\partial x_i}\left(D_s \frac{\partial \rho_s}{\partial x_i} \right). \tag{5}$$

Каждая из этих систем по отдельности не описывает поставленную задачу. Но при численном решении последовательное применение разностных схем для каждой из этих подзадач дает приближенное решение полной задачи в соответствии с принципом суммарной аппроксимации.

Для решения первой задачи использован метод С. К. Годунова.

Методы численного решения (4–5) подробно описаны в [Холодов, 1980; Магомедов, Холодов, 1988; Воробьев, Холодов, 1996; Холодов, Холодов, 2006].

Используемые технологии и оборудование

При проведении численных расчетов были использованы различные технологии распараллеливания программы. При этом вычисления проводились как на центральных процессорах, так и на графических ускорителях. Проведено сравнение производительности различных технологий многопоточных параллельных вычислений.

NVIDIA CUDA™

CUDA (сокр. от англ. Compute Unified Device Architecture, дословно — унифицированная вычислительная архитектура устройств) — архитектура (совокупность программных и аппаратных средств), позволяющая производить на GPU вычисления общего назначения, при этом GPU фактически выступает в роли мощного сопроцессора [NVIDIA CUDA C Programming Guide].

CUDA дает разработчику возможность по своему усмотрению организовывать доступ к набору инструкций графического ускорителя и управлять его памятью, организовывать на нем сложные параллельные вычисления. Графический ускоритель с поддержкой CUDA становится мощной программируемой открытой архитектурой, подобно сегодняшним центральным процессорам. Всё это предоставляет в распоряжение разработчика низкоуровневый распределяемый высокоскоростной доступ к оборудованию, делая CUDA необходимой основой для построения серьезных высокоуровневых инструментов, таких как компиляторы, отладчики, математические библиотеки, программные платформы.

Технология NVIDIA CUDA не предлагает замену традиционному CPU и не оспаривает его первенства и главенства: GPU выступает в роли мощного сопроцессора, то есть помощника центрального процессора.

Технология NVIDIA CUDA поддерживается графическими процессорами ускорителей GeForce (начиная с восьмого поколения — GeForce 8 Series, GeForce 9 Series, GeForce 200 Series), Nvidia Quadro и Tesla.

KhronosGroup OpenCL™

OpenCL — стандарт разработки приложений для гетерогенных систем. OpenCL изначально задумывался как единый стандарт для написания приложений, которые должны исполняться в системе, где установлены процессоры, ускорители и платы расширения различной архитектуры [The OpenCL Specification].

Первая версия стандарта была опубликована в конце 2008 года и с тех пор уже успела претерпеть несколько ревизий.

Почти сразу после того, как стандарт был опубликован, компания NVidia заявила о поддержке OpenCL в рамках GPU Computing SDK поверх CUDA Driver API [developer.nvidia.com].

На текущий момент OpenCL поддерживается также AMD GPU, начиная с серии HD4000 и CPU архитектур x86 и x86–64 [http://developer.amd.com/].

OpenMP™

OpenMP (Open Multi-Processing) — набор директив компилятора, библиотечных процедур и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с единой памятью на языках C, C++ и Fortran.

Разработку спецификации OpenMP ведут несколько крупных производителей вычислительной техники и программного обеспечения, чья работа регулируется некоммерческой организацией OpenMP Architecture Review Board (ARB) [OpenMP Specifications].

OpenMP реализует параллельные вычисления на основе многопоточности, в которой «главная» (master) нить исполнения создает набор подчиненных (slave) нитей и задача распределяется между ними. Предполагается, что потоки выполняются параллельно на машине, имеющей несколько процессорных ядер с общим доступом к памяти. Ядра могут быть частями как одного, так и нескольких процессоров.

Задачи, выполняемые потоками параллельно, так же как и данные, требуемые для выполнения этих задач, описываются с помощью специальных директив препроцессора соответствующего языка — прагм.

Оборудование

Вычисления производились на вычислительном комплексе следующей конфигурации:

- 2 x CPU Intel Xeon X6550 (2,4 ГГц, 4 ядра, Hyper-Threading), 16 Гб DDR3,
- GPU NVIDIA Tesla C1060 (1,3 ГГц, 240 ядер, 4 Гб GDDR3),
- GPU NVIDIA Tesla C2050 (1,15 ГГц, 448 ядер, 3 Гб GDDR5),
- GPU NVIDIA GeForce GTX 480 (1,4 ГГц, 480 ядер, 1,5 Гб GDDR5).

На всех видеокартах производилось тестирование технологий CUDA и OpenCL, кроме того было выполнено тестирование производительности CPU с использованием технологий OpenCL и OpenMP.

Алгоритмы

Для технологий OpenCL и CUDA, работающих на видеокартах, была применена оптимизация работы с памятью. Память видеокарт неоднородна по скорости доступа. Для уменьшения количества обращений к глобальной памяти видеокарты используется кэширование ячеек сетки в разделяемую память видеокарты, при этом кэшируется сразу блок ячеек. Форма и размер блока зависят от модели используемой видеокарты и размеров вычислительной сетки.

Для проверки выполнения условия Куранта на каждом шаге по времени необходимо найти максимальную скорость звука по всем ячейкам, для этого применен модифицированный алгоритм редукционного суммирования [Боресков, Харламов, 2010].

Из-за большого сходства технологий OpenCL и CUDA появилась возможность адаптировать один и тот же код, исполняемый на видеокарте, для работы с обеими версиями программы, исполняемыми на хосте. Такой подход повышает читаемость кода и удобство разработки. Был разработан интерфейс, позволяющий легко переключаться между OpenCL и CUDA.

Тестовые расчеты

В качестве теста программы использовалась модельная задача [The Athena Code Test Page]. В прямоугольной области размером $0.3 \times 0.3 \times 0.3$ имеется выделенный регион ($0 \le x + y \le 0.15$, $0 \le z \le 0.3$).

Внутри региона задаются следующие начальные условия: плотность $\rho_{in} = 0.125$, давление $p_{in} = 0.14$.

Вне выделенного региона — плотность $\rho_{out} = 1$, давление $p_{out} = 1$.

Граничные условия области — жесткие стенки. Показатель адиабаты газа $\gamma = 1.4$, молярная масса $\mu = 0.029$.

В данной области строились двухмерные и трехмерные сетки с разным количеством ячеек. Вычисления проводились до достижения времени $t = 0.2$.

Результаты тестовых расчетов

В качестве эталонного решения использовалось решение [Боресков, Харламов, 2010]. На рис. 1 показано распределение плотности газа в начальный и конечный моменты времени.

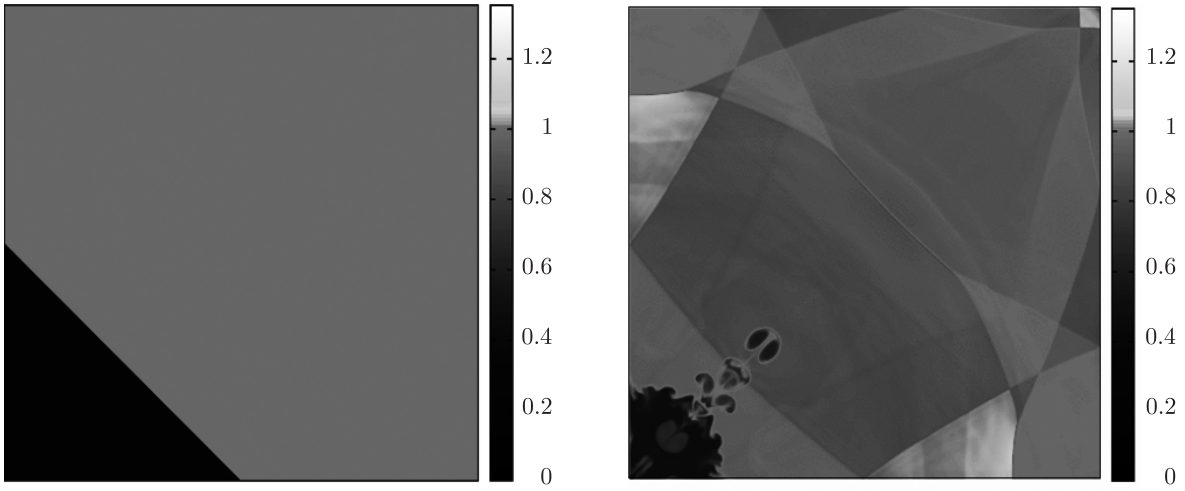

Рис. 1. Распределение плотности газа в моменты времени $t = 0$ и $t = 0.2$

Была получена зависимость времени исполнения от количества используемых нитей исполнения OpenMP (рис. 2). Прямой линией показано максимально возможное ускорение, пропорциональное числу нитей. Максимальное практическое ускорение для этой технологии на двух четырехядерных процессорах оказалось близко к 8. Этот результат получается при количестве нитей исполнения больше 13, что вполне объяснимо наличием на используемом процессоре технологии Hyper-Treading.

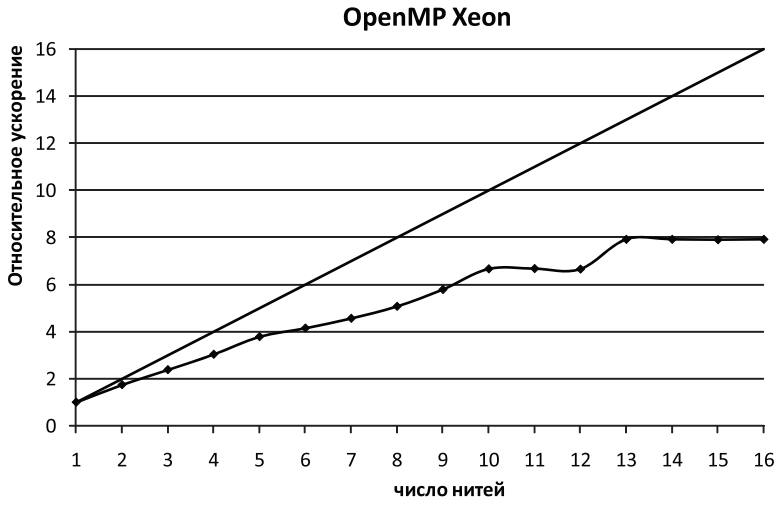

Рис. 2. Зависимость относительного ускорения от числа нитей исполнения OpenMP

Также были получены зависимости скорости счета от размера и формы сетки (рис. 3–4, таблица 1). За точку отсчета была взята скорость счета по технологии OpenMP с одной нитью.

Таблица 1. Описание клеточных автоматов типа «Жизнь»

Размер сетки, ячеек	Относительное ускорение							
	CUDA	OpenCL	CUDA	OpenCL	CUDA	OpenCL	OpenCL	OpenMP
	GTX480	GTX480	C2050	C2050	C1060	C1060	CPU	CPU
Двухмерные сетки								
625	1,42	1,23	1,08	1,02	0,66	0,59	32,71	8,00
1250	1,90	1,67	1,49	1,35	0,95	0,85	29,44	8,00
2500	2,16	1,97	1,76	1,59	1,12	1,01	22,79	8,00
5000	3,81	3,41	3,05	2,73	1,93	1,72	19,05	8,00
10000	5,62	4,98	4,48	3,98	2,85	2,55	15,02	8,00
20000	9,43	8,37	7,44	6,65	4,70	4,23	13,13	8,00
40000	20,71	18,81	15,97	14,44	10,03	9,07	13,48	8,00
80000	37,99	33,92	28,27	25,55	15,28	13,22	12,65	8,00
160000	63,14	58,84	47,47	43,96	20,85	16,38	14,35	8,00
320000	74,94	68,88	54,53	50,10	21,39	16,83	13,50	8,00
640000	69,94	64,99	49,45	46,09	20,62	16,78	12,17	8,00
1280000	79,83	72,27	54,59	49,61	21,88	17,32	11,59	8,00
2560000	77,12	70,11	51,78	47,38	20,41	16,93	11,01	8,00
Трехмерные сетки								
15625	12,51	10,95	10,10	8,76	5,36	4,96	19,83	8,00
120000	46,77	41,30	36,07	31,91	12,18	10,19	10,06	8,00
125000	70,18	62,51	53,34	48,79	21,39	13,94	14,36	8,00
240000	63,75	55,96	45,92	40,36	16,97	12,85	10,72	8,00
480000	67,34	59,78	49,77	43,95	17,19	13,15	9,64	8,00
1000000	72,21	63,80	51,22	45,06	16,77	13,00	9,04	8,00
1000000	74,06	65,62	51,70	45,31	18,19	14,67	9,75	8,00
2000000	73,84	65,27	52,07	45,92	17,29	13,30	8,99	8,00
4000000	74,22	65,51	52,32	46,03	17,17	13,26	8,83	8,00
8000000	74,67	65,93	52,60	46,31	17,20	13,32	8,80	8,00

Из приведенных результатов тестовых расчетов следует, что применять видеокарты для решения данной задачи имеет смысл при сетках с большим числом ячеек. Также можно заметить, что скорость расчета зависит не только от размера сетки, но и от способа разбиения расчетной области на ячейки (при одинаково полном числе ячеек). Скачки в производительности, представленные на рисунке 4, связаны с тем, как распределялись ячейки разностной сетки по осям координат. Эта зависимость объясняется функционированием алгоритма кэширования из глобальной в разделяемую память видеокарты.

Важным результатом является сравнение скорости работы OpenMP и OpenCL на одном и том же центральном процессоре. OpenCL на всех тестах показал более высокую производительность, причем в некоторых случаях — более чем 16-кратное ускорение. Объясняется это использованием более полного набора команд процессора. Реализация OpenCL использует расширенный набор команд SSE3, в то время как OpenMP компилируется под набор команд 80386.

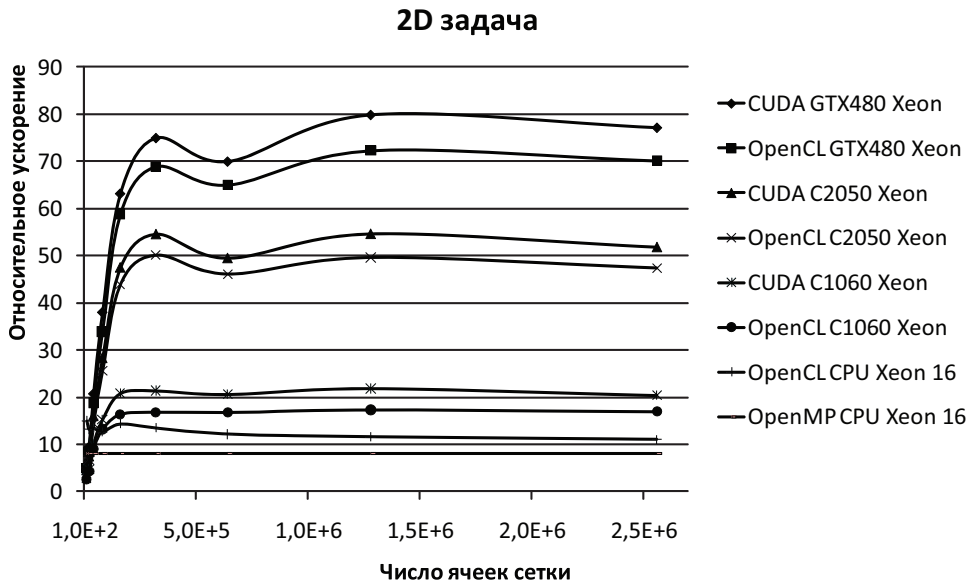

Рис. 3. Зависимость относительной скорости счета от размера двумерной сетки

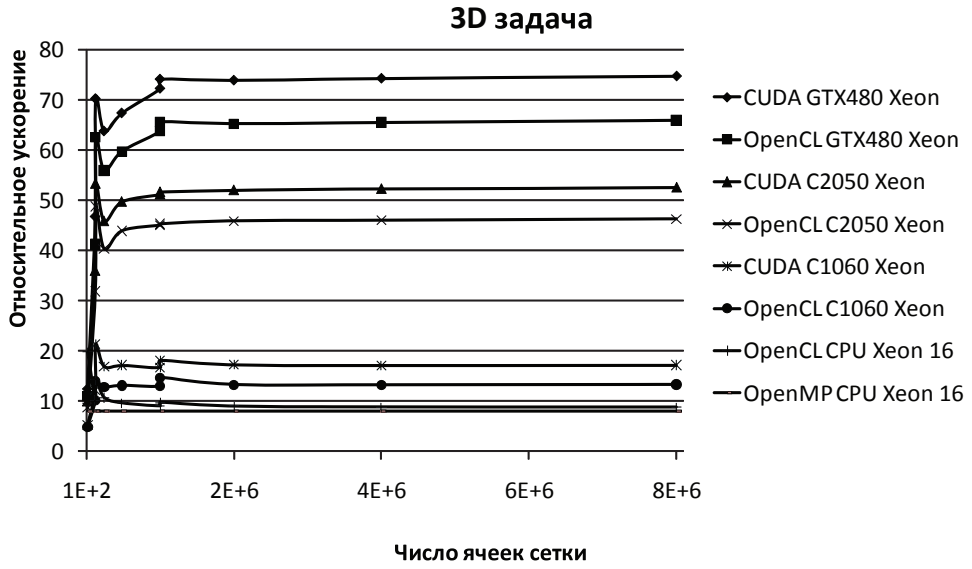

Рис. 4. Зависимость относительной скорости счета от размера трехмерной сетки

Заключение

Показана принципиальная возможность использования графических карт для решения задачи численного моделирования многокомпонентных газовых потоков. При этом достигнуто значительное ускорение. Следует отметить, что метод С. К. Годунова не является идеальным с точки зрения распараллеливания на видеокарте, так как алгоритм решения задачи Римана о распаде разрыва содержит большое количество ветвлений, что может снижать производительность работы видеокарты.

Реализован общий интерфейс, позволяющий выполнять расчеты на различных гетерогенных платформах, содержащих как центральные, так и графические процессоры. При этом возможно выбирать технологию решения задачи исходя из конфигурации установленного оборудования.

Список литературы

Боресков А. В., Харламов А. В. Основы работы с технологией CUDA. — М.: ДМК Пресс, 2010. — 232 с.

Воробьев О. В., Холодов Я. А Об одном методе численного интегрирования одномерных задач газовой динамики // Математическое моделирование. — 1996. — Т. 8, № 1. — С. 77–92.

Годунов С. К. Разностный метод численного расчета разрывных решений уравнений гидродинамики // Мат. сб. — 1959. — Т. 47(89), № 3. — С. 271–306.

Нигматулин Р. И. Динамика многофазных сред, часть I. — М.: Наука, 1987. — 464 с.

Магомедов К. М., Холодов А. С. Сеточно-характеристические численные методы. — М.: Наука, 1988. — 287 с.

Уоллис Г. Одномерные двухфазные течения. — М.: Мир, 1972. — 220 с.

Холодов А. С. О построении разностных схем повышенного порядка точности для уравнений гиперболического типа // Журнал выч. математики и мат. физики. — 1980. — Т. 20, № 6. — С. 1601–1620.

Холодов А. С., Холодов Я. А. О критериях монотонности разностных схем для уравнений гиперболического типа // Журнал выч. математики и мат. физики. — 2006. — Т. 46, № 9. — С. 1560–1588.

NVIDIA Corporation NVIDIA CUDA C Programming Guide Version 3.2 — 2010.

Khronos OpenCL Working Group The OpenCL Specification Version: 1.1 Revision: 36 — 2010.

OpenMP Architecture Review Board OpenMP Specifications Version: 3.0 — 2008.

OpenCL Zone — 2010 AMD Corporation, 2010
`http://developer.amd.com/zones/OpenCLZone/Pages/default.aspx`

OpenCL — 2010 NVIDIA Corporation, 2010.
`http://www.developer.nvidia.com/object/opencl.html`

The Athena Code Test Page. — Edgewall Software.
`http://www.astro.princeton.edu/~jstone/Athena/tests/`