

Раздел III. Использование суперЭВМ в математическом моделировании

УДК 519.684.4

Д.Н. Морозов, Б.Н. Четверушкин, Н.Г. Чурбанова, М.А. Трапезникова

МОДЕЛИРОВАНИЕ ЗАДАЧ ФИЛЬТРАЦИИ НА ГИБРИДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Рассматриваются проблемы использования гибридных вычислительных систем сверхвысокой производительности для решения задач математической физики. Описывается модульный комплекс программ для моделирования процессов многофазной фильтрации слабосжимаемых жидкостей (учитываются капиллярные и гравитационные силы), позволяющий задействовать весь потенциал многоядерных суперкомпьютеров, содержащих графические ускорители вычислений. На примере тестовых расчетов задач просачивания и нефтедобычи показано, что логическая простота предложенных вычислительных алгоритмов и надлежащая программная реализация обеспечивают высокое ускорение вычислений.

Многоядерная архитектура; графические ускорители; течение в пористой среде; двухфазная жидкость; модульное программирование.

D.N. Morozov, B.N. Chetverushkin, N.G. Churbanova, M.A. Trapeznikova

SIMULATION OF FILTRATION PROBLEMS ON HYBRID COMPUTER SYSTEMS

In the paper the challenge of using high-performance hybrid computer systems is considered when solving problems of mathematical physics. A modular program system for multiphase slightly compressible fluid filtration process simulation (the capillary and gravity forces are taken into account) is described for involving the entire potential of multicore supercomputers with graphics accelerators. By means of test predictions for infiltration and oil recovery problems it is shown that the logical simplicity of proposed computational algorithms and the competent software implementation provide high speed-up of computations.

Multicore architecture; graphics accelerators; flow in a porous medium; two-phase fluid; modular programming.

Введение. Гибридные вычислительные системы (ГВС) предоставляют уникальные вычислительные возможности для моделирования задач математической физики, позволяют использовать расчетные сетки с количеством узлов, необходимым для проведения высокоточных расчетов. При этом разработка программ для ГВС – самостоятельная, достаточно трудоемкая задача. Большинство классических алгоритмов малоэффективны, поэтому их приходится модифицировать с учетом гибридной архитектуры. Кроме того, средства разработки программ для ГВС, такие как технология CUDA, появились относительно недавно, находятся в стадии совершенствования и не имеют достаточно широкого набора библиотек в отличие от программного обеспечения для классических кластеров. Потому особенно актуальными становятся логически простые и легко модифицируемые для различных архитектур алгоритмы, например, основанные на явных разностных схемах.

Совместное использование для расчётов в гибридных кластерах CPU и GPU предъявляет новые требования к эффективной реализации параллельных алгоритмов. Кроме необходимости минимизации обменов между узлами кластера, добавляются требования минимизации обменов между CPU и GPU и эффективного использования памяти.

В данной работе на примере задач двухфазной фильтрации рассматриваются вопросы использования гибридных вычислительных систем в математическом моделировании.

1. Математическая модель и вычислительный алгоритм. В [1, 2] предложен новый подход к моделированию течений слабосжимаемых жидкостей в пористых средах, преимуществом которого является численная реализация с помощью явных конечно-разностных схем. В качестве математической модели фильтрации использовалась система уравнений, построенная по аналогии с кинетически-согласованными разностными схемами и квазигазодинамической системой [3]. От традиционной модели [3] она отличается модифицированным уравнением неразрывности, полученным при помощи метода дифференциальных приближений и принципа минимальных размеров [4, 5] из классического уравнения. Модифицированное уравнение содержит дополнительные вторые производные по времени и пространству, умноженные на малые параметры, имеющие смысл соответствующих минимальных характерных величин [3,6]:

$$m \frac{\partial(\rho_\alpha S_\alpha)}{\partial t} + \tau \frac{\partial^2(\rho_\alpha S_\alpha)}{\partial t^2} + \operatorname{div}(\rho_\alpha \mathbf{u}_\alpha) = q_\alpha + \operatorname{div} m \frac{l_\alpha c_\alpha}{2} \operatorname{grad}(\rho_\alpha S_\alpha). \quad (1)$$

В систему уравнений входит обобщенный закон Дарси для каждой фазы:

$$\mathbf{u}_\alpha = -K \frac{k_\alpha(S_w)}{\mu_\alpha} (\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}), \quad (2)$$

уравнение состояния:

$$\rho_\alpha = \rho_{0\alpha} [1 + \beta_\alpha (p_\alpha - p_{0\alpha})], \quad (3)$$

а также соотношения между насыщенностями и давлениями фаз:

$$S_w + S_n = 1, \quad p_n - p_w = p_c(S_w). \quad (4)$$

Здесь α – индекс, обозначающий фазу (w – вода, n – NAPL, от англ. Non-Aqueous Phase Liquid); ρ – плотность флюида; p – давление; S – насыщенность; \mathbf{u} – скорость фильтрации; K и $k(S_w)$ – абсолютная и относительная проницаемости; β – коэффициент сжимаемости жидкости; m – пористость; μ – динамическая вязкость; \mathbf{g} – ускорение свободного падения; q – источник жидкости; $p_c(S_w)$ – капиллярное давление; p_0 и ρ_0 – характерные значения давления и плотности; l – минимальная характерная длина; τ – минимальное характерное время; c – величина порядка скорости звука во флюиде.

Алгоритм состоит:

- ◆ из вычисления плотностей и скоростей фаз на текущем временном шаге из уравнений (3) и (2);
- ◆ вычисления $\rho_\alpha S_\alpha$ для обеих фаз на следующем временном шаге с помощью явной схемы для уравнений (1);
- ◆ вычисления давлений и насыщенностей на следующем временном шаге при помощи полученной в каждой расчетной точке из уравнений (1)–(4) системы уравнений:

$$\begin{cases} \rho_{0w} \left[1 + \beta_w (p_w^{N+1} - p_{0w}) \right] (1 - S_n^{N+1}) = (\rho_w S_w)^{N+1}, \\ \rho_{0n} \left[1 + \beta_n (p_w^{N+1} + p_c^{N+1} - p_{0n}) \right] S_n^{N+1} = (\rho_n S_n)^{N+1}, \end{cases} \quad (5)$$

которую можно решать, например, методом Ньютона.

1. Реализация параллельного алгоритма на гибридной вычислительной системе. Для проведения расчетов задач фильтрации на ГВС был создан комплекс программ на языке C++ с использованием библиотек CUDA и MPI. Комплекс ориентирован на класс задач, в которых аппроксимация производится на ортогональных сетках в прямоугольных областях с помощью явных разностных схем. В основу положен модульный принцип: комплекс состоит из четырех уровней – управляющего, расчетного, коммуникационного и уровня тестирования, который используется при отладке. Управляющий уровень содержит функции инициализации расчетов, выделения/освобождения памяти, сохранения результатов в файлы графиков, сохранения/восстановления системы на конкретный момент времени. Расчетный уровень содержит вычислительные функции решаемой задачи на выбранном вычислителе (например, CPU или GPU). В расчетный уровень входят модули для расчетов на CPU или GPU двухфазной и трехфазной задач фильтрации в различных постановках. Коммуникационный уровень содержит функции для обменов информацией между расчетными узлами кластера. Уровень тестирования содержит функции модульного тестирования¹ и проверки корректности вычисляемых результатов.

Каждый уровень состоит из модулей. Чтобы перейти от решения двухфазной задачи фильтрации к решению трехфазной задачи, необходимо изменить только один модуль из расчетного слоя. Чтобы производить расчеты на GPU, а не на CPU, также надо заменить только один модуль расчетного слоя. Таким образом, осуществлена возможность использования комплекса для расчетов как на персональном компьютере (с использованием CPU или GPU), так и на традиционных MPI-кластерах и кластерах с гибридной архитектурой.

Кроме того, модульная структура позволяет использовать при необходимости для каждого модуля свой компилятор, например, собирать код на CUDA с помощью компилятора nvcc, код на MPI – с помощью mpicc, а также включать модули на различных языках программирования. Таким образом, комплекс защищен от проблемы устаревания языка – при использовании на гибридных кластерах более современного языка программирования, он будет поддержан комплексом без существенных доработок.

Для получения максимальной эффективности используются различные типы памяти GPU: насколько это максимально возможно, используется регистровая память. Если её недостаточно, то данные размещаются в глобальной памяти. Для переменных, используемых только при чтении (таких, как параметры задачи, постоянные свойства сред, константы), используется константная память. Такой подход позволяет значительно уменьшить время доступа к информации, а значит, увеличить эффективность программного комплекса.

2. Результаты расчетов. Для оценки эффективности разработанного программного комплекса были проведены расчеты трехмерной тестовой задачи о просачивании двухфазной (вода – тетрахлорэтилен) жидкости в однородную по-

¹ Идея модульного тестирования (или unit-тестирования) состоит в том, чтобы задавать тесты для каждой функции. При внесении изменений в код данный подход оберегает от ошибок в уже оттестированных местах программы.

ристую среду с использованием модели и алгоритма, описанных в разд. 1 данной статьи. Постановка задачи в двумерной расчетной области подробно описана в работе [2].

Расчеты проводились на гибридном вычислительном кластере К-100. Этот суперкомпьютер содержит 64 вычислительных узла, на каждом из которых установлено по 2 шестиядерных процессора Intel Xeon X5370, 3 графических ускорителя NVidia Fermi C2050 (в каждом по 448 ядер GPU и 2,5 Гб памяти) и 96 Гб оперативной памяти. Вычислительные узлы соединены посредством коммуникационной системы «МВС-Экспресс» и внутренней сети Infiniband, скорость передачи данных между узлами – до 700 Мбайт/с, латентность – порядка 1,2 мкс.

Тестовые расчеты задачи просачивания продемонстрировали следующие результаты по быстродействию. Максимальное число точек расчетной сетки, которое умещается в памяти одного графического ускорителя, составляет порядка 250^3 (15 млн). На такой сетке 500 шагов по времени на одном графическом ускорителе рассчитываются за 174 секунды, на трех графических ускорителях одного узла – за 127 секунд, на одном ядре CPU – за 18 878 секунд, на одном шестиядерном CPU – за 3 554 секунды, а на узле, состоящем из двух центральных процессоров, – за 1 781 секунду.

Таким образом, ускорение одного GPU по сравнению с одним ядром CPU составляет **108,5**, ускорение одного GPU по сравнению с одним шестиядерным CPU – **20,4**. Один узел кластера К-100, состоящий из трех графических плат, рассчитал задачу одного узла, состоящего из двух шестиядерных процессоров, быстрее **в 14 раз**.

В следующем эксперименте количество точек расчетной сетки было увеличено в 100 раз и составило около 1,5 млрд. Были проведены расчеты 100 шагов по времени и получены следующие результаты: 80 графических ускорителей произвели расчет за 80 секунд, а 80 ядер CPU – за 5 744 секунды, таким образом, ускорение составило **71,8 раза**.

Хотя и для графических ускорителей, и для CPU используются одни и те же коммуникационные модули (в программном комплексе реализованы коммуникации на основе MPI и SHMEM-express), но эффективность расчетов на нескольких CPU по сравнению с одним CPU несколько выше, чем эффективность расчетов на нескольких GPU по сравнению с одним GPU. Это связано с дополнительными временными затратами на загрузку/выгрузку данных из памяти GPU в оперативную память и обратно.

Заключение. Результаты расчетов тестовых задач многофазной фильтрации, полученные с помощью разработанного универсального комплекса программ, показывают, что хотя использование графических ускорителей накладывает ряд серьезных ограничений на программную реализацию, тем не менее удачный выбор модели и численного алгоритма и надлежащая их адаптация к архитектуре ГВС позволяют получать на порядок большие ускорения, чем при использовании классических многопроцессорных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Четверушкин Б.Н., Морозов Д.Н., Трапезникова М.А., Чурбанова Н.Г., Шильников Е.В. Об одной схеме для решения задач фильтрации // Математическое моделирование. – 2010. – Т. 22, № 4. – С. 99-109.
2. Морозов Д.Н., Трапезникова М.А., Четверушкин Б.Н., Чурбанова Н.Г. Использование явных схем для моделирования процесса двухфазной фильтрации // Математическое моделирование. – 2011. – Т. 23, № 7. – С. 52-60.
3. Басниев К.С., Кочина И.Н., Максимов В.М. Подземная гидромеханика. – М.: Недра, 1993.

4. *Четверушкин Б.Н.* К вопросу об ограничении снизу на масштабы в механике сплошной среды // *Время, хаос, математические проблемы.* – М.: Изд-во МГУ, 2009. – Вып. 4. – С. 75-96.
5. *Morozov D.N., Chetverushkin B.N., Churbanova N.G., Trapeznikova M.A.* An Explicit Algorithm for Porous Media Flow Simulation using GPUs // *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press, Stirlingshire, UK, 2011.* – P. 19
6. *Боресков А.В., Харламов А.А.* Основы работы с технологией CUDA. – М.: ДМК-Пресс, 2010. – 232 с.

Статью рекомендовал к опубликованию к.ф.-м.н.; доцент Е.В. Шильников.

Морозов Дмитрий Николаевич – Московский физико-технический институт; e-mail: dmitry.morozov@phystech.edu; 141700, Московская область, г. Долгопрудный, Институтский пер., 9; тел.: +74954084554; аспирант.

Четверушкин Борис Николаевич – Институт прикладной математики им. М.В. Келдыша РАН; e-mail: chetver@imamod.ru; 125047, г. Москва, Миусская пл., 4; тел.: +74999781314; академик; директор.

Чурбанова Наталья Геннадиевна – e-mail: nata@imamod.ru; 125047, г. Москва, Миусская пл., 4; тел.: +74999781314; с.н.с.

Трапезникова Марина Александровна – e-mail: marina@imamod.ru; 125047, г. Москва, Миусская пл., 4; тел.: +74999781314; с.н.с.

Morozov Dmitry Nikolaevich – Moscow Institute of Physics and Technology; e-mail: dmitry.morozov@phystech.edu; 9, Institutskii per., Dolgoprudny, 141700, Moscow Region, Russia; phone: +74954084554; postgraduate student.

Chetverushkin Boris Nikolaevich – Keldysh Institute of Applied Mathematics; e-mail: chetver@imamod.ru; 4, Miusskaya sq., Moscow 125047, Russia; phone: +74999781314; academician; director.

Churbanova Natalia Gennadievna – e-mail: nata@imamod.ru; 4, Miusskaya sq., Moscow, 125047, Russia, phone: +74999781314; senior scientist.

Trapeznikova Marina Aleksandrovna – e-mail: marina@imamod.ru chetver@imamod.ru; 4, Miusskaya sq., Moscow, 125047, Russia; phone: +74999781314; senior scientist.

УДК 004.272.2

И.Г. Данилов

ОБ ОДНОМ ПОДХОДЕ К РЕАЛИЗАЦИИ ПРОГРАММНОЙ ТРАНЗАКЦИОННОЙ ПАМЯТИ ДЛЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

Транзакционная память (Transactional Memory, TM) предоставляет неблокирующий синхронизационный управляющий механизм для многопоточных приложений. В настоящее время существует большое количество реализаций (программных, аппаратных и гибридных) данного механизма для систем с общей памятью. Актуальным является исследование возможности применения транзакционной памяти к построению распределённых вычислительных систем. В данной работе описывается система программной транзакционной памяти DSTM_PI, с помощью которой можно запускать многопоточное приложение, написанное на языке C, на многоядерном кластере.

Распределённые вычисления; распределённая транзакционная память.