

УДК 519.6

DOI: 10.17223/19988605/51/13

Е.В. Семёнов, А.В. Старченко, Е.А. Данилкин, С.А. Проханов**ВЫБОР ТЕХНОЛОГИИ РАСПАРАЛЛЕЛИВАНИЯ ЧИСЛЕННОГО РЕШЕНИЯ
КОНВЕКТИВНО-ДИФфуЗИОННОГО УРАВНЕНИЯ НА ГИБРИДНОЙ
МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ***Исследование выполнено за счет гранта Российского научного фонда (проект № 19-71-20042).*

Представлены результаты применения трех популярных технологий параллельного программирования – Message Passing Interface (MPI), Open Multi-Processing (OpenMP), Open Accelerators (OpenACC) – с целью определения наиболее производительной для параллельной версии метеорологической модели TSUNM3. Сравнение выполнено на примере численного решения неоднородного нестационарного трехмерного конвективно-диффузионного уравнения. Расчеты выполнялись на вычислительном кластере с многоядерными центральными процессорами и графическими видеокартами.

Ключевые слова: распараллеливание; параллельные вычисления; MPI; OpenMP; OpenACC; конвективно-диффузионное уравнение; явно-неявные разностные схемы.

В настоящее время при исследовании окружающей среды наряду с экспериментальными методами большое внимание уделяется применению методов математического моделирования. Именно математическое моделирование позволяет изучать и прогнозировать развитие атмосферных, гидрологических и геологических процессов на Земле без активного воздействия на них. Основу многих математических моделей механики сплошных сред, гидрометеорологии, экологии и других наук составляют неоднородные нестационарные трехмерные уравнения «конвекции–диффузии», которые описывают изменение таких параметров изучаемой среды, как плотность, компоненты скорости, температура, концентрация, турбулентность. Современные математические модели представляют собой систему из нескольких конвективно-диффузионных уравнений, дополненную замыкающими алгебраическими соотношениями, а также начальными и граничными условиями.

Для решения рассматриваемых систем дифференциальных уравнений применяются методы конечных разностей, конечного объема или конечных элементов. Приближенное решение ищется на весьма подробной сетке, состоящей из нескольких миллионов вычислительных узлов, или ячеек. Причем часто при оперативном моделировании атмосферных или гидрологических процессов необходимо в кратчайшие сроки вычислений на компьютере представить результаты прогностического моделирования.

Возможность ускорить процесс получения численного решения дает применение многопроцессорной и многоядерной вычислительной техники, снабженной графическими ускорителями [1, 2]. Разнообразие в архитектуре и компоновке высокопроизводительных вычислительных систем порождает разнообразие инструментов для построения параллельных программ. На сегодняшний день в области научных вычислений самыми популярными и узнаваемыми для написания параллельных программ являются технологии MPI, OpenMP и OpenACC. Это подталкивает исследователей к оценке перспективности использования разных подходов и их комбинаций для решения своих задач.

В работе [3] на примере модели атмосферы UM (Unified Model UK Met Office) рассматривается эффективность использования гибридного подхода MPI + OpenMP в сравнении с MPI. Также исследуется эффективность выполнения параллельной программы при фиксированном количестве процес-

сов, но при изменяющемся количестве задействованных ядер на каждом процессоре. Это обусловлено тем, что в вычислительных задачах особое внимание уделяется работе с памятью, и увеличение количества активных ядер в процессоре приводит к увеличению их конкуренции в борьбе за ресурсы процессора. Показано, что использование только технологии распараллеливания MPI оказывается эффективнее на всем диапазоне рассматриваемого количества процессов (до 1 536). В то же время использование только 12 из 16 ядер на каждом процессоре также позволяет повысить эффективность вычислений на 10–15%. Это происходит за счет увеличения пропускной способности памяти в расчете на одно активное ядро, сокращения издержек на коммуникации при уменьшении количества процессов MPI и активных ядер, увеличения частоты работы ядер в режиме turbo boost [3].

В работе [4] рассматривается реализация блока переноса примесей в моделях атмосферы и океана на основе гибридного программирования MPI + OpenMP. В работе показано, что при небольшом количестве задействованных MPI-процессов более целесообразно дальнейшее наращивание их количества. В этом случае дополнительное ускорение достигается за счет размещения данных локальных массивов в более быстрой кэш-памяти процессора. При достаточно большом числе задействованных MPI-процессов, наоборот, более эффективным методом ускорения расчетов становится увеличение числа нитей OpenMP. В работе делается акцент на то, что гибридный подход MPI + OpenMP является наиболее универсальным с точки зрения использования ресурсов современных суперкомпьютеров, позволяя оптимизировать число MPI-процессов и число нитей OpenMP с учетом задачи и архитектуры вычислительных элементов.

В работе [5] анализируются возможности повышения производительности вычислительных моделей горения на многоядерных компьютерах. Показано, что использование явного распределения данных по потокам оказывается более эффективным по сравнению с непосредственным применением директив OpenMP для автоматического распараллеливания циклов.

В работе [6] представлены результаты исследования параллельных реализаций схем расщепления на основе технологий MPI и OpenMP для решения трехмерной задачи теплопереноса. Проводится сравнение следующих параллельных реализаций: MPI, MPI + OpenMP и MPI + OpenMP с выделением потоков-«почтальонов». Под MPI + OpenMP реализацией понимается непосредственно применение автоматического распараллеливания циклов внутри MPI-процессов. Основная идея гибридной MPI + OpenMP реализации с «почтальонами» заключается в выделении на каждом из узлов с общей памятью одного OpenMP потока, отвечающего за выполнение обменов данными между процессами на фоне вычислений. Результаты показали, что использование подхода с выделением потоков-«почтальонов» значительно повышает эффективность простой гибридной реализации, но уступает простой MPI реализации.

Таким образом, проведенный анализ использования технологий распараллеливания MPI и OpenMP как отдельно, так и совместно показал, что наибольшей эффективности можно достичь при применении технологии MPI на большом количестве используемых процессов. При небольшом количестве многоядерных центральных процессоров на общей памяти разумнее использовать OpenMP технологию как более простую с точки зрения применения и не менее эффективную при небольшом числе задействованных вычислительных ядер. Постоянное развитие вычислительной техники (процессоров, сопроцессоров, графических процессоров), компиляторов и технологий параллельного программирования требует систематического проведения исследований по выбору наиболее эффективных технологий распараллеливания алгоритмов решения прикладных задач вычислительной математики.

Целью данной работы является проведение вычислительных экспериментов на гибридной кластерной вычислительной системе по определению наиболее перспективного использования параллельных вычислительных технологий (MPI, OpenMP, OpenACC) для получения эффективных параллельных программ для решения нестационарного трехмерного неоднородного уравнения «конвекции–диффузии». Полученные результаты будут использованы при создании параллельной версии развиваемой в Томском государственном университете мезомасштабной метеорологической модели высокого разрешения TSUNM3 для прогнозирования опасных погодных явлений и качества атмосферного воздуха над городом [7].

1. Математическая формулировка и численный метод

1.1. Математическая постановка задачи

Рассмотрим обобщенное дифференциальное уравнение конвективно-диффузионного переноса:

$$\frac{\partial \rho \Phi}{\partial t} + \frac{\partial \rho U \Phi}{\partial x} + \frac{\partial \rho V \Phi}{\partial y} + \frac{\partial \rho W \Phi}{\partial z} = \frac{\partial}{\partial x} \left(K_{xy} \frac{\partial \Phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_{xy} \frac{\partial \Phi}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z \frac{\partial \Phi}{\partial z} \right) + S_\Phi. \quad (1)$$

Здесь t, x, y, z – время и координаты (ось Ox направлена на восток, Oy – на север, Oz – вертикально вверх); U, V, W – компоненты вектора скорости; ρ – плотность; K_{xy}, K_z – коэффициенты турбулентного обмена; S_Φ – источниковый член. В (1) могут рассматриваться в качестве $\Phi = 1$ ((1) принимает вид уравнения неразрывности), $\Phi = U, V, W, \Theta, q_v, q_{rain}$ (компоненты скорости, потенциальная температура воздуха, абсолютная влажность воздуха и концентрация дождевых капель). Для учета рельефа подстилающей поверхности $h(x, y)$ к уравнению (1) применяется преобразование вида:

$$x' = x; \quad y' = y; \quad z' = \frac{z - h(x, y)}{H - h(x, y)} H. \quad (2)$$

В качестве граничных условий для (1) на подстилающей поверхности используются граничные условия третьего рода:

$$z = h(x, y): K_z \frac{\partial \Phi}{\partial z} = \alpha(\Phi - \Phi_w). \quad (3)$$

На верхней границе – граничные условия второго рода:

$$z = H: K_z \frac{\partial \Phi}{\partial z} = \gamma_\Phi. \quad (4)$$

На боковых границах – граничные условия «радиационного» типа [8]:

$$x = -L; x = L: \frac{\partial \Phi}{\partial t} + C_\Phi^x \frac{\partial \Phi}{\partial x} = \frac{\partial \Phi_S}{\partial t} + C_\Phi^x \frac{\partial \Phi_S}{\partial x}; \quad (5)$$

$$y = -L; y = L: \frac{\partial \Phi}{\partial t} + C_\Phi^y \frac{\partial \Phi}{\partial y} = \frac{\partial \Phi_S}{\partial t} + C_\Phi^y \frac{\partial \Phi_S}{\partial y}. \quad (6)$$

Здесь индекс S соответствует параметрам, определяемым из модели глобального масштаба [9]; C_Φ^x, C_Φ^y – фазовая скорость [8]; $2L$ – размер области по горизонтали.

Начальные условия для уравнения (1) определяются в результате интерполяции Φ_S , вычисляемой на более грубой сетке по модели глобального масштаба [9], на более подробную сетку мезомасштабной метеорологической модели.

1.2. Численный метод решения задачи

Для решения задачи (1)–(6) используются структурированные сетки с равномерным шагом по горизонтальным направлениям Ox и Oy и сгущающиеся к поверхности Земли по вертикальному. При аппроксимации дифференциальной постановки задачи (1)–(6) используется метод конечного объема со вторым порядком аппроксимации по пространственным переменным с применением оригинальной монотонизированной схемы для конвективных членов, построенной с использованием интерполяционных весовых сплайнов [10].

Рассматриваемые явно-неявные аппроксимации во времени (Адамса–Бэшфорда и Кранка–Никольсон) обеспечивали второй порядок точности при интегрировании уравнения (1) по времени. В итоге полученная разностная схема – аналог (1) – имела следующий вид:

$$\frac{\rho_P^{k+1} \Phi_P^{k+1} - \rho_P^k \Phi_P^k}{\tau} Vol_P + \frac{3}{2} \iiint_{Vol_P} \left[\frac{\partial \rho U \Phi}{\partial x} + \frac{\partial \rho V \Phi}{\partial y} + \frac{\partial \rho W \Phi}{\partial z} - \frac{\partial}{\partial x} \left(K_{xy} \frac{\partial \Phi}{\partial x} \right) - \frac{\partial}{\partial y} \left(K_{xy} \frac{\partial \Phi}{\partial y} \right) - S_\Phi \right]^k dx dy dz -$$

$$-\frac{1}{2} \iiint_{Vol_p} \left[\frac{\partial p U \Phi}{\partial x} + \frac{\partial p V \Phi}{\partial y} + \frac{\partial p W \Phi}{\partial z} - \frac{\partial}{\partial x} \left(K_{xy} \frac{\partial \Phi}{\partial x} \right) - \frac{\partial}{\partial y} \left(K_{xy} \frac{\partial \Phi}{\partial y} \right) - S_{\Phi} \right]^{k-1} dx dy dz - \frac{1}{2} \iiint_{Vol_p} \left[\frac{\partial}{\partial z} \left(K_z \frac{\partial \Phi}{\partial z} \right) \right]^{k+1} dx dy dz - \frac{1}{2} \iiint_{Vol_p} \left[\frac{\partial}{\partial z} \left(K_z \frac{\partial \Phi}{\partial z} \right) \right]^k dx dy dz = 0. \quad (7)$$

Здесь k – номер слоя по времени; (x_p, y_p, z_p) – координаты середины конечного объема; Vol_p – значение конечного объема; здесь и далее нижний индекс p (point) указывает, что переменная определяется в центре конечного объема. Величина шага интегрирования по времени τ выбиралась из условия устойчивости разностной схемы. Рассматриваемая разностная схема является экономичной и для вычисления значений сеточной функции $\{\Phi_p^{k+1}\}$ позволяет использовать метод прогонки. Следует отметить, что неявная аппроксимация для вертикального диффузионного переноса, который имеет важное значение в пограничном слое атмосферы, позволяет уйти от более жесткого ограничения на шаг интегрирования по времени.

Вычислительное ядро программы расчета значений сеточной функции $\{\Phi_p^{k+1}\}$ на новом $(k + 1)$ -м слое по времени представлено на рис. 1. Пунктирными линиями выделен блок, который может быть преобразован в подпрограмму TDMA(I, J) для вычисления $\{\Phi_p^{k+1}\}$ на отдельной вертикальной сеточной линии с помощью метода прогонки.

Для оценки времени работы программной реализации (7) рассматривалась задача о переносе примеси от постоянно действующего источника в области с размерами $L = 25\,000$ м и $H = 600$ м. Источник примеси расположен в точке с координатами $(0,014L, 0,014L, 0,0101H)$. Компоненты скорости считались постоянными: $U = 1$ м/с, $V = 1$ м/с, $W = 0$ м/с. Коэффициенты диффузии также принимались постоянными $K_{xy} = K_z = 100$ м²/с. Шаг по времени выбирался равным 6 с, а время окончания моделирования $TimeFin = 500\tau$. Расчет проводился на сетке $256 \times 256 \times 32$ на узлах вычислительного кластера ТГУ Cyberia со следующими характеристиками: 48Gb RAM, 2xIntel® Xeon® CPU 5670, 2,93 ГГц. Использовался компилятор Intel Fortran. Время счета последовательной программы (см. рис. 1) составило 107 с. Точность проведения расчета контролировалась по известному аналитическому решению рассматриваемой задачи.

```

Time = 0
Do While (Time < TimeFin)
  Do I = 1, Nx
    Do J = 1, Ny
      !-----TDMA (I, J)-----
      P(0) = !из разностного аналога формулы (3)
      Q(0) = !из разностного аналога формулы (3)
      Do K = 1, Nz
        !расчет коэффициентов разностной схемы и
        !прогоночных коэффициентов P и Q из разностной схемы (7)
      End Do !цикл K
      F(Nz+1, I, J) = Q(Nz) / (1.0 - P(Nz))
      Do K = Nz, 0, -1
        F(K, I, J) = P(K) * F(K+1, I, J) + Q(K)
      End Do !цикл K
      !задание боковых граничных условий из (5) и (6)
      !-----TDMA (I, J)-----
    End Do !цикл J
  End Do !цикл I
  F0(:, :, :) = F(:, :, :) !обмен значениями
  Time = Time + tau
End Do !While

```

Рис. 1. Вычислительное ядро численного метода (7) решения уравнения переноса
 Fig. 1. The computational kernel of the numerical method (7) of solving the transport equation

2. Параллельные алгоритмы для численного метода

2.1. Технология Message Passing Interface

При применении этой технологии в качестве основного подхода распараллеливания для вычислительной системы с распределенной памятью выбрана двумерная (по осям Ox и Oy) декомпозиция сеточной области на подобласти (рис. 2).

При такой декомпозиции каждому процессорному элементу (ПЭ), задействованному в вычислениях, распределяются значения сеточной функции $\{\Phi_P^k\}$, принадлежащие локальной подобласти, для вычисления значений на следующем слое по времени $\{\Phi_P^{k+1}\}$ с использованием метода прогонки.

Из-за используемого шаблона явно-неявной разностной схемы (7) (см. рис. 2, выделено в нижнем правом углу) для вычисления очередного приближения в приграничных узлах каждой локальной подобласти необходимо знать значения сеточной функции $\{\Phi_P^k\}$ соседней граничащей подобласти. Для этого создаются фиктивные ячейки (на рис. 2 светлые значки) для хранения данных с процессорного элемента, рассчитывающего $\{\Phi_P^{k+1}\}$ в соседней подобласти декомпозиции, и организуются пересылки этих граничных значений, необходимых для обеспечения однородности вычислений в каждой локальной подобласти [11]. Для передачи данных другим процессорным элементам и получения необходимых для продолжения вычислений данных от них в данной работе используется интерфейс передачи сообщений MPI.

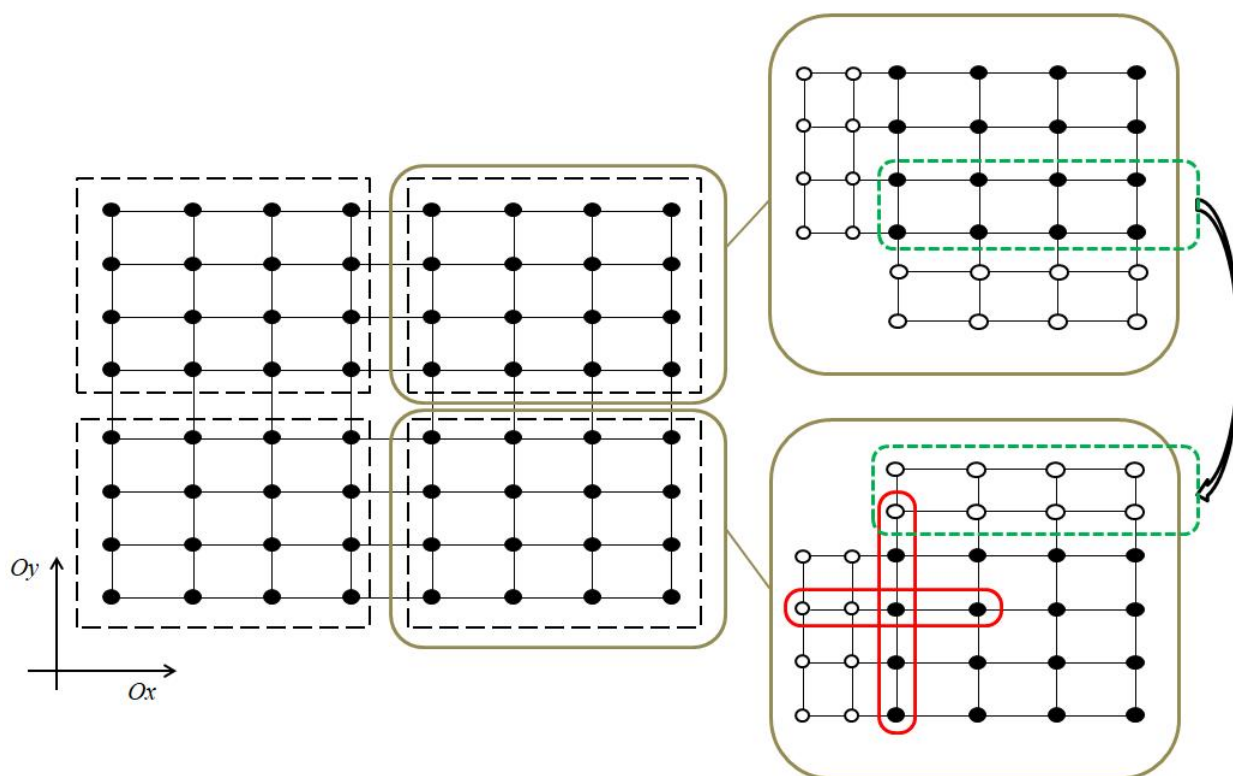


Рис. 2. Двумерная декомпозиция сеточной области вдоль направлений Ox и Oy и организация передачи данных между подобластями
 Fig. 2. Two-dimensional decomposition of the grid area along the Ox and Oy directions and the organization of data transfer between the subdomains

При разработке параллельной программы дополнительно используется ряд функций MPI для создания и тестирования двумерной декартовой логической топологии, создания новых типов данных для обеспечения эффективной пересылки между процессорами, а также функция SENDRECV (в процедуре MPE_SEND_F) для проведения парных блокирующих пересылок значений сеточной функции в соседних подобластях декомпозиции (рис. 3).

```

Time = 0
Do While (Time < TimeFin)
  Do I = 2, Px-1 !- число узлов в подобласти по x (Рис.2)
    Do J = 2, Py-1 !- число узлов в подобласти по y (Рис.2)
      CALL TDMA(I,J)
    End Do !цикл J
  End Do !цикл I
  Call MPE_SEND_F !обмен значений между ПЭ
  F0(:, :, :) = F(:, :, :) !обмен значениями между слоями k и k+1
  Time = Time + tau
End Do !While
    
```

Рис. 3. Фрагмент параллельной части MPI-программы для решения уравнения переноса
 Fig. 3. Fragment of the parallel part of the MPI program for solving the transport equation

Проведенные на вычислительном кластере ТГУ Cyberia (использовались узлы с указанными выше характеристиками) расчеты показали достаточно высокую эффективность (не ниже 70% на 128 процессах) выбранного способа распараллеливания явно-неявного вычислительного алгоритма (рис. 4, светлые значки). Для сравнения на этом же рисунке приведены значения ускорения и эффективности, полученные с использованием технологии OpenMP (см. п. 2.2). Значения ускорения рассчитывались по формуле $S_p = T_1/T_p$, а эффективность параллельной программы вычислялась как $E_p = S_p/p$. Здесь T_1 , T_p – время счета параллельной программы на 1 и p процессах. Из рис. 4 также видно, что время счета MPI-программы снижается практически линейно. При количестве процессов параллельной программы $p = 128$ время счета снижается в 90 раз по сравнению с временем счета последовательной программы.

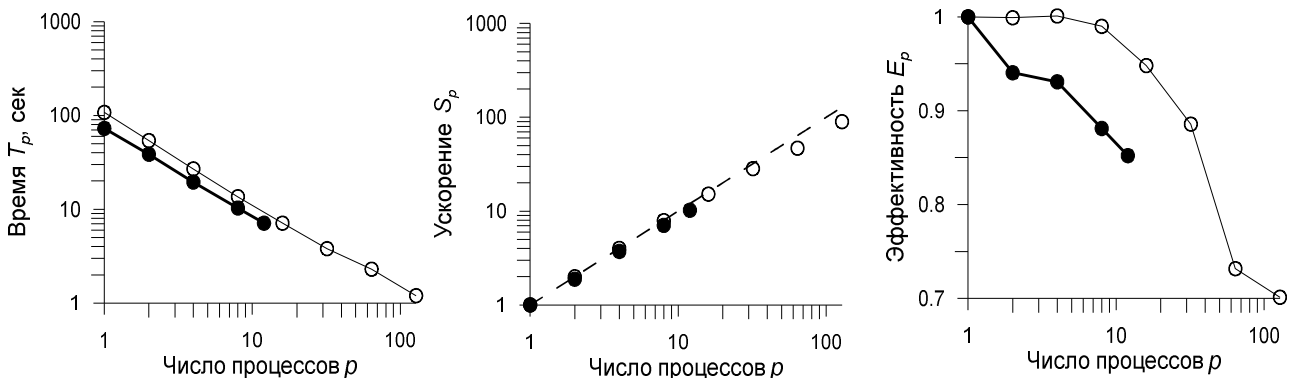


Рис. 4 Время счета, ускорение и эффективность MPI (o) и OpenMP (•) программ.
 Штриховая линия представляет график идеального ускорения параллельных программ
 Fig. 4 Time of calculation, acceleration and efficiency of MPI (o) and OpenMP (•) programs.
 The dashed line represents the graph of the ideal acceleration of parallel programs

Этот результат чрезвычайно важен при разработке программ для проведения прогностических расчетов, когда требуется за минимальное время выполнить прогноз опасного погодного явления (ливень, интенсивный снегопад, заморозки, град и т.п.), тем самым повышая уровень заблаговременной предупредительности и оставляя больше времени для подготовки и снижения последствий.

2.2. Технология Open Multi-Processing (OpenMP)

Данная технология распараллеливания применяется для многопроцессорных (многоядерных) вычислительных систем с общей оперативной памятью. Основная задача, которая стоит перед программистом, выбравшим эту технологию распараллеливания, заключается в распределении вычислений между параллельно выполняемыми потоками обработки общих разделяемых данных. Это реализуется с помощью специальных директив препроцессора используемого языка программирования, вызова библиотечных процедур и задания переменных окружения [12].

В рассматриваемом случае при создании параллельного многопоточного приложения особое внимание нужно уделить основному вычислительному блоку (см. рис. 1), в котором производится перевычисление значений сеточной функции $\{\Phi_p^{k+1}\}$ на каждом следующем слое по времени, а также распределению вычислений, выполняемых для каждого фиксированного значения переменных циклов (I, J) , между задействованными в расчетах многоядерными процессорами. Для этого в последовательную версию программы в соответствующих местах нужно разместить соответствующие директивы OpenMP (рис. 5). Заметим, что трудоемкость преобразования последовательной программы в параллельную с помощью технологии OpenMP значительно ниже, чем при разработке MPI параллельной программы. Более того, для того чтобы размещенные в OpenMP-программе директивы заработали, нужно при компиляции программы использовать некоторые дополнительные опции. Без них директивы OpenMP при компиляции будут игнорироваться.

```

Time = 0
Do While (Time < TimeFin)
$OMP PARALLEL PRIVATE(I,J)
$OMP DO
  Do I = 1,Nx
    Do J = 1,Ny
      CALL TDMA(I,J)
    End Do !цикл J
  End Do !цикл I
$OMP END DO
!обмен значениями
$OMP DO
  Do I = 0,Nx+1
    Do J = 0,Ny+1
      Do K = 0,Nz+1
        FO(K,I,J) = F(K,I,J)
      End Do !цикл K
    End Do !цикл J
  End Do !цикл I
$OMP END DO
$OMP END PARALLEL
  Time = Time + tau
End Do !While

```

Рис. 5. Фрагмент параллельной части OpenMP-программы для решения уравнения переноса
 Fig. 5. Fragment of the parallel part of the OpenMP program for solving the transport equation

При проведении расчетов на одном вычислительном узле кластера ТГУ Cyberia с двумя процессорами Intel Xeon 5670, каждый из которых имеет 6 ядер, были получены следующие результаты ускорения и эффективности разработанной OpenMP-программы (см. рис. 4). Из рисунка видно, что в относительных величинах ускорения S_p и эффективности E_p , а также по возможности привлечения других вычислительных узлов кластера OpenMP-программа уступает MPI-программе, однако в абсолютных значениях время работы OpenMP-программы для каждого параллельного запуска несколько меньше (различие уменьшается с ростом количества процессов). Тем не менее с использованием относительно простой технологии параллельного программирования OpenMP на одном вычислительном узле кластера Cyberia можно получить 10-кратное ускорение параллельной программы по сравнению с последовательной.

2.3. Технология Open Accelerators (OpenACC)

Данная технология параллельного программирования используется для создания параллельных программ на гетерогенных вычислительных системах с центральными и графическими процессорами. Эта технология параллельного программирования, как и технология OpenMP, с помощью директив компилятора позволяет отмечать участки программы, которые следует выполнять параллельно на

графическом процессоре, обозначает общие и локальные переменные [13]. Здесь также при параллельном программировании большое внимание уделяется распределению циклически выполняющихся вычислений графическому процессору (рис. 6).

```

Time = 0
!$ACC DATA COPYIN(F0) COPYOUT (F)
Do While (Time<TimeFin)
!$ACC KERNELS LOOP
  Do I = 1,Nx
    Do J = 1,Ny
      Do K = 1,Nz
        !расчет коэффициентов разностной схемы и
        !коэффициентов метода прогонки из разностного аналога (7)
      End Do !цикл K
    End Do !цикл J
  End Do !цикл I
!$ACC END KERNELS LOOP
!$ACC KERNELS LOOP
  Do I = 1,Nx
    Do J = 1,Ny
      Do K = 1,Nz
        !расчет прогоночных коэффициентов P и Q по формулам метода прогонки
      End Do !цикл K
      F(Nz+1,I,J) = Q(I,J,Nz)/(1.0 - P(I,J,Nz))
      Do K = Nz,0,-1
        F(K,I,J) = P(I,J,K) * F(K+1,I,J) + Q(I,J,K)
      End Do !цикл K
    End Do !цикл J
  End Do !цикл I
!$ACC END KERNELS LOOP
!$ACC KERNELS
  !задание боковых граничных условий (5) и (6). Обмен значениями
!$ACC END KERNELS
  Time = Time + tau
End Do !While
!$ACC END DATA

```

Рис. 6. Фрагмент параллельной части OpenACC-программы для решения уравнения переноса
 Fig. 6. Fragment of the parallel part of the OpenACC program for solving the transport equation

Результаты счета параллельной программы, созданной с помощью технологии OpenACC и компилятора Portland Group Fortran, на одном вычислительном узле кластера ТГУ Cyberia с 14-ядерным процессором Intel Xeon E5-2695v3 и графическим процессором NVIDIA Tesla K80 показали время выполнения 3 с, что существенно меньше времени выполнения последовательной программы и в несколько раз меньше времени выполнения MPI- и OpenMP-программ на 16 процессах на узлах вычислительного кластера Cyberia.

Заключение

Для выбора наиболее перспективной технологии параллельного программирования для разработки высокопроизводительного программного обеспечения для численного решения неоднородного нестационарного трехмерного конвективно-диффузионного уравнения с помощью метода конечного объема и явно-неявных разностных схем были проведены вычислительные эксперименты на вычислительном кластере ТГУ Cyberia. В качестве технологий параллельного программирования рассматривались международные стандарты программирования Message Passing Interface – для многопроцессорных систем с распределенной памятью, Open Multi-Processing – для многопроцессорных систем с общей памятью, и Open Accelerators – для гибридных вычислительных систем с графическими процессорами.

Расчеты, проведенные на структурированных разностных сетках с количеством узлов более 2 млн, показали, что наиболее высокопроизводительную программу для решения рассматриваемой задачи можно получить с помощью технологии MPI. На 128 процессах (10–20 CPUs, связанных компьютерной сетью) она дает ускорение вычислений почти в 90 раз по сравнению с расчетом последовательной программы. Более «бюджетные» варианты: использование технологии OpenACC для графических процессоров NVIDIA (1 CPU + 1 GPU на одном вычислительном узле) при тех же условиях дает ускорение вычислений почти в 30 раз; применение технологии OpenMP (2 CPUs на одном вычислительном узле) обеспечивает ускорение в 10 раз.

ЛИТЕРАТУРА

1. Powers, J.G., Klemp J.B., Skamarock W.C., Davis C.A., Dudhia J., Gill D.O., Coen J.L., Gochis D.J., Ahmadov R., Peckham S.E., Grell G.A., Michalakes J., Trahan S., Benjamin S.G., Alexander C.R., Dimego G.J., Wang W., Schwartz C.S., Romine G.S., Liu Z., Snyder C., Chen F., Barlage M.J., Yu W., Duda M.G. The Weather Research and Forecasting Model: Overview, System Efforts and Future Directions // *Bull. Amer. Meteor. Soc.* 2017. V. 98. P. 1717–1737. DOI: 10.1175/BAMS-D-15-00308.1
2. Ridwan R., Kistijantoro A.I., Kudsy M., Gunawan D. Performance evaluation of hybrid parallel computing for WRF model with CUDA and OpenMP // 2015 3rd Int. Conf. on Information and Communication Technology (ICoICT), Nusa Dua. 2015. P. 425–430. DOI: 10.1109/ICoICT.2015.7231463
3. Bermous I., Steinle P. Efficient performance of the Met Office Unified Model v8.2 on Intel Xeon partially used nodes // *Geosci. Model Dev.* 2015. V. 8. P. 769–779.
4. Мортиков Е.В. Программная реализация блока переноса примесей в климатических моделях на основе гибридного программирования MPI-OpenMP // Суперкомпьютерные дни в России : тр. междунар. конф. 2016. С. 521–529.
5. Кандрюкова Т.А., Лаевский Ю.М. О численном моделировании фильтрационного горения газов на многоядерных вычислительных системах // *Сибирский журнал индустриальной математики.* 2014. Т 17, № 1. С. 55–64.
6. Воронин К.В. Численное исследование mpi/openmp реализации на основе асинхронной работы с потоками для трехмерной схемы расщепления в задачах теплопереноса // *Сибирский журнал индустриальной математики.* 2014. Т 17, № 2. С. 41–49.
7. Starchenko A.V., Bart A.A., Bogoslovskiy N.N., Danilkin E.A., Terenteva M.A. Mathematical modelling of atmospheric processes above an industrial centre // *Proc. of SPIE 9292, 20th Int. Symposium on Atmospheric and Ocean Optics: Atmospheric Physics.* 2014. V. 9292. P. 929249-1–929249-30.
8. Carpenter K. Note on the Paper «Radiation Condition for the Lateral Boundaries of Limited-Area Numerical Models» by Miller, M. and Thorpe, A. (vol. 107. pp. 615–628) // *J. of Royal Meteorology Society.* 1982. V. 108. P. 717–719.
9. Tolstykh M.A. Semi-Lagrangian high-resolution atmospheric model for numerical weather prediction // *Russian Meteorology and Hydrology.* 2001. № 4. P. 1–9.
10. Семёнова А.А., Старченко А.В. Разностная схема для нестационарного уравнения переноса, построенная с использованием локальных весовых интерполяционных кубических сплайнов // *Вестник Томского государственного университета. Математика и механика.* 2017. № 49. С. 61–74. DOI: 10.17223/19988621/49/6
11. Старченко А.В., Берцун В.Н. Методы параллельных вычислений. Томск : Изд-во Том. ун-та, 2013. 234 с.
12. Антонов А.С. Параллельное программирование с использованием технологии OpenMP. М. : Изд-во МГУ, 2009. 77 с.
13. Романенко А.А. Особенности адаптации программ под GPU с использованием технологии OpenACC. Новосибирск : ИЦ НГУ, 2016. 33 с.

Поступила в редакцию 15 июля 2019 г.

Semenov E.V., Starchenko A.V., Danilkin E.A., Prohanov S.A. (2020) THE CHOICE OF TECHNOLOGY FOR PARALLELIZATION OF NUMERICAL THE SOLUTION OF THE CONVECTIVE-DIFFUSION EQUATION ON HYBRID MULTIPROCESSOR COMPUTER SYSTEM. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 51. pp. 111–120

DOI: 10.17223/19988605/51/13

This work is devoted to comparison of three technologies for building parallel algorithms, by the example of numerical solution of the nonstationary 3D convection-diffusion equation. This equation is the basis of many models of continuum mechanics, hydrometeorology, ecology, and others. When doing operational modeling of atmospheric or hydrological processes, it is necessary to present the results of prognostic modeling in the shortest possible time computing (on a computer). The use of multiprocessor and multicore computing equipment gives a great opportunity to accelerate the process of obtaining a numerical solution.

This work is aimed to carry out computational experiments on a hybrid computing system in order to determine the most promising use of parallel computing technologies (MPI, OpenMP, OpenACC) to obtain efficient parallel programs for solving the nonstationary 3D convection-diffusion equation.

When applying MPI technology as the main approach of parallelization for a distributed-memory computer system, a two-dimensional grid domain decomposition into subdomains was chosen. The calculations done on TSU Cyberia computing cluster showed sufficiently high efficiency of the chosen method for parallelization of the explicit-implicit computational algorithm.

Open Multi-Processing technology is used for multiprocessor (multicore) computer systems with shared RAM. The results showed that OpenMP-program is inferior to MPI-program in terms of acceleration; however, in absolute values, operate time of OpenMP-program is slightly shorter for each parallel start (the difference decreases with the increase in the number of processes).

The calculation results of the parallel program created using OpenACC technology on one computing node of TSU Cyberia cluster showed about 3 seconds execution time, which is significantly shorter than the execution time of the sequential program, and several times shorter than that of MPI- and OpenMP-programs on 12 or 16 processes on computing nodes of Cyberia cluster.

The calculations done on structured difference grids with more than 2 million nodes showed that the most high-performance program for solving the task in question can be obtained using MPI technology. It accelerates calculations in almost 90 times in comparison with the calculation of the sequential program on 128 processes (10-20 CPUs connected with computer network). More “low-end” options include the use of OpenACC technology for NVIDIA graphic processors (1 CPU+1GPU on one computing node) which accelerates calculations in almost 30 times under the same conditions, and the use of OpenMP technology (2 CPUs on one computing node) accelerates calculations in 10 times.

Keywords: parallel computing; MPI; OpenMP; OpenACC; convective-diffusion equation; explicit-implicit differencing schemes.

SEMENOV Evgenii Vitalevich (Post-graduate Student, Department of Computational Mathematics and Computer Modelling of National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: semyonov@math.tsu.ru

STARCHENKO Alexander Vasilevich (Professor, Doctor of Physical and Mathematical Sciences, Head of Department of Computational Mathematics and Computer Modelling of National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: starch@math.tsu.ru

DANILKIN Evgeniy Alexandrovich (Candidate of Physics and Mathematics, Assistant Professor of the Department of Computational Mathematics and Computer Modelling, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: ugin@math.tsu.ru

PROHANOV Sergey Anatolevich (Programmer of the Interregional Supercomputer Center of the National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: viking@mail.tsu.ru

REFERENCES

1. Powers, J.G., Klemp, J.B., Skamarock, W.C., Davis, C.A. et al. (2017) The Weather Research and Forecasting Model: Overview, System Efforts and Future Directions. *Bulletin of American Meteorology Society*. 98. pp. 1717–1737. DOI 10.1175/BAMS-D-15-00308.1
2. Ridwan, R., Kistijantoro, A.I., Kudsy, M. & Gunawan, D. (2015) Performance evaluation of hybrid parallel computing for WRF model with CUDA and OpenMP. *3rd International Conference on Information and Communication Technology (ICoICT)*. Nusa Dua. pp. 425–430. DOI: 10.1109/ICoICT.2015.7231463
3. Bermous, I. & Steinle, P. (2015) Efficient performance of the Met Office Unified Model v8.2 on Intel Xeon partially used nodes. *Geoscientific Model Development*. 8. pp. 769–779. DOI: 10.5194/gmd-8-769-2015
4. Mortikov, E.V. (2016) [Software implementation of impurity transfer unit in climate models based on hybrid programming MPI-OpenMP]. *Superkomp'yuternye dni v Rossii* [Russian Supercomputing Days]. Proc. of the International Conference. Moscow State University. pp. 521–529.
5. Kandryukova, T.A. & Laevskii, Yu.M. (2014) Simulating the filtration combustion of gases on multi-core computers. *Journal of Applied and Industrial Mathematics*. 8(2). pp. 218–226. DOI: 10.1134/S1990478914020082
6. Voronin, K.V. (2014) A numerical study of an MPI/OpenMP implementation based on asynchronous threads for a three-dimensional splitting scheme in heat transfer problems. *Journal of Applied and Industrial Mathematics*. 8(3). pp. 436–443. DOI: 10.1134/S199047891403017X
7. Starchenko, A.V., Bart, A.A., Bogoslovsky, N.N., Danilkin, E.A. & Terenteva, M.A. (2014) Mathematical modelling of atmospheric processes above an industrial centre. *Proc. of SPIE 9292, 20th Int. Symposium on Atmospheric and Ocean Optics: Atmospheric Physics*. 9292. pp. 929249-1–929249-30. DOI: 10.1117/12.2075164
8. Carpenter, K. (1982) Note on the Paper “Radiation Condition for the Lateral Boundaries of Limited-Area Numerical Models” by Miller, M. and Thorpe, A. (vol. 107. pp. 615–628). *Journal of Royal Meteorology Society*. 108. pp. 717–719.
9. Tolstykh, M.A. (2001) Semi-Lagrangian high-resolution atmospheric model for numerical weather prediction. *Russian Meteorology and Hydrology*. 4. pp. 1–9.
10. Semyonova, A.A. & Starchenko, A.V. (2017) The finite-difference scheme for the unsteady convection-diffusion equation based on weighted local cubic spline interpolation. *Vestnik Tomskogo gosudarstvennogo universiteta. Matematika i mekhanika – Tomsk State University Journal of Mathematics and Mechanics*. 49. pp. 61–74. DOI: 10.17223/19988621/49/6.
11. Starchenko, A.V. & Bertsun, V.N. (2013) *Metody paralel'nykh vychisleniy* [Parallel Computing Methods]. Tomsk: Tomsk State University.
12. Antonov, A.S. (2009) *Paralel'noe programmirovaniye s ispol'zovaniem tekhnologii OpenMP* [Parallel programming using OpenMP technology]. Moscow: Moscow State University.
13. Romanenko, A.A. (2016) *Osobennosti adaptatsii programm pod GPU s ispol'zovaniem tekhnologii OpenACC* [Features of the adaptation programs under the GPU using technology OpenACC]. Novosibirsk: Novosibirsk State University.