Параллельная реализация метода конечных элементов для начально-краевой задачи мелкой воды*

Е. Д. Карепова, В. В. Шайдуров
Учреждение Российской академии наук
Институт вычислительного моделирования СО РАН, Красноярск, Россия
e-mail: jane@icm.krasn.ru, shaidurov@icm.krasn.ru

Проведено исследование эффективности нескольких параллельных реализаций алгоритма численного решения начально-краевой задачи для уравнений мелкой воды, выполненных с помощью библиотеки MPI для языка Си. Рассмотрены два подхода к декомпозиции вычислительной области и две схемы реализации двухточечных обменов в алгоритме. Приведены сравнительные результаты ускорения вычислений в зависимости от количества процессов, способа реализации коммуникаций, способа декомпозиции вычислительной области, архитектуры кластерной системы.

Ключевые слова: модели мелкой воды, декомпозиция согласованной триангуляции, метод конечных элементов, блокирующие/неблокирующие передачи, ускорение вычислений.

1. Моделирование поверхностных волн в водоемах методом конечных элементов

Модели мелкой воды хорошо описывают большой круг природных явлений, таких как крупномасштабные поверхностные волны, возникающие в морях и океанах, цунами, приливные течения, поверхностный и русловой сток, гравитационные колебания поверхности океанов [1, 2]. В работах [2 – 4] рассмотрено численное моделирование поверхностных волн в больших акваториях с учетом сферичности Земли и ускорения Кориолиса на основе уравнений мелкой воды. В [2] для дифференциальной постановки задачи выведены априорные оценки, обеспечивающие устойчивость решения и однозначную разрешимость задачи. В [3, 4] для этой же задачи построен метод конечных элементов (МКЭ), для которого получены необходимые априорные оценки. Там же приведены результаты численных экспериментов на модельных сетках для акваторий Охотского моря и Мирового океана.

Поскольку для расчетов на реальных акваториях объем вычислений велик, то оправданно применение высокопроизводительных ЭВМ. В [7] приведены некоторые данные по ускорению параллельного алгоритма для данной задачи, требующие некоторого уточнения. Цель данной работы — тщательное исследование эффективности реализации параллелизма по данным для рассматриваемой задачи и особенностей поведения кластерных систем различной архитектуры при расчетах.

*Работа выполнена при поддержке РФФИ (грант № 08-01-00621).

Рассмотрим задачу в следующей постановке. Для стандартной сферической системы координат (r, λ, θ) с началом в центре земного шара будем использовать вместо угла θ географическую широту $\varphi = \pi - \theta$, так что $0 \le \varphi < \pi$. Через λ обозначена географическая долгота $0 \le \lambda \le 2\pi$. Полагаем всюду $r = R_E$, где R_E — радиус Земли, который считается постоянным. Пусть Ω — заданная область на сфере с границей $\Gamma{=}\Gamma_1 \cup \Gamma_2$, где Γ_1 — часть границы, проходящая вдоль берега, а $\Gamma_2 = \Gamma \setminus \Gamma_1$ — часть границы, проходящая по морю. Обозначим через m_1 и m_2 характеристические функции соответствующих участков границы. Для простоты считается, что полюса $\varphi = 0$ и $\varphi = \pi$ не входят в Ω. Относительно неизвестных функций $u = u(t, \lambda, \varphi)$, $v = v(t, \lambda, \varphi)$ и $\xi = \xi(t, \lambda, \varphi)$ запишем в $\Omega \times (0, T)$ уравнения баланса импульсов и уравнение неразрывности [2]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= lv + mg\,\frac{\partial \xi}{\partial \lambda} - R_f u + f_1, \\ \frac{\partial v}{\partial t} &= -lu + ng\,\frac{\partial \xi}{\partial \varphi} - R_f v + f_2, \\ \frac{\partial \xi}{\partial t} &= m\left(\frac{\partial}{\partial \lambda}(Hu) + \frac{\partial}{\partial \varphi}\left(\frac{n}{m} Hv\right)\right) + f_3, \end{aligned} \tag{1}$$

где u, v — компоненты вектора скорости $\mathbf{U}$ по осям λ и φ соответственно; ξ — отклонение свободной поверхности от невозмущенного уровня; $H(\lambda, \varphi) > 0$ — глубина водоема в точке (λ, φ); функция $R_f = r_*|\mathbf{U}|/H$ учитывает силу трения о дно, r_* — коэффициент трения; $l = -2\omega \cos\varphi$ — параметр Кориолиса; $m = 1/(R_E \sin\varphi)$; $n = 1/R_E$; g — ускорение силы тяжести; $f_1 = f_1(t, \lambda, \varphi)$, $f_2 = f_2(t, \lambda, \varphi)$ и $f_3 = f_3(t, \lambda, \varphi)$ — заданные функции внешних воздействий.

Граничные условия рассмотрим в следующем виде [2]:

$$HU_n + \beta m_2 \sqrt{gH}\,\xi = m_2 \sqrt{gH}\, d \quad \text{на} \quad \Gamma \times (0, T), \tag{2}$$

где $U_n = \mathbf{U} \cdot \mathbf{n}$, $\mathbf{n} = \left(n_1, \dfrac{n}{m} n_2\right)$ — вектор внешней нормали к границе; $\beta \in [0, 1]$ — заданный параметр, $d = d(t, \lambda, \varphi)$ — заданная граничная функция, определенная на границе Γ_2.

Зададим также начальные условия

$$u(0, \lambda, \varphi) = u_0(\lambda, \varphi),\ v(0, \lambda, \varphi) = v_0(\lambda, \varphi),\ \xi(0, \lambda, \varphi) = \xi_0(\lambda, \varphi). \tag{3}$$

Рассмотрим некоторую согласованную триангуляцию $\mathcal{T} = \{\omega_i\}\big|_{i=1}^{N_{el}}$ области Ω, состоящую из невырожденных треугольников с прямолинейными сторонами в координатах λ и φ и содержащую область Ω. Сетка в общем случае может быть неструктурированной, но обязательно является согласованной, т. е. каждое ребро является либо стороной двух треугольников, либо стороной одного треугольника, находящейся на границе области (рис. 1). Пусть $\bar{\Omega}_h$ — множество узлов (т. е. вершин треугольных элементов) общим числом N_{nd}, Ω_h — множество внутренних узлов. Для каждого узла $z_j \in \bar{\Omega}_h$ введем базисную функцию $\Psi_j(\lambda, \varphi)$, которая равна единице в z_j, равна нулю во всех остальных узлах $\bar{\Omega}_h$ и линейна в каждом треугольнике. Обозначим линейную оболочку этих функций через $H_h(\Omega_h) = \operatorname{span}\{\Psi_j\}_{j=1}^{N_{nd}}$.

Для действительных вектор-функций $\mathbf{\Phi}_h = (u^h, v^h, \xi^h)$, $\hat{\mathbf{\Phi}}_h = (\hat{u}^h, \hat{v}^h, \hat{\xi}^h) \in \mathbf{H}_h(\Omega_h) \equiv (H_h(\Omega_h))^3$ рассмотрим [2–4] дискретный аналог скалярного произведения

$$(\mathbf{\Phi}_h, \hat{\mathbf{\Phi}}_h)_h = \sum_{i=1}^{N_{el}} \frac{1}{3} S_i \sum_{j=0}^{2} R_E^2 \sin(\varphi_{ij}) \left(H_{ij}(u_{ij}\hat{u}_{ij} + v_{ij}\hat{v}_{ij}) + g\xi_{ij}\hat{\xi}_{ij}\right). \tag{4}$$

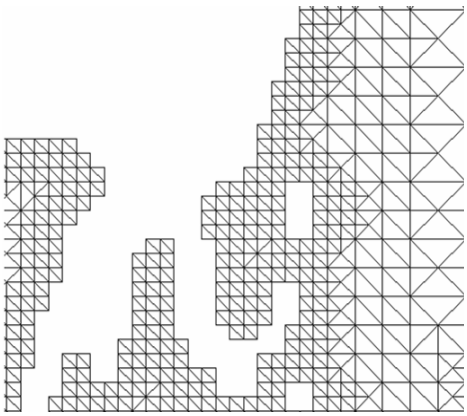

Рис. 1. Фрагмент триангуляции

Здесь через S_i обозначена площадь i-го треугольного элемента, вершины которого занумерованы через 0, 1, 2; следовательно $f_{ij} = f(\lambda_{ij}, \varphi_{ij})$ — значение функции в j-й вершине i-го элемента.

Для дискретизации по времени системы (1)–(2) разобьем временной отрезок $[0, T]$ на K интервалов: $0 = t_0 < t_1 < \cdots < t_K = T$ с шагом $\tau = T/K$. Аппроксимируем производные по времени левыми разностями. В результате на каждом временном интервале (t_k, t_{k+1}) получим полудискретный аналог системы (1)–(2), для которого в терминах скалярного произведения (4) сформулируем метод Бубнова—Галеркина [3, 4]. Для аппроксимации интегралов при получении билинейной и линейных форм используем формулу трапеций и ее двумерный аналог.

Занумеровав узлы $\bar{\Omega}_h$ от 1 до N_{nd}, запишем задачу в векторно-матричной форме: для фиксированного момента времени t^{k+1} найдем вектор $\mathbf{V}^{k+1} = (u_1, \ldots, u_{N_{nd}}, v_1, \ldots, v_{N_{nd}}, \xi_1, \ldots, \xi_{N_{nd}})$, удовлетворяющий системе линейных алгебраических уравнений

$$A^{k+1}\mathbf{V}^{k+1} = \mathbf{F}^{k+1}. \tag{5}$$

В работе [3] показан второй порядок сходимости решений в норме, порождаемой скалярным произведением (4) на равномерной сетке.

Для решения системы (5) использовался итерационный метод Якоби, который обладает хорошим параллелизмом, а диагональное преобладание для его сходимости легко обеспечивается выбором шага по времени τ. В векторно-матричной форме метод Якоби запишется в следующем виде:

$$\mathbf{\Phi}^{(\nu+\mathbf{1})} = \mathbf{\Phi}^{(\nu)} - D^{-1}\left(A\mathbf{\Phi}^{(\nu)} - \mathbf{F}\right). \tag{6}$$

Здесь ν — номер итерации, индексы $(k+1)$ для шага по времени опущены, однако компоненты глобальной матрицы жесткости A и вектора $\mathbf{F}$ правой части выражения зависят от времени и должны пересчитываться в начале каждого временного шага.

Отметим некоторые особенности реализации, диктуемые методом конечных элементов. Глобальная матрица жесткости A зависит от времени и должна пересчитываться на каждом временном шаге. Однако для реализации метода Якоби на конечных элементах не требуется явного хранения глобальной матрицы жесткости. В программе насчитываются только элементы локальных матриц жесткости (причем лишь их диагональные элементы зависят от времени и перевычисляются на каждом временном шаге). Вычис-

ление невязки $A\mathbf{\Phi}^{(\nu)} - \mathbf{F}$ в (6) производится по треугольникам с использованием элементов локальных матриц. Сходимость метода определялась малостью разности $\mathbf{\Phi}^{(\nu)}$ для двух соседних итераций в дискретном аналоге равномерной нормы:

$$\max_{1\le i\le N_{nd}} |\Phi_i^{(\nu+1)} - \Phi_i^{(\nu)}| \le \varepsilon. \tag{7}$$

2. Параллельный алгоритм

Используя явный параллелизм по данным, исходную расчетную область можно разбить на несколько частично перекрывающихся подобластей. Расчеты в каждой подобласти выполняются независимо друг от друга в рамках итерации Якоби. После каждой итерации Якоби необходимо проводить согласование данных в перекрытиях. Имеют место по крайней мере два следующих варианта разбиения.

1. *Декомпозиция с теневыми гранями.* Исходная область включает взаимно перекрывающиеся подобласти, определяемые шаблоном. При этом невязка в граничных точках подобласти i-го процесса насчитывается в подобластях соседних процессов. С учетом семиточечного шаблона и согласованности триангуляции достаточно перекрытия областей в два слоя расчетных точек. Поскольку алгоритм вычисления невязки требует хранения в каждой граничной точке подобласти семи коэффициентов матрицы жесткости, трех значений вектора решения текущей и предыдущей итераций и одного значения правой части, то избыточность хранения информации для p-процессов составляет $56(p-1)N_{bnd} * \mathsf{SizeOfDouble}$ байт. Здесь через N_{bnd} обозначено количество точек на разрезе, $\mathsf{SizeOfDouble}$ — количество байт, занимаемых переменной, хранимой с двойной точностью.

2. *Декомпозиция без перекрытий.* Исходная область разрезается на подобласти, пересекающиеся только по границам разреза. Для каждой граничной точки подобласти невязка насчитывается частично только по тем треугольникам, которые лежат в подобласти. При обмене данными после каждой итерации Якоби требуется дополнительное суммирование для значений невязки в граничных для подобласти точках. Этот способ декомпозиции более экономичен по памяти, прост в программировании, однако предполагает дополнительные арифметические операции на каждой итерации Якоби. Очевидно его достоинство для неструктурированных сеток, когда границы подобластей не являются последовательным множеством точек.

В рамках выбранной схемы распределения данных все процессы осуществляют одни и те же вычисления, но над разными подобластями. Структура обменов, за исключением первого и последнего процессов, также является однородной. Обмены необходимы на каждой итерации метода Якоби.

Реализация параллельной программы осуществлялась на языке программирования Си с применением функций библиотеки передачи сообщений MPI.

Выполним теоретические оценки возможного ускорения каждого из параллельных алгоритмов, следуя [8].

Обозначим время выполнения одной арифметической операции t_{op}, а время выполнения пересылки одного значения — t_{comm}. Пусть N_{nd} — общее количество точек сетки расчетной области, s — количество операций, выполняемых в одной расчетной точке на каждой итерации Якоби, k — количество шагов по времени, ν — среднее количество итераций Якоби на каждом временном шаге. Тогда общий объем вычислений V_{calc} в ал-

горитме определяется соотношением $V_{calc} = k\nu s N_{nd}$, а время выполнения алгоритма на одном процессоре соответственно можно оценить следующим образом: $T_1 \sim k\nu s N_{nd} t_{op}$.

Потенциальное ускорение алгоритма оценивается как отношение времени вычисления на одном процессоре T_1 к времени вычислений на p-процессах T_p: $S_p = \dfrac{T_1}{T_p}$. Выполним теоретические оценки ускорения для каждого рассматриваемого случая декомпозиции области, учитывая время, затрачиваемое на выполнение обменов, а также на возможные накладные расходы при вычислениях в перекрывающихся подобластях.

Предположим, что при декомпозиции области нам удается равномерно распределить весь объем вычислений V_{calc} по процессорам. Тогда для ускорения параллельного алгоритма можно записать следующее:

$$S_p = \frac{T_1}{T_1/p + T_{over} + T_{comm}}. \tag{8}$$

Здесь T_1 — время вычислений на одном процессоре, T_{over} — время на дополнительные вычисления, связанные с декомпозицией области, T_{comm} — время, требуемое для обменов.

Как следует из принятой нами схемы распределения данных, на каждой итерации метода Якоби требуются:

1) глобальная операция приведения для вычисления равномерной нормы разности между двумя последовательными приближениями итераций Якоби;

2) обмен граничными элементами вектора решений на разрезах;

3) дополнительные вычисления, порождаемые распределенностью данных.

Пусть g — количество дополнительных вычислений, связанных с декомпозицией области, которые необходимы для одной граничной в подобласти точки. Декомпозиция с теневыми гранями не требует прямых дополнительных вычислений, т. е. $g = 0$. При декомпозиции без теневых граней каждый процесс, вычисляя невязку в граничной в подобласти точке, "обрабатывает" свою часть треугольников. В этом случае после обмена требуется дополнительное суммирование частей невязок, насчитанных в соседнем процессоре. Поскольку в каждой точке вычисляется невязка для трех компонент вектора (u, v, ξ), то $g = 3$. В результате объем дополнительных вычислений $V_{over} \sim 2k\nu g N_{bnd}(p-1)$. Поскольку суммирование процессы выполняют асинхронно и независимо, то время, затрачиваемое на дополнительные вычисления в N_{bnd} граничных точках подобласти, можно оценить следующим образом:

$$T_{over} \sim 2k\nu g N_{bnd} t_{op}.$$

Для вычисления критерия остановки итерационного процесса на каждой итерации требуется вычислить глобальный максимум по всем процессам. Предположим, что реализация глобальных операций приведения в MPI выполняется по оптимальному алгоритму сдваивания, что дает время выполнения одной операции приведения $T^1_{comm} = (t_{op} + t_{comm}) \log_2 p$. Пусть также для каждой граничной точки требуется пересылка m значений (в нашем случае $m = 3$). Объем пересылаемых данных каждым процессом соседу можно оценить как

$$V_{comm} \sim k\nu m N_{bnd}.$$

При использовании библиотеки MPI возможно два принципиально разных способа организации обменов — с использованием блокирующих или неблокирующих функций передачи данных. Оценим время, необходимое для обменов в обоих случаях.

Блокирующие передачи. На рис. 2 показана схема реализации обменов по цепочке процессов с помощью функций совмещенных приема-передачи MPI_Sendrecv(...), отражающая влияние блокирующих передач на производительность на примере восьми процессов. На схеме операции отправки данных правому и левому соседу обозначены через Snd_R и Snd_L соответственно, а операции приема от правого и левого соседа — через Rcv_R и Rcv_L. Сначала все процессы, кроме последнего, отправляют данные своим правым соседям и от них же ожидают поступления данных. Затем все процессы, кроме первого, посылают данные своим левым соседям и ожидают поступления данных от них же. Поскольку на первом этапе последний процесс не передает данные вправо, то он успешно выполняет операцию приема от шестого (левого) процесса и передает ему свои данные, в то время как процессы с нулевого по пятый простаивают в ожидании приема данных своими правыми соседями. На втором этапе шестой процесс получает данные от пятого и осуществляет передачу ему своих данных, а процессы с нулевого по четвертый простаивают. Таким образом происходит разрешение всех обменов по цепочке. Все обмены завершатся за $p+1$ этап.

В общем случае обмены с использованием блокирующих передач потребуют $2(p-1)$ пересылок V_{comm} единиц данных, где p — количество используемых процессов. Тогда $T^2_{comm} = 2(p-1)k\nu m N_{bnd} t_{comm}$ — общие затраты на обмены в блокирующем режиме.

С учетом (8) и всех проведенных оценок ускорение при использовании блокирующих передач будет определяться следующим соотношением:

$$S^{bl}_p = \frac{1}{\dfrac{1}{p} + 2\dfrac{g}{s}R + \dfrac{\log_2 p}{sN_{nd}}(1+\varkappa) + 2(p-1)\dfrac{m}{s}R\varkappa}. \tag{9}$$

Здесь R — отношение количества граничных точек в подобласти к общему числу точек расчетной области: $R = N_{bnd}/N_{nd}$, $\varkappa$ — отношение времени пересылки одного значения к времени выполнения одной арифметической операции: $\varkappa = t_{comm}/t_{op}$.

Неблокирующие передачи. В общем случае время, затрачиваемое на обмены с использованием неблокирующих передач, не зависит от количества участвующих

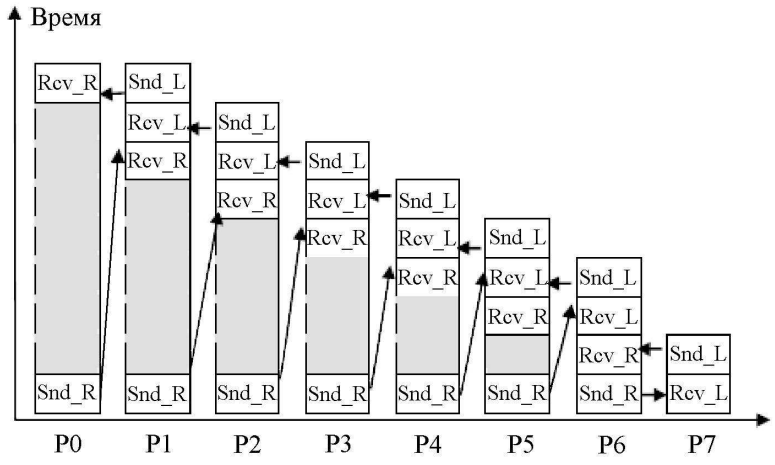

Рис. 2. Схема реализации блокирующих обменов

в обменах процессов: $T^2_{comm} = 2V_{comm}t_{comm} = 2k\nu mN_{bnd}t_{comm}$. Оценка (8) в случае неблокирующих обменов имеет вид

$$S_p^{unbl} = \frac{1}{\frac{1}{p} + 2\frac{g}{s}R + \frac{\log_2 p}{sN_{nd}}(1+\varkappa) + 2\frac{m}{s}R\varkappa}. \tag{10}$$

Из (9), (10) следует, что для достаточно мелких сеток потенциальное ускорение близко к линейному на достаточно большом диапазоне количества процессов. Величина ускорения определяется двумя параметрами. Первый из них $R = N_{bnd}/N_{nd}$ характеризует декомпозицию расчетной области. Приемлемое ускорение обеспечивается малостью R, поэтому при построении декомпозиций сложных вычислительных областей наряду с требованием равенства вычислительной нагрузки на процесс необходимо обеспечивать минимальную протяженность границы подобласти для каждого процесса. Второй параметр $\varkappa = t_{comm}/t_{op}$ характеризует коммуникационную среду. Этот параметр описывает вычислительную сеть очень условно, но он показывает, что для приемлемого ускорения следует выбирать архитектуру кластера с небольшими значениями $\varkappa$. Расчеты показали, что неоднородность архитектуры высокопроизводительной системы приводит к непредсказуемому поведению $\varkappa$ в зависимости от количества процессов, конкретных вычислительных узлов, на которых будет выполняться задача, сложившейся общей ситуации на кластере (т. е. задач других пользователей). Отметим также, что величина T_{over} на несколько порядков меньше, чем другие слагаемые знаменателя в (8) и не зависит от количества процессов. С учетом экономии памяти и легкости реализации на неструктурированных сетках это дает преимущество декомпозиции без перекрытий. Кроме того, неблокирующие передачи, даже в ситуации, когда нет совмещения вычислений и обменов, несколько предпочтительней блокирующих пересылок.

3. Вычислительный эксперимент

3.1. Модельная область

Для численного исследования ускорения параллельного алгоритма рассмотрим следующую модельную задачу. Пусть Ω — "квадрат" на сфере: $\Omega = [0, \pi/10] \times [\pi/2, \pi/2 + \pi/10]$. Границы Ω считаются "твердыми". В Ω рассмотрена задача с известным точным решением [3]. В расчетной области построены две равномерные квадратные сетки 401×401 и 801×801 точек с соответствующими согласованными триангуляциями. В вычислительных экспериментах было сделано 1000 шагов по времени.

Вычислительный эксперимент проведен на трех высокопроизводительных комплексах.

Во-первых, вычисления выполнялись на 99-ядерном кластере Института вычислительного моделирования СО РАН. Кластер МВС-1000/ИВМ (собственная сборка ИВМ СО РАН [12]) содержит 27 вычислительных узлов AMD Athlon64/3500+/1Гб (однопроцессорные, одноядерные); 12 вычислительных узлов AMD Athlon64 X2 Dual Core/4800+/2Гб (однопроцессорные, двухъядерные); 12 вычислительных узлов 2XDual-Core AMD Opteron Processor 2216/4Гб (двухпроцессорные, двухъядерные). Управляющий узел, сервер доступа и файловый сервер — Athlon64/3500+/1Гб с общей дисковой памятью 400 Гб под управлением ОС Gentoo Linux. Управляющая сеть — FastEthernet, сеть передачи данных — GigaEthernet.

Вторым кластером, на котором проведены серии расчетов, является кластер Томского государственного университета SKIF Cyberia, содержащий 283 двухпроцессорных двухъядерных узла (1132 ядра) IntelXeon5150 2.66 Ггц с суммарным объемом дисковой памяти 1136 Гб и объемом дискового пространства 22.56 Тб. Внешняя дисковая система хранения данных имеет объем 10 Тб. Все узлы объединены высокопроизводительной сетью передачи данных InfiniBand.

Наконец, эксперименты были частично повторены на высокопроизводительном вычислительном комплексе Информационного вычислительного центра Новосибирского государственного университета, на базе кластера из 64 восьмиядерных вычислительных узлов, основанных на блэйд-серверах Hewlett-Packard BL460c. Все вычислительные узлы объединены высокопроизводительной сетью InfiniBand DDR 4x. В кластере используется параллельная файловая система хранения SFS емкостью 24 Тб, основанная на технологии Lustre, скорость чтения СХД — более 1 Гб/с, записи — 800 Мб/с.

Отличительной чертой первого кластера является его гетерогенная архитектура. Кластер SKIF Cyberia и HP-кластер ИВЦ НГУ отличаются в интересующем нас плане количеством ядер на узел.

Исследования проводились на грубой и мелкой сетке — 401×401 и 801×801 точек соответственно. Ясно, что потенциальное ускорение параллельной программы зависит от размерности сетки. Исследования зависимости ускорения проводились вплоть до 32 процессов, где на рассматриваемых сетках алгоритм хорошо масштабируем. Временные характеристики замерялись средствами MPI каждым процессом в отдельности, за измеряемый показатель принималось максимальное значение. Все временные характеристики осреднялись по результатам нескольких десятков расчетов, исключая экстремальные значения. Дисперсия измерений, кроме некоторых расчетов на кластере ИВМ СО РАН с гетерогенной архитектурой, как правило, была незначительна. Численные эксперименты показали несомненное преимущество однородной архитектуры высокопроизводительной вычислительной системы.

Результаты серии расчетов на всех кластерных системах на грубой модельной сетке 401×401 точек начиная с девяти процессов показали на неблокирующих режимах обмена для обоих вариантов декомпозиции ускорение выше линейного, что можно объяснить эффектом “попадания в кэш”. Однако для кластера ИВМ СО РАН при блокирующих обменах эффект попадания в кэш также наблюдается, но начиная с 12 процессов рост ускорения прекращается. Существенных отличий ускорений для разных видов декомпозиции не наблюдается.

На рис. 3 приведены зависимости ускорения вычислений от количества используемых процессов для мелкой сетки, полученные на кластерах ИВМ СО РАН и SKIF Cyberia. Для сравнения представлен график потенциального ускорения согласно оценке (10). Чтобы не загромождать рисунок, не показаны графики ускорений, полученных для декомпозиции с теневыми гранями, поскольку они практически совпадают с представленными. В экспериментах на кластере ИВМ СО РАН общий тренд ускорения совпадает с теоретическими оценками, однако его рост носит сильно неустойчивый характер. Расчеты, проведенные на кластере SRIF Cyberia, показывают классическую картину ускорения, подтверждающую линейный характер его роста с увеличением количества процессов с эффективностью около единицы (эффективность расчета для 32 узлов ≈ 0.85). Эксперименты на кластере НГУ подтверждают эти результаты. Таким образом, неустойчивый характер ускорения в расчетах, проведенных на кластере ИВМ СО РАН, объясняется скорее всего его неоднородной архитектурой.

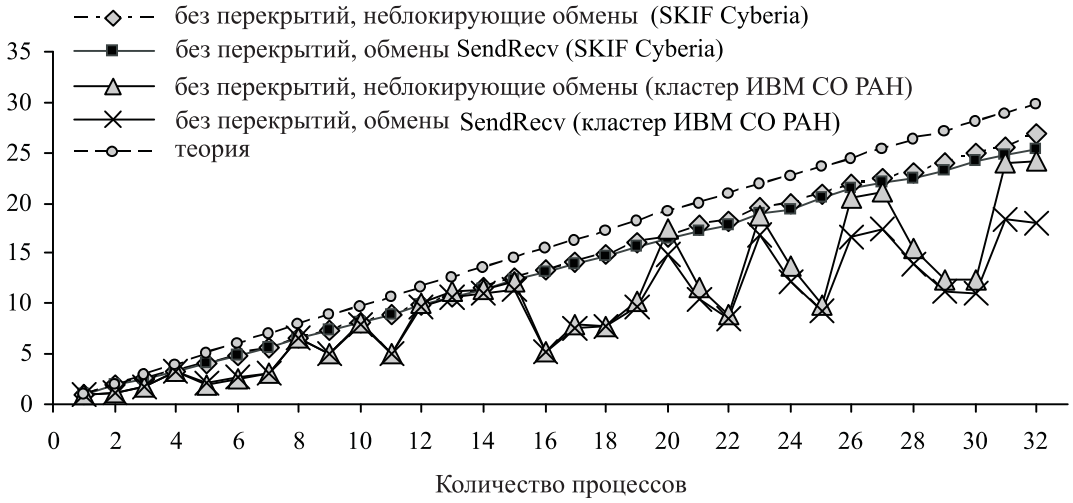

Рис. 3. Зависимость ускорения вычислений (ось ординат) от количества доступных процессов; сетка 801×801

Отметим, что тип декомпозиции существенно не влияет на ускорение параллельной программы, а неблокирующие операции следует признать более надежными и эффективными.

Поскольку время выполнения программы складывается из времени вычислений, времени коммуникаций и дополнительного времени для операций, связанных с распределенностью данных, то были проведены серии расчетов, в которых названные составляющие были измерены отдельно. На рис. 4 приведены результаты этих экспериментов для случая декомпозиции без перекрытий и неблокирующего режима двухточечных обменов. Для того чтобы была возможность сравнить результаты расчетов с теоретическими оценками, рассмотрены зависимости безразмерных величин — отношения времени выполнения вычислений и времени выполнения обменов для p-процессов к времени выполнения всего алгоритма для одного процесса: $\beta_{calc} = T_p^{calc}/T_1$ и $\beta_{comm} = T_p^{comm}/T_1$. Время на дополнительные операции не зависит от количества участвующих в вычислениях процессов и на пять-шесть порядков меньше суммарного времени вычислений и коммуникаций. Время выполнения вычислений (см. рис. 4 вверху) для гомогенных архитектур совпадает с теоретическими оценками. Ввиду неоднородности узлов кластера ИВМ СО РАН время вычислений в этом случае уменьшается немонотонно.

Схема двухточечных обменов алгоритма, диктуемая декомпозицией, начиная с трех процессов неизменна: каждый процесс, за исключением первого и последнего, на каждой итерации Якоби пересылает для компонент скорости и возвышения свободной поверхности невязку в граничных точках своим левым и правым соседям (шесть операций посылок и шесть операций получения). В результате время выполнения обменов в неблокирующем режиме не зависит от количества процессов, участвующих в расчетах. Для кластера с неоднородной архитектурой наблюдается ожидаемое немонотонное поведение исследуемой величины. Однако немонотонное поведение наблюдается и в случае расчетов для гомогенных архитектур. Исследование времени выполнения обменов показало следующее: 1) время обменов минимально и не зависит от количества процессов, участвующих в обменах, если загружены все ядра на узле по одному процессу на ядро (для SKIF Cyberia количество процессов кратно четырем, для HP-кластера НГУ — восьми); 2) при существовании в расчетах узлов, не полностью загруженных,

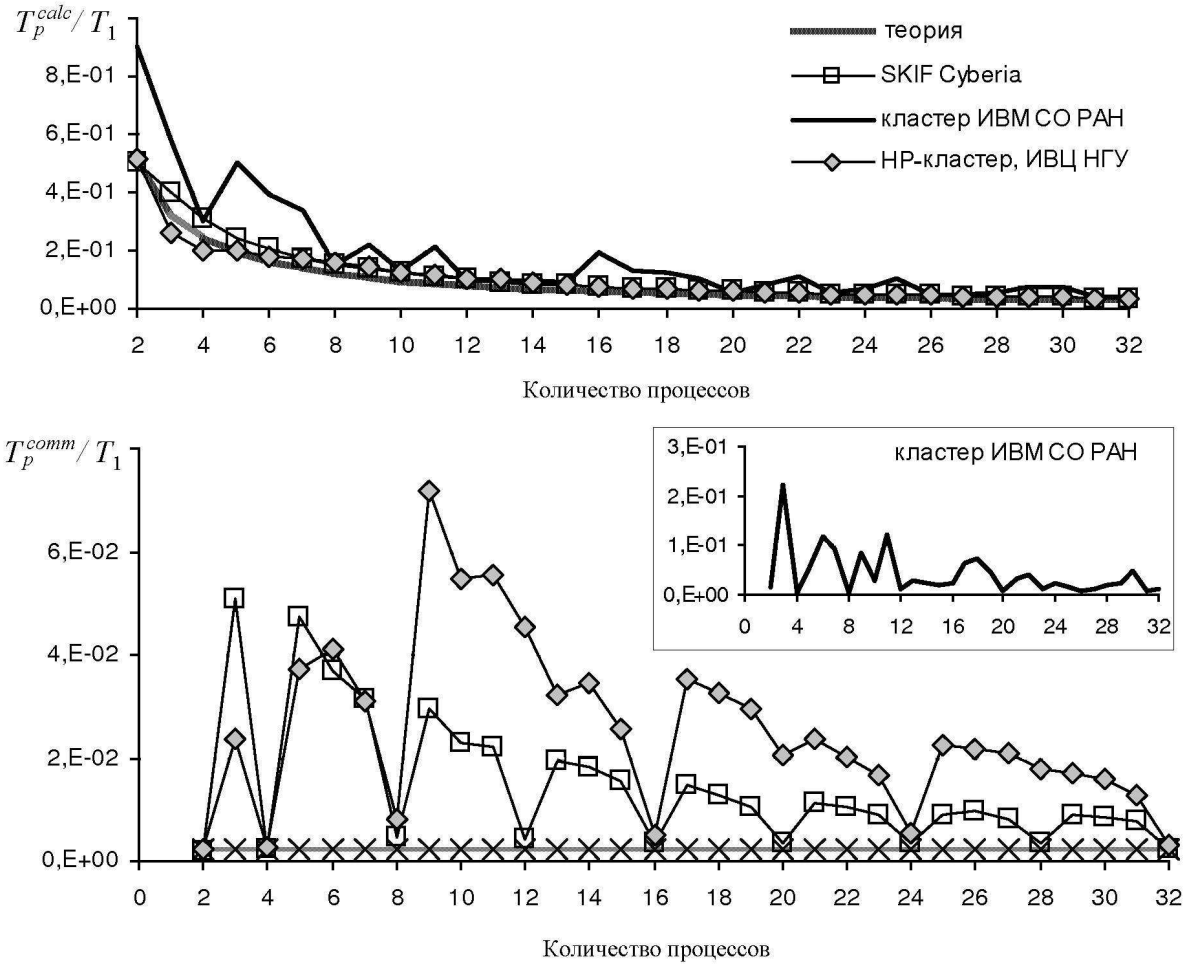

Рис. 4. Исследование времени счета и времени коммуникаций; неблокирующие обмены, декомпозиция без перекрытий

время обменов тем больше, чем больше количество простаивающих ядер; 3) время, затраченное на обмены, уменьшается с ростом количества задействованных процессов (т. е. с уменьшением времени вычислений).

3.2. Эксперименты для акватории Охотского моря

Тестовые расчеты для акватории Охотского моря проводились на сетках, подготовленных С.Ф. Пятаевым и И.В. Киреевым на основе открытой батиметрической базы данных ETOPO2 [6, 11].

Численный эксперимент соответствует начальным данным с локальным подъемом уровня, описываемым гауссовской функцией

$$\xi(0,\lambda,\varphi) = A\exp\bigl(-(\lambda-\lambda_0)^2/(2D)^2 - (\varphi-\varphi_0)^2/(2D)^2\bigr).$$

В расчетах было принято $A = 10$, $\lambda_0 = 149.1°$ восточной долготы, $\varphi_0 = 53.1°$ северной широты, $D = 0.005$, коэффициент трения $r_* = 0.0026$. Для скоростей предполагались нулевые начальные данные: $u(0,\lambda,\varphi) = v(0,\lambda,\varphi) = 0$.

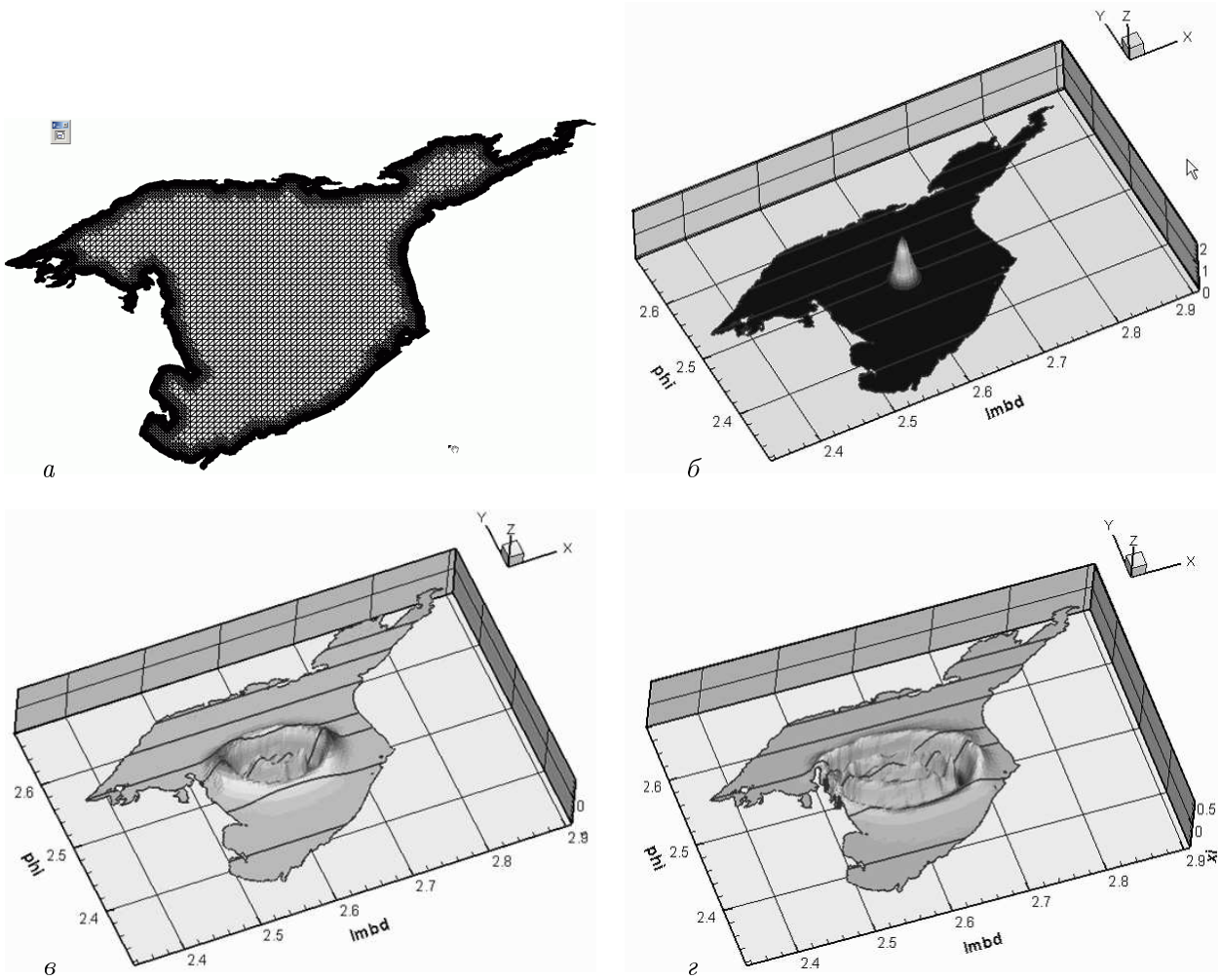

Рис. 5. Численные эксперименты в акватории Охотского моря на восьми процессах: *а* — общий вид сетки; *б* — начальное возмущение; уровень свободной поверхности через 42 мин (*в*), через 67 мин (*г*)

В построенной сетке из общего числа граничных участков около 6.4 % проходят по морю. На таких участках $m_2 = 1$, и в краевом условии (2) полагали $\beta = 1$, $d(t, \lambda, \varphi) \equiv 0$.

Результаты численного моделирования показаны на рис. 5, *б* — *г*, где представлена функция $\xi(t, \lambda, \varphi)$ для некоторых моментов времени. Линиями показаны границы, получившиеся при декомпозиции области на восемь частей. Поскольку сетка сильно сгущается в приграничных областях, ширина полосы, которая отводится одному процессу, в прибрежной зоне значительно уже, чем в море. Следует отметить, что декомпозиция области с теневыми гранями в этом случае является трудоемким процессом, поэтому не проводилась.

Заключение

На примере численного решения краевой задачи для уравнений мелкой воды в работе рассмотрены технологические аспекты разработки масштабируемых параллельных алгоритмов для кластерных вычислительных систем с использованием библиотеки MPI.

Поскольку при дискретизации задачи использовался метод конечных элементов с организацией вычислений по треугольным элементам, были рассмотрены два естественных подхода к декомпозиции области — без перекрытий и с теневыми гранями.

Теоретические оценки показали, что алгоритм обладает значительным объемом потенциального параллелизма и хорошей с точки зрения распараллеливания структурой, что в зависимости от количества используемых процессов дает ускорение, близкое к линейному.

Численные эксперименты показали, что использование неблокирующего режима обменов, которое допускается алгоритмом, является безусловно более эффективным. Как явное преимущество следует отметить простоту организации параллельного алгоритма при декомпозиции без перекрытия подобластей на неструктурированных триангуляциях реальных акваторий.

Поскольку расчеты были проведены на трех высокопроизводительных ВС двух различных архитектур, то показано преимущество однородного устройства кластера (SRIF Cyberia) над гетерогенным. Кроме того, продемонстрирована неустойчивость ускорения при неоднородном составе кластера, которая не присуща ни алгоритму, ни реализации.

Авторы благодарят коллективы МВЦ ТГУ и ИВЦ НГУ, а также проф. А.В. Старченко и Д.Л. Чубарова за предоставленную возможность проведения серии вычислительных экспериментов на кластере SRIF Cyberia и кластере ИВЦ НГУ.

Список литературы

[1] Марчук Г.И., Каган Б.А. Динамика океанских приливов. Л.: Гидрометеоиздат, 1983.

[2] Agoshkov V.I. Inverse problems of the mathematical theory of tides: boundary-function problem // Rus. J. Numer. Anal. Math. Modelling. 2005. Vol. 20, N 1. P. 1–18.

[3] Kamenshchikov L.P., Karepova E.D., Shaidurov V.V. Simulation of surface waves in basins by the finite element method // Ibid. 2006. Vol. 21, N 4. P. 305–320.

[4] Kamenshchikov L.P., Karepova E.D., Shaidurov V.V. Numerical solution of the boundary problem foe shallow water equations for modelling surface waves in World ocean by finite elements methods // Finite Difference Methods: Theory and Appl. Proc. of Fourth Intern. Conf. FDM:T&A'06. Bulgaria: Rousse, 2007. P. 227–233.

[5] Каменщиков Л.П., Карепова Е.Д., Шайдуров В.В. Моделирование поверхностных волн в водоемах методом конечных элементов на вычислительном кластере // Избр. материалы Четвертой школы-семинара "Распределенные и кластерные вычисления". Красноярск: ИВМ СО РАН, 2005. С. 114–125.

[6] Каменщиков Л.П., Карепова Е.Д., Пятаев С.Ф., Шайдуров В.В. Моделирование гравитационных волн в Мировом океане методом конечных элементов с распараллеливанием // Избр. материалы Шестой школы-семинара "Распределенные и кластерные вычисления". Красноярск: ИВМ СО РАН, 2006. С. 52–64.

[7] Карепова Е.Д., Шайдуров В.В., Вдовенко М.С. Параллельные реализации метода конечных элементов для краевой задачи для уравнений мелкой воды // Вестник ЮУрГУ. Серия "Математическое моделирование и программирование". 2009. Т. 17(150), вып. 3. С. 73–85.

[8] Ортега Дж. Введение в параллельные и векторные методы решения линейных систем: Пер. с англ. М.: Мир, 1991.

[9] Balay S. Efficient management of parallelism in object-oriented numerical software libraries, modern software tools in scientific computing / Eds. S. Balay, W.D. Gropp, L.C. McInnes and others // Modern Software Tools for Scientific Computing. Birkhauser Press, 1997. P. 163–202.

[10] McBryan O.A. An overwiev of message passing environments // Parallel Computing. 1994. Vol. 20. P. 417–441.

[11] National Geophysical Data Center. `http://www.ngdc.noaa.gov/ngdc.html`

[12] Исаев С.В., Малышев А.В., Шайдуров В.В. Развитие Красноярского центра параллельных вычислений // Вычисл. технологии. 2006. Т. 11, спецвыпуск: Избранные докл. X Российской конф. "Распределенные информационно-вычислительные ресурсы". С. 27–33.

Поступила в редакцию 27 октября 2009 г.