

ПРОГРАММНЫЙ АЛГОРИТМ УСКОРЕННОГО РЕШЕНИЯ СИСТЕМЫ УРАВНЕНИЯ МЕЛКОЙ ВОДЫ НА КОМПЬЮТЕРАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ

В.В. Калмыков

Явные схемы широко используются в параллельных вычислениях за счет простой реализации и естественного разбиения вычисляемой области между процессорами. В то же время, условие Куранта для явных схем диктует жесткое ограничение на шаг по времени, что ведет к увеличению числа итераций цикла и соответствующему росту времени счета. Кроме вычислительных затрат возникают коммуникационные проблемы, связанные с обменов данными между процессорами на каждом временном шаге, что при росте размера задачи и соответствующем росте числа процессоров ведет к тому, что значительная часть времени тратится на операции межпроцессорного обмена. В данной работе предложен метод ускорения явной схемы для уравнения мелкой воды, основанный на изменении области расчета и соответствующем уменьшении числа коммуникаций. Метод дает ощутимые преимущества в скорости выполнения по сравнению с естественной реализацией явной схемы.

Введение

Блок решения уравнения мелкой воды является частью модели гидродинамики Мирового океана, разрабатываемой в ИВМ РАН [1]. Из-за разницы в скоростях бароклинных и баротропных волн, нет возможности решать динамическую систему уравнений модели одновременно для обоих типов волн, так как условие Куранта накладывает жесткое ограничение на шаг по времени, что ведет к уменьшению временного шага для всей модели. Поэтому нахождение баротропной скорости вынесено в отдельную систему уравнений мелкой воды, решаемую с меньшим на порядок временным шагом. Уменьшение шага соответствует увеличению числа итераций цикла, что, за счет необходимости межпроцессорных обменов на каждой итерации, ведет к тому, что простейшая явная схема решения уравнения мелкой воды становится узким местом программы с точки зрения параллельной реализации.

Данная статья организована следующим образом. В разделе **Постановка задачи** приводится используемая система уравнений и соответствующая численная схема. В разделе **Тестовые расчеты** приведены параметры используемого суперкомпьютера и показаны результаты работы функции на различном числе процессоров до внесения изменений. Далее, в секции **Изменение структуры функции** приведен метод разрешения узкого места программы. Сравнительные результаты начальной и модифицированной функций представлены в разделе **Результаты ускорения**. В секции **Выводы** представлены некоторые заключительные замечания.

Постановка задачи

Исходная система уравнений мелкой воды (1.1)-(1.3) приведена ниже. Она получается из интегрирования по вертикали уравнений движения и неразрывности. Все необходимые выкладки и полную систему уравнений можно найти в [2].

$$(1.1) \quad U_t - fV = -gH \frac{1}{a \cos \varphi} \frac{\partial \eta}{\partial \lambda} + R_{x1}$$

$$(1.2) \quad V_t - fU = -gH \frac{1}{a} \frac{\partial \eta}{\partial \varphi} + R_{y1}$$

$$(1.3) \quad \frac{\partial \eta}{\partial t} + \frac{1}{a \cos \varphi} \frac{\partial U}{\partial \lambda} + \frac{1}{a \cos \varphi} \frac{\partial U \cos \varphi}{\partial \varphi} = \rho_f^{-1} W$$

Здесь f - сила Кориолиса, H - функция глубины океана, η - функция уровня, W - интенсивность потока пресной воды через границу, ρ_f - плотность пресной воды. Переменными R_{x1} , R_{y1} для наглядности записи обозначены оставшиеся члены, не зависящие от искомым U , V , η .

Для решения системы (1.1)-(1.3) применим явную аппроксимацию по времени:

$$(1.4) \quad \frac{\eta^{p+1} - \eta^p}{\tau} + \left(\frac{1}{a \cos \varphi} \delta_\lambda (\bar{U}^\varphi) \right)^p + \left(\frac{1}{a \cos \varphi} \delta_\varphi (a \cos \varphi \bar{V}^\lambda) \right)^p = \frac{1}{\rho_f} W^l$$

$$(1.5) \quad \frac{U^{p+1} - U^p}{\tau} - fV^p = -gH \left(\frac{1}{a \cos \varphi_j} \delta_\lambda (\bar{\eta}^\varphi) \right)^{p+1} + rx$$

$$(1.6) \quad \frac{V^{p+1} - V^p}{\tau} + fU^p = -gH \left(\frac{1}{a} \delta_\varphi(\bar{\eta}^\lambda) \right)^{p+1} + r\gamma$$

, где p - целочисленный момент времени для интегрирования баротропной системы с временным шагом τ , δ - оператор разностного дифференцирования.

Разрешим (1.4)-(1.6) уравнения относительно $\eta^{p+1}, U^{p+1}, V^{p+1}$:

$$(1.7) \quad \eta^{p+1} = -\tau \left(\frac{1}{a \cos \varphi} \delta_\lambda(\bar{U}^\varphi) + \frac{1}{a \cos \varphi} \delta_\varphi(a \cos \varphi \bar{V}^\lambda) \right)^p = \tau \frac{1}{\rho_f} W^l + \eta^p$$

$$(1.8) \quad U^{p+1} = -\tau gH \left(\frac{1}{a \cos \varphi_j} \delta_\lambda(\bar{\eta}^\varphi) \right)^{p+1} + \tau r x + U^p + \tau f V^p$$

$$(1.9) \quad V^{p+1} = -\tau gH \left(\frac{1}{a} \delta_\varphi(\bar{\eta}^\lambda) \right)^{p+1} + \tau r \gamma + V^p - \tau f U^p$$

Расчет баротропной части уравнений ведется с шагом $\tau \ll \Delta t$ на временном интервале $2 \Delta t$: $(t^l, t^l + 2 \Delta t)$. Количество шагов интегрирования равно $2 pl$, где $pl = \Delta t / \tau$.

Начальные условия:

$$U^{p=0} = U^l$$

$$V^{p=0} = V^l$$

$$\eta^{p=0} = \xi^l$$

Решение баротропной системы осредняется на интервале $(t^l, t^l + 2 \Delta t)$, т.е. отфильтровываются быстрые баротропные колебания:

$$\eta^{l+1} = \frac{1}{2 pl + 1} \sum_{p=0}^{2 pl} \eta^p$$

$$U^{l+1} = \frac{1}{2 pl + 1} \sum_{p=0}^{2 pl} U^p$$

$$V^{l+1} = \frac{1}{2 pl + 1} \sum_{p=0}^{2 pl} V^p$$

Таким образом, естественная реализация явной схемы:

1. Получение с помощью межпроцессорных обменов значения функций в приграничных ячейках от 8 соседей: северного, южного, восточного, западного, северо-западного, северо-восточного, юго-западного и юго-восточного.
2. Расчет всех необходимых величин и явное вычисление неизвестных U, V, η на следующем шаге по формулам (1.7)-(1.9).

Тестовые расчеты

Расчеты проводились на суперкомпьютере «МВС-100К», входящем в состав Межведомственного суперкомпьютерного центра РАН (www.jccc.ru).

Вычислительные мощности суперкомпьютера состоят из 1460 вычислительных модулей (11680 процессорных ядер). Базовый вычислительный модуль «МВС-100К» представляет собой сервер HP Proliant, содержащий:

- два четырехядерных микропроцессора Intel Xeon, работающих на частоте 3 ГГц;
- оперативную память DDR2 объёмом не менее 4 Гбайт;
- жёсткий диск объёмом не менее 36 Гбайт;
- интерфейсную плату HP Mezzanine Infiniband DDR;
- два интегрированных контроллера Gigabit Ethernet.

Вычислительные модули связаны между собой высокоскоростной коммуникационной сетью Infiniband DDR, транспортной и управляющей сетями Gigabit Ethernet.

Для расчетов использовались сетки 1/8 и 1/12 градуса, что соответствует массивам размерности 2880x1440 и 4320x2160. Расчет велся на 480 шагов по времени, причем для первой сетки число $2 pl$ баротропных шагов равнялось 120, для второй - 180. Итого, имеем 57600 и 86400 шагов цикла для каждой из сеток соответственно.

На **Рисунке 1** приведен график ускорения работы функции относительно числа процессоров для двух размеров сеток. Число процессоров для первой сетки равнялось 128, 288, 512, 800, 1800, 2592; для второй - 288, 512, 800, 1800, 2592, 4608. Черной линией показано теоретическое линейное ускорение. Оно берется

относительно 128 и 288 процессоров для соответствующих типов сеток.

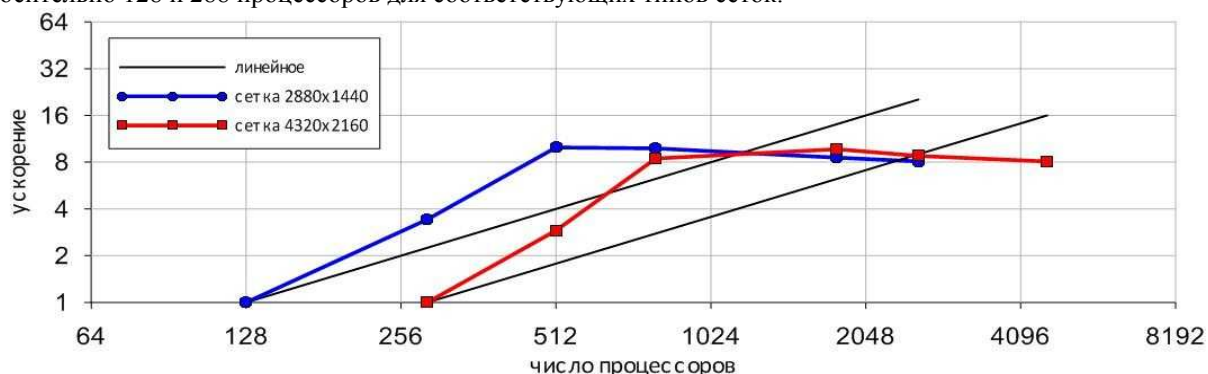


Рис. 1 Зависимость ускорения от числа процессоров для двух размеров сеток в начальной реализации явной схемы решения системы уравнений мелкой воды

По графику можно отследить две тенденции. Во-первых, значительный рост ускорения на 288, 512 и 800 процессорах для соответствующих типов сеток. Дело в том, что ускорение отсчитывается от минимального числа процессоров (128 и 288), когда массивы данных функции еще не полностью используют возможности кэш-памяти. По мере роста числа процессоров, размеры областей уменьшаются, что ведет к более эффективному использованию кэша и появлению суперлинейного ускорения.

Во-вторых, падение ускорения при увеличении числа процессоров, начиная с 512 и 1800 процессоров соответственно. Поскольку области, а значит и размеры вычислений, продолжают уменьшаться, причину повышения времени счета следует искать в коммуникационных задержках при получении на каждом шаге приграничных ячеек.

Данная проблема отмечена, например, в работе (*B. Sanders, J. Pau, 2003*) [3]. Авторы использовали одномерное разбиение области и 3 типа обмениваемых данных между процессорами-соседями, что соответствует 6 сообщениям за один шаг цикла. Они получили очень низкое ускорение и связали результаты с большим числом межпроцессорных обменов (21600 сообщений за 3600 шагов по времени).

В своей следующей работе (*J. Pau, B. Sanders, 2006*) [4], авторы попытались решить проблему коммуникаций за счет перекрытия вычислений и обменов. Они рассчитывали приграничные ячейки, асинхронно отправляли их, и, затем, досчитывали внутреннюю область. В конце шага проверяли, что все обмены завершились и выполняли следующую итерацию цикла. Данный метод также не принес существенно лучших результатов ускорения, к тому же, усложнил код.

Другой метод был предложен в работе (*H. Kuo et al., 2010*) [5]. Авторы предлагают новый параллельный метод коллокаций и тестируют его на примере решения системы уравнений мелкой воды. Однако, их метод хотя и дает суперлинейное ускорение для небольшого (меньшего 8) числа процессоров, авторы отмечают, что при дальнейшем увеличении процессорной решетки затраты на коммуникации будут быстро расти. Уже на 16 процессорах ускорение падает в 4 раза по сравнению с результатами на 8 процессорах.

Важно отметить, что процедура решения системы уравнений мелкой воды является частью обширной и сложной программы модели Мирового океана. Используемые в процедуре массивы значений различных функций подчиняются двумерной декомпозиции области, используемой во всей модели для всех глобальных массивов. Поэтому, использование неявного метода решения или сложного метода декомпозиции области приведет к нетривиальным обменам данными между процессорами, которые необходимо затронут еще и массивы внешних для данной процедуры модельных величин, используемых при решении системы уравнений мелкой воды. Это приведет к дополнительным обменам и существенному усложнению идеи естественного параллелизма.

Поэтому, требуется, не уходя от явной схемы, найти метод разрешения «узкого», с точки зрения коммуникаций, места программы, при этом сохраняя естественное двумерное распределение расчетной области между процессорами.

Изменение структуры функции

Для реализации явного метода в функции используется схема 1 шаг вычислений / межпроцессорный обмен. На **Рисунке 2** для наглядности изображен одномерный вариант этой схемы. Баротропные временные шаги отсчитываются по вертикальной оси, горизонтальная ось содержит ячейки сетки по широте. Красные узлы получены с помощью межпроцессорных обменов, в зеленых находится подсчитанная баротропная скорость. Расчетная область состоит из 5 ячеек. Для двумерной области аналогичные обмены происходят еще и по второй горизонтальной координате.

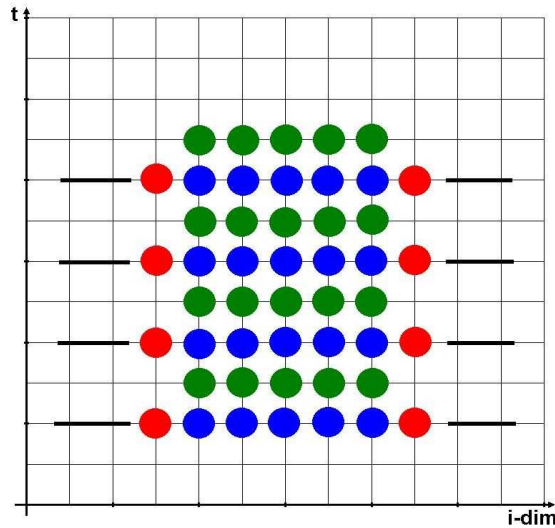


Рис. 2 Естественная схема работы функции

Теперь введем числовой параметр p , обозначающий количество ячеек, на которое будет увеличена область в каждом направлении. Далее, заменим схему работы функции с 1 шаг/межпроцессорный обмен на схему p шагов/ межпроцессорный обмен. Таким образом, решаются те же уравнения, но на расширенной на p ячеек области, и с каждым шагом эта область сужается на одну ячейку по всем направлениям.

То есть, вычисленные приграничные ячейки на данном шаге станут мнимыми ячейками-соседями (которые для начальной схемы обновляются с помощью обменных процедур) для следующего шага, а никаких межпроцессорных коммуникаций не потребуется.

В результате, через p шагов, область сузится до обычного расчетного прямоугольника данного процессора, и внутри него уже будет вычислена искомая баротропная скорость.

Далее, отрабатывает обменная процедура, процессор опять получает расширенную область, и продолжает аналогичные вычисления, всего - $2pl/p$ циклов, где $2pl$ - общее число баротропных шагов. То есть, увеличивая расчетную область на p ячеек в каждом направлении, мы уменьшаем число обменов в p раз.

Схема работы измененной функции в одномерном случае представлена на **Рисунке 3**. Расчетная область по-прежнему состоит из 5 ячеек, значение параметра расширения p равно 3.

Очевидно, начальную версию функции можно рассматривать как модифицированную с параметром p , равным 1.

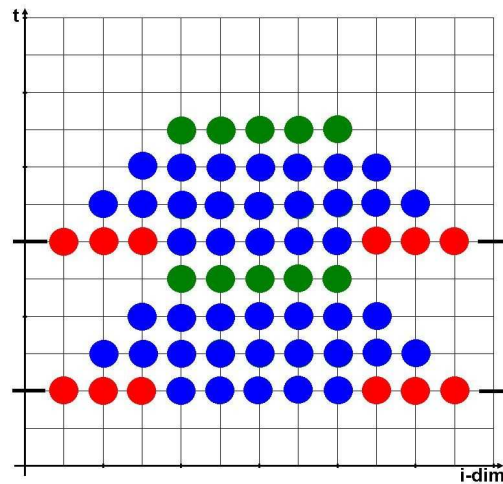


Рис. 3 Модифицированная схема работы функции

Таким образом, требуется:

- изменить цикл по времени со схемы 1 шаг/межпроцессорный обмен на схему p шагов/ межпроцессорный обмен за счет введения вложенного цикла
- организовать расширенные межпроцессорные обмены для массивов, участвующих в цикле вычислений
- организовать расширенные межпроцессорные обмены для данных, используемых, но не изменяющихся в цикле вычислений. К таким данным относится, например, массивы длин

граней ячеек, нужных при вычислении потоков – $dx_nw_v[]$, $dx_nw_t[]$ и т.д.

Если расчет происходит на ограниченной сушей расчетной области, то на этом изменения заканчиваются.

Если суша находится с севера и юга, а на восточной и западной границе поставлены циклические граничные условия, естественные для океана, то потребуются лишь добавить циклических соседей процессорам у границы, что не вызывает затруднений.

А вот в случае, когда расчетная область представляет собой весь Мировой океан, который с точки зрения массива имеет открытые границы с севера, запада и востока, могут потребоваться дополнительные усложнения при организации обменов на севере. Примером может служить используемая в нашей модели комбинация широтно-долготной и биполярной сеток (**Рисунок 4**), выбранная для обхода известных вычислительных проблем на Северном полюсе. Полюса сетки лежат на суше и измельчения ячеек, а, соответственно и численных трудностей на расчетной части области не возникает. Для такой сетки, очевидно, потребуются склейка на севере, через линию, соединяющую два ее полюса. Обсуждение технических сложностей такого обмена для данных, участвующих в решении системы громоздко и выходит за рамки данной статьи.

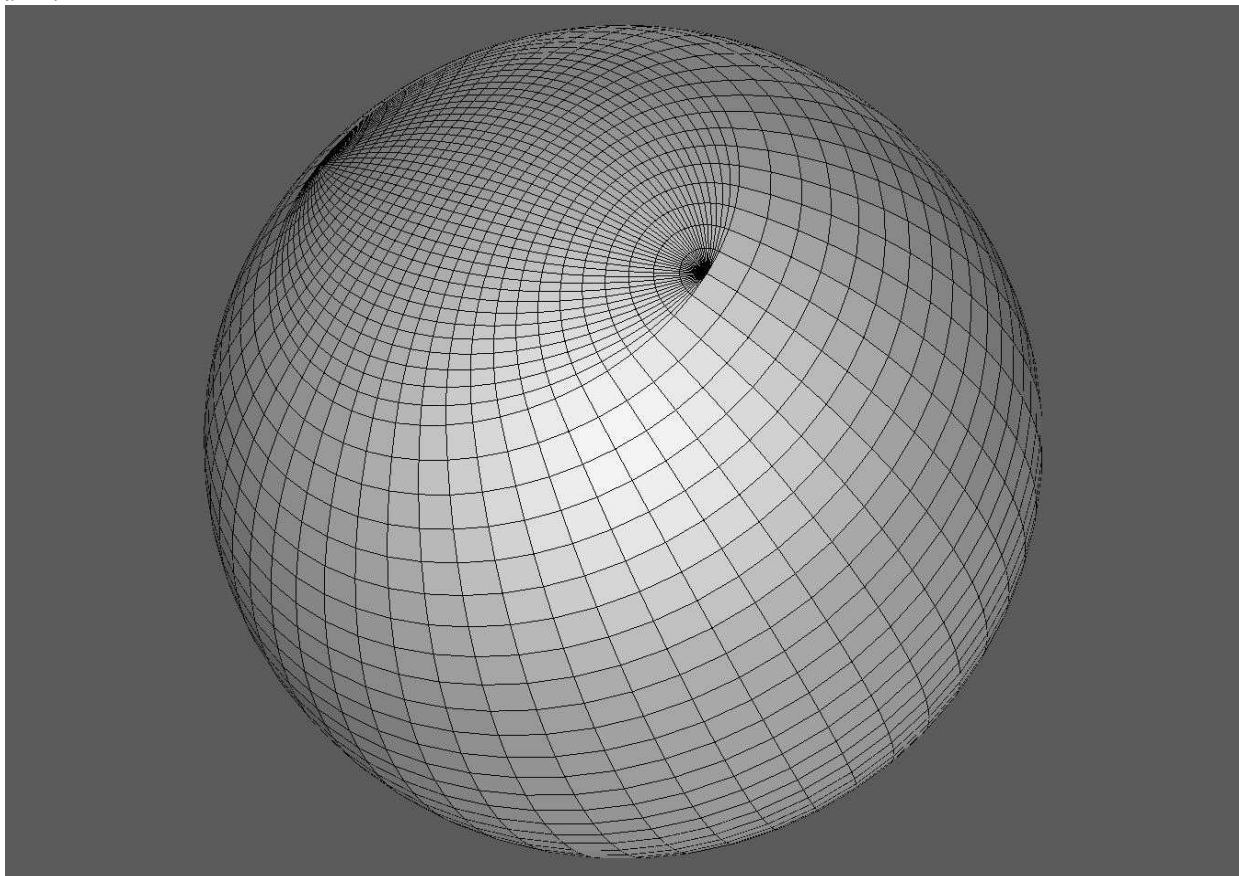


Рис. 4 Модельная сетка

Результаты ускорения

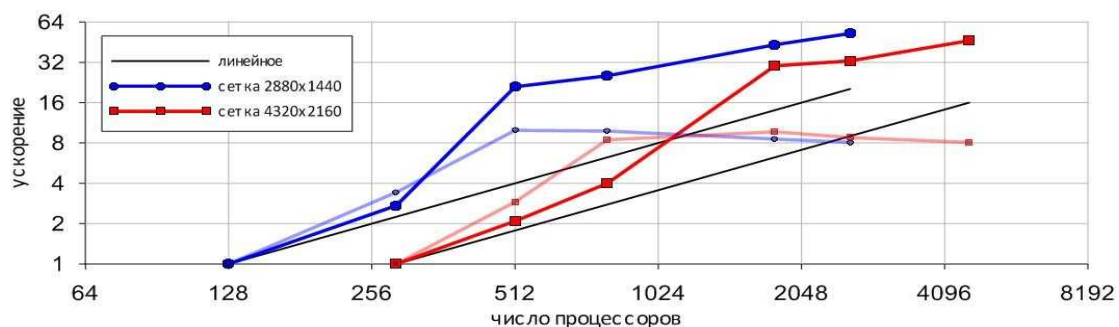


Рис. 5 Зависимость ускорения от числа процессоров для двух размеров сеток и двух версий реализации явной схемы решения системы уравнений мелкой воды
Модифицированная версия тестировалась на том же суперкомпьютере, параметры задачи не менялись:

сетки 1/8 и 1/12 градуса, 480 шагов по времени, 120 и 120 баротропных шагов соответственно. Параметр расширения области p брался равным 10. На **Рисунке 5** представлены результаты ускорения функции относительно числа процессоров, вычисляемые относительно 128 и 288 процессоров начальной версии. Для сравнения, более прозрачными линиями обозначены результаты ускорения начальной версии, приведенные ранее на **Рисунке 1**.

По графику можно отследить несколько тенденций.

Во-первых, начальная версия дает лучшие результаты на небольшом числе процессоров. Это логично, так как модифицированная версия использует область, расширенную на 10 ячеек в каждом направлении. Для небольшого числа процессоров, когда коммуникационные задержки еще невелики, это ведет к увеличению расчетной области и соответствующему росту времени вычислений. По этой же причине ускорение отсчитывается от времени работы начальной версии, дабы искусственно не завышать показатели.

Поэтому, для максимального ускорения на всех размерах процессорных решеток можно до некоторого числа процессоров использовать значение p , равное 1, а далее – p , равное 10. Хотя нас больше интересуют результаты в правой части графика, так как модель Мирового океана с хорошим разрешением требует значительных вычислительных мощностей.

Во-вторых, ясно, что при минимальном числе процессоров, от которых отсчитывается ускорение, данные не полностью используют возможности кэш-памяти. Вследствие этого, дальнейшее уменьшение областей при отсутствии коммуникационных проблем ведет к значительному росту ускорения.

При использовании схемы в составе модели Мирового океана, общее ускорение модели будет не таким высоким, так как помимо функции решения уравнения мелкой воды модель содержит множество блоков, что приуменьшает относительную роль баротропной функции в общих результатах. Однако, при отсутствии других «узких» параллельных алгоритмов, порядок суммарного времени будет сохраняться.

Выводы

На примере системы уравнений мелкой воды предложен и протестирован программный алгоритм ускорения явной схемы.

Результаты ускорения представляются удовлетворительными, что позволяет использовать метод для сокращения коммуникационных расходов в широком круге задач. Метод был введен в разрабатываемую нами модель Мирового океана, где также показал хорошие результаты.

Ясно, что применять алгоритм имеет смысл лишь в функциях, по своей структуре перенасыщенных обменными процедурами. То есть, замена всех явных схем в программе описанным образом приведет к неоправданному усложнению кода и вряд ли повысит эффективность, так как в крупных моделях между функциями межпроцессорной коммуникации содержится достаточное количество локальных вычислений и обмены в этом смысле довольно редки. А вот для функций вида шаг/обмен, где под шагом подразумевается вычисление нескольких потоков и операторов, метод может дать существенные результаты. Особенно, при использовании большого числа процессоров, что становится необходимостью при разработке современных глобальных моделей высокого пространственного разрешения.

ЛИТЕРАТУРА:

1. A.S. Sarkisyan, R.I. Ibrayev, High resolution and four-dimensional analysis as a prospect for ocean modelling, Russ. J. Numer. Anal. Math. Modelling, Vol. 25, No 5, pp. 477-496, 2010
2. Р.А. Ибраев, Математическое моделирование термодинамических процессов в Каспийском море. – М.: ГЕОС, 2008
3. B. Sanders, J. Pau, Parallel implementation of an explicit finite-volume shallow-water model, 16th ASCE Engineering Mechanics Conference, 2003
4. B. Sanders, J. Pau, Performance of parallel implementations of an explicit finite-volume shallow-water model, Journal of Computing in Civil Engineering, ASCE, 2006
5. H. Kuo, Y. Chang, A new parallel domain-decomposed Chebyshev collocation method for atmospheric and oceanic model, Journal of Advances in Modelling Earth Systems, 2010
6. В.В. Воеводин, Параллельные вычисления. СПб.: БХВ-Петербург, 2004