ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ MPI/OPENMP-РЕАЛИЗАЦИИ С ВЫДЕЛЕНИЕМ ПОТОКОВ-«ПОЧТАЛЬОНОВ» ТРЕХМЕРНОЙ СХЕМЫ РАСЩЕПЛЕНИЯ ДЛЯ ЗАДАЧИ ТЕПЛОПЕРЕНОСА

***К.В. Воронин*[1,2]**

[1]*Институт вычислительной математики и математической геофизики СО РАН*
[2]*Новосибирский госуниверситет*

Представлены результаты исследования параллельных алгоритмов реализации векторных схем расщепления на основе технологий MPI и OpenMP для решения трехмерных задач теплопереноса. Проводится сравнение параллельных алгоритмов, использующих только MPI, простой MPI/OpenMP и MPI/OpenMP с выделением потоков-«почтальонов». Основная идея MPI/OpenMP-алгоритма с «почтальонами» заключается в выделении на каждом из узлов с общей памятью одного OpenMP-потока, отвечающего за реализацию обмена данными между процессами. При использовании такого подхода вычисления выполняются одновременно с обменами данными. Результаты проведенного исследования позволили заключить: несмотря на то, что использование «гибридного» подхода с выделением потоков-«почтальонов» позволяет значительно ускорить эффективность «прямолинейного» MPI/OpenMP-варианта, такой подход уступает «чистой» MPI-реализации для рассматриваемого класса алгоритмов.

Введение

При решении больших прикладных задач математического моделирования физических процессов естественным образом возникает потребность в использовании современных суперкомпьютеров с общей и разделенной памятью. Целью использования суперкомпьютеров является не только преодоление ограничения на оперативную память на одном вычислительном узле, но и ускорение общего времени выполнения алгоритма за счет распараллеливания вычислений на все доступные ядра. Одним из наиболее распространенных подходов является использование технологии MPI для проведения расчетов на системах с распределенной памятью. В то же время современные вычислительные узлы на самом деле всегда состоят из нескольких вычислительных ядер с общей памятью. Поэтому кажется естественным использовать «гибридные» алгоритмы на основе MPI и OpenMP, где коммуникации между узлами осуществляются с помощью MPI, а внутри узлов работают OpenMP-потоки на общей памяти. Достаточно хорошо известно, что «чистая» MPI-реализация для алгоритмов, где распределение данных по процессам возникает естественным образом (явные схемы для параболических уравнений, метод разделения переменных для эллиптических уравнений, методы декомпозиции области и т.п.), показывает высокую эффективность (почти линейное шкалирование времени от числа MPI-процессов). Для достижения сравнимой производительности для OpenMP-реализации внутри одного вычислительного узла обычно осуществляется значительная модификация кода с введением локальных массивов для каждого из OpenMP-потоков. В данной работе исследован другой способ повышения эффективности MPI/OpenMP-реализации - основная идея заключается в выделении на каждом из узлов с общей памятью одного потока («почтальона»), отвечающего за выполнение обменов данными между процессами (между узлами – MPI). Данный подход позволяет организовать одновременное выполнение полезных вычислений и обменов данными

[1]. Для этого необходимо дополнительно разбить данные на каждом узле на части меньшего размера. Эффективность подобной гибридной MPI/OpenMP-реализации сильно зависит, в том числе, и от размера этих частей, который подбирался (экспериментально) оптимальным для рассматриваемых задач. В итоге, общее время работы алгоритма сводится к максимуму из времени, затрачиваемого на вычисления, и времени, затрачиваемого на обмен данными.

Исследование эффективности такого гибридного подхода с выделением потоков-«почтальонов» было проведено для трехмерной схемы расщепления в смешанном методе конечных элементов в задачах теплопереноса [2, 3]. Одной из характерных особенностей данной схемы является то, что благодаря использованию параллелепипедальных сеток и конечных элементов низкого порядка реализация данной схемы на каждом временном шаге сводится к выполнению (на каждом из дробных шагов) независимых прогонок вдоль координатных линий сетки. Данная схема является лишь одним из примеров векторных схем расщепления для задачи теплопереноса, записанной в терминах «температура – тепловой поток», поэтому все сделанные выводы об эффективности использования гибридных алгоритмов имеют достаточно общий характер и распространяются на весь класс схем, предложенных в рамках упомянутого подхода. В настоящее время векторные схемы расщепления в смешанном МКЭ применяются, например, для моделирования геотермальных процессов в литосфере [4].

Тезисы организованы следующим образом – в первом разделе приведены кратко постановка задачи и вид трехмерной схемы расщепления, для которой был построен параллельный алгоритм. Во втором разделе описаны нумерация данных и схема распределения данных по MPI-процессам, а также пересылки, необходимые для реализации одного временного шага схемы. В разделе 3 изложены основные особенности исследуемой гибридной MPI/OpenMP-реализации. Наконец, в разделе 4 представлены сравнительные результаты, полученные при проведении численных экспериментов на кластере Сибирского суперкомпьютерного центра (ССКЦ СО РАН) [5], для гибридной реализации с «почтальонами», прямолинейной MPI/OpenMP-реализации и «чистой» MPI-реализации. В заключении еще раз формулируются основные результаты работы.

1. Постановка задачи и схема расщепления

Рассмотрим следующую систему дифференциальных уравнений первого порядка, описывающую процесс распространения тепла в трехмерной области Ω:

$$\begin{aligned} c_p \rho \frac{\partial T}{\partial t} + \nabla \cdot \mathbf{w} &= f \\ \frac{1}{\lambda} \mathbf{w} &= -\nabla T \end{aligned} \qquad x \in \Omega, t > 0.$$

Здесь T и $\mathbf{w}$ – искомые функции температуры и теплового потока, c_p, ρ и λ – коэффициенты теплоемкости, плотности и теплопроводности соответственно, f – распределенные внутри области источники тепла. Первое уравнение представляет собой закон сохранения энергии, второе – закон Фурье. К системе присоединяются начальные данные для температуры, на границе ставятся краевые условия Неймана или Дирихле. После стандартных преобразований можно получить смешанную слабую (по пространству) постановку задачи

$$\begin{aligned} \int_\Omega c_p \rho \frac{\partial T}{\partial t} q dx + \int_\Omega \nabla \cdot \mathbf{w} q dx &= \int_\Omega f q dx, \forall q \in L_2(\Omega) \\ \int_\Omega \frac{1}{\lambda} \mathbf{w} \cdot \mathbf{u} dx &= \int_\Omega T \nabla \cdot \mathbf{u} dx - \int_{\partial\Omega} T \mathbf{u} \cdot \mathbf{n} d\gamma, \forall \mathbf{u} \in \mathbf{H}_{div}(\Omega) \end{aligned},$$

где искомые функции T и w ищутся как элементы функциональных пространств $C^1(0,T;L_2(\Omega))$ и $C(0,T;\mathbf{H}_{div}(\Omega))$ соответственно. Для пространственной аппроксимации применяется смешанный метод конечных элементов с элементами Равьяра-Тома для вектор-функций и кусочно-постоянными элементами для скаляров.

Таким образом, возникает следующая система обыкновенных дифференциальных уравнений

$$M\frac{\partial T_h}{\partial t}+\mathbf{B}\mathbf{w}_h=F_h \quad t>0,$$
$$\mathbf{A}\mathbf{w}_h=\mathbf{B}^T T_h$$

где матрица масс для потока $\mathbf{A}$ – блочно-диагональная с трехдиагональными блоками, M – диагональная матрица масс для температуры, прямоугольные матрицы $\mathbf{B}$ и $\mathbf{B}^T$ соответствуют операторам градиента и дивергенции.

Использованная в данной работе схема расщепления основана на схеме Дугласа и Гана для сеточной дивергенции потока. Эта схема имеет второй порядок точности по времени и пространству и обеспечивает абсолютно устойчивое вычисление температуры [4]. Реализация схемы на одном шаге по времени сводится к пяти дробным шагам, на каждом из которых необходимо выполнять прогонки вдоль линий сетки, правая часть для прогонки аппроксимирует вторые смешанные производные по пространству от промежуточных потоковых переменных.

2. Распределение данных по MPI-процессам

Для всех сеточных функций используется следующая нумерация: скалярные величины нумеруются в порядке возрастания индексов *y* - *z* - *x*; *x*-компоненты векторных величин – в порядке возрастания индексов *x* - *y* - *z*, *y*-компоненты – *y* - *z* - *x*, *z*-компоненты – *z* - *y* - *x*. При такой нумерации естественным образом организовывается распределение данных по процессам - данные «нарезаются» по внешнему индексу.

Для реализации данной схемы требуется три обмена данными типа «all-to-all» на каждом шаге по времени – пересылки компонент потоков по *x*, *y* и *z*. Выполнение каждого из дробных шагов схемы сводится к обращению трехдиагональных матриц на ассемблируемых векторах правой части, т.е. к выполнению независимых прогонок вдоль линий сетки, что позволяет на каждом MPI процессе выполнять решение соответствующих систем маленького размера независимо. В силу несимметричности схемы выбор внешних индексов нумерации был оптимальным с точки зрения количества необходимых обменов данных на каждом временном слое.

3. MPI/OpenMP-реализация с «почтальонами»

Коммуникации между вычислительными узлами осуществляются с помощью процедур MPI, поэтому данные между вычислительными узлами распределяются так, как это описано в предыдущем разделе. На каждом из вычислительных узлов используется распараллеливание с помощью OpenMP. Главным для исследуемой MPI/OpenMP-реализации с «почтальонами» является организация наложения вычислений и обменов данными на каждом из вычислительных узлов. С этой целью при выполнении прогонок вдоль каждого из направлений данные на каждом процессе дополнительно разбиваются на части меньшего размера. Каждая такая часть обрабатывается параллельно потоками-«решателями», в то время как поток-«почтальон» выполняет динамически пересылку обработанных на каждом MPI-процессе к текущему моменту времени частей.

Выбор размера маленькой части для обработки каждым из OpenMP-потоков-«решателей» должен быть оптимальным с точки зрения следующих ограничений. С

одной стороны, слишком маленький размер блока может привести к зависимости по памяти и многократной перекачке данных между кэшами каждого из потоков, с другой – слишком большой размер блока увеличивает время возможного простоя из-за необходимости синхронизации и уменьшает возможный выигрыш во времени из-за уменьшения гибкости гибридной части алгоритма. При исследовании эффективности алгоритма размер маленькой части подбирался экспериментально. Также необходимо отметить, что, помимо разбиения процесса вычислений на части, следует разбивать и процесс сборки векторов правой части для прогонок, причем размер частей разбиения для правой части также зависит от размера кэша и от размера блоков для вычислений (для тех слагаемых, в которых участвуют данные, полученные в ходе межузловых коммуникаций).

Кратко основная идея подобной MPI/OpenMP-реализации может быть также сформулирована в виде следующего псевдокода:

```
if (my_thread_ID != 0) {
 // "solver"-threads:
 for (int count = 0; count < chunks_number; count++) { /* loop for chunks */
 #pragma omp parallel
 /* solving each small chunk in parallel */
 foo1();
 }
 #pragma omp flush
 /*telling the "postman"-thread that several chunks are ready to be exchanged*/
 foo2();
}
else { //my_threadID = 0
 // "postman"-thread:
 while /*not all chunks are exchanged */ {
 #pragma omp flush
 /* checking if there are ready chunks to be exchanged, if yes - transfer */
 foo3();
 }
}
```

4. Сравнение MPI-, MPI/OpenMP- и MPI/OpenMP-реализации с «почтальонами»

В данном разделе представлены результаты сравнения MPI-реализации с «прямолинейной» MPI/OpenMP- и MPI/OpenMP-реализацией с «почтальонами». Под «прямолинейной» MPI/OpenMP-реализацией подразумевается простое использование *#pragma omp* директив в MPI-коде. Представленные результаты были получены на кластере ССКЦ СО РАН [5] на двойных блейд-серверах HP BL2x220 G7, ОП модуля – 24 Гбайт, каждый узел состоит из двух 6-ядерных процессоров Intel Xeon X5670 2.93 GHz (Westmere). Расчеты проводились на тестовом решении, соответствующем правой части формулой $f = \left(f_{i,j,k}\right)$, где $f_{i,j,k} = h_x^i h_y^j h_z^k \frac{t}{t+1}\frac{i+j+k}{i+j+k+2}10^{-4}$.

Для гибридных MPI/OpenMP-реализаций число MPI-процессов совпадало с числом вычислительных узлов. Т.к. основное отличие от MPI-реализации состоит в использовании общей памяти внутри узла, использовались только два вычислительных узла. При этом для MPI/OpenMP-реализации с «почтальонами» был проведен ряд оптимизаций, связанных с локальной перенумерацией данных. На рис.1 приведены результаты сравнения MPI-, MPI/OpenMP- и MPI/OpenMP-реализаций для сетки 384x384x384, при этом оптимальный размер частей для реализации с «почтальонами», на которые делились данные внутри узла, был подобран экспериментально.

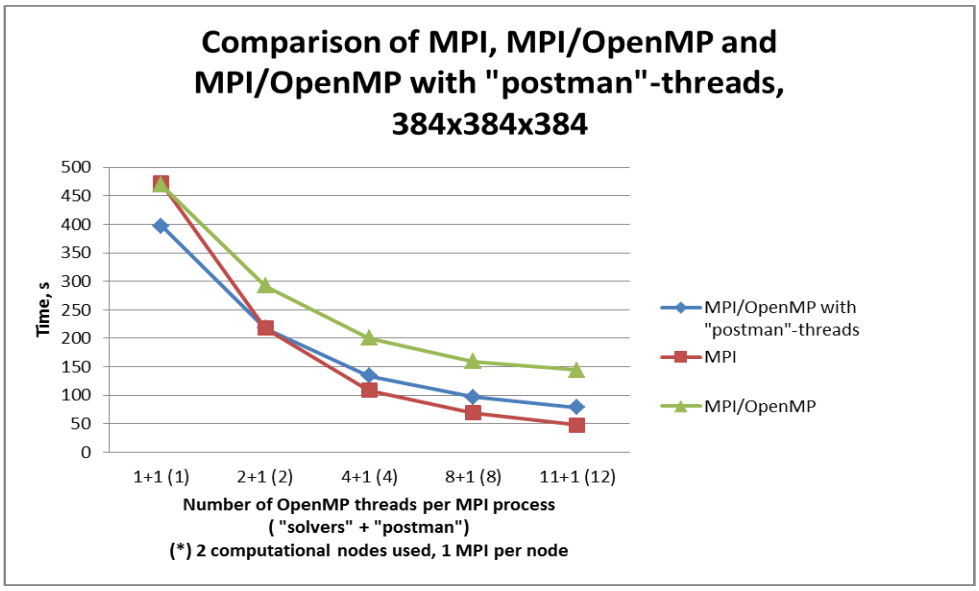

Рис. 1. Зависимость времени выполнения от числа OpenMP-потоков на вычислительном узле (числа MPI-процессов на узле для MPI-реализации), 2 вычислительных узла, сетка 384x384x384

Как можно заметить на рис. 1, наложение вычислений и обменов данными позволило существенно повысить эффективность простой MPI/OpenMP-реализации, однако MPI-реализация все равно показывает лучший результат. Это происходит не столько за счет возникновения дополнительных накладных расходов организации «гибридности» кода, сколько из-за того, что значительно увеличивается время, затрачиваемое на пересылки (т.к. в данном случае было использовано всего 2 MPI-процесса для MPI/OpenMP и 24 для MPI). Следует также заметить, что чистая MPI-реализация показывают очень высокую эффективность (80% при использовании 12 процессов на узле по сравнению с одним). Более того, при увеличении размера задачи шкалирование MPI-реализации еще улучшается. Необходимо отметить также одну из важных особенностей MPI/OpenMP-реализации с «почтальонами», а именно сильную зависимость эффективности от размера частей, на которые дополнительно делятся данные на каждом процессе. При этом априорный выбор оптимального размера требует тщательного анализа особенностей устройства общей памяти на вычислительном узле.

Заключение

Результаты проведенного численного исследования позволяют сделать вывод, что для рассматриваемого класса алгоритмов использование гибридного MPI/OpenMP-подхода с выделением потоков-«почтальонов» значительно улучшает «простую» (без локальных массивов для каждого OpenMP-потока) MPI/OpenMP-реализацию, но не приводит к повышению эффективности по сравнению с реализацией на основе MPI.

Данная работа поддержана грантами РФФИ №13-01-00019 и №12-01-31046.

Литература

1. Rabenseifner R., Hager G., Jost G. Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes // Proceedings of the 2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing. 427-436 (2009).

2. Воронин К.В., Лаевский Ю.М. Схемы расщепления в смешанном методе конечных элементов решения задач теплопереноса // Матем. Моделирование. 24(8). 109-120 (2012).
3. Воронин К.В., Лаевский Ю.М. О схемах расщепления в смешанном методе конечных элементов // Сиб. журн. вычисл. Математики. 15(2). 101-107 (2012)..4. Верниковская А.Е., Даценко В.М., Верниковский В.А., Матушкин Н.Ю., Лаевский Ю.М., Романова И.В., Травин А.В., Воронин К.В., Лепехина Е.Н. Эволюция магматизма и карбонатит-гранитная ассоциация в неопретерозойской активной континентальной окраине Сибирского кратона: термохронологические реконструкции // Доклады Академии наук. 448(5). 555-562 (2013).
5. Сибирский суперкомпьютерный центр – http://www2.sscc.ru/.