

Параллельная объектно-ориентированная среда для моделирования процессов атмосферного переноса при аварийных выбросах *

А.В.МуРИН

*Институт исследования природных
и техногенных катастроф УдГУ, Ижевск, Россия*

27 декабря 2003 г.

В работе представлен открытый пакет программ с распараллеливанием на суперкомпьютерах, предназначенный для вычислительных экспериментов с моделями переноса примеси при залповых аварийных выбросах. Обсуждается глобальное распараллеливание многовариантных расчетов и локальное распараллеливание решения краевых задач краткосрочного переноса на основе методов декомпозиции области. Рассматривается объектно-ориентированная модель вычислительного ядра и переносимая программная реализация.

Введение

Изучение процессов переноса примесей в нижнем слое атмосферы играет исключительно важную роль в приложениях. В частности, моделирование указанного процесса широко используется в задачах охраны окружающей среды [1] и задачах анализа аварийного риска [2]. Интенсивные исследования в этой области, начавшиеся в середине прошлого века и продолжающиеся по настоящее время, привели к накоплению большого багажа знаний, сконцентрированных в разнообразных моделях процессов атмосферного рассеяния (например, только в [3] зарегистрировано свыше 90 моделей).

Ресурсоемкость и вычислительная сложность существующих моделей варьируется в очень широких пределах (от нескольких минут работы современной ЭВМ для простейших гауссовых моделей до нескольких суток для трехмерных численных моделей) и имеет постоянную тенденцию к возрастанию. Несмотря на потенциальное ускорение вычислений, которое сегодня может обеспечить использование многопроцессорных ЭВМ, параллельные вычисления в задачах рассеяния примесей все еще не очень широко распространены и фактически отсутствуют среди инженерных приложений. Одной из главных причин этого является необходимость внесения серьезных изменений в алгоритмы для того, чтобы трансформировать последовательную программу в параллельную, особенно работающую в гетерогенной распределенной среде (кластеры и решетки). Не способствует прогрессу и реализация многих моделей при помощи методов структурного программирования, а

*Работа выполнена при частичной финансовой поддержке РФФИ (грант 01-01-96444), научной программы “Университеты России” (грант УР.03.01.015) и МНТЦ (грант 2065).

также привязка к одной программно-аппаратной платформе. Вместе с тем, с вычислительной точки зрения, разнообразные модели имеют в своей основе много общего, различаясь в физическом наполнении, численной схеме или функциональности.

Основная цель данной работы — создание программного комплекса нового поколения, предназначенного для проведения вычислительных экспериментов с моделями краткосрочного рассеяния примесей в пограничном слое атмосферы (ПСА) при залповых аварийных выбросах. Отличительными чертами предлагаемого подхода являются: открытость и расширяемость, мобильность, глубокая параллелизация вычислительных процессов.

Структура работы такова. В п.1 рассматриваются общие черты моделей рассеяния примесей в эйлеровом описании. В п.2 производится отображение вычислительных алгоритмов на архитектуру многопроцессорной ЭВМ. В п.3 строятся объектно-ориентированные вычислительные модели рассеяния. В п.4 приводится пример реализации комплекса.

1. Вычислительные модели переноса примеси

Одной из основных математических моделей рассеяния является уравнение конвекции-диффузии, описывающее в эйлеровом представлении эволюцию концентрационного поля $c_i(\mathbf{x}, t)$ пассивной примеси в пространственной области Ω

$$\frac{\partial c_i}{\partial t} + \nabla \mathbf{u} c_i = \nabla \mathbf{K} \nabla c_i + f_i, \quad \mathbf{x} \in \Omega, \quad t \in [0, T], \quad (1)$$

$$c_i \Big|_{t=0} = c_i^0(\mathbf{x}), \quad (\mathbf{u} c_i - \mathbf{K} \frac{\partial c_i}{\partial \mathbf{n}}, \mathbf{n}) \Big|_{\Gamma_{gr}} = j_i, \quad c_i \Big|_{\Gamma \setminus \Gamma_{gr} \rightarrow \infty} = 0, \quad (2)$$

где $\mathbf{x} = \{x_\alpha\}$, $\Omega = \Gamma_1 \times \Omega_\perp$, $\Omega_\perp = \Gamma_2 \times \Gamma_3$, $\Gamma_\alpha = [0 \leq x_\alpha \leq l_\alpha]$, $\alpha = 1, 2, 3$, Γ — граница Ω , Γ_{gr} — подстилающая поверхность, i — номер компонента, $i = 1, \dots, n$. Введена декартова система координат, ось x_1 направлена в сторону движения потока воздушных масс.

Физическое содержание задачи определяется путем выбора соответствующих поля ветра $\mathbf{u} = (u_1, u_2, u_3)^{-1}$, поля коэффициентов турбулентного обмена среды $\mathbf{K} = (K_1, K_2, K_3)^{-1}$, функций (в общем случае нелинейных) источников-стоков f_i и поглощения-эмиссии j_i .

Вычислительное содержание задачи фактически определяется методом решения (1-2). Можно выделить три основных класса вычислительных моделей.

Гауссовы модели (G-модели). Решение строится на основе функции влияния источника G (функции Грина) и может быть представлено в виде

$$c_i(\mathbf{x}, t) = \int_{\Omega} \int_0^t G_i(\mathbf{x} - \mathbf{x}', t - t') f_i(\mathbf{x}', t') d\mathbf{x}' dt'. \quad (3)$$

Численные стохастические модели частиц (S-модели) являются одними из перспективных моделей в задаче переноса примесей. Поле $c(\mathbf{x}, t)$ определяется как плотность распределения координат частиц с законом движения

$$d\mathbf{x}_\mu = (\mathbf{u} + \nabla \mathbf{K}) dt + \Sigma d\mathbf{W}, \quad \mathbf{x}_\mu \Big|_{t=0} = \mathbf{x}_\mu^0, \quad (4)$$

где μ — номер частицы, $(\Sigma)_{\alpha\beta} = \sqrt{2K_\alpha} \delta_{\alpha\beta}$, $\beta = 1, 2, 3$, $d\mathbf{W}$ — приращение стандартного винеровского процесса с независимыми компонентами W_α . Интересно отметить, что

лагранжевы модели (\mathcal{L} -модели) имеют аналогичную вычислительную структуру. Следовательно на основе адекватной программной реализации \mathfrak{S} -модели можно построить и разнообразные \mathcal{L} -модели.

Численные сеточные модели (\mathfrak{M} -модели) представляют собой наиболее обширный класс и позволяют получить концентрационное поле при произвольных физических параметрах. В качестве примера рассмотрим вычислительную модель [4] на основе методов суммарной аппроксимации. Пусть $A_\alpha = -\nabla_\alpha u_\alpha$, $D_\alpha = \nabla_\alpha K_\alpha \nabla_\alpha$, $f_\alpha c = f(c)/3$. На отрезке $0 \leq t \leq T$ введем равномерную сетку с шагом τ . Тогда цепочка уравнений

$$\frac{\partial \zeta_\beta}{\partial t} = L_\beta \zeta_\beta, \quad t_j < t \leq t_{j+1}, \quad t_j = j \cdot \tau, \quad \beta = 1, 2, \dots, 9, \quad (5)$$

$$\zeta_1(\mathbf{x}, t_j) = c(\mathbf{x}, t_j), \quad \zeta_\beta(\mathbf{x}, t_j) = \zeta_{\beta-1}(\mathbf{x}, t_{j+1}), \quad \beta = 2, \dots, 9, \quad (6)$$

аппроксимирует исходную задачу (1), причем решением является $c(\mathbf{x}, t) = \zeta_9(\mathbf{x}, t)$. Введены обозначения $L_1 = f_1$, $L_2 = D_3$, $L_3 = A_3$, $L_4 = f_2$, $L_5 = D_2$, $L_6 = A_2$, $L_7 = f_3$, $L_8 = D_1$, $L_9 = A_1$. Пусть $\Omega_h = \{\mathbf{x}^q = (x_1^q, x_2^q, x_3^q), x_\alpha^q = x_\alpha^{q\alpha-1} + h_\alpha^{q\alpha}, q = (i, j, k)\}$ — неравномерная структурированная сетка в Ω , $h_\alpha^{q\alpha}$ — шаги сетки. Сетка Ω_h разбивает Ω на контрольные объемы, каждый из которых имеет форму прямоугольного параллелепипеда. Пространственная дискретизация (5) основывается на методе контрольного объема, который приводит к локально-консервативным схемам. Для аппроксимации (5) при $\beta = 3, 6, 9$ используется явная интегро-интерполяционная схема с коррекцией потоков точности $O(\tau, h^4)$ [5], нивелирующая “паразитную” численную диффузию при расчете адвекции. Для аппроксимации (5) при $\beta = 2, 5, 8$ используется неявная схема точности $O(\tau, h^2)$. В результате дискретизации получим цепочку линейных алгебраических уравнений

$$\zeta^{n+\beta/9} = S_\beta \zeta^{n+(\beta-1)/9}, \quad \zeta^n = c^n, \quad c^{n+1} = \zeta^{n+1}, \quad (7)$$

где S_β — оператор перехода схемы. Получающиеся трехдиагональные системы алгебраических уравнений решаются методом прогонки.

Отметим, что путем комбинации вычислительных моделей трех указанных классов могут быть синтезированы гибридные модели рассеяния.

2. Алгоритмы параллелизации

Для параллелизации необходимо отобразить структуру вычислительных алгоритмов на архитектуру многопроцессорной ЭВМ. Пусть $P = \{p\}$ множество процессоров ЭВМ.

Глобальная параллелизация. Решение задачи (1-2) при использовании \mathfrak{S} - и \mathfrak{G} -моделей не занимает особо много машинного времени современных однопроцессорных ЭВМ. Однако в таких задачах как анализ чувствительности модели к входным данным или верификация по экспериментальным данным требуются многократные прогоны модели. Эффективное решение многовариантных задач такого рода может быть достигнуто при помощи глобальной параллелизации по основным входным данным.

Пусть $t = t(\mathbf{p})$ — вычислительная задача, $\mathbf{p} = (p_1, p_2, \dots, p_n)$ — вектор входных параметров, $p_i \in \mathbb{P}_i$, $\mathbf{p} \in \mathbb{P}$, $\mathbb{P} = \mathbb{P}_1 \times \mathbb{P}_2 \times \dots \times \mathbb{P}_n$, \mathbb{P}_i — конечное множество. На множестве \mathbb{P} определим некоторое отношение линейного нестрого порядка \preceq . Тогда алгоритм глобальной параллелизации может быть представлен в следующем виде.

Шаг I. Преобразование \mathbb{P} в линейный список \mathbb{L} при помощи отношения \preceq , определяющего приоритеты входных параметров.

Шаг II. Формирование очереди заданий: разбиение \mathbb{L} на непересекающиеся подписки $\mathbb{L} = \bigcup_{j=0}^m \mathbb{L}_j$ методом рекурсивного деления пополам, $m \gg |P|$.

Шаг III. Централизованная динамическая балансировка. Выделенный процессор 0 (мастер), последовательно выбирает задание \mathbb{L}_j из очереди \mathbb{L} и размещает его для выполнения на первом незагруженном процессоре p (рабочем) из пула P . Рабочий, получив задание, решает $t(\mathbf{p})$ для $\forall \mathbf{p} \in \mathbb{L}_j$.

Локальная параллелизация. Решение (1-2) при помощи \mathfrak{N} -модели представляет собой весьма трудоемкую операцию. Рассмотрим один из алгоритмов параллелизации [4] на основе методов декомпозиции области. Особенностью задачи краткосрочного моделирования поля примеси, возникшего в результате аварийного выброса, является анизотропия Ω вдоль x_1 ($l_1 \gg l_2 > l_3$), причем $u_1 \geq 0$. Введем разбиение $\Gamma_1 = \bigcup_p \Gamma_1^p$, $\Gamma_1^p \cap \Gamma_1^{p'} = \emptyset$, $\Omega^p = \Gamma_1^p \times \Omega_{\perp}$, $\Omega_h^p = \{\mathbf{x}^q | \mathbf{x}^q \in \Omega^p\}$. Подобласть Ω_h^p поставим в соответствие с процессором p . Эффективный параллельный алгоритм может быть построен при учете того обстоятельства, что физические возмущения распространяются преимущественно вдоль выделенного направления x_1 . Если для аппроксимации задачи (5) при $\beta = 8$ использовать схему типа “бегущего счета”, то возможно явно определять значения $\zeta^{n+8/9}$ в подобласти Ω_h^p требуя для шивки только граничные значения c^{n+1} из подобласти Ω_h^{p-1} . Тогда алгоритм локальной параллелизации может быть представлен в следующем виде.

Шаг I. Решить трехдиагональную систему (7) при $\beta = 1 - 7$ локально для $\forall p \in P$.

Шаг II. Для $\forall p \in P$, $p \neq 0$: передать граничные значения $\zeta^{n+7/9}$ для $p - 1$.

Шаг III. Для $\forall p \in P$, $p \neq 0$: принять от $p - 1$ граничные значения c^{n+1} .

Шаг IV. Решить двухдиагональную систему (7) при $\beta = 8, 9$ локально для $\forall p \in P$.

Шаг V. Для $\forall p \in P$, $p \neq |P| - 1$: передать для $p + 1$ граничные значения c^{n+1} .

Отметим, что схема бегущего счета обладает нежелательным свойством условной аппроксимации исходной дифференциальной задачи. Это приводит к ограничениям на τ в зависимости от шага h . Тем не менее, при значениях \mathbf{u} и \mathbf{K} , отвечающих процессам переноса в нижней атмосфере, параллельный алгоритм позволяет вести устойчивый счет при ограничениях на шаг τ , более слабых по сравнению с ограничениями явных схем.

3. Объектно-ориентированная модель

Перейдем к разработке логической модели программной среды. При построении необходимо учитывать тот факт, что вычислительные модели не являются статическими объектами, они подвержены постоянной эволюции и изменению [6]. Программа должна обеспечить возможность для такой эволюционной изменчивости моделей.

Традиционные методы процедурного программирования используют процедуры для отражения инвариантов предметной области. Процедуры соответствуют некоторым процессам. Взаимосвязи между процедурами являются опорой процедурной модели. Однако абстрактные процедуры имеют тенденцию к изменению: эволюция модели приводит к добавлению новых аспектов, что влечет серьезные изменения и дробление процедур. В свою очередь это приводит к ветвлению программы и существенному изменению ее логики.

Одним из перспективных подходов к разработке больших программных комплексов вычислительного моделирования является объектно-ориентированный подход (ООП). Суть ООП заключается в построении такой логической модели, которая близко соответству-

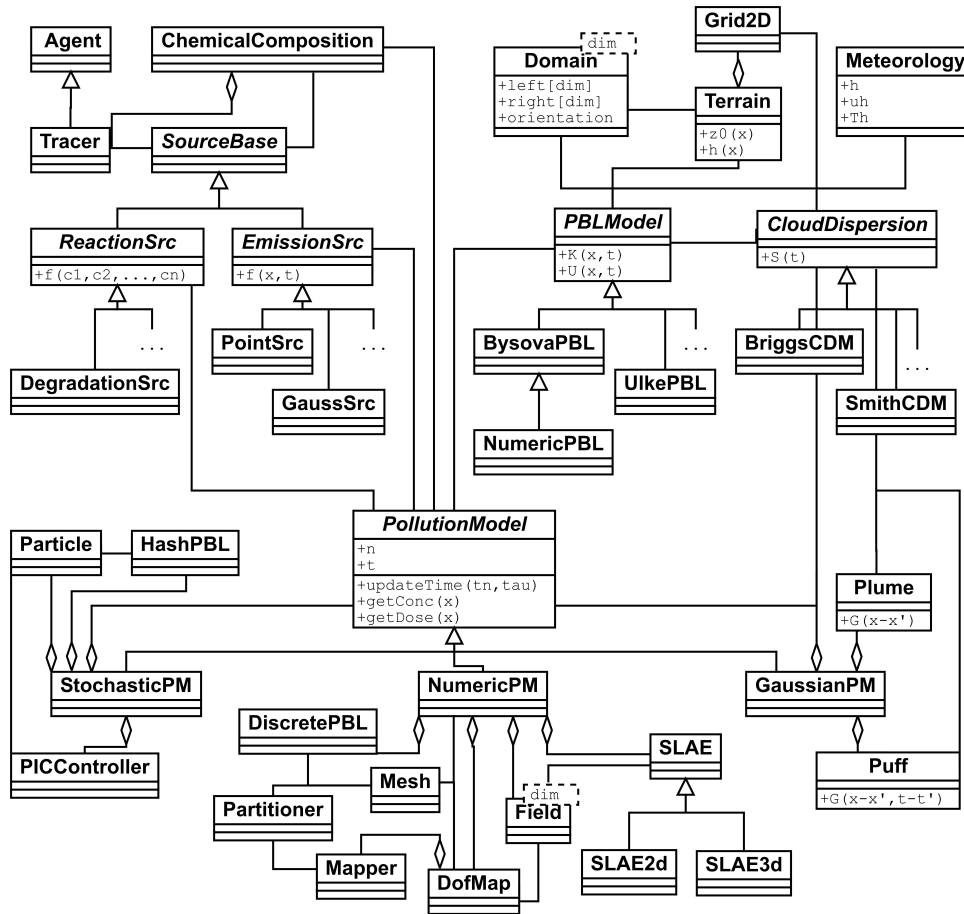


Рис. 1. Диаграмма классов объектно-ориентированной модели рассеяния примесей.

ет инвариантам задачи. В большинстве случаев, предметная область довольно статична по отношению к классам объектов. Следовательно, абстрактные модели, основанные на предметных классах, имеют внутреннюю гибкость к неизбежным эволюционным изменениям. Таким образом, ООП позволяет разрабатывать программные среды, оперирующие стабильными физическими и математическими абстракциями.

Обратим внимание на отсутствие в настоящее время сколько-нибудь систематического изложения объектной модели рассеяния примесей в атмосфере. На рис. 1 приведены основные классы объектов и их взаимосвязи, выделенные в результате анализа вычислительных моделей рассеяния примесей. Были определены следующие группы классов объектов. Первая группа представлена управляющими классами, ответственными за реализацию алгоритма глобальной параллелизации и синхронизацию данных между распределенными объектами, классами таймеров и др. Во вторую группу входят классы, задающее физическое содержание задачи: химический состав, источники, характеристики ПСА. В третью группу входят классы, определяющие вычислительное содержание. Наконец, четвертая группа состоит из классов объектов, используемых для построения приложений высокого уровня: классы представления и анализа результатов моделирования; классы для верификации моделей по экспериментальным данным. Остановимся только на второй и третьей группах.

Вторая группа образована из двух изолированных подгрупп “Источники” и “ПСА”.

Источники. Класс **Agent** есть абстракция компонента примеси и обеспечивает доступ к различным физико-химическим свойствам (молекулярному весу, коэффициенту диффузии и др.). Класс **Tracer** дополняет путем наследования **Agent** аспектами, связанными с задачами рассеяния примеси: коэффициентами гравитационного оседания, функцией эмиссии-поглощения j_i и др. Класс **ChemicalComposition** является контейнером для объектов класса **Tracer** и определяет химический состав концентрационного поля. Функция f_i представлена в модели базовым классом **SourceBase**, инкапсулирующим время активации, продолжительность действия и др. Выделены два базовых подкласса источников, наследующих свои свойства от **SourceBase**: класс **EmissionSource**, отвечающий источникам вида $f_i = f(\mathbf{x}, t)$, а также класс **ReactionSource**, отвечающий за нелинейные стоки и источники вида $f_i = f(c_1, \dots, c_n)$. Все конкретные реализации функции f_i образуются путем наследования от двух указанных базовых классов.

ПСА. Абстракцией области Ω является шаблонный класс **Domain<dim>**, где **dim** — размерность пространства. Класс **Terrain** инкапсулирует информацию о свойствах подстилающей поверхности (шероховатость, землепользование, рельеф и др.). Свойства поверхности аппроксимированы на двумерной сетке, представленной классом **Grid2D**. Класс **Meteorology** отвечает измерению совокупности метеоэлементов в отдельной пространственной точке. Характеристики ПСА (например, поля **u** и **K**) инкапсулированы в абстрактном классе **PBLModel**. Все реализации моделей ПСА наследуют интерфейс от **PBLModel**. Кроме того, выделен отдельный абстрактный класс **CloudDispersion**, обеспечивающий доступ к данным о дисперсии облака в ПСА и необходимый для построения функции Грина в \mathcal{G} -моделях.

Основой третьей группы “*Вычислительные модели*” является класс **PollutionModel**, предоставляющий абстрактный интерфейс модели рассеяния. Все реализации моделей наследуют интерфейс и отношения от **PollutionModel**.

\mathcal{G} -модели представлены классом **GaussianPM**, являющимся контейнером для объектов классов **Puff** и **Plume**. Последние суть абстракции функций Грина G мгновенного и продолжительных точечных источников, соответственно. При построении функции G используется информация о дисперсиях облака **CloudDispersion**. Концентрационное поле вычисляется в узлах сетки **Grid2D**.

Абстракцией \mathcal{S} -моделей является класс **StochasticPM**. Модель строится из ансамбля частиц **Particle**, диффундирующих в полях **U**, **K**. Класс **HashPBL** обеспечивает быстрый доступ к полям ветра и коэффициентов обмена путем дискретизации на сетке и линейной интерполяции. Класс **PICController** служит для реализации метода частиц в ячейках, используемого для вычисления концентрации в выделенной точке пространства.

Основой реализации \mathcal{N} -моделей, представленных классом **NumericPM**, служат классы **Mesh** и **Field<dim>**, отвечающие разностной сетке Ω_h и сеточной функции, соответственно. Метод декомпозиции области реализуется при помощи класса **Partitioner**, отвечающего за разделение сетки на подсетки методом рекурсивного деления пополам. В результате работы **Partitioner** создается объект класса **Mapper**, ставящий в соответствие процессору p подмножество узлов сетки Ω_h^p . Объекты класса **DoiMap**, пользуясь **Mesh**, **Mapper** и **ChemicalComposition**, ставят в соответствие номерам процессора, узла сетки и компонента локальные и глобальные индексы **Field<dim>**. Дискретные поля **U**, **K**, определенные в центрах граней контрольных объемов **Mesh**, инкапсулированы в объектах класса **DiscretePBL**. Объекты приведенных классов обеспечивают всю необходимую инфраструктуру для реализации той или иной параллельной численной модели. Дополняют модель классы **SLAE2d** и **SLAE3d** — абстракции двух- и трехдиагональных систем

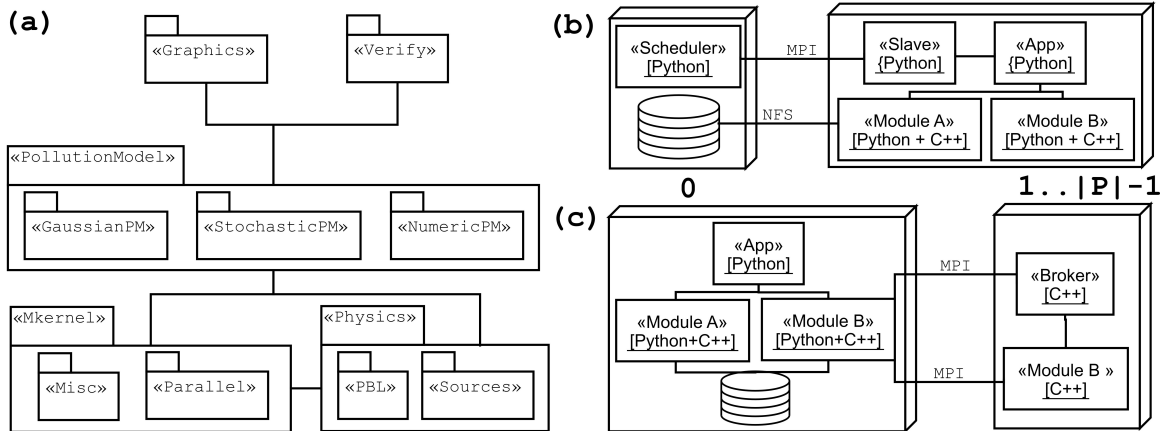


Рис. 2. Архитектура модулей (a) и архитектура процессов: (b) глобальная параллелизация, (c) локальная параллелизация.

линейных алгебраических уравнений, соответственно. Численная схема расщепления (5-6) реализуется как набор методов **NumericPM**, каждый из которых отвечает оператору перехода (7).

4. Пример реализации

Приведенная объектно-ориентированная вычислительная модель может быть реализована тем или иным способом. Рассмотрим один из возможных путей реализации.

На рис. 2a приведена архитектура модулей программной среды. Классы отдельных групп образуют изолированные подсистемы, обеспечивающие некоторую функциональность. Подсистемы подразделяются на слабосвязанные модули, отвечающие подгруппам. Модуль состоит из ядра и оболочки. Ядро реализовано на языке C++ и представляет собой динамически загружаемую библиотеку. Оболочка суть “обертка” над интерфейсной частью ядра, реализованная на языке Python. Интерпретатор Python играет роль интегрирующей платформы. Предлагаемый подход позволяет использовать в гибком интерпретаторе языка сверхвысокого уровня Python вычислительно эффективное ядро. Объекты предметной области в сочетании с динамической природой, ясным синтаксисом и мощной семантикой Python образуют прикладную вычислительную среду. Такая среда является по построению открытой, прозрачной, расширяемой, производительной и мобильной.

Архитектура процессов, определяющая отображение подсистем на структуру многопроцессорной ЭВМ, проиллюстрирована на рис. 2. При глобальной параллелизации вычислительного процесса (см. рис. 2b), процессор 0 выполняет процесс-мастер “Scheduler”. На процессорах $1 \dots |P| - 1$ выполняются процессы-рабочие “Slave”. Каждый “Slave” решает задачу в определенной подобласти входных параметров, передавая аргументы сценарию-приложению “App”. В свою очередь, “App” координирует работу вычислительных модулей, отвечающих задаче. Выходные данные помещаются в общее сетевое хранилище данных; управляющие коммуникации между “Scheduler” и “Slave” основаны на интерфейсе MPI. При локальной параллелизации (рис. 2c), процессор 0 выполняет сценарий “App”, в котором используются последовательные и параллельные вычислительные модули. На процессорах $1 \dots |P| - 1$ выполняются процессы-рабочие с фиксированной структурой управ-

ления: менеджер запросов “Broker” анализирует запросы, поступающие от “App”, и передает управление методам параллельных объектов. Среди последних можно выделить два вида: реплицированные и распределенные объекты. К первому виду принадлежат объекты, создающие общий вычислительный контекст задачи и имеющие идентичную копию $\forall p \in P$ (например, объекты классов **ChemicalComposition** или **ReactionSource**). Ко второму виду относятся объекты, связанные с реализацией методов декомпозиции (например, объекты классов **Mesh** и **Field<dim>**), и, следовательно, данные которых уникальны для каждого процесса. Все коммуникации осуществляются посредством MPI и скрыты внутри методов параллельных объектов. Указанная архитектура процессов добавляет к программной среде свойство масштабируемости и возможность работы на программно-аппаратных платформах с произвольной организацией памяти.

Заключение

В работе рассмотрена логическая и вычислительная структура моделей рассеяния примесей в пограничном слое атмосферы при залповых аварийных выбросах. В результате теоретического анализа создана объектно-ориентированная модель — устойчивая основа для создания расширяемых параллельных вычислительных моделей переноса. Предложенный подход к реализации обеспечивает открытость, прозрачность, мобильность, производительность и масштабируемости создаваемой среды моделирования нового поколения. В дальнейшем планируется развивать вычислительную среду по ряду направлений, в частности — по пути интеграции с параллельными мезометеорологическими моделями.

Список литературы

- [1] Берлянд М.Е. Прогноз и регулирование загрязнения атмосферы. — Л.: Гидрометеоиздат, 1985. — 272 с.
- [2] Колодкин В.М., Мурин А.В., Петров А.К., Горский В.Г. Количественная оценка риска химических аварий. /Под ред. В. М. Колодкина. — Ижевск: Изд-во Удм. Ун-та, 2001. — 218 с.
- [3] MDS – MODEL DOCUMENTATION SYSTEM. — [HTTP://ETC-ACC.EIONET.EU.INT/DATABASES/MDS.HTML](http://etc-acc.eionet.eu.int/databases/mds.html).
- [4] Мурин А.В. Математическое моделирование на параллельных системах последствий химических аварий. — Автореф. дисс. на соискание уч. ст. канд. физ.-мат. наук. — Ижевск, 2002. — 24 с.
- [5] WALCEK C. J., ALEKSIC N. M. A SIMPLE BUT ACCURATE MASS CONSERVATIVE, PEAK-PRESERVING, MIXING RATIO BOUNDED ADVECTION ALGORITHM WITH FORTRAN CODE //ATMOS. ENV., 1998. — VOL. 32. N 22. — PP. 3863-3880.
- [6] Хокни Р., Иствуд Дж. Численное моделирование методом частиц. — М.: Мир, 1987. — 640 с.