

УДК 533.6:681.3

МУЛЬТИПРОЦЕССОРНОЕ КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ГРАВИТАЦИОННОЙ ГАЗОВОЙ ДИНАМИКЕ

Г. А. Тарнавский¹, А. Г. Тарнавский²

Настоящая работа ориентирована на создание новых современных компьютерных технологий и методов параллельного программирования для повышения эффективности решения фундаментальных научных и прикладных проблем в области космической газодинамики, связанных с большим объемом вычислений. Основное внимание уделено теоретическим вопросам и их практическому использованию при проведении декомпозиции и параллелизации успешно функционирующих методов и реализующих их алгоритмов решения сложных интегродифференциальных систем уравнений. Рассмотрены различные способы распараллеливания, которые определяются особыми видами декомпозиции полной задачи на ряд одновременно выполняемых подзадач (декомпозиция по физико-математическим процессам, по расчетным геометрическим областям, по технологии счета, по главным входным параметрам) и проанализирована степень их эффективности для определения оптимального способа параллелизации. Проведенные теоретические исследования используются при создании комплекса компьютерных программ нового поколения для решения задач космической газодинамики.

1. Введение. Одной из областей знаний, решение задач которой вызывает настоятельную необходимость в создании и развитии компьютерных систем параллельного программирования [1], является астрофизика. Среди множества ее актуальных проблем [2] можно выделить проблему образования звездных и планетарных систем из протовещества и их дальнейшего развития. К этой проблеме непосредственно примыкают вопросы [3] о допланетном синтезе первичного органического вещества на Земле и об установлении наиболее вероятных условий абиогенного синтеза органических соединений в Солнечной системе.

Прогресс современной вычислительной техники с использованием технологий параллельного счета на многопроцессорных ЭВМ делает возможным компьютерное моделирование реальных динамических процессов, наблюдаемых в галактиках: образование спиральных структур, шаровых скоплений, аккреционных дисков и т.п. Сложный комплекс физико-химических, механических и даже философских [4] проблем в целях оптимизации путей решения необходимо требует структурирования полной задачи на ряд подзадач — некоторых сегментов, охватывающих определенную часть проблемы. Эти сегменты, с одной стороны, должны быть достаточно автономны и иметь минимальное число связей с другими сегментами, чтобы, например, коренной пересмотр физико-химических концепций одного сегмента “не обрушил” теорию и алгоритмику решения других подзадач. С другой стороны, предмет исследования сегмента должен быть достаточно обширен, чтобы представлять не только некоторую структурную часть проблемы, не имеющую смысла вне ее, но и быть в большой мере самодостаточным, индивидуально вызывающим особый научный интерес. Таким структурным разделом большой астрофизической проблемы является сегмент “космическая газодинамика”.

И фундаментальные научные проблемы, от особенностей распространения ударных волн в протозвездном веществе до физико-химических процессов, происходящих в межгалактических газовых средах, и прикладные задачи, от проектирования летательных аппаратов различного назначения до расчетов их движения в атмосферах Земли и планет Солнечной системы, предъявляют исключительно высокие требования к компьютерным ресурсам (быстродействию и памяти). Достигнутый к настоящему времени уровень численного моделирования таких задач, как с использованием систем уравнений Больцмана или их высоких приближений типа модели Барнетта, Навье–Стокса, Эйлера, так и в упрощенной постановке вязкого ударного слоя и т.п., требует многочасовой работы суперкомпьютеров высокой производительности. Дальнейшее продвижение исследований в этой области и, соответственно, повышение требований

¹ Институт теоретической и прикладной механики СО РАН, ул. Институтская, 4/1, 630090, Новосибирск; e-mail: tarnav@itam.nsc.ru

² Новосибирский государственный университет, ул. Пирогова, 2, 630090, Новосибирск; e-mail: tarnavsky@inbox.ru

к компьютерным ресурсам делают уже очевидным исчерпанность пути использования в этой области все более и более мощных суперкомпьютеров с последовательным выполнением операций и требуют перехода к использованию многопроцессорных систем параллельных вычислений. В свою очередь, использование мультипроцессорных систем требует не только создания новых и реорганизации уже хорошо отработанных на широком классе задач алгоритмов и реализующих их комплексов основных и поддерживающих компьютерных программ, но также вызывает необходимость изменения всей стратегии и разработки новой тактики решения больших задач.

Настоящая работа представляет собой комплексный теоретический анализ и ориентирована на создание новых современных вычислительных технологий и методов параллельного программирования в области космической газодинамики. Предлагаемая разработка опирается на опыт [5–11] создания информационно-вычислительной системы “Поток-3”, предназначенной для исследования физики и механики процессов обтекания тел различной конфигурации гиперзвуковым потоком вязкого теплопроводного газа в широком диапазоне определяющих параметров и высот полета как в атмосфере Земли, так и в газовых средах с аномальными свойствами, которые моделируют условия в атмосферах Венеры, Марса и Юпитера.

2. Стратегия параллелизации. Весьма важным этапом создания параллельных вычислительных комплексов в любой предметной области, в том числе и в области космической газодинамики, является этап теоретической разработки глобальной стратегии параллелизации всей методологии компьютерного моделирования изучаемого физического (химического, биологического и т.п.) процесса. Эта глобальная стратегия опирается на принцип декомпозиции полной задачи на ряд одновременно выполняемых подзадач. Можно определить четыре вида декомпозиции: физико-математическую, геометрическую (крупномасштабную и мелкомасштабную), технологическую и декомпозицию “по главным входным параметрам”.

Физико-математическая декомпозиция задачи является одним из глобальных видов структурирования высокого уровня и связана с расщеплением исследуемого физического процесса по составляющим его подпроцессам и, соответственно, с сегментацией общего математического алгоритма решения полной задачи на ряд алгоритмов решения составляющих подзадач. При изучении процессов, происходящих в газовых средах с учетом широкого спектра физических явлений, могут быть промоделированы отдельно (параллелизованы) подпроцессы “газовая динамика”, “термодинамика”, “теплопередача”, “гравитация”, “метрика пространства”, “динамика твердой фазы” и т.п. с их соответствующим интерфейсом.

Геометрическая декомпозиция задачи (здесь и в дальнейшем имеется в виду в основном крупномасштабная декомпозиция) и последующая параллелизация вычислений, так же как и физико-математическая, является одним из видов глобального структурирования высокого уровня и позволяет существенно снизить величину астрономического времени, требуемого на решение. Геометрическая декомпозиция заключается в разделении всей области интегрирования на карту подобластей (сегментов), решение в которых производится одновременно и независимо с последующей соответствующей сшивкой решений на границах прилегающих друг к другу сегментов.

Дополнительно к физико-математическому и геометрическому виду декомпозиции полной задачи на ряд подзадач с их параллельным выполнением может быть рассмотрен и использован еще один вид декомпозиции — технологическая декомпозиция, которая, в свою очередь, может быть разделена на три подвида: высокого, среднего и низшего уровня (подробнее см. [5, 9]). Вообще говоря, это достаточно разные типы декомпозиции, объединяемые лишь тем фактом, что они все представляют только технические способы проведения параллельного счета. Для оптимизации численного решения полной задачи обеспечивающий это решение комплекс программ, как правило, структурирован и состоит из набора действий, имеющих определенный алгоритмический смысл, вычислительно реализуемый замкнутыми автономными подпрограммами. Такое структурирование и является технологической декомпозицией (сегментацией) высокого уровня. Данные подпрограммы на однопроцессорных ЭВМ, естественно, выполняются последовательно, даже если в этом нет необходимости и часть из них может выполняться одновременно (параллельно).

Далее, внутри каждой подпрограммы можно выделить часть операций, которые являются независимыми и могут выполняться параллельно. Такое глубокое сегментирование операций, вплоть до арифметических операций во внутренних циклах, определяется [5] как технологическая декомпозиция низшего уровня, а соответствующее распаралеливание — технологической параллелизацией низшего уровня.

Технологическая параллелизация среднего уровня находится между технологическими параллелизациями высшего и низшего уровней и является специальной организацией вычислений достаточно часто встречающихся математических элементов, таких как полиномиальные интерполяции, свертка, умножение матриц, прогонка [6, 7] и т.д. Однако анализу эффективности именно этих видов распаралеливания

с их применением в вычислительных комплексах научных и прикладных программ уделяется мало внимания, поскольку такая параллелизация пока практически не используется вследствие больших теоретических и реализационных трудностей. Это связано с тем, что, например, реализация крупномасштабной декомпозиции не требует реконфигурирования процессорного пространства: в течение всего выполнения задачи каждому сегменту расчетной области соответствует один и тот же процессор-вычислитель. Более того, даже процессоры-маршрутизаторы выполняют одну и ту же функцию с одними и теми же структурами данных и их обменом между одними и теми же процессорами-вычислителями, поскольку геометрическая сегментация является выбранной однозначно, постоянной и не меняющейся в процессе счета.

При технологической декомпозиции среднего и более глубокого, низшего уровня процессорное пространство необходимо должно реконфигурироваться с изменением функций значительного числа процессоров, поскольку такие операции, как прогонка [10, 11] или сложение векторов, являются существенно математически разнородными и соответственно существенно разнородными по их компьютерному исполнению. Следует отметить, что несмотря на сложность практической реализации этого вида распараллеливания (технология параллельного программирования такого класса процессов приведена, например, в [12]), его применение в каждом из сегментов крупномасштабной геометрической параллелизации может быть в комплексе чрезвычайно эффективным.

Последним, четвертым, достаточно специфическим видом декомпозиции является так называемая “траекторная” задача, когда требуется провести цикл расчетов при изменении одного или нескольких глобальных входных параметров, в частности для задач космической газодинамики — определенного набора значений гравитирующей массы M и температуры T протопланетного облака. Такая задача хорошо коррелирует с архитектурой многопроцессорной системы SIMD-типа с распределенной памятью: “единые инструкции — множественные данные” (весь полный пакет программ, в комплексе моделирующих механику и физику явления, — заданный набор значений M и/или T). Следует подчеркнуть возникающую при этом проблему сверхбольшого объема полученной информации, анализ которой представляет собой далеко не простую и практически пока не решенную задачу.

Ниже все эти виды декомпозиции будут рассмотрены особо.

3. Физико-математическая декомпозиция. Для понимания сущности этого вида декомпозиции и организации соответствующих параллельных вычислений следует кратко изложить (подробно см. [13]) физико-математические основы моделирования задач космической газодинамики.

Газовая динамика. Физические процессы, протекающие в газовых средах, определяются фундаментальными законами сохранения массы, импульса, энергии, которые могут быть записаны в дифференциальной форме в дивергентном виде:

$$\frac{\partial \mathbf{U}}{\partial t} + \sum_{k=1}^3 \nabla_k \mathbf{F}^k(\mathbf{U}) = \mathbf{A}, \quad (1)$$

где $\mathbf{U} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)$ — вектор консервативных переменных, представляющий плотности сохраняющихся величин (массы, импульса, энергии); $\mathbf{F}$ — вектор их потоков; $\mathbf{A}$ — вектор массовых сил; ∇ — дифференциальный оператор “набла”, в частности, в декартовой системе координат $\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right)$.

Система (1), эволюционная по времени t , в пространстве трех измерений (x, y, z) состоит из пяти уравнений, связывающих сохраняемую величину U_i с ее потоком через границы или источником ее производства внутри рассматриваемой локальной области:

$$\frac{\partial \rho}{\partial t} + \sum_{k=1}^3 \nabla_k (\rho v_k) = 0, \quad (2)$$

$$\frac{\partial \rho v_i}{\partial t} + \sum_{k=1}^3 \nabla_k (\rho v_i v_k) + \nabla_i p = F_i; \quad i = 1, 2, 3, \quad (3)$$

$$\frac{\partial \rho E}{\partial t} + \sum_{k=1}^3 \nabla_k v_k (\rho E + p) = - \sum_{k=1}^3 \nabla_k q_k + Q + \sum_{k=1}^3 v_k F_k, \quad (4)$$

$$\mathbf{F} = \mathbf{F}_f + \mathbf{F}_g. \quad (5)$$

Здесь ρ — плотность, $\mathbf{v}$ — вектор скорости, $E = e + v^2/2$ — массовая плотность полной энергии, e — внутренняя энергия, p — давление, $\mathbf{F}$ — суммарный вектор внешних сил, $\mathbf{q}$ — тепловой поток, Q — локальное энерговыделение, $\mathbf{vF}$ — удельная работа внешних сил. Смысл $\mathbf{F}_f$ и $\mathbf{F}_g$ (фотонная и гравитационная составляющие) будет пояснен ниже.

Программный сегмент (автономный модуль) “Газовая динамика”, реализующий алгоритмику решения уравнений (1)–(5), характеризуется следующим списком параметров. Входные параметры подразделяются на две группы:

1) собственные параметры модуля ρ , $\mathbf{v}$, e и их комбинация E , насчитанные на предыдущем слое по времени;

2) получаемые от других сегментов параметры p , $\mathbf{q}$, Q , $\mathbf{F}_f$, $\mathbf{F}_g$.

Выходными параметрами модуля являются ρ , $\mathbf{v}$, e — величины на следующем слое по времени. Кроме того, входными для этого и всех других сегментов являются сеточные параметры — число расчетных узлов и их координаты, а также (не для всех модулей) шаг по времени.

Термодинамика. Система (2)–(5) замыкается уравнением состояния

$$p = Z(\rho, T) \frac{R}{\mu} \rho T, \quad (6)$$

где R — универсальная газовая постоянная ($R = 8.31$ Дж K^{-1} моль $^{-1}$), T — абсолютная термодинамическая температура, μ — молекулярный (атомарный) вес газовой среды. При $Z = 1$ уравнение (6) переходит в уравнение состояния идеального газа Клапейрона–Менделеева.

Весьма удобным для использования является уравнение состояния, связывающее давление с плотностью и внутренней энергией:

$$p = p(\rho, e). \quad (7)$$

В классической аэродинамике уравнение (7) имеет вид

$$p = (\gamma - 1)\rho e, \quad (8)$$

где γ — показатель адиабаты газовой среды, величина, как правило, постоянная во всей области течения газа.

При моделировании задач космической газодинамики уравнение состояния (6) может весьма сильно отклоняться от идеального вследствие различных физических процессов — возбуждения внутренних степеней свободы (вращательных, колебательных, электронных), диссипации и рекомбинации, ионизации и т.п. При структурировании программного комплекса на сегменты, моделирующие (отдельно в вычислительном смысле) различные физические процессы, удобно использовать метод эффективного показателя адиабаты. Сущность этого метода (подробнее см. [14, 15]) заключается в использовании формы записи уравнения состояния в классическом виде (8), в котором γ является существенно переменной величиной:

$$\gamma = \gamma(\rho, T). \quad (9)$$

Этот метод эффективен для моделирования нестационарных течений газа с большими перепадами p и T в расчетной области в случае равновесных или квазиравновесных процессов. (Существенно неравновесные процессы требуют дополнительных связей между программными сегментами. Этот вопрос лежит вне рамок настоящей работы.)

Программный сегмент (автономный модуль) “Термодинамика”, реализующий алгоритмику решения уравнений (6)–(9), оперирует в следующей параметрической среде: входные параметры — ρ , e , выходные — p , T , Z , μ . Генерируемые этим сегментом параметры Z и μ не требуются для функционирования других модулей, но полезны при анализе полученной информации.

Теплопередача. Тепловой поток $\mathbf{q}$ в уравнении баланса энергии (4) определяется двумя физическими механизмами — теплопроводностью газа и переносом тепла излучением:

$$\mathbf{q} = \mathbf{q}_t + \mathbf{q}_f. \quad (10)$$

Первый из них связан с разностью температур в слоях газа и в стационарном (квазистационарном) случае описывается законом Фурье

$$\mathbf{q}_t = -\kappa \nabla T, \quad (11)$$

в котором коэффициент теплопроводности κ может быть вычислен по значениям ρ и T при постулировании, например, максвелловского распределения молекул по скоростям, или взят непосредственно из каких-либо термодинамических таблиц. Механизм передачи тепла излучением описывается следующими уравнениями. Тепловой поток $\mathbf{q}_f$ определяется решением системы

$$\nabla \mathbf{q}_f = \frac{c}{l} (U - U_0), \quad (12)$$

$$\nabla U = -\frac{3}{cl} \mathbf{q}_f, \quad (13)$$

где l — росселандов пробег (средняя длина пробега квантов излучения), c — скорость света ($3 \cdot 10^8$ м/с), $(U - U_0)$ — отклонение плотности энергии от равновесного, которое определяется по закону Стефана–Больцмана

$$U_0 = \beta T^4, \quad (14)$$

где постоянная Стефана–Больцмана $\beta = 7.56 \cdot 10^{-16}$ Дж м⁻³ К⁻⁴.

Используя формализм классической термодинамики, можно получить уравнение состояния излучающего газа (имеется в виду только фотонный газ, без атомарной составляющей)

$$p_f = U/3, \quad (15)$$

где плотность энергии излучения U , определяемая из системы (12)–(13) исключением $\mathbf{q}_f$, является аналогом удельной внутренней энергии ϵ в (7), p_f — давление излучения.

В равновесном случае, приложив закон Стефана–Больцмана (14) к уравнению состояния (15), можно представить давление излучения в виде функции температуры

$$p_f(T) = \beta T^4/3. \quad (16)$$

В общем (15) или частном (16) виде давление p_f должно быть учтено в общей системе уравнений (2)–(5) как одна из составляющих ($\mathbf{F}_f$) суммарного вектора ($\mathbf{F}_\Sigma = \mathbf{F}_f + \dots$):

$$\mathbf{F}_f = \rho \nabla p_f. \quad (17)$$

Программный сегмент (автономный модуль) “Теплопередача”, реализующий алгоритмику решения уравнений (10)–(17), оперирует в следующей параметрической среде: входные параметры — k , l , T , выходные — $\mathbf{q}_t$, $\mathbf{q}_f$ (и их комбинация $\mathbf{q}$), $\mathbf{F}_f$.

Гравитация. Гравитационное воздействие на газовую среду определяется двумя физическими механизмами — внешним источником тяготения и собственной самогравитацией газа:

$$\Phi = \Phi_e + \Phi_i, \quad (18)$$

где Φ_e , Φ_i , Φ — соответственно гравитационные потенциалы внешнего источника, внутренний (газовой среды), суммарный.

Внешний источник тяготения может находиться как внутри, так и вне области моделирования (в последнем случае имеет место достаточно специфично поставленная задача). Обычно исследуются задачи в приближении “точечного” источника с размером, существенно меньшим характерного размера задачи. В этом случае может быть использован ньютоновский потенциал

$$\Phi_e = -G \frac{M}{r}, \quad (19)$$

где $G = 6.67 \cdot 10^{-11}$ м³ кг⁻¹ сек⁻² — гравитационная постоянная. M — масса источника тяготения, r — расстояние от него до элемента газовой среды.

Гравитационный потенциал $\Phi_i(x, y, z)$ газовой среды с распределенной массой, характеризующейся функцией плотности $\rho(x, y, z)$ в декартовой или иной системе координат, определяется из решения уравнения Пуассона

$$\Delta \Phi_i = 4\pi G(\rho + \rho_0). \quad (20)$$

Величина ρ_0 в (20) позволяет учитывать так называемую “репульсивную” силу, некоторый гравитационный “фон”, частично нейтрализующий поле газовой среды. Заметим, что в большинстве случаев принимается $\rho_0 = 0$, если при моделировании какой-либо галактической системы не рассматривается влияние как других галактических систем, так и невидимой (темной) материи, учет влияния которой, несмотря на абсолютную неясность ее природы, начинает появляться в настоящее время в некоторых работах.

Гравитационная составляющая $\mathbf{F}_g$ вектора внешних сил $\mathbf{F}$ в системе уравнений (3)–(5) определяется гравитационным потенциалом (18):

$$\mathbf{F}_g = -\rho \nabla \Phi = -\rho \nabla (\Phi_e + \Phi_i). \quad (21)$$

При этом в (21) могут быть учтены по (19), (20) потенциалы Φ_e и Φ_i как по отдельности, так и вместе, что определяется физическим смыслом решаемой задачи.

Работа внешних сил (гравитационное трение) $\mathbf{v} \mathbf{F}_g$ вносит лишь небольшой вклад в общий баланс энергии (5), за исключением задач с очень большими градиентами плотности и ее величиной, что может привести к образованию высокоградиентных полей тяготения.

Программный сегмент (автономный модуль) “Гравитация”, реализующий алгоритмику решения уравнений (18)–(21), оперирует в следующей параметрической среде: входные параметры — ρ (и ρ_0 , если производится его учет), выходные — Φ_e, Φ_i (их комбинация Φ и его координатная производная $\mathbf{F}_g$).

Метрика пространства. Всякое гравитационное поле является не чем иным, как изменением метрики пространства-времени, соответственно чему оно определяется метрическим тензором g_{ik} . То же самое относится и к неинерциальным системам отсчета, которые эквивалентны гравитационным полям (А. Эйнштейн, Общая теория относительности, 1916). При моделировании задач космической газодинамики может быть необходим учет искривления (неевклидовости) пространства при наличии сильных внешних полей тяготения или даже самогравитации газового облака, если его масса (плотность или размер) достаточно велика. Эти эффекты могут быть узко локализованы (“черные дыры”) и, соответственно, могут приводить к существенно неравномерным распределениям газодинамических параметров, в особенности для высокоскоростных течений газа, характеризующихся сложными динамическими картинами ударно-волновых структур.

Задающий свойства риманова пространства метрический тензор g_{ik} может быть найден (в принципе; в реальности его вычисление требует как специального моделирования, так и существенных вычислительных затрат) из основных уравнений общей теории относительности (уравнений гравитационного поля) — уравнений Эйнштейна

$$R_{ik} - \frac{1}{2} g_{ik} R = \kappa T_{ik}, \quad (22)$$

где $\kappa = \frac{8\pi G}{c^4}$ — постоянная Эйнштейна, c — скорость света, T_{ik} — тензор энергии-импульса материи (включая электромагнитное поле), R — скалярная кривизна пространства (инвариант), а тензор Риччи R_{ik} получен путем свертывания тензора кривизны пространства R_{jlmk} :

$$R = g^{ik} R_{ik}; \quad R_{ik} = g^{jm} R_{jimk}; \quad (23)$$

$$R_{ikjm} = \frac{1}{2} \left(\frac{\partial^2 g_{im}}{\partial x^k \partial x^j} + \frac{\partial^2 g_{kj}}{\partial x^i \partial x^m} - \frac{\partial^2 g_{ij}}{\partial x^k \partial x^m} - \frac{\partial^2 g_{km}}{\partial x^i \partial x^j} \right) + g_{np} (\Gamma_{kj}^n \Gamma_{im}^p - \Gamma_{km}^n \Gamma_{ij}^p). \quad (24)$$

Символы Кристоффеля, использованные в (24), имеют вид

$$\Gamma_{km}^i = \frac{1}{2} g^{is} \left(\frac{\partial g_{ks}}{\partial x^m} + \frac{\partial g_{ms}}{\partial x^k} - \frac{\partial g_{km}}{\partial x^s} \right). \quad (25)$$

Одним из приближенных способов учета искривления пространства является использование метрики Шварцшильда для гравитационного поля центрально-симметричного изолированного тела. Соответствующий ей метрический тензор диагонален и его компоненты равны:

$$g_{00} = 1 + 2\Phi/c^2, \quad g_{11} = g_{22} = g_{33} = 1/g_{00}, \quad (26)$$

где Φ — гравитационный потенциал.

Подчеркнем, что обсуждение проблем общей теории относительности ни в коей мере не является предметом данной работы (подробнее см., например, [2, 16, 17]); рассматриваются только возможные способы использования ее теоретических аспектов (в том или ином приближении) при создании специализированного программного сегмента вычислительного комплекса, опирающегося на возможности современной компьютерной техники, для моделирования задач астрофизики.

Программный сегмент (автономный модуль) “Метрика пространства”, реализующий алгоритмику решения уравнений (22)–(26), имеет только один входной параметр — Φ (а при расширении круга моделируемых задач — электромагнитные поля $\mathbf{E}$ и $\mathbf{H}$) и один выходной — g_{ik} .

4. Распараллеливание по физическим процессам. На основе выполненной физико-математической декомпозиции задачи, алгоритмов ее решения и реализующих их программных кодов может быть проведена процедура параллелизации вычислений, называемая “распараллеливанием по физическим процессам”. На рис. 1 представлена последовательная схема операций вычислительного комплекса. Ею инициализирующая система (ИС) реализует подготовку к запуску алгоритмов решения приведенных выше уравнений: осуществляет ввод определяющих задачу параметров (как индивидуальных, требуемых именно этой конкретной задачей, так и мировых констант), генерирует дискретную расчетную сетку, общую для всех модулей, создает банк начальных данных (БНД) для сегмента “Газовая динамика” (1ГД), производит запись в общий банк текущих данных (ОБТД) и передает управление модулю 1ГД. На рис. 1 и других тонкими стрелками обозначены потоки данных, жирными — потоки команд (переход управления).

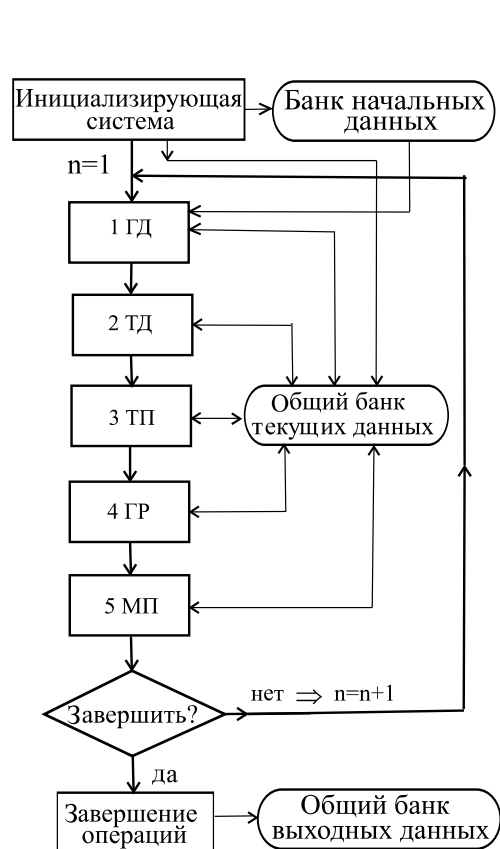


Рис. 1. Физико-математическая сегментация. Последовательная схема вычислений

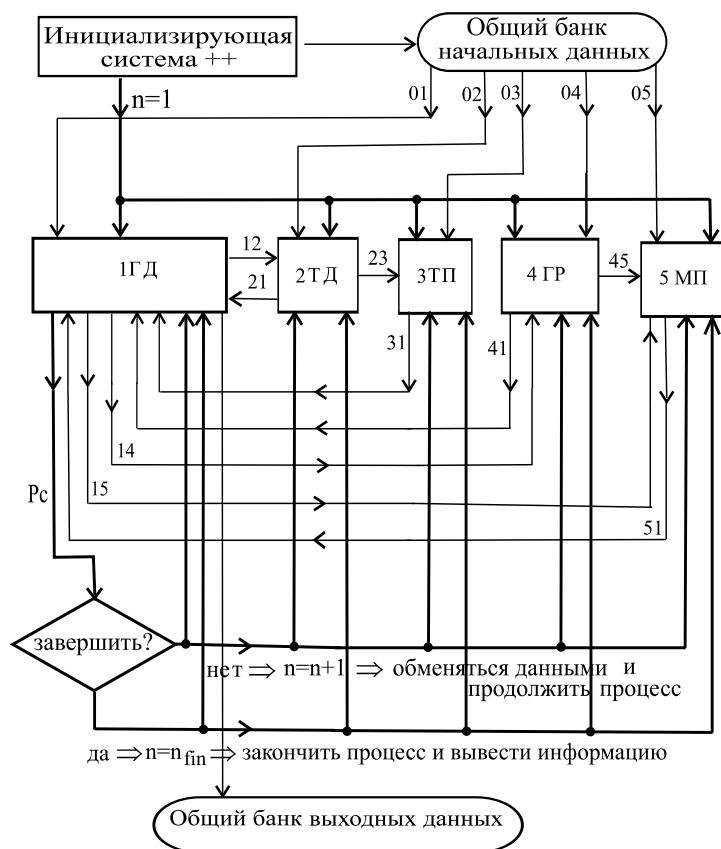


Рис. 2. Физико-математическая сегментация. Параллельная схема вычислений

Данный модуль, получая необходимые входные параметры (при первом обращении к нему — из БНД, затем — из ОБТД), исполняет свой алгоритм, размещает сгенерированные им выходные параметры в ОБТД. Далее управление передается последовательно в модули “Термодинамика” (2ТД), “Теплопередача” (3ТП), “Гравитация” (4ГР), “Метрика пространства” (5МП). Таким образом, вся схема

вычислений реализуется в виде цепочки исполнения команд

$$1ГД \longrightarrow 2ТД \longrightarrow 3ТП \longrightarrow 4ГР \longrightarrow 5МП, \quad (27)$$

обеспечивающих решение уравнений соответственно (1)–(5), (6)–(9), (10)–(17), (18)–(21), (22)–(26). Потоки данных являются стандартными по направлению, но длины и структуры их, естественно, индивидуальны для каждого модуля — входные параметры вызываются из ОБТД и туда же помещаются выходные:

$$ОБТД \longrightarrow МОДУЛЬ \longrightarrow ОБТД, \quad (28)$$

где символом “МОДУЛЬ” обозначен любой модуль из списка (1ГД, 2ТД, 3ТП, 4ГР, 5МП).

После исполнения последнего модуля (5МП) проводится анализ критерия достижения результата и в зависимости от этого управление или вновь возвращается модулю 1ГД (так называемый “переход на следующий слой по времени”) и далее вычисления повторяются по цепочке (27), или операции завершаются и создается общий банк выходных данных (ОБВД). Фактически ОБВД содержит те же данные, что и ОБТД после последнего обращения к нему, но имеет другую структуру, удобную для анализа результатов расчета (цифровые таблицы и 1D-, 2D- и 3D-графики).

При параллельном счете схемы (27)–(28) должны быть существенно реорганизованы. Для определенности ограничимся системами с распределенной памятью. Назначим исполнение каждого модуля в качестве отдельного процесса на своей собственной машине (или группе машин; об этом ниже). Может быть предложена структура, приведенная на рис. 2. Так же, как и ранее, запуск вычислительного комплекса производится инициализирующей системой (ИС++), сконструированной из ИС последовательного счета добавлением двух новых функций (два плюса в аббревиатуре). Первая функция ИС++ заключается в генерировании начальных данных — входных параметров для всех модулей, а не только для модуля 1ГД. Для этого в ИС++ включается весь программный комплекс с последовательным (однопроцессорным) выполнением операций, показанный на рис. 1. Вся его цепочка (27) исполняется только один раз для генерации всех данных, необходимых для инициализации каждого модуля, со своими индивидуальными входными параметрами. При этом ОБТД последовательного счета становится общим начальным банком данных (ОНБД) параллельного счета (рис. 2) и функционирует под управлением главной машины. Вторая дополнительная функция ИС++ связана с распределением процессорного пространства и определяется конкретным типом используемой МВС и ее операционной среды.

После конфигурирования процессорного пространства, генерации параметров и заполнения ОБНД главная машина (хост) инициирует (запускает) одновременно все процессы исполнения модулей 1ГД, 2ТД, 3ТП, 4ГР, 5МП на собственных машинах (группах машин). По окончании выполнения операций, существенно разных по времени, каждый процесс сообщает об этом в хост. Кроме того, выбранный в качестве основного модуль 1ГД направляет в хост группу данных, по которым производится контроль достижения требуемого результата. Если результат не достигнут, то производится “переход на новый слой по времени”: из главной машины во все остальные выдается команда об обмене данными и, после этого, о выполнении нового такта операций со вновь полученными от других модулей параметрами. Потоки данных на рис. 2 промаркированы двузначными цифрами: первая обозначает номер отправляющей, вторая — номер получающей машины. Заметим, что в отличие от последовательной схемы (рис. 1) здесь не создается ОБТД, а каждый модуль организует свой собственный локальный банк данных, содержащий (а также получающий и принимающий) только те данные, которые необходимы именно этому модулю.

Файлы информации, которыми обмениваются процессы, одинаковы по структуре (массивы одной размерности), но существенно различаются по длине (количеству передаваемых/получаемых массивов).

Обратим внимание не только на значительное число пересылок, но и на существование “встречных” потоков 12 и 21, 14 и 41. Все это предъявляет особые требования к синхронизации процедур приема/передачи сообщений во избежание неприятных ситуаций: тупиков или зависаний (подробнее см. [1, 5]).

Если критерий решения задачи выполнен, то главная машина рассылает всем процессам команду окончания операций и команду передачи данных в одну из машин, например, выполнявшую процесс 1ГД, для создания общего банка выходных данных и последующего анализа полученной информации.

Такой вид декомпозиции и соответствующий ему тип параллелизации счета, существенно ускоряющий процесс вычислений, вместе с тем ставит вопрос об эффективности использования задействованного сегмента мультимпьютера. Это связано с тем, что времена исполнения модулей весьма различаются.

Список входных данных

Имя модуля	Поток	Массив
1ГД	21	p
	31	$\mathbf{q}, \mathbf{F}_f$
	41	$\mathbf{F}_g$
	51	g_{ik}
2ТД	12	ρ, ϵ
	32, 42, 52	нет
3ТП	23	T
	13, 43, 53	нет
4ГР	14	ρ
	24, 34, 54	нет
5МП	15, 25, 35	нет
	45	Φ

Список выходных данных

Имя модуля	Поток	Массив
1ГД	12	ρ, ϵ
	14	ρ
	13, 15	нет
2ТД	21	p
	23	T
	24, 25	нет
3ТП	31	$\mathbf{q}, \mathbf{F}_f$
	32, 34, 35	нет
4ГР	41	$\mathbf{F}_g$
	45	Φ
	42, 43	нет
5МП	51	g_{ik}
	52, 53, 54	нет

Представим время исполнения одного модуля t_i в виде

$$t_i = t_c a_i, \quad i \in [1, 5], \quad (29)$$

где t_c — некоторое характерное время (общий масштаб), а значение индекса i соответствует введенному ранее первому символу аббревиатуры модуля ($i = 1$ соответствует модулю 1ГД; $i = 2$ — модулю 2ТД и т.д.). Поскольку производятся лишь оценки, то конкретная величина t_c в (29) для дальнейшего значения не имеет; важны лишь относительные (друг к другу) значения t_i , определяемые коэффициентами a_i (округленными до порядков).

Так, при моделировании коллапса газового облака с идеальным уравнением состояния в ньютоновском поле тяготения и евклидовом пространстве [13] имеют место для модулей 1ГД, 2ТД, 3ТП, 4ГР, 5МП соответственно коэффициенты

$$a_1 = 100, \quad a_2 = 1, \quad a_3 = 10, \quad a_4 = 0.1, \quad a_5 = 0. \quad (30)$$

Значения коэффициентов (30) показывают, что во время счета каждого шага (такта) алгоритма все процессы будут ожидать окончания первого, и при этом хорошего ускорения решения полной задачи достичь невозможно.

С изменением физики моделирования космических процессов значения в (30) могут существенно изменяться. Так, учет неидеальности уравнения состояния может привести к $a_2 = 10$ или даже к $a_2 = 100$, в зависимости от длины учитываемого вириального разложения. Более детальный учет влияния излучения приводит к $a_3 = 100$, изменение модели тяготения с ньютоновского на самогравитирующее поле сразу резко увеличивает a_4 до 1000. Значение a_5 в (30) для пространства с метрикой $g_{ik} = 1$ означает, что операции в 5МП вообще не производятся, приближение Шварцшильда с метрикой (26) приводит к $a_5 = 1$, а необходимость решения уравнения Эйнштейна (22) в римановом пространстве (23)–(25) может потребовать необозримых затрат времени ($a_5 \gg 100$).

Этот анализ, при желании эффективно использовать ресурсы суперкомпьютера (что, вообще говоря, вовсе не обязательно — см. дискуссию в [9]), ставит две новые проблемы. Первая — о дальнейшей параллелизации вычислений в каждом из модулей (именно в каждом, поскольку невозможно априори определить, в каком из них число операций будет максимальным, если физико-математическое наполнение вычислительного комплекса достаточно универсально). Вторая — о необходимости разработки специальных технологий гибкого реагирования на “изменение физико-математической обстановки” и создания реконфигурируемого процессорного пространства для обеспечения примерно одинаковой нагрузки на каждый его элемент. Вторая проблема успешно решается специалистами в области параллельного программирования (см. [12]). Пути решения первой проблемы рассматриваются ниже.

5. Технологическая декомпозиция алгоритма и распараллеливание программного кода.

Алгоритм любого из модулей 1ГД, 2ТД, 3ТП, 4ГР, 5МП может быть записан в общем виде:

$$W = W\left(F, \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial^2 F}{\partial x^2}, \frac{\partial^2 F}{\partial y^2}\right), \quad (31)$$

где $F(x, y)$, $W(x, y)$ — непрерывные функции. Для краткости изложения сущности этого вида распараллеливания рассматривается только двумерная задача; в пространственном случае параллелизация проводится аналогично.

Проведем “расщепление” (31) по координатным направлениям

$$W = W^x + W^y \quad (32)$$

так, чтобы разделить члены, содержащие различные координатные производные:

$$W^x = W^x\left(F, \frac{\partial F}{\partial x}, \frac{\partial^2 F}{\partial x^2}\right), \quad (33)$$

$$W^y = W^y\left(F, \frac{\partial F}{\partial y}, \frac{\partial^2 F}{\partial y^2}\right). \quad (34)$$

Заметим, что рассматриваемые алгоритмы космической газодинамики (во всяком случае — основные) всегда допускают представление (33)–(34), в крайнем случае с использованием дополнительных преобразований в специальные системы координат.

При построении вычислительного алгоритма реальному физическому пространству (x, y) соответствует дискретное сеточное пространство

$$\{R_{ij}\} = i \otimes j, \quad i \in [1, I] \quad j \in [1, J], \quad (35)$$

в узлах которого определены значения дискретных массивов x_i , y_j , F_{ij} и W_{ij} , соответствующие непрерывным координатам x, y и функциям $F(x, y)$ и $W(x, y)$. С учетом традиционных аппроксимаций непрерывных производных их дискретными аналогами алгоритмы модулей (33)–(34) соответственно могут быть записаны в символическом виде:

$$W_{ij}^x = i\text{-OPERATIONS}(F_{ij}, F_{i\pm 1, j}, F_{i\pm 2, j}, x_i, x_{i\pm 1}, x_{i\pm 2}), \quad (36)$$

$$W_{ij}^y = j\text{-OPERATIONS}(F_{ij}, F_{i, j\pm 1}, F_{i, j\pm 2}, y_j, y_{j\pm 1}, y_{j\pm 2}), \quad (37)$$

где F , x , y — входные, W^x и W^y — выходные массивы программных сегментов.

Применение метода расщепления (32)–(34) детерминирует в общей стратегии параллелизации вычислительного комплекса тактику локального (технологического по терминологии [5]) распараллеливания программных сегментов — модулей, реализующих в виде отдельных подпрограмм операции (36)–(37) в виде С- и L-типов (“column” и “lines”) параллельного счета [11]. В сущности, эти способы параллелизации опираются на методику выделения или специальной организации в алгоритмике решения задачи особых структур — циклов ParDo [1].

C-тип параллелизации. Данный тип распараллеливания предназначен для ускорения вычислений массива W_{ij}^x . В операциях (36) особое, даже ключевое, значение имеет то обстоятельство, что при расчете W_{ij}^x имеет место сцепление операций по первому (i) и их независимость по второму (j) индексу. Это означает возможность проведения операций одновременно по всем колонкам (столбцам) расчетной сетки (35). Циклы вычислений в (36) должны быть записаны в последовательности

$$\begin{aligned} &\text{do } j=1, J \\ &\quad \text{do } i=1, I \\ &\quad \quad \dots i\text{-OPERATIONS} \dots \\ &\quad \text{end do} \\ &\text{end do} \end{aligned} \quad (38)$$

В (38) все шаги внешнего цикла независимы друг от друга и могут быть выполнены на разных машинах. Таким образом, общее сеточное пространство R_{ij} может быть разделено на P_C подпространств (“вертикальных колонок”), не обязательно одинаковых, в каждой из которых операции (36) являются независимыми процессами ${}^P W_{ij}^x$ и могут быть проведены параллельно:

$$W_{ij}^x = \sum_{p=1}^{P_C} {}^P W_{ij}^x. \quad (39)$$

При этом, естественно, для (39) должно выполняться условие

$$P_C \leq J. \quad (40)$$

Вариация P_C в пределах (40) может быть использована для общей балансировки числа операций в ветвях графа параллелизации с целью синхронизирования времени исполнения программных сегментов (29)–(30).

L-тип параллелизации. Данный тип распараллеливания предназначен для ускорения вычислений массива W_{ij}^y . В операциях (37) особое значение имеет то обстоятельство, что при расчете W_{ij}^y имеет место, наоборот, сцепление операций по второму (j) и их независимость по первому (i) индексу. Это означает возможность проведения операций одновременно по всем полосам (строкам) расчетной сетки (35). Циклы вычислений в (37) должны быть записаны в последовательности

$$\begin{aligned} &\text{do } i=1, I \\ &\quad \text{do } j=1, J \\ &\quad \quad \dots j\text{-OPERATIONS} \dots \\ &\quad \text{end do} \\ &\text{end do} \end{aligned} \quad (41)$$

В (41) все шаги внешнего цикла независимы друг от друга и могут быть выполнены на разных машинах.

Таким образом, общее сеточное пространство R_{ij} может быть разделено на P_L подпространств (“горизонтальных полос”), не обязательно одинаковых, в каждой из которых операции (37) являются независимыми процессами ${}^P W_{ij}^y$ и могут быть проведены параллельно:

$$W_{ij}^y = \sum_{p=1}^{P_L} {}^P W_{ij}^y. \quad (42)$$

При этом, естественно, для (42) должно выполняться условие

$$P_L \leq I. \quad (43)$$

Вариация P_L в пределах (43) может быть использована для общей балансировки числа операций в ветвях графа параллелизации с целью синхронизирования времени исполнения программных сегментов (29)–(30).

На рис. 3 показана схема параллелизации этого типа. Исполнение расщепления (32) вследствие независимости процессов организуется на двух группах машин (или на двух машинах, если дальнейшее распараллеливание не производится). Первая из этих двух групп реализует С-тип, а вторая — L-тип параллелизации, и включают в себя, соответственно, $P_C + 1$ и $P_L + 1$ машин. Особые процессоры-диспетчеры P_0 каждой группы производят сегментацию R -пространства на P_C и P_L подпространств, т.е. разделяют все задействованные массивы на подмассивы и рассылают их в соответствующие машины, которые производят операции (36) и (37) и затем возвращают подмассивы диспетчерам, составляющим по (39) и (42) полные массивы.

Размеры колонок и полос, в принципе, могут быть разными, но разбиение области на почти одинаковые сегменты оптимально, поскольку при этом число операций в них также почти одинаково, и, соответственно, время ожидания общего завершения всех процессов минимизируется.

Особенности практической реализации С- и L-типов распараллеливания подробно исследованы в [11], где показано, что, несмотря на исходную математическую тождественность, L-тип распараллеливания значительно более сложен (если используется язык FORTRAN, и наоборот при использовании языка СИ) в практической реализации, чем С-тип, требует существенного внимания от программиста и более затратен по компьютерным ресурсам (времени реконфигурации вычислительного пространства R_{ij} на подпространства и обратно, точнее — сегментации и последующего объединения массивов ${}^P W_{ij}^x$ в W_{ij}^x и ${}^P W_{ij}^y$ в W_{ij}^y и далее в W_{ij}).

6. Геометрическая декомпозиция. Крупномасштабное распараллеливание. Рассмотрим и кратко проанализируем еще один вид декомпозиции задачи — геометрическую декомпозицию и соответствующее ей так называемое крупномасштабное (крупноблочное) распараллеливание. Геометрическая декомпозиция заключается в разделении всей области интегрирования на карту подобластей (сегментов),

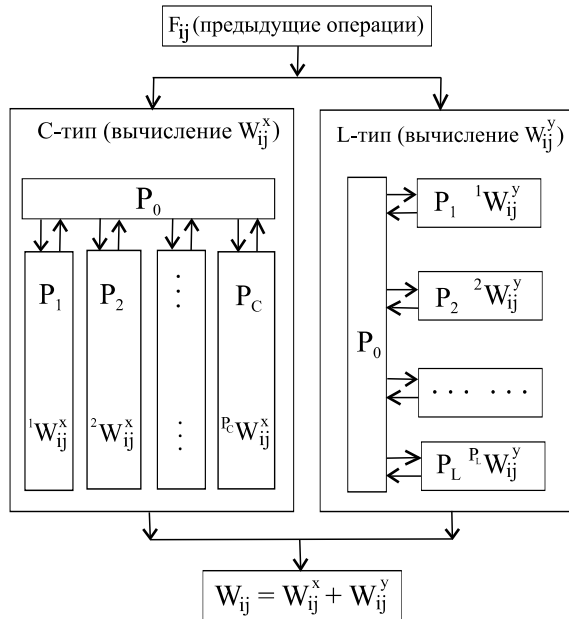


Рис. 3. Технологическая сегментация.
Параллельная схема вычислений

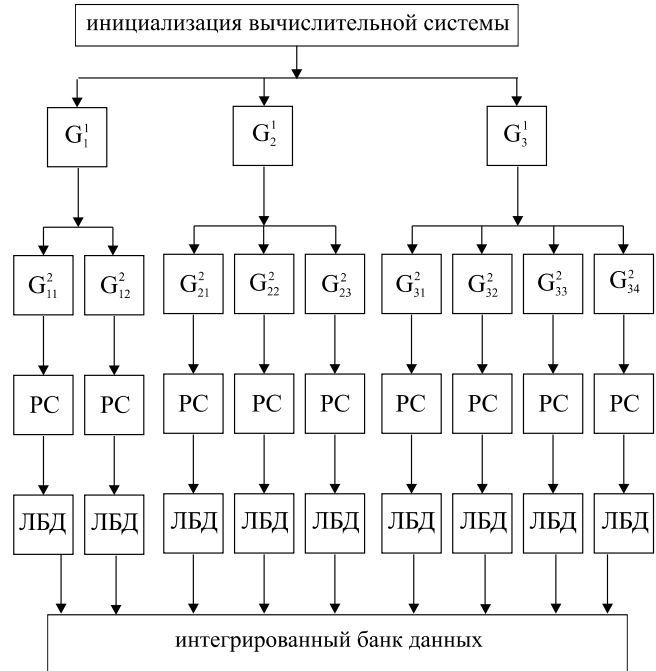


Рис. 4. Сегментация по главным входным параметрам. Глобальная параллелизация вычислительной системы

решение в которых производится одновременно и независимо с последующей соответствующей сшивкой решений на границах прилегающих друг к другу сегментов.

Этот тип параллелизации хорошо реализуется на SIMD-системах с распределенной памятью: “единые инструкции” (пакет программ, единообразных в каждом сегменте) — “множественность данных” (численные решения, существенно различные в разных сегментах). Следует отметить, что при геометрической декомпозиции возникает целый ряд сложных проблем, связанных с тем, что пространственная сегментация существенно зависит от математической постановки задачи: вида систем уравнений и используемых численных алгоритмов, которые должны удовлетворять определенным требованиям [5, 18].

Вследствие этого организация обмена данными граничащих сегментов наталкивается на трудности, иногда весьма существенные, трудности как математические, определяемые положением границ интерфейса (зоны эллиптичности, параболичности или гиперболичности исходных дифференциальных систем требуют особых подходов), так и технические, связанные с маршрутизацией данных. Вообще говоря, часть из этих проблем можно было бы снять, реализуя методологию геометрической декомпозиции на параллельных вычислительных системах с общей памятью, однако при этом, естественно, возникают другие проблемы. Заметим, что этот вид декомпозиции является одним из наиболее идейно простых способов параллелизации вычислительного процесса и наиболее часто применяется при распараллеливании методов расчета научных и прикладных задач. Здесь следует подчеркнуть и особо выделить то, что зачастую вся параллелизация начинается на крупномасштабной геометрической сегментации и на ней же заканчивается; дальнейшего распараллеливания операций в сегментах не производится [19, 20].

Приблизительно оценим эффективность этого вида распараллеливания. Пусть для решения полной задачи должна быть использована расчетная сетка (для простоты оценки — кубическая) в I^3 узлов. В каждом из этих узлов должно быть выполнено N операций со скоростью V . Тогда общее время расчета составит

$$T = NI^3V^{-1}.$$

Пусть в распоряжении вычислителя имеются P машин. Следовательно, возможно провести сегментацию (“нарезку”) всей области на P подобластей. Пусть эти подобласти одинаковы и также имеют кубическую форму с гранью $I_1 = I/P^{-1/3}$. Время расчета в подобласти определится, очевидно, как

$$T_1 = NI^3V_1^{-1}P^{-1}. \quad (44)$$

Следует обратить особое внимание, что в (44) участвуют значения скорости операций в отдельном кубике V_1 . Абсолютно неверно полагать, что $V_1 = V$. В общем случае

$$V_1 = k(P)V,$$

где k — некоторый специфический коэффициент “замедления”, определяемый конкретным алгоритмом и зависящий от числа сегментов. Например, для явных схем число операций в каждом узле сетки практически не зависит от других узлов, т.е. $k = 1$ при сегментировании на любое число подобластей. Для неявных схем, в особенности при использовании процедур прогонки (скалярной, векторной или тем более матричной) этот коэффициент может быть существенно меньше единицы (подробнее см. вычислительные эксперименты в [7] по “разрезанию” операций прогонки на P машин). Каждый кубик должен провести обмен (прием и передачу) данных, для газовой динамики обычно 8 величин в узле, с соседними на всех своих гранях. Пусть скорость пересылки одного параметра (весьма и весьма усредненная величина) составляет V_* . Тогда время приема и передачи всех данных на всех 6 гранях кубика составит

$$T_* = 2 \cdot 6 \cdot 8 \cdot I_1^2 \cdot V_*^{-1} \approx 100 I^2 P^{-2/3} V_*^{-1}.$$

Определим показатель эффективности E как отношение суммарного времени расчета и обменов $T_{1*} = T_1 + T_*$ в одном кубике ко времени T расчета полной задачи в несегментированной области. Сделаем важное замечание: время T_{1*} , вообще говоря, не вполне адекватно отражает время расчета сегментированной на кубики полной задачи, поскольку времена T_{1*} в каждом из кубиков могут существенно различаться (см. многоцелевые вычислительные эксперименты в [7]). На замедление расчета в отдельных кубиках влияет множество факторов, как статистически случайных, связанных с нахождением в мультипроцессоре других задач, так и систематических, определяемых разными техническими характеристиками отдельных машин, что всегда имеет место в большом комплексе. Поэтому, в принципе, для предварительного анализа можно использовать T_{1*} , но следует понимать, что на самом деле имеется в виду $\max (T_{1*})_i$ по всем i задействованным машинам. Итак,

$$E = \frac{T_{1*}}{T} = \frac{1}{P} \left[\frac{1}{k(P)} + 100 \frac{P^{1/3}}{IV_*} \frac{V}{N} \right]. \quad (45)$$

Разумеется, (45) есть существенно приближенная оценка, однако она весьма полезна для общего анализа, особенно в ключевой период принятия решения, какой тип распараллеливания реализовывать при минимуме ресурсов (технических и “человеческого фактора”) и максимуме желаемых, в том числе и в перспективе.

Анализ (45) по минимизации E приводит к следующим выводам. Если значение относительного показателя операций N/V в узле велико, то имеет смысл проводить этот тип параллелизации. То же самое можно сказать и в отношении сеточного параметра I , но его влияние не так существенно и сказывается только при использовании сеток очень большой размерности. При невысокой скорости V_* (низкой пропускной способности каналов) вряд ли можно рекомендовать распараллеливание. Но самым существенным является вопрос о числе машин P . На первый взгляд, чем больше P , тем эффективнее параллелизация. Однако влияние $k(P)$ таково, что может привести не только к ускорению, но даже и замедлению процесса решения всей задачи [9].

В заключение этого пункта следует сказать несколько слов о возможности использования еще одного вида параллелизации — мелкомасштабной технологической декомпозиции и “мелкозернистого” распараллеливания. Существо этого вида параллелизации, самого глубокого по уровню, заключается в распараллеливании арифметических операций внутри каждой подпрограммы алгоритма. Такое распараллеливание MIMD-типа “множество инструкций — множество данных” не унифицируется вообще и в каждой подпрограмме имеет свой характер. Применение этого вида параллелизации весьма трудоемко и в большой мере зависит от количества участвующего в проекте персонала и его квалификации, однако может дать высокое ускорение вычислительного процесса. При этом следует учесть возможное появление большого числа конфликтов, зависаний и даже тупиков при больших трудностях отладки.

7. Глобальная параллелизация по главным входным параметрам. Рассмотрим физико-математическую сущность этого вида распараллеливания. Вычислительный комплекс обеспечивает моделирование процессов космической (гравитационной) газовой динамики в широком диапазоне вариации параметров. Список этих параметров и их значения определяются конкретным типом решаемой задачи. Так, для задачи о коллапсе протопланетного облака главными входными параметрами являются начальная температура газа T и его масса M , определяющая конфигурацию и интенсивность поля тяготения.

При моделировании взрыва Сверхновой, помимо T и M , важными входными параметрами являются также начальная угловая скорость вращения звезды Ω и ее радиус R .

Как правило, для изучения динамики протекающих процессов и сценария их возможного финала необходимо провести значительное количество расчетов с вариацией всех параметров T , M , Ω , R . Такой цикл расчетов может быть организован в едином запуске с параллелизацией “по глобальным параметрам”, схема которой приведена на рис. 4. Введем обозначение списка главных параметров, вариации которых требует проводящееся исследование, например,

$$G^n = |T, M, \Omega, R, \dots|, \quad n \in [1, N]. \quad (46)$$

В (46) индексу n соответствует конкретный входной параметр, а значение N (длина списка) определяется физическим смыслом решаемой задачи и возможностью мультипроцессорной системы. Пусть требуется провести комплексный расчет K вариантов значений первого параметра G^1 , обозначаемых как

$$G_k^1 = |G_1^1, G_2^1, \dots, G_K^1|, \quad k \in [1, K]. \quad (47)$$

На рис. 4 показан ряд уровней распараллеливания, образующий некоторый односвязный граф: древовидную ветвящуюся непересекающуюся систему связей, выходящих из единого корня — входа в программный комплекс.

Для каждого из вариантов G_k^1 (т.е. для каждой главной ветви графа рис. 4) требуется провести расчет нескольких вариантов значений второго параметра G^2 из (46), который с учетом (47) обозначим через G_{kl}^2 . Индексы G_{kl}^2 имеют следующий графический смысл: верхний индекс указывает номер уровня графа (т.е. физический смысл данного параметра в (46)), первый нижний индекс отмечает принадлежность к той или иной главной ветви графа (содержит указание на числовое значение первого параметра), второй нижний индекс нумерует l -ветвь на k -той главной ветви (в иллюстрирующем двухуровневом графе рис. 4 это уже “лист”, последний уровень). Заметим, что число подвариантов в варианте в общем случае может быть разным на разных ветвях графа:

$$G_{kl}^n : \quad (\forall k \in [1, K] \implies l \in [1, L(k)]). \quad (48)$$

В терминах (47)–(48) конкретно изображенный на рис. 4 граф означает заказ на проведение одно-временного расчета девяти вариантов задачи: два подварианта со значениями массы $M = M_{11}$ и M_{12} , каждый из них при значении температуры $T = T_1$; три подварианта со значениями массы $M = M_{21}$, M_{22} и M_{23} , каждый из них при значении температуры $T = T_2$; четыре подварианта со значениями массы $M = M_{31}$, M_{32} , M_{33} и M_{34} , каждый из них при значении температуры $T = T_3$.

После организации файлов вариантов входных параметров с “индивидуальными” значениями каждый файл (окончательная, наиболее удаленная от корня графа структура) передает данные в программный комплекс РС и инициирует его процессорную систему. Результаты выполнения каждого процесса помещаются в собственный локальный банк данных (ЛБД). Затем необходимая для дальнейшего анализа информация объединяется из всех ЛБД в интегрированный банк данных. Подчеркнем еще раз, что обмена данными в процессе исполнения всего задания в целом не происходит ни между процессорными подгруппами, ни, тем более, между процессорными блоками. В этом состоит сущность данного типа параллелизации — распараллеливание по глобальным параметрам для программного комплекса, т.е. по основным входным параметрам (как правило, скалярным), определяющим физико-математический смысл задачи. Вообще говоря, процессорное время, затрачиваемое на исполнение операций решения одиночной задачи (независимого процесса) существенно зависит от используемых начальных данных: может оканчиваться целесообразным вначале заканчивать процесс, например, G_{11}^n , и затем принимать его результаты в качестве стартовых условий для процесса G_{12}^n (см. [5, 13]). Однако этот вопрос лежит вне рамок настоящей работы, и здесь будет предполагаться полная независимость всех процессов G_{kl}^n . В этом случае имеет место параллелизация SIMD-типа: единые инструкции (весь программный комплекс, приведенный на рис. 2) — множественность данных вообще без обмена между процессами.

Данный тип параллелизации, т.е. распараллеливание по глобальным параметрам, с точки зрения собственно системы МРІ принципиальных трудностей не вызвал, не было также проблем и конфликтов в отладке, однако при разработке выявился ряд технически сложных проблем унифицированного входа в систему, конфигурации процессорного пространства и организации автоматизированного сбора со всех задействованных процессоров полученной информации в интегрированный банк данных вычислительной системы. Кратко рассмотрим проблемы организации входа в систему и ветвление по параметрам.

Из более чем 50 глобальных входных параметров, определяющих физический, математический, алгоритмический и режимно-технический смысл задачи, были отобраны 12 “основных”, распараллеливание по которым (перебор вариантов) наиболее целесообразно. Максимальная глубина распараллеливания N , т.е. количество уровней параллелизации (см. рис. 4), для первого этапа была выбрана равной 4 (отладка проводилась, как правило, для $N = 2$). Был построен граф параллелизации, позволяющий в произвольном порядке размещать основные параметры по уровням: для приведенных выше примеров вначале определялся список температур газа, для которых необходимо получить решение, а затем для каждого элемента этого списка определялся свой список масс газового облака, причем эти списки могли быть разными для каждого элемента как по длине (количеству вариантов), так и по их конкретным значениям. Возможна и обратная параллелизация: вначале определяется список масс газового облака, для которых необходимо получить решение, а затем для каждого элемента этого списка — список температур газа.

Следует отметить, что устойчивое функционирование этого графа в смысле разумной постановки задачи и запуска системы специалистами в области вычислительной аэродинамики (выбранными в качестве тест-группы), не связанными с разработкой данного комплекса, обеспечивалось только до уровня $N = 2$, что, по-видимому, и должно быть реализовано в системе, ориентированной на широкие круги пользователей, поскольку использование более глубоких уровней вызывало существенные затруднения. Однако собственно идеология и наработки, полученные в ходе создания и опытной эксплуатации распараллеливания по глобальным параметрам с использованием графа многоуровневого ветвления, могут быть полезны при создании параллельных вычислительных систем в других областях знания.

Другой трудной проблемой этого типа параллелизации была проблема организации потоков выходной информации. Даже отдельная задача в прикладном аспекте представляет большой файл данных, требующих структуризации, например “заголовок” (список метаданных, т.е. параметров решенной задачи, полностью ее характеризующий и позволяющий однозначно ее идентифицировать и отличить от других; это особо важно при проведении не одиночных расчетов, а циклов для комплексного изучения какой-либо научной или прикладной проблемы), “том” (графика или таблицы), “глава” (таблицы различных физических величин), “страница” (часть “главы”, представляющая физическую функцию в какой-либо подобласти параметров) и т.д., вплоть до “строки” или “колонки” (см. любые таблицы по газовой динамике). Это структурирование должно обеспечивать в дальнейшем достаточно простой доступ к полученной информации и возможность сравнения данных из разных “томов”. В принципе здесь может быть использована какая-нибудь уже отработанная СУБД типа Oracle, Access или Paradox, однако этот вопрос еще требует своего решения. В представляемой разработке использовалась специальная система маркировки выходной информации, применяемой при записи файлов во временный банк данных проводимого расчета, содержащая главный и вспомогательный ключи. Главный ключ — индивидуальный (неповторяющийся) идентификатор проводимого расчета составлялся из семи символов: одной буквы и шести цифр. Буква A, S, C (Astra, System of planets, Cloud in galaxy) являлись краткими наименованиями баз данных, в которых размещались результаты расчетов; первые три цифры показывали виртуальный номер процессора, на котором проводился расчет варианта, а последние три цифры обеспечивали сквозную нумерацию варианта в A-, S- или C-базе данных, составляющей интегрированный банк данных. При такой структуре имен имеют место некоторые ограничения: максимально используемое число процессоров не должно превышать 999, число записей в одном томе базы данных также не должно превышать 999. При работе с базами имена файлов (каналов передачи информации) составлялись из восьми символов, первый из которых — буквы W, R, U или P — означали тип потока данных (Write — запись, Read — чтение, Universal — запись и чтение, Print — файл печати данных), остальные семь совпадали с символами главного ключа. Так, например, символ WA001004 означает, что канал предназначен для записи в базу данных результатов расчета эволюции звезды, полученных на процессоре с виртуальным номером 001, и эта запись размещается в БД под номером 004. Главный ключ дополняется вспомогательными ключами — метаданными проведенного расчета (главными входными параметрами, определяющими физический смысл задачи). Главный ключ однозначно идентифицирует запись в БД, в отличие от вспомогательных ключей, поскольку в БД, естественно, может содержаться несколько записей с одним и тем же значением, например T.

Вообще говоря, эта система, далекая от завершения, хотя и позволила определенным образом более или менее осознанно управлять потоками информации и выполнять селекцию получаемых выходных данных различных процессов (а их может быть значительное количество), оказалась громоздкой и не вполне удобной при эксплуатации, требовала внимания администратора системы и высокой дисциплины вычислителей, поскольку в противном случае в БД появлялись записи с одинаковыми именами. Это вызывало необходимость создания временных БД, их просмотра администратором и только затем переноса записей в БД долгосрочного хранения.

В целом проблеме фиксации информации, получаемой на мультипроцессорных вычислительных системах при решении прикладных задач, в настоящее время практически не уделяется должного внимания. Особенно это касается решения задач в режиме удаленного доступа с необходимостью пересылки по Сети весьма больших по объему файлов. Аналогичными являются и проблемы визуализации (графического представления) решений при интегрировании данных, полученных на различных сегментах процессорного пространства, однако этот вопрос слишком объемен и лежит вне рамок настоящей работы.

8. Заключение. В работе проведен многосторонний анализ фундаментальных проблем вычислительного инструментария параллельного счета для моделирования пространственных нестационарных задач космической (гравитационной) газовой динамики. Рассмотрены различные способы распараллеливания, которые определяются особыми видами декомпозиции полной задачи на ряд одновременно выполняемых подзадач. Исследовано четыре существенно различных типа параллелизации: глобальное распараллеливание программного комплекса по главным входным параметрам; крупноблочное распараллеливание по физическим процессам; геометрическое крупномасштабное сегментирование; мелкозернистое распараллеливание процедур основного итерационного ядра системы.

Предназначенное для локального ускорения операций мелкозернистое распараллеливание дополнительно обеспечивает важную функцию динамической балансировки загрузки использующихся элементов мультипроцессорной системы вследствие существенно разной скорости операций в различных крупноблочных (физических или геометрических) сегментах и выравнивания времени исполнения этих операций. Применение такого комплексного подхода означает необходимость дополнения функций иницирующей системы вычислительного комплекса принципиально новым механизмом организации автоматической, *без участия программиста*, связи между *цифровыми значениями* главных входных параметров, определяющих физический смысл конкретной решаемой задачи, и *числом машин* в каждом отдельном процессорном блоке с учетом общего выделенного ресурса на всю задачу, причем соотношение между ними должно быть *“плавающим”*, *меняющимся* от задачи к задаче без кардинального переустройства процедуры распределения ресурса.

Модернизация и кардинальные изменения программного комплекса [13] с использованием новых компьютерных технологий параллельного счета и создание на его основе вычислительного комплекса третьего поколения позволит существенно продвинуться в область уже достаточно физически и математически проработанных (и, тем более, за ее пределы), но пока технически недоступных задач, представляющих исключительный интерес в данной области знания.

В заключение авторы считают своим приятным долгом выразить признательность Г. С. Хакимзянову и С. Н. Коробейникову за полезные обсуждения и постоянное внимание к работе, В. Д. Кургузову и В. В. Пикалову за помощь в оформлении статьи.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (код проекта 04-07-90002).

СПИСОК ЛИТЕРАТУРЫ

1. Воеводин В. В., Воеводин В. В. Параллельные вычисления. СПб: БХВ-Петербург, 2002.
2. Гинзбург В. Л. Какие проблемы физики и астрофизики представляются сейчас особенно важными и интересными? // Успехи физ. наук. 1999. **169**, № 4. 419–441.
3. Пармон В. Н. Пребиотическая фаза зарождения жизни // Вестник РАН. 2002. **72**, № 11. 976–983.
4. Контримавичюс В. Л. Истоки учения о ноосфере // Вестник РАН. 2003. **73**, № 11. 1002–1009.
5. Тарнавский Г. А., Шпак С. И. Декомпозиция методов и распараллеливание алгоритмов решения задач аэродинамики и физической газовой динамики: вычислительная система “ПОТОК-3” // Программирование. 2000. № 6. 45–57.
6. Тарнавский Г. А., Шпак С. И. Схемы распараллеливания операций решения систем алгебраических уравнений методом многомерной скалярной прогонки // Вычисл. методы и программирование. 2000. **1**, № 1. 21–29.
7. Вшивков В. А., Тарнавский Г. А., Неупокоев Е. В. Параллелизация алгоритмов прогонки: многоцелевые вычислительные эксперименты // Автометрия. 2002. **38**, № 4. 74–86.
8. Тарнавский Г. А., Тарнавский А. Г. Современные компьютерные технологии и неединственность решений задач газовой динамики // Симметрия и дифференциальные уравнения / Под ред. В. К. Андреева. Красноярск: ИВТ СО РАН, 2002. 209–213.
9. Тарнавский Г. А., Вшивков В. А., Тарнавский А. Г. Параллелизация алгоритмов и кодов вычислительной системы “Поток-3” // Программирование. 2003. № 1. 24–44.
10. Тарнавский Г. А., Корнеев В. Д. Распараллеливание программного комплекса математического моделирования высокоскоростных течений реального газа // Автометрия. 2003. **39**, № 3. 72–83.
11. Тарнавский Г. А., Корнеев В. Д., Вайнер Д. А., Покрышкина Н. М., Слюняев А. Ю., Танасейчук А. В., Тарнав-

- ский А.Г. Вычислительная система “Поток-3”: опыт параллелизации программного комплекса. Часть I. Идеология распараллеливания // Вычисл. методы и программирование. 2003. 4, № 1. 37–48.
12. Малышкин В.Э., Цыгулин А.А. ParaGen — генератор параллельных программ, реализующих численные модели // Автометрия. 2003. 39, № 3. 124–135.
 13. Тарнавский Г.А., Аульченко С.М., Вишнев В.А. Математическое моделирование нестационарных трехмерных процессов в космической газодинамике // Вычисл. методы и программирование. 2003. 4, № 2. 127–155.
 14. Тарнавский Г.А., Шпак С.И. Эффективный показатель адиабаты в задачах гиперзвукового обтекания тел реальным газом // Теплофизика и аэромеханика. 2001. 8, № 1. 41–58.
 15. Тарнавский Г.А., Шпак С.И. Способы расчета эффективного показателя адиабаты при компьютерном моделировании гиперзвуковых течений // Сиб. ж. индустриальной математики. 2001. 4, № 1(7). 177–197.
 16. Ландау Л.Д., Лившиц Е.М. Теория поля. М.: Наука, 1967.
 17. Блинные С.И., Высоцкий М.И., Окунь Л.Б. Скорости $c/\sqrt{3}$ и $c/\sqrt{2}$ в общей теории относительности // Успехи физ. наук. 2003. 173, № 10. 1131–1136.
 18. Белоцерковский О.М. Математическое моделирование на суперкомпьютерах (опыт и тенденции) // Ж. вычисл. матем. и мат. физ. 2000. 40, № 8. 1221–1236.
 19. Rognlien T.D., Xu X.Q., Hindmarsh A.C. Application of parallel implicit methods to edge-plasma numerical simulations // J. Comput. Phys. 2002. 175, N 1. 249–268.
 20. Marcos C., Barge P., Marcos R. Dust dynamics in protoplanetary disks: parallel computing with PVM // J. Comput. Phys. 2002. 176, N 1. 274–294.

Поступила в редакцию
03.02.2005
