

Сравнение эффективности применения различных математических постановок для суперкомпьютерного 3D моделирования сейсмических волновых полей*

А.Ф. Сапегина

Институт вычислительной математики и математической геофизики СО РАН

Моделирование распространения сейсмических волн в средах, характерных для вулканических структур, требует разработки суперкомпьютерных технологий для сокращения времени расчетов и получения возможности моделирования «больших» 3D упругих сред в целях создания систем вибросейсмического мониторинга сейсмоопасных объектов. В данной работе, в рамках применения методики со-дизайна, проводится анализ эффективности параллельной реализации двух трехмерных постановок динамической теории упругости. Проведено сравнительное исследование времени работы, масштабируемости и количества используемой памяти программного обеспечения, разработанного для обоих подходов и оптимизированного под архитектуру суперкомпьютера, оснащенного GPU.

Ключевые слова: упругие волны, 3D моделирование, со-дизайн, разностные схемы, гибридный кластер, GPU.

1. Введение

Разработка эффективного программного инструментария для современных суперкомпьютерных систем отдельная сложная научная задача. Ее решение становится все более зависимым от архитектуры суперкомпьютера, подчиняя себе не только выбор алгоритмов для решения поставленной задачи под вычислительную архитектуру, но требующее со-дизайна алгоритмов на всех стадиях решения задачи – от постановки до выбора инструментов разработки. Понятие со-дизайна в контексте математического моделирования физических процессов понимается как построение физико-математической модели явления, численного метода, параллельного алгоритма и его программной реализации, эффективно использующей архитектуру суперкомпьютера. Со-дизайн успешно применяется при разработке программного обеспечения для моделирования различных физических процессов на суперкомпьютерах. Например, при решении задач физики плазмы [1], астрофизики [1,2] и многих других. При таком подходе актуальным становится проведение сравнения эффективности использования различных физических и математических постановок при разработке программного обеспечения.

В данной работе проводится сравнение эффективности параллельной реализации двух трехмерных алгоритмов для гибридного суперкомпьютера, оснащенного графическими ускорителями. Исследуемые алгоритмы осуществляют численное моделирование сейсмических волновых полей в 3D упругих средах, характерных для вулканических структур, на основе разных постановок динамической теории упругости. Первый алгоритм решает задачу, поставленную в терминах скоростей смещения и напряжения, а второй решает эту же задачу, но записанную в терминах смещений.

Моделирование сейсмических волновых полей в 3D упругих средах может быть актуальным для создания систем вибросейсмического мониторинга вулканических структур. Создание таких систем требует проведения предварительных и сопутствующих комплексных исследований конкретных вулканов. Обычно рельеф исследуемого объекта достаточно сложен и не позволяет расположить на нем площадную систему наблюдения для решения обратной задачи геофизики, поэтому приходится решать прямую задачу, варьируя геометрию и подбирая параметры моделируемой среды так, чтобы результаты численного и натурального экспериментов были близки.

* Работа поддержана проектами РФФИ 16-07-00434, 16-29-15120 и 16-01-00455.

Сложность и масштаб моделируемой области диктуют необходимость создания высокоэффективного программного обеспечения, которое позволяло бы достаточно быстро решать прямую задачу геофизики для «больших» 3D упругих сред. Заметим, что быстроедействие особенно актуально для проведения сопутствующего моделирования при осуществлении мониторинга и прогноза извержения реального вулкана. Таким образом, необходимо создание программного инструментария, использующего суперкомпьютерные мощности и позволяющего достаточно быстро осуществить перебор предполагаемых данных о строении вулканической структуры, для интерпретации натурального эксперимента.

Для исследования эффективности алгоритмов, с точки зрения используемой памяти и быстрогодействия на суперкомпьютерных системах с графическими ускорителями, нами рассмотрены две постановки задачи динамической теории упругости, описанные в следующем разделе. На их основе разработано оптимизированное под архитектуру GPU и гибридного кластера в целом программное обеспечение и проведено сравнение времени работы полученных реализаций.

2. Описание математических постановок задачи динамической теории упругости

Численное моделирование распространения сейсмических волн в сложно построенных упругих неоднородных средах можно проводить на основе решения полной системы уравнений теории упругости с соответствующими начальными и граничными условиями, записанной в терминах вектора скоростей смещений $\vec{u} = (u, v, w)^T$ и тензора напряжений $\vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$. В качестве области моделирования выступает изотропная 3D неоднородная сложно построенная упругая среда, представляющая собой параллелепипед, одна из граней которого является свободной поверхностью (плоскость $z = 0$). Рассмотрим прямоугольную декартову систему координат так, что ось Oz направлена вертикально вниз, а оси Ox и Oy лежат на свободной поверхности. Уравнения в векторной форме могут быть представлены в следующем виде:

$$\rho \frac{\partial \vec{u}}{\partial t} = [A] \vec{\sigma} + \vec{F}(t, x, y, z), \quad (1)$$

$$\frac{\partial \vec{\sigma}}{\partial t} = [B] \vec{u}, \quad (2)$$

$$A = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial z} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}, \quad B = \begin{bmatrix} (\lambda + 2\mu) \frac{\partial}{\partial x} & \lambda \frac{\partial}{\partial y} & \lambda \frac{\partial}{\partial z} \\ \lambda \frac{\partial}{\partial x} & (\lambda + 2\mu) \frac{\partial}{\partial y} & \lambda \frac{\partial}{\partial z} \\ \lambda \frac{\partial}{\partial x} & \lambda \frac{\partial}{\partial y} & (\lambda + 2\mu) \frac{\partial}{\partial z} \\ \mu \frac{\partial}{\partial y} & \mu \frac{\partial}{\partial x} & 0 \\ \mu \frac{\partial}{\partial z} & 0 & \mu \frac{\partial}{\partial x} \\ 0 & \mu \frac{\partial}{\partial z} & \mu \frac{\partial}{\partial y} \end{bmatrix},$$

где t — время, $\rho(x, y, z)$ — плотность, $\lambda(x, y, z)$, $\mu(x, y, z)$ — параметры Ламе.

Предполагается, что параметры упругой среды зависят от трех пространственных переменных X , Y и Z .

Начальные условия имеют вид:

$$\begin{aligned} \sigma_{xz}|_{t=0} = 0, \quad \sigma_{yz}|_{t=0} = 0, \quad \sigma_{xy}|_{t=0} = 0, \quad \sigma_{xx}|_{t=0} = 0, \quad \sigma_{yy}|_{t=0} = 0, \quad \sigma_{zz}|_{t=0} = 0, \\ u(x, y, z)|_{t=0} = 0, \quad v(x, y, z)|_{t=0} = 0, \quad w(x, y, z)|_{t=0} = 0. \end{aligned} \quad (3)$$

В качестве граничных условий на свободной поверхности выступают:

$$\sigma_{xz}|_{z=0} = 0, \sigma_{yz}|_{z=0} = 0, \sigma_{zz}|_{z=0} = 0. \quad (4)$$

При численном моделировании на основе решения системы (1–4) на каждом шаге по времени требуется хранить в памяти ЭВМ как минимум 12 трехмерных массивов содержащих информацию о неизвестных $\vec{u} = (u, v, w)^T$, $\vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$ и о параметрах среды ρ , λ , μ . При моделировании «больших» 3D сред в таком случае для проведения расчета требуется большой объем оперативной памяти.

Это количество можно сократить, перейдя к постановке динамической теории упругости в терминах вектора смещений $\vec{U} = (U, V, W)^T$ и уменьшив, таким образом, количество неизвестных. В этом случае для той же моделируемой области основные уравнения будут выглядеть следующим образом:

$$\rho \frac{\partial^2 \vec{U}}{\partial t^2} = [C] \vec{U} + \vec{F}(t, x, y, z), \quad (5)$$

$$C = \begin{bmatrix} (\lambda + 2\mu) \frac{\partial^2}{\partial x^2} + \mu \left(\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) & (\lambda + \mu) \frac{\partial^2}{\partial x \partial y} & (\lambda + \mu) \frac{\partial^2}{\partial x \partial z} \\ (\lambda + \mu) \frac{\partial^2}{\partial y \partial x} & (\lambda + 2\mu) \frac{\partial^2}{\partial y^2} + \mu \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right) & (\lambda + \mu) \frac{\partial^2}{\partial y \partial z} \\ (\lambda + \mu) \frac{\partial^2}{\partial z \partial x} & (\lambda + \mu) \frac{\partial^2}{\partial z \partial y} & (\lambda + 2\mu) \frac{\partial^2}{\partial z^2} + \mu \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \end{bmatrix}$$

В обеих постановках предполагается, что правая часть (массовая сила) имеет следующий вид: $\vec{F}(t, x, y, z) = F_x \vec{i} + F_y \vec{j} + F_z \vec{k}$, где \vec{i} , \vec{j} , \vec{k} — единичные направляющие векторы координатных осей.

3. Метод решения задачи

В случае трехмерных динамических задач теории упругости наиболее «гибким» и широко применяемым является конечно-разностный метод. Для решения задачи (1–4) в данной работе используется хорошо себя зарекомендовавшая конечно-разностная схема на сдвинутых сетках, наиболее подробно описанная в статьях [3–5]. Расчет сеточных коэффициентов ρ , λ , μ , которые могут иметь разрывы, проводится на основе интегральных законов сохранения. Используемая конечно-разностная схема имеет второй порядок аппроксимации по времени и пространству [3]. Рассматриваются только равномерные сетки. Для примера представим несколько конечно-разностных уравнений используемой схемы:

$$\frac{\rho_{i,j,k} + \rho_{i-1,j,k}}{2} \frac{u_{i-\frac{1}{2},j,k}^{n+1} - u_{i-\frac{1}{2},j,k}^n}{\tau} = \frac{(\sigma_{xxi,j,k}^{n+\frac{1}{2}} - \sigma_{xxi-1,j,k}^{n+\frac{1}{2}})}{h} + \frac{(\sigma_{xyi-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \sigma_{xyi-\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}})}{h} +$$

$$+ \frac{(\sigma_{xzi-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \sigma_{xzi-\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}})}{h} + f_{xi,j,k}^n,$$

$$\frac{(\sigma_{xzi-\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} - \sigma_{xzi-\frac{1}{2},j,k-\frac{1}{2}}^{n-\frac{1}{2}})}{\tau} = \text{Ml}_{i-\frac{1}{2},j,k-\frac{1}{2}} \left(\frac{u_{i-\frac{1}{2},j,k}^n - u_{i-\frac{1}{2},j,k-1}^n}{h} + \frac{w_{i,j,k-\frac{1}{2}}^n - w_{i-1,j,k-\frac{1}{2}}^n}{h} \right)$$

где $\text{Ml}_{i-\frac{1}{2},j,k-\frac{1}{2}} = \left(\frac{1}{4} \left(\frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i-1,j,k}} + \frac{1}{\mu_{i,j,k-1}} + \frac{1}{\mu_{i-1,j,k-1}} \right) \right)^{-1}$.

Для решения задачи в терминах смещений используется аналогичная конечно-разностная схема на сдвинутых сетках. Конечно-разностное уравнение для нахождения компоненты U выглядит следующим образом:

$$\begin{aligned} & \frac{\rho_{i-1,j,k} + \rho_{i,j,k}}{2} \frac{U^{n+1}_{i-\frac{1}{2},j,k} - 2U^n_{i-\frac{1}{2},j,k} + U^{n-1}_{i-\frac{1}{2},j,k}}{\tau^2} = (\lambda_{i,j,k} + 2\mu_{i,j,k}) \frac{U^n_{i+\frac{1}{2},j,k} - U^n_{i-\frac{1}{2},j,k}}{h^2} + \\ & + \lambda_{i,j,k} \left(\frac{V^n_{i,j+\frac{1}{2},k} - V^n_{i,j-\frac{1}{2},k}}{h^2} + \frac{W^n_{i,j,k+\frac{1}{2}} - W^n_{i,j,k-\frac{1}{2}}}{h^2} \right) - (\lambda_{i-1,j,k} + 2\mu_{i-1,j,k}) \frac{U^n_{i-\frac{1}{2},j,k} - U^n_{i-\frac{3}{2},j,k}}{h^2} - \\ & - \lambda_{i-1,j,k} \left(\frac{V^n_{i-1,j+\frac{1}{2},k} - V^n_{i-1,j-\frac{1}{2},k}}{h^2} + \frac{W^n_{i-1,j,k+\frac{1}{2}} - W^n_{i-1,j,k-\frac{1}{2}}}{h^2} \right) + \\ & + M2_{i-\frac{1}{2},j+\frac{1}{2},k} \left(\frac{U^n_{i-\frac{1}{2},j+1,k} - U^n_{i-\frac{1}{2},j,k}}{h^2} + \frac{V^n_{i,j+\frac{1}{2},k} - V^n_{i-1,j+\frac{1}{2},k}}{h^2} \right) - \\ & - M2_{i-\frac{1}{2},j-\frac{1}{2},k} \left(\frac{U^n_{i-\frac{1}{2},j,k} - U^n_{i-\frac{1}{2},j-1,k}}{h^2} + \frac{V^n_{i,j-\frac{1}{2},k} - V^n_{i-1,j-\frac{1}{2},k}}{h^2} \right) + \\ & + M1_{i-\frac{1}{2},j,k+\frac{1}{2}} \left(\frac{U^n_{i-\frac{1}{2},j,k+1} - U^n_{i-\frac{1}{2},j,k}}{h^2} + \frac{W^n_{i,j,k-\frac{1}{2}} - W^n_{i-1,j,k-\frac{1}{2}}}{h^2} \right) - \\ & - M1_{i-\frac{1}{2},j,k-\frac{1}{2}} \left(\frac{U^n_{i-\frac{1}{2},j,k} - U^n_{i-\frac{1}{2},j,k-1}}{h^2} + \frac{W^n_{i,j,k-\frac{1}{2}} - W^n_{i-1,j,k-\frac{1}{2}}}{h^2} \right), \\ \text{где } M2_{i-\frac{1}{2},j-\frac{1}{2},k} & = \left(\frac{1}{4} \left(\frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i-1,j,k}} + \frac{1}{\mu_{i,j-1,k}} + \frac{1}{\mu_{i-1,j-1,k}} \right) \right)^{-1}. \end{aligned}$$

Отметим, что количество операций для расчета неизвестных на одном шаге по времени для этой конечно-разностной схемы больше чем для предыдущей, что должно сказаться на скорости расчета.

4. Параллельная реализация и оптимизации

Современные суперкомпьютеры все чаще оснащены ускорителями. Этот тренд в том числе задается лидерами списка TOP500. Разработка программного кода для таких гибридных систем требует дополнительных знаний и времени, но позволяет получить значительный прирост производительности. Графические ускорители хорошо подходят для решения конечно-разностных уравнений, благодаря своей массивно-параллельной архитектуре и быстрому доступу к памяти устройства. Поэтому в данной работе разработка программ велась с учетом особенностей архитектуры гибридного кластера с графическими ускорителями НКС-30Т+GPU ССКЦ ИВМиМГ СО РАН (<http://www2.sssc.ru>). Он состоит из 40 вычислительных узлов HP SL390s G7, каждый узел содержит два 6-ядерных CPU Xeon X5670 и три карты NVIDIA Tesla M2090 на архитектуре Fermi, у каждой 1 GPU с 512 ядрами и 6 ГБ оперативной памяти GDDR5. Суммарно НКС-30Т+GPU содержит 80 процессоров (480 ядер) CPU и 120 процессоров (61440 ядер) GPU. Пиковая производительность 85 Тфлопс. Программы написаны на языке программирования C++ с использованием технологий CUDA (Compute Unified Device Architecture) и MPI (Message

Passing Interface), которые дают возможность задействовать одновременно большое количество параллельных процессов и в итоге получить максимальную эффективность.

Для эффективного использования гибридной архитектуры необходимо распараллеливание и оптимизация используемых в ходе моделирования алгоритмов, основанная на знании архитектуры кластера и его составляющих и необходимых программных средств. Так как методы решения сравниваемых постановок задачи теории упругости имеют сходную природу, а также для корректности сравнения, адаптация и оптимизация этих методов проводится одинаковыми путями.

Для распараллеливания используется декомпозиция расчетной области на слои вдоль направления одной из координатных осей (рис. 1). Каждый слой рассчитывается на отдельном узле, где, в свою очередь, он разбивается еще на подслои вдоль другой координатной оси (для обеспечения возможности лучшего масштабирования) по числу графических ускорителей на узле.

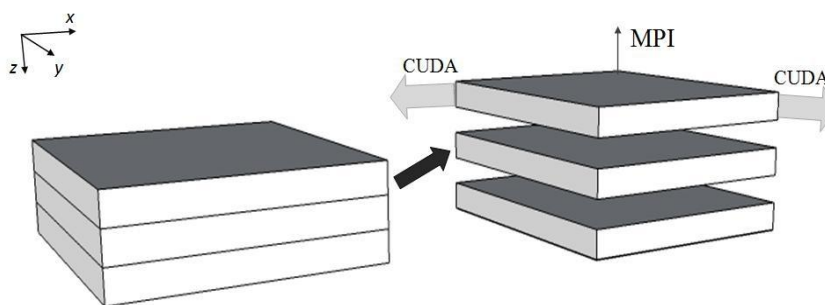


Рис. 1. Декомпозиция расчетной области

При такой реализации каждая графическая карта рассчитывает свою сеточную область внутри подслоя на каждом временном шаге независимо от других, за исключением точек, находящихся на границе между двумя соседними областями. Эти точки являются общими для каждой из областей и для продолжения счета необходимо производить обмен информацией об искомых величинах между «соседями». Отметим, что размер данных для обмена одинаков в обоих подходах. Для обмена между вычислительными узлами используется асинхронная неблокированная передача данных с помощью соответствующих функций MPI. Обмен между графическими картами производится средствами CUDA.

Эффективное использование графических ускорителей требует оптимизации работы с памятью. Для уменьшения времени доступа к глобальной памяти производится одинаковое оптимальное расположение всех используемых трехмерных массивов и соответствующее распределение работ между нитями. Специально выделены и сохранены в константную память графической карты все основные константы, используемые на каждом шаге по времени. Не смотря на то, что решаемая задача оперирует трехмерными массивами, можно попытаться использовать более быструю разделяемую память графической карты [6], для уменьшения количества чтений из глобальной памяти GPU. Эта оптимизация будет выполнена в дальнейшем и предположительно принесет большее ускорение для расчета смещений, так как в нем используется меньшее количество данных, соответственно потребуется меньше копирований в разделяемую память.

В ходе оптимизации исследована зависимость быстродействия программного обеспечения от размерности и размера блока нитей, задаваемых для осуществления вычислений на GPU, как для расчета скоростей смещения и напряжения, так и для смещений. В итоге расчетная область на каждом GPU разбивается на трехмерную сетку блоков, размер которых по компоненте x должен быть кратен длине $warp$ 'а (количество физически одновременно исполняющихся нитей на GPU). Конкретные размеры блока подбираются эмпирически для каждого алгоритма. Результаты некоторых измерений указаны в табл. 1. Эта не сложная, с точки зрения изменения кода, оптимизация позволяет ускорить работу программы в разы при эффективном использовании глобальной памяти графического ускорителя. Возможно, полученные выводы можно обобщить на реализацию всех конечно-разностных методов.

Также отметим, что в обеих реализациях в памяти GPU хранятся не исходные сеточные коэффициенты $\lambda_{i,j,k}$, $\mu_{i,j,k}$ и $\rho_{i,j,k}$, а их модификации, используемые в расчетной схеме на каждом шаге по времени, для устранения повтора одних и тех же вычислений. Таким образом, реализация для расчета скоростей смещения и напряжения требует выделения памяти для хранения 26 трехмерных массивов, а реализация для расчета смещений всего 14 массивов. Это позволяет уменьшить количество используемой памяти практически в два раза, что очень актуально для расчетов на вычислительных комплексах, содержащих малое количество узлов.

Таблица 1. Зависимость быстродействия программного обеспечения от размера блока нитей

Размер блока, кол-во нитей по соответствующему направлению $x*y*z$	Время расчета скоростей смещения и напряжения, с (сетка 500*500*600 узлов по пространству и 1000 шагов по времени)	Время расчета смещений, с (сетка 600*600*600 узлов по пространству и 1000 шагов по времени)
4*4*4	789,4	
8*8*8	506,3	422,8
2*16*2	1429,2	
16*2*2	352,7	
16*4*4	358,5	317,3
4*32*4		629,2
32*2*2	290,1	
32*4*4	290,9	287,5
64*2*2	272,9	260,1
64*4*2	273,8	
128*2*2	264,0	258,4
128*1*1	277,2	258,1
256*1*1	269,9	257,7
512*1*1	300,1	256,9

5. Сравнение эффективности параллельных реализаций

Для анализа эффективности параллельной реализации двух трехмерных постановок динамической теории упругости исследована масштабируемость разработанного программного обеспечения и проведено сравнение времени расчетов на одинаковых по размеру и параметрам средах.

Под слабой масштабируемостью будем понимать сохранение времени счета одного шага одного и того же объема задачи при одновременном увеличении количества графических карт. Результаты исследований приведены на рис. 2, 3 в виде графиков. Под эффективностью в данном случае понимается отношение времени расчета на n узлах в n раз большей задачи к времени расчета на одном узле исходной задачи. Как при расчете смещений, так и при расчете скоростей смещения и напряжения эффективность падает до уровня 87% и слабо колеблется рядом с этим значением при увеличении количества графических карт до 45 (15 узлов кластера НКС-30Т+GPU). Сходное поведение времени расчета для обоих алгоритмов объясняется одинаковым количеством передаваемых данных между вычислительными узлами ($3N^2$ для задачи N^3).

Для сравнения быстродействия обоих подходов проведены расчеты одинаковых сред с размером сетки 1500*700*2100 узлов по пространству и 1000 шагов по времени. Взята сетка близкая к максимально входящей в память 45 графических карт (15 вычислительных узлов кластера НКС-30Т+GPU) при расчете скоростей смещения и напряжения. При расчете смещений на такой же по размеру сетке используется практически в два раза меньше оперативной памяти, и он может быть проведен минимум на 8 узлах вместо 15. Результаты измерений представлены в табл. 2. Расчеты смещений и скоростей смещения и напряжения на одинаковом количестве узлов занимают примерно одинаковое количество времени (расчет смещений проходит чуть быстрее). При уменьшении количества узлов примерно в 2 раза с сохранением размера расчет-

ной сетки, что возможно только для расчета смещений, время расчета увеличивается примерно в 1,4 раза.

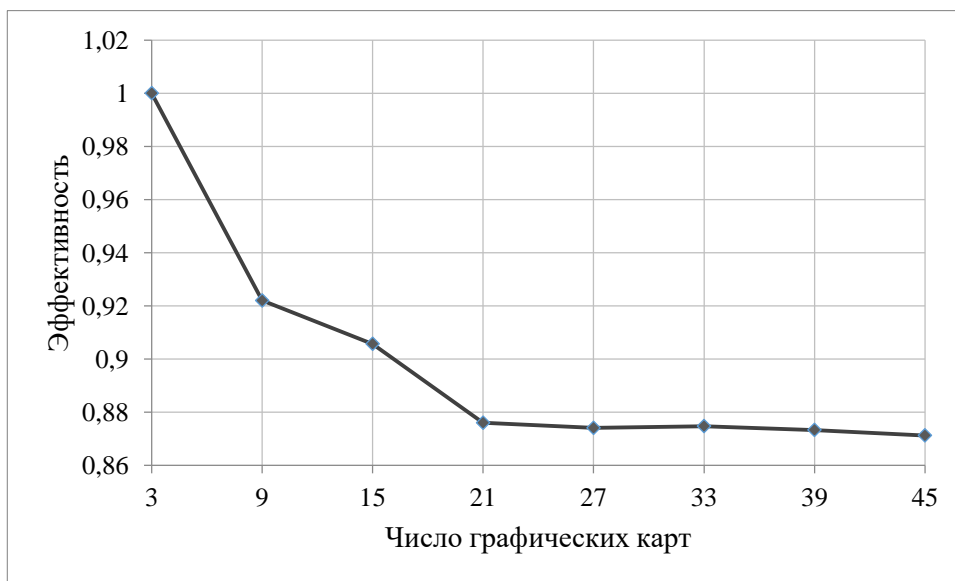


Рис. 2. График исследования масштабируемости для алгоритма расчета смещений

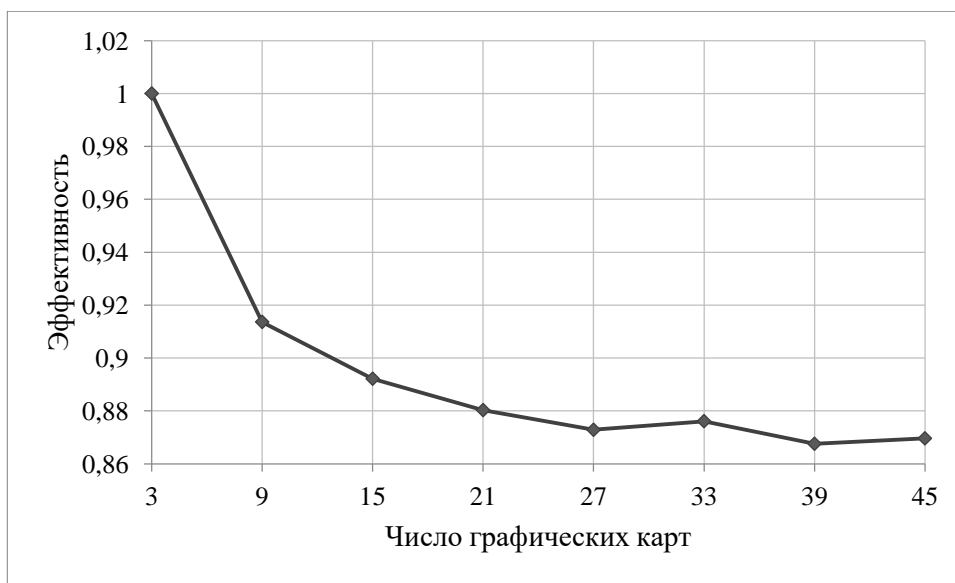


Рис. 3. График исследования масштабируемости для алгоритма расчета скоростей смещения и напряжения

Таблица 2. Сравнение быстродействия алгоритмов

	Кол-во узлов	Время, с
Расчет скоростей смещения и напряжения	15	183,1
Расчет смещений	15	174,8
Расчет смещений	8	247,4

Проведенные эксперименты демонстрируют, что подход, основанный на расчете смещений, быстрее и при этом позволяет проводить расчеты для очень больших сеток, запрашивая меньшее количество свободных узлов. Это позволяет быстрее получать доступ к ним в условиях очереди на кластере, обеспечивая при этом приемлемое время расчета (несколько часов для реальной полномасштабной задачи).

Отметим, что для уменьшения требуемых графических ускорителей для расчета скоростей смещения и напряжения возможно создание программной реализации с хранением промежуточных расчетов в памяти узла. Это, соответственно, потребует осуществлять множественное копирование подмассивов с искомыми неизвестными и параметрами среды из памяти узла в память ускорителя и назад на каждом шаге по времени, что приведет к существенно большему увеличению времени расчета, чем использование постановки в смещениях.

Для иллюстрации работоспособности разработанного программного обеспечения проведено численное моделирование распространения упругих волн для приближенной модели вулкана Эльбрус (рис.4). С описанием геофизической модели вулкана и результатами численных экспериментов можно подробнее познакомиться в работах [7,8].

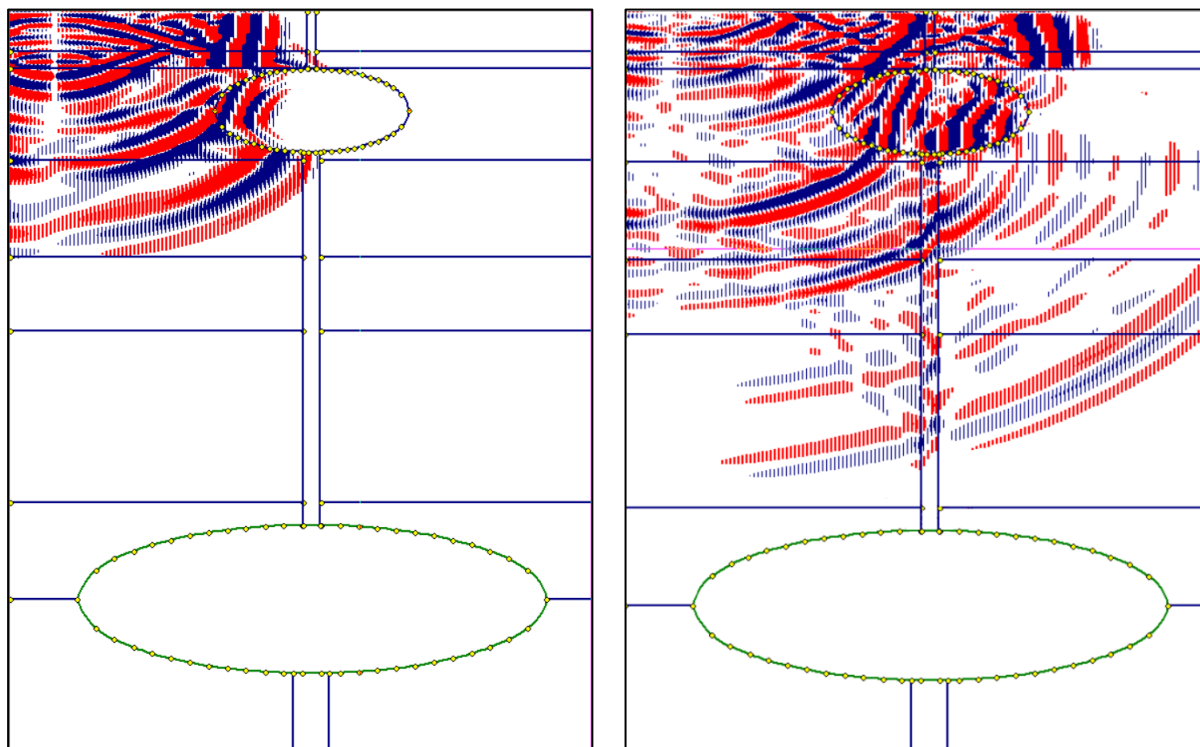


Рис. 4. Результаты численного моделирования для приближенной модели вулкана Эльбрус. На снимках волнового поля представлена компонента u вектора скоростей смещений в плоскости Oxz в разные моменты времени.

6. Заключение

В рамках методики со-дизайна проведено сравнение разработанных эффективных параллельных реализаций решения задачи динамической теории упругости, записанной в терминах скоростей смещения и напряжения и в терминах смещений для гибридного суперкомпьютера, оснащенного графическими ускорителями. Исследовано время работы созданных параллельных программ и их масштабируемость.

На основе полученных результатов можно дать рекомендации по предпочтению в использовании подхода, основанного на расчете смещений. Хотя он дает небольшой выигрыш во времени по сравнению с подходом, основанном на расчете скоростей смещения и напряжения, но позволяет решать большие 3D задачи динамической теории упругости на существенно меньшем числе графических ускорителей за приемлемое время.

Литература

1. Glinskiy B.M., Kulikov I.M., Snytnikov A.V., Romanenko A.A., Chernykh I.G., Vshivkov V.A. Co-design of Parallel Numerical Methods for Plasma Physics and Astrophysics // Supercomputing Frontiers and Innovations. 2014. Vol. 1. No 3. P. 88–98.

2. Kulikov I.M. GPUPEGAS: a new GPU-accelerated hydrodynamic code for numerical simulations of interacting Galaxies // *The Astrophysical Journal Supplement Series*. 2014. Vol. 214, 12. 12 p.
3. Bihn M., Weiland T. A Stable Discretization Scheme for the Simulation of Elastic Waves. *Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics (IMACS 1997)*. Berlin, 1997. Vol. 2. P. 75–80.
4. Караваев Д.А. Параллельная реализация метода численного моделирования волновых полей в трехмерных моделях неоднородных сред // *Вестник Нижегородского университета им. Н.И. Лобачевского*. 2009. № 6(1). С. 203–209.
5. Глинский Б.М., Караваев Д.А., Ковалевский В.В., Мартынов В.Н. Численное моделирование и экспериментальные исследования грязевого вулкана «Гора Карabetова» вибросейсмическими методами // *Вычислительные методы и программирование*. 2010. Т. 11. С. 95–104.
6. Norimitsu Nakata, Takeshi Tsuji, Toshifumi Matsuoka. Acceleration of computation speed for elastic wave simulation using a Graphic Processing Unit // *Exploration Geophysics*. 2011. Vol. 42(1). P. 98–104.
7. Глинский Б.М., Мартынов В.Н., Сапетина А.Ф. Технология суперкомпьютерного 3D моделирования сейсмических волновых полей в сложно построенных средах // *Вестник Южно-Уральского Государственного Университета*. Серия: Вычислительная математика и информатика. 2015. Т. 4, № 4. С. 101–116.
8. Глинский Б.М., Мартынов В.Н., Сапетина А.Ф. 3-D моделирование сейсмических волновых полей в средах, характерных для вулканических структур // *Математические заметки СВФУ*. 2015. Т. 22, № 3(87). С. 84–98.

Comparing efficiency of using different mathematical statements of a problem for 3D simulation of seismic wave field at supercomputer

A.F. Sapetina

Institute of Computational Mathematics and Mathematical Geophysics SB RAS

Simulation of seismic wave propagation in media typical of volcanic structures needs developing of supercomputing technologies for reduction of the computation time and simulation “big” 3D elastic media in order to create systems of vibroseis monitoring for quake-prone objects. Analysis of efficiency of two parallel implementations solving a 3D elastodynamic problem written in different terms is carried out under co-design. Software is developed for the both approaches and optimized for architecture of a supercomputer equipped with GPU. Comparative study of the software computation time, a size of memory usage and scalability is realized.

Keywords: elastic waves, 3D simulation, co-design, finite difference schemes, hybrid cluster, GPU.

References

1. Glinskiy B.M., Kulikov I.M., Snytnikov A.V., Romanenko A.A., Chernykh I.G., Vshivkov V.A. Co-design of Parallel Numerical Methods for Plasma Physics and Astrophysics // Supercomputing Frontiers and Innovations. 2014. Vol. 1. No 3. P. 88–98.
2. Kulikov I.M. GPUPEGAS: a new GPU-accelerated hydrodynamic code for numerical simulations of interacting Galaxies // The Astrophysical Journal Supplement Series. 2014. Vol. 214, 12. 12 p.
3. Bihn M., Weiland T.A. Stable Discretization Scheme for the Simulation of Elastic Waves. Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics (IMACS 1997). Berlin, 1997. Vol. 2. P. 75–80.
4. Karavaev D.A. Parallel'naya realizatsiya metoda chislennogo modelirovaniya volnovykh poley v trekhmernykh modelyakh neodnorodnykh sred [Parallel implementation of wave field numerical modeling method in 3D models of inhomogeneous media]. Vestnik Nizhegorodskogo universiteta imeni N.I. Lobachevskogo [Vestnik of Lobachevsky State University of Nizhni Novgorod]. 2009. No. 6(1). P. 203–209.
5. Glinskiy B.M., Karavaev D.A., Kovalevskiy V.V., Martynov V.N. Chislennoe modelirovanie i eksperimental'nye issledovaniya gryazevogo vulkana «Gora Karabetova» vibroseismicheskimi metodami [Numerical modeling and experimental research of the «Karabetov Mountain» mud volcano by vibroseismic methods]. Vychislitel'nye metody i programmirovaniye [Numerical Methods and Programming]. 2010. Vol. 11. P. 95–104.
6. Norimitsu Nakata, Takeshi Tsuji, Toshifumi Matsuoka. Acceleration of computation speed for elastic wave simulation using a Graphic Processing Unit // Exploration Geophysics. 2011. Vol. 42. P. 98–104.
7. Glinskiy B.M., Martynov V.N., Sapetina A.F. Tehnologiya superkomp'yuternogo 3D modelirovaniya sejsmicheskikh volnovykh poley v slozhno postroennykh sredah [Technology of supercomputer simulation of seismic wave fields in complicated media]. Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika [Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering]. 2015. No. 4, Vol. 4. P. 101–116.

8. Glinskiy B.M., Martynov V.N., Sapetina A.F. 3-D modelirovanie sejsmicheskikh volnovykh polej v sredah, harakternykh dlja vulkanicheskikh struktur [3D Modeling of Seismic Wave Fields in a Medium Specific to Volcanic Structures]. *Matematicheskie zametki SVFU [Mathematical notes of NEFU]*. 2015. No. 22. Vol 3(87). P. 84–98.