

Программный комплекс для моделирования сейсмических и геофизических процессов с применением современных технологий для высокопроизводительных вычислений.

Хохлов Н. И.¹ and Петров И. Б.¹

¹Московский физико-технический институт

В работе рассматривается применение различных современных технологий высокопроизводительных вычислений для распараллеливания программного комплекса, предназначенного для решения задач сейсмологии и геофизики. Рассматриваются технологии как для центральных процессоров (CPU), так и для графических процессоров (GPU). Приведены сравнительные результаты применения технологий MPI, OpenMP, CUDA, OpenCL. Тестирование проводилось на платформах AMD и Intel для центральных процессоров и платформах AMD и nVidia для графических процессоров. В качестве примеров работы разработанного программного комплекса приводится ряд примеров расчета задач сейсмологии, сейсмостойкости и геофизики. Комплекс рассчитан на работу используя двумерные и трехмерные структурные блочные сетки с наличием неоднородностей. Для численного интегрирования применяются сеточно-характеристические и конечно-объемные методы 2-4 порядка точности.

Код распараллелен используя различные современные технологии для высокопроизводительных вычислительных систем. В настоящее время достигнута эффективность распараллеливания до 70 % используя технологию MPI при масштабировании до 16 тысяч вычислительных ядер. В системах с общей памятью алгоритм распараллелен используя технологию OpenMP. Также код распараллелен используя технологию CUDA, что дает ускорение до 50 раз по сравнению с одним ядром CPU. Программа может использовать несколько карточек в рамках одного узла. Для графических процессоров отличных от семейства карточек NVidia, код распараллелен используя технологию OpenCL. Что дает ускорение до 50 раз на графических ускорителях от AMD.

В данной работе рассмотрены результаты одного и того же алгоритма используя различные технологии. Приведены тесты распараллеливания до 16 тысяч ядер CPU и восьми устройств CUDA.

Сформулируем основные уравнения линейной динамической теории упругости, которым подчиняется состояние бесконечно малого объема линейно-упругой среды. Рассмотрим нестационарные уравнения теории упругости для случая трех переменных, в некоторой ортонормированной системе координат (x_1, x_2, x_3) :

$$\begin{aligned}\rho \dot{v}_i &= \nabla_j \sigma_{ij}, \\ \dot{\sigma}_{ij} &= q_{ijkl} \dot{\epsilon}_{kl} + F_{ij},\end{aligned}\tag{1}$$

здесь ρ – плотность среды, v_i – компоненты вектора скорости смещения, σ_{ij} и ϵ_{ij} – компоненты тензоров напряжений Коши и деформации, ∇_j – ковариантная производная по j -й координате, F_{ij} – добавочная правая часть. Вид компонент тензора четвертого порядка q_{ijkl} определяется реологией среды. Для линейно-упругого случая они имеют вид:

$$q_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}).$$

В этом соотношении, которое обобщает закон Гука, λ и μ – параметры Ламе, а δ_{ij} – символ Кронекера.

Алгоритм распараллелен в системах с общей памятью используя технологию OpenMP. Результат распараллеливания финальной реализации приведен на рис. 1.

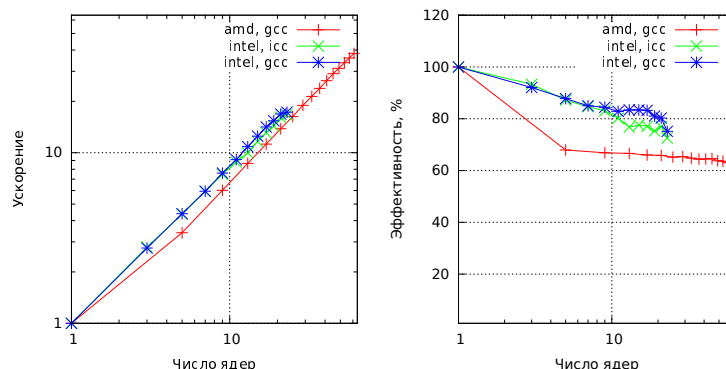


Рис. 1: Графики ускорение и эффективности от числа ядер для OpenMP реализации

Для работы в системах с распределенной памятью программный комплекс распараллелен используя технологию MPI. При распараллеливании применялся классический алгоритм для явных сеточных методов, основанный на принципе геометрического параллелизма. Расчетная сетка разделялась между процессами на возможно более равные части с перекрытием равным половине ширины шаблона разностной схемы. На каждом шаге происходит обмен данными из приграничных ячеек. Обмер организован используя функции MPI Isend/Irecv.

Одно из основных требований к алгоритму – работа на большом числе вычислительных ядер (тысячи) для обеспечения приемлемого времени расчета на больших задачах. На рис. 2 приведены результаты тестирования ускорения и эффективности алгоритма при увеличении числа расчетных ядер от 128 до 16384.

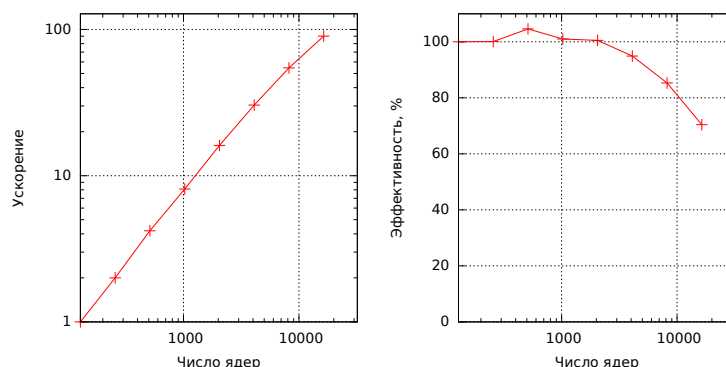


Рис. 2: Графики ускорения и эффективности MPI реализации

Алгоритм также был распараллелен на графических GPGPU процессорах NVidia используя технологию CUDA. Потребовалось полное переписывание части расчетного модуля под данную архитектуру. Однако это дало существенное ускорение работы алгоритма, по сравнению с CPU версией. Так применение данной технологии позволило получить ускорение до 44 раз по сравнению с одним ядром CPU Intel Xeon E5-2697. Использование нескольких устройств GPU позволяет получить дополнительное ускорение.

Помимо технологии CUDA алгоритм также был распараллелен используя технологию OpenCL, что позволило использовать для ускорения расчетов графические ускорители от AMD.

Результат ускорения по сравнению с одним ядром CPU приведен на рис. 3.

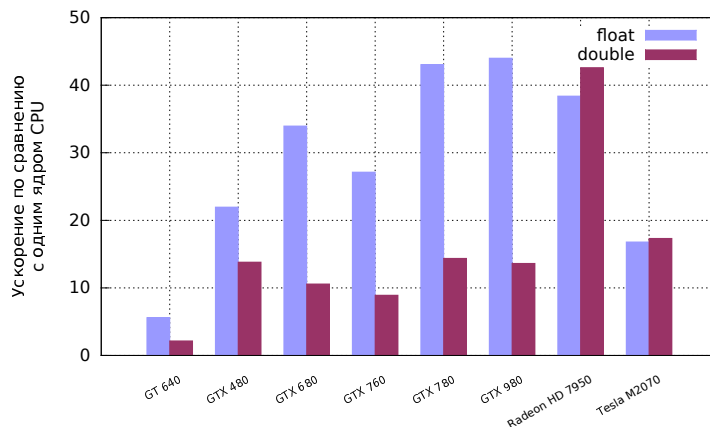


Рис. 3: Ускорение на графических устройствах

При этом для устройства от AMD (Radeon HD 7950) реализация была на OpenCL, для других графических ускорителей на CUDA. Разница скорости работы на устройствах от NVidia реализация на CUDA и OpenCL незначительна. На устройствах для настольных компьютеров от NVidia ускорение значительно падает при переходе от float к double, это связано с особенностью архитектуры. На карте Tesla M2070 такого резкого падения в скорости работы не наблюдается. Следует заметить, что графический ускоритель для настольных компьютеров от AMD Radeon HD 7950 лишен этого недостатка и дает примерно одинаковое ускорение на числах с двойной и одинарной точностью.

Результат ускорения работы на нескольких GPU устройствах приведен на рис. 4.

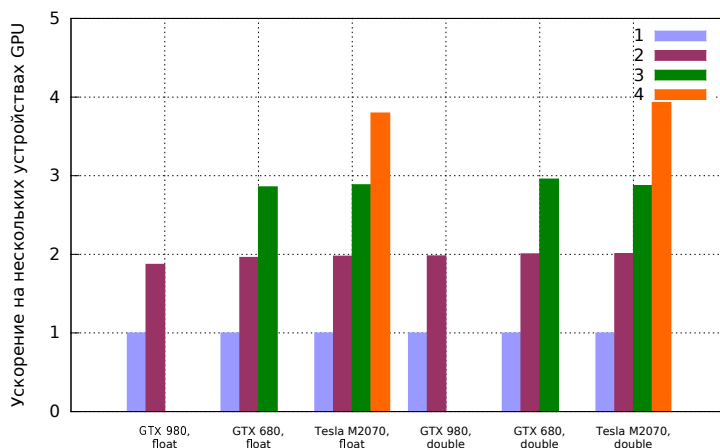


Рис. 4: Ускорение на нескольких графических устройствах

Для обмена данными между устройствами используется технология CUDA 6 обмена данными минуя память хоста. Это позволило получить существенное ускорение на нескольких устройствах, по сравнению с работой на одном. Кроме того, работа на нескольких устройствах позволяет считать большие задачи, за счет распределения расчетной сетки между устройствами.

В работе представлен результат применения широкого круга современных технологий написания параллельных приложений под различные архитектуры для задач моделирования процессов сейсмологии с применением сеточно-характеристического и конечно-объемных методов. Полученные результаты говорят о эффективной реализации алгоритма под различные архитектуры. Распараллеливание на большое число ядер позволяет решать задачи, которые раньше решить было либо очень проблематично, либо невозможно из-за недостатка вычислительных ресурсов. В дальнейшем планируется перенести рабочий алгоритм на архитектуру MIC от Intel (Xeon Phi) и гибридный параллелизм GPU+MPI.