

Моделирование волновых процессов в задачах сейсморазведки на суперкомпьютере «Ломоносов» с использованием GPU *

А.П. Бурцев¹, Н.Н. Попова¹, Е.А. Курин²

Московский государственный университет им. М.В. Ломоносова¹, ООО «ГЕОЛАБ-ИТ»²

В данной работе рассматриваются вопросы, посвящённые моделированию распространения акустических волн в гетерогенной среде на системах, оснащённых графическими ускорителями, на примере одной из задач сейсморазведки. Приводятся описание и результаты исследования эффективности параллельного алгоритма моделирования распространения акустических волн в неоднородной среде, использующего графические ускорители, для суперкомпьютера «Ломоносов».

Ил.: 3 рис. Библиогр.: 2 назв.

1. Введение

В настоящее время различные высокопроизводительные системы используются во многих областях. Одной из таких сфер применения является сейсморазведка. Данная область сталкивается с необходимостью обработки больших объёмов данных. Сейчас геологи собирают сотни гигабайт данных результатов сейсморазведки даже с небольших площадей, а требуется исследовать сотни и сотни квадратных километров. Чтобы максимально ускорить процесс анализа полученных данных необходимо использование супервычислителей: многоядерных машин, кластеров, грид-систем, а в процессах обработки, моделирования и визуализации как исходных, так и преобразованных данных требуется использование различных параллельных алгоритмов. Одной из важнейших задач сейсморазведки является задача моделирования волнового поля. Эта задача возникает как на этапе построения (уточнения) модели среды, как одна из частей задачи, так и на этапе верификации полученных результатов. Для моделирования поля давления, в акустическом приближении, необходимо: определить область моделирования, задать модель среды - скорость распространения волны и плотность в каждой точке расчётной области, определить положения источников возмущения волнового поля и функцию их воздействия, выбрать особенности поведения волны на границах области. Результатом моделирования являются значения давления в каждой точке расчётной области в каждый момент времени моделирования.

2. Описание алгоритма

В работе [1] рассматривалось применение различных подходов к разработке параллельных алгоритмов моделирования распространения акустических волн в однородной среде. В данной работе рассматривается усовершенствованная (относительно предыдущей работы) математическая модель описывающая распространения акустических волн в неоднородной среде.

Решается следующее волновое уравнение [2]:
$$\frac{1}{\rho \cdot v^2} \frac{\partial^2 p}{\partial t^2} = \operatorname{div} \left(\frac{1}{\rho} \operatorname{grad} (p) \right) + S, \quad c$$
 начальными условиями — $p|_{t=t_0} = p_0$ и граничными — $p|_{\Gamma=\Omega} = p_{\Gamma}$, где: ρ — плотность среды. v — скорость акустической волны. p — давление волны в момент

* Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации (Государственный контракт № 07.514.12.4007) и РФФИ (гранты №11-07-00756 и №11-07-00614).

времени t , S — внешняя сила (функция источника). Рассматривается двумерная задача. При решении используется явная конечно-разностная схема 4-ого порядка по пространственным координатам и 2-ого по времени. На границе среды реализованы отражающие граничные условия — элементы граничного слоя всегда не возмущены (поле давлений равно 0). Результирующая формула давления в момент времени $t+1$:

$$p(x, z, t+1) = \rho v^2 \left(\Delta p(x, z, t) + S(x, z, t) + \left\langle \text{grad}\left(\frac{1}{\rho}\right), \text{grad}(p) \right\rangle \right) + \Delta t^2 + 2p(x, z, t) - p(x, z, t-1)$$

Параллельный алгоритм состоит из следующих шагов:

1. Контроль входных данных и распределение нагрузки между процессами.
2. Расчёт нового состояния поля давлений (цикл по времени):
 1. Копирование готовых данных.
 2. Расчёт новых значений поля давлений.
 3. Передача «теневого» граничных областей соседним процессам.

Все процессы могут выполнять свою работу параллельно, но на этапе обмена «теньевыми» границами требуется синхронизация. Копирование готовых данных на жёсткий диск может осуществляться параллельно с проведением расчётов, но в данной работе этот вариант не рассматривался.

Для распределения данных между MPI-процессами используется простая геометрическая декомпозиция данных: каждый процесс вычисляет значения поля давлений в своей прямоугольной области. Нагрузка распределяется равномерно между всеми процессами. Данные соседних процессов пересекаются, т. е. часть данных, используемых одним процессом, вычисляется другим. На Рис. 1 приведён пример для 4 процессоров и граничного (общего) слоя из одного элемента.

Технология CUDA используется на уровне узла для проведения процедуры расчёта независимо по каждой точке расчётной области. В CPU-версии программы этот уровень параллелизма реализован с использованием директив OpenMP.

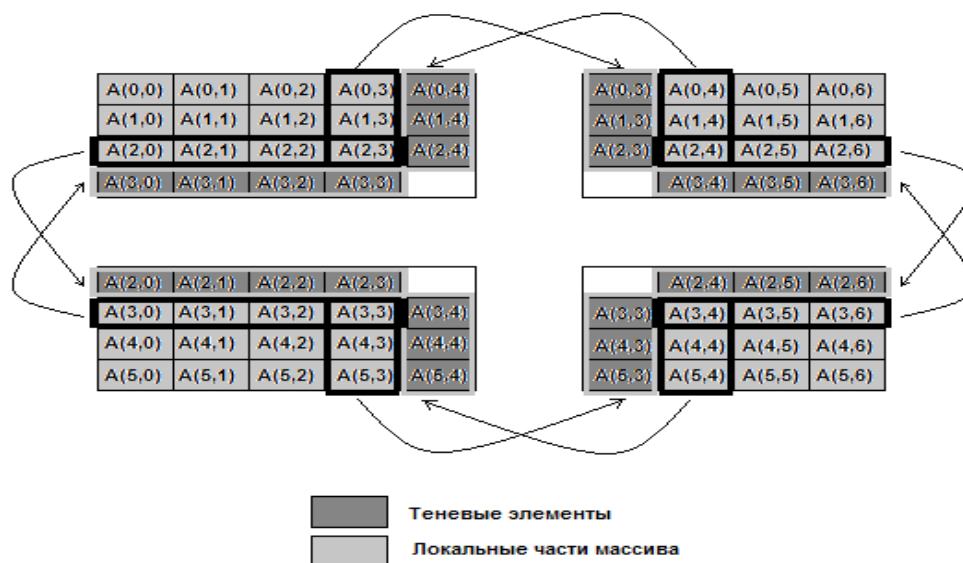


Рис. 1. Схема распределения данных для четырёх процессоров.

3. Вычислительный эксперимент

Тесты на суперкомпьютере «Ломоносов» разбиты по числу используемых ресурсов на две группы: одна вычислительная стойка (1-16 MPI-процессов), несколько стоек (64-256 MPI-процессов). При выполнении на один процессорный узел назначался один MPI-процесс, таким образом каждый процесс имеет выделенный графический ускоритель. Для первой группы размер входных данных не менялся с увеличением числа MPI-процессов. Для второй группы размер входных увеличивался пропорционально росту числа MPI-процессов.

На Рис. 2 изображены графики зависимости времени выполнения программы от числа процессов для данных фиксированного размера (8000x8000x100 точек) и двух различных версий программы: MPI+CUDA (для графических процессоров), MPI+OpenMP. Из графиков видно, что время выполнения GPU-версии меньше чем у CPU-версии для 1-2 процессов, сравнивается при 4 и становится большим при 8-16. Это можно объяснить возникновением коммуникационных издержек и снижением нагрузки на один графический сопроцессор, что существенно снижает производительность программы.

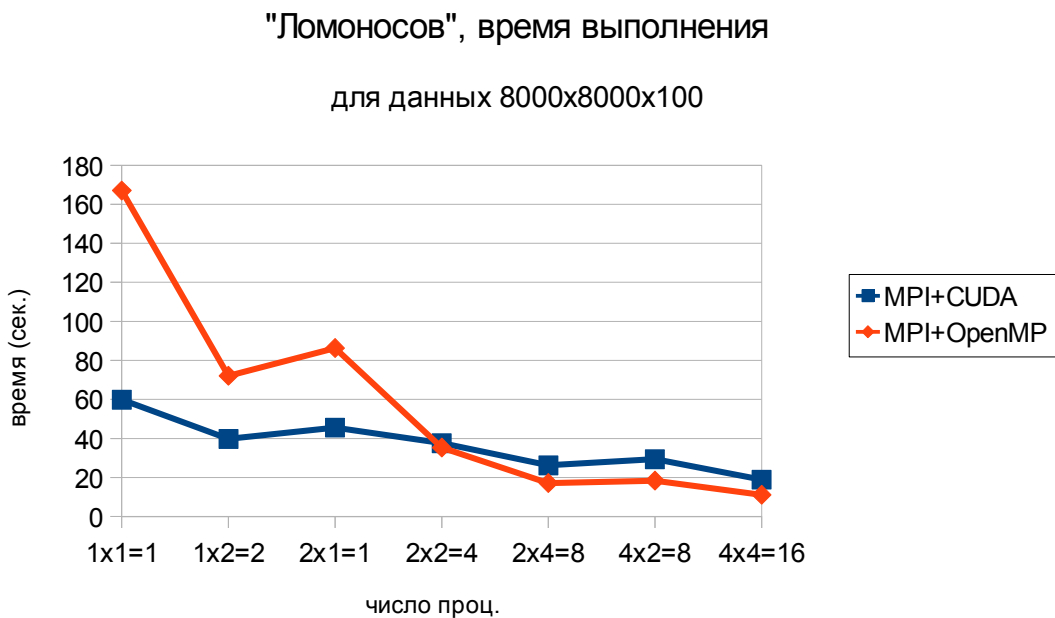


Рис. 2. Сравнение CPU и GPU версий программы для одной процессорной стойки.

На Рис. 3 изображён график зависимости времени выполнения программы от числа процессов для фиксированной нагрузки на один процесс 4000x4000x100 точек. Из него видно, что версия, использующая графические ускорители, обладает сравнительно лучшей динамикой (с увеличением числа процессов, время выполнения замедляется медленнее), чем версия, использующая только ресурсы центрального процессора, т. к. время выполнения GPU-версии растёт медленнее, чем CPU-версии.

"Ломоносов", время выполнения

для данных 4000x4000x100

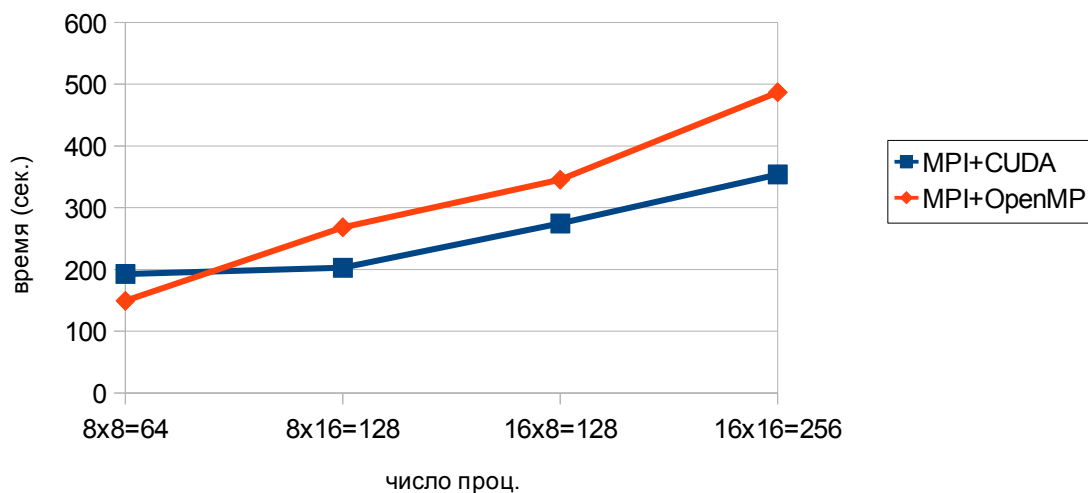


Рис. 3. Сравнение CPU и GPU версий программы для нескольких процессорных стоек.

4. Заключение

В качестве выводов можно отметить, что использование графических ускорителей на кластерах для решения поставленной задачи оправдано, если объём вычислительной нагрузки на одну видеокарту достаточно велик. В рамках дальнейшего исследования планируется:

1. Перейти к более точной физической (и математической) модели, что существенно повышает точность расчётов за счёт использования уравнений теории упругости.
2. Понизить влияние коммуникационных расходов за счёт совмещения процесса обмена данными с процессом расчёта, что можно достичь за счёт специальной декомпозиции данных.
3. Разработать версию программы, использующую сопроцессоры Intel MIC, т. к. это ещё один вариант сопроцессоров который может повысить эффективность расчётов.

Литература

1. Бурцев А. П. Исследование различных подходов к разработке параллельных алгоритмов моделирования распространения акустических волн для задач сейсморазведки. // Программные системы и инструменты. Тематический сборник № 13. / А.П. Бурцев, Е. А. Курин. - М.: МГУ, 2012. сс. 56-61.
2. Тихонов А.Н. Уравнения математической физики. / А.Н. Тихонов, А.А. Самарский. - 7-е изд. - М.: Изд-во МГУ: Наука, 2004. 742 с.