

О технологии параллельного компьютерного моделирования на многопроцессорных вычислительных комплексах кластерного типа

Г.Г. ШВАЧИЧ, А.А. ШМУКИН

Национальная металлургическая академия Украины

Предложен подход к разработке современных вычислительных математических технологий нахождения решения многомерных нестационарных задач. Процесс моделирования реализован на основе применения многопроцессорных вычислительных систем кластерного типа. Освещаются вопросы конструирования вычислительно кластера.

Запропоновано підхід до розробки сучасних обчислювальних математичних технологій знаходження розв'язку багатовимірних нестационарних задач. Процес моделювання реалізовано на основі застосування багатопроцесорних обчислювальних систем кластерного типу. Висвітлюються питання конструювання обчислювально кластера.

Approach is offered to development of modern calculable mathematical technologies of finding and decision multi-dimensional non stationary tasks. The process of design is realized on basis application of the multiprocessor computer systems of cluster type. The questions of constructing are lighted calculable cluster.

Актуальность темы исследований

В настоящее время машинное моделирование является одним из наиболее распространенных и мощных методов исследования сложных систем. При этом нынешний этап научных исследований отличается тем, что имеет место огромная вычислительная мощность современных средств вычислительной техники, при помощи которых проводится соответствующий этап математического моделирования. Однако существует широкий класс задач, решение которых при помощи классического (последовательное) моделирования (особенность его состоит в том, что реализуется на однопроцессорном компьютере) занимает неприемлемо долгое время (недели и месяцы). К таким задачам относятся, например, численное моделирование процессов гидродинамики и металлургической теплофизики [1,2], задачи распознавания образов, оптимизационные задачи с большим числом параметров и т.д. Например, в [3] освещаются вопросы моделирования с помощью метода конечных элементов процесса волочения проволоки в роликовых волоках. Задачу моделирования авторы осуществляли на основе использования коммерческого программного обеспечения FORGE3 фирмы Transvalor (Франция). На компьютере с процессором Celeron 2.4 задача решалась около 12 часов.

Время решения подобных задач можно существенно сократить, если для моделирования использовать многопроцессорные ЭВМ.

Для того чтобы в полной мере использовать преимущества, предоставляемые такими ЭВМ, необходимо решить следующие задачи:

- сконструировать алгоритмы решения задач с учётом возможностей параллельной обработки данных несколькими процессорами одновременно;

- реализовать процесс вычислений таким образом, чтобы каждый процессор использовался наиболее полно, и при этом суммарное время решения задачи стремилось к минимуму.

Проблемы, возникающие при разработке параллельных вычислительных систем, отвечающих заданным характеристикам, как правило, являются первостепенными и требуют глубокого изучения и исследования. Действительно, распределенное (параллельное) компьютерное моделирование охватывает весь спектр современной вы-

числительной техники: суперкомпьютеры, кластерные вычислительные системы, локальные и глобальные сети. Кроме того, распределенное моделирование позволяет решать задачи, требующие большого количества процессорного времени, интегрировать математические модели, которые обрабатываются на различных (в том числе и географически отдаленных) вычислительных системах.

В настоящее время в мире наблюдается стремительный рост числа многопроцессорных вычислительных систем кластерного типа и их суммарной производительности. Одновременно растет потребность в имитационных моделях сложных систем, требующих большого количества вычислительных ресурсов. Однако широкому внедрению машинного моделирования для многопроцессорных вычислительных систем препятствует отсутствие или недоступность систем распределенного моделирования. В этой связи, проблемы конструирования вычислительных кластеров, а также разработки вычислительных алгоритмов для параллельного процессора являются актуальными и первостепенными.

В данной статье освещаются вопросы конструирования параллельного вычислительного кластера, а также математического моделирования широкого класса задач на параллельных вычислительных системах *кластерного типа*.

Анализ последних достижений и современное состояние проблемы исследований

Впервые в классификации вычислительных систем термин «кластер» ввела компания Digital Equipment Corporation (DEC). Определение кластера было сформулировано следующим образом: кластер – это группа вычислительных машин, которые связаны между собой и функционируют, как один узел обработки информации. Существуют и другие определения «кластера» [4,5], которые связаны с особенностями сегодняшнего развития средств информационных технологий, но они несут, скорее всего, производный характер от определения компании DEC. В данной статье, используя термин «кластер», мы будем подразумевать как определение компании DEC, так и определения, сформулированные в [4,5].

В настоящее время развитие высокопроизводительных вычислительных систем развивается по двум основным направлениям:

1. Параллельные компьютеры с общей памятью (архитектура SMP).

Вычислительные кластеры архитектуры SMP – Symmetric Microprocessors отличаются тем, что их вся оперативная память разделена несколькими одинаковыми процессорами. Это обстоятельство снимает проблемы вычислительных кластеров предыдущего класса, но добавляет новые – число процессоров, имеющих доступ к общей памяти, по техническим причинам нельзя сделать большим, при этом отмечают, что здесь имеет место проблема масштабирования [4,6].

2. Массивно-параллельные компьютеры с распределенной памятью (архитектура MPP).

Идея построения вычислительных кластеров архитектуры MPP – Massively Parallel Processing состоит в следующем: берутся серийные микропроцессоры, снабженные каждый своей локальной памятью, и связываются быстродействующими каналами связи.

У такой архитектуры достоинств много: при необходимости можно добавлять процессоры, увеличивая производительность такого кластера; если ограничены финансовые возможности или заранее известна требуемая вычислительная мощность, то легко подбирать требуемую конфигурацию кластера. Однако здесь есть и один существенный недостаток. Дело в том, что межпроцессорное взаимодействие в процессорах такой архитектуры идет намного медленнее, чем происходит локальная обработка данных самими процессорами, т.е. здесь отмечают, что в ряде случаев может возникать проблема так называемой «медленной» памяти. К данному классу можно отнести компьютеры Intel Paragon, IBM SP1, Parsytec и др. Заметим, что к этому классу относят и сети компьютеров, которые все чаще рассматривают как недорогую альтернативу крайне дорогих суперкомпьютеров.

Эффективность суперкомпьютерных установок определяется LINPAK – производительностью (LINPAK – тест, основанный на решении систем линейных алгебраических уравнений большой размерности), которая измеряется в GFlops (GFlops = 10^9 «плавающих» операций/с). Заметим, что один из наиболее мощных суперкомпьютеров установлен в Ливерморском национальном исследовательском центре (<http://lnl.org>) включает 8192 процессора с пиковой производительностью 12×10^{12} операций/с. С 1993 года ведется список TOP500 (<http://top500.org>) самых производительных суперкомпьютеров в мире. В 25-й редакции такого списка отмечается, что все, находящиеся в верхних строках списка, вычислительные системы реализуют параллельную архитектуру. Кроме того, 96% всех представленных в этом списке суперкомпьютеров обладают LINPAK – производительностью 1-10 тыс. GFlops. Графическое представление производительности суперкомпьютеров можно интерпретировать узкой областью прорыва на фоне обширного фундамента поисковых работ. Такой характер распределения соответствует наиболее динамично развивающейся отрасли науки и техники.

Структуру мощностей кластерных вычислительных систем обычно интерпретирует тремя уровнями (табл. 1).

В данной статье освещается вычислительный кластер третьего уровня (табл. 1) MPP архитектуры и особенности его применения для решения широкого класса многомерных нестационарных задач. Кроме того,

авторы предлагают свой подход к устранению недостатков вычислительных кластеров MPP архитектуры.

Таблица 1.

Уровень решаемых задач	Область применения	Пиковая производительность	Примерная стоимость
Суперкомпьютеры планетарного масштаба	Решение вычислительных проблем глобального уровня	Десятки-сотни тысяч GFlops	Десятки – сотни млн. долларов US
Профессиональные кластеры	Решение проблем государственного уровня	Десятки – тысячи GFlops	Десятки – сотни тыс. долларов US
Персональные вычислительные кластеры	Решение проблемно-ориентированных задач, разработка библиотек	Единицы GFlops	Уровень тысячи долларов US

Особенности конструирования вычислительного кластера

Конструктивно вычислительный кластер представляет собой вычислительную систему, построенную из стандартных вычислительных узлов, объединенных быстродействующей низколатентной (малоинерционной) компьютерной сетью. Заметим, что при освещении вопросов конструирования и применения вычислительных кластеров часто рассматривают понятие *латентности*. Под латентностью здесь понимают время самого простого взаимодействия узлов кластера через коммуникационную среду. Эти понятия вводятся для оптимизации сетевых взаимодействий [6].

Практическая реализация вычислительного кластера предполагает наличие главного (управляющего, MASTER) узла и некоторых подчиненных (SLAVE) вычислительных узлов. Как правило, компиляция и сборка исполняемых кодов программ осуществляется MASTER-машиной, иницирующей соответствующие вычислительные процессы и рассылающей исполняемые коды программ по SLAVE-машинам. Данные MASTER-машиной рассылаются путем применения какого-либо из интерфейсов: MPI (Message Passing Interface) или PVM (Parallel Virtual Machine).

На рис. 1 представлена структурная схема разработанного авторами вычислительного кластера. Кластер собран из 5 системных блоков идентичной конфигурации: Sempron 2000, 256 Mb RAM DDR 400, HDD Samsung 40 Gb, Mb KT-600 ECS, Radeon 9200, Codegen 300 W, net card Realtek 8139. Системные блоки подключены в сеть по технологии Ethernet 100BaseT через сетевой коммутатор

(switch), физический тип соединения – «звезда». Кластер работает под управлением ОС Linux. Связь между узлами кластера осуществляется посредством протоколов TCP/IP с использованием технологии MPI.

В зависимости от составленного алгоритма вычислений процесс работы такого кластера может развиваться по одному из направлений:

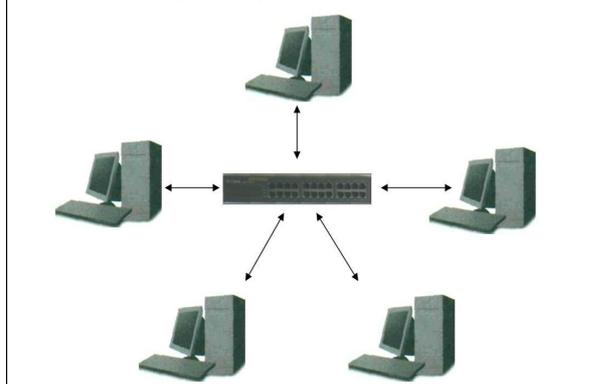


Рис. 1. Структурная схема вычислительного кластера

- на всех компьютерах вычислительного кластера запускается одна также программа;

- MASTER-машина готовит данные, необходимые SLAVE-машинам и рассылает их в виде MPI - сообщений, а затем собирает с них обработанную информацию для интерпретации.

Заметим, что внутри общей программы MASTER-машина и SLAVE-машины идентифицируют себя посредством специальной переменной – идентификатора.

Авторы статьи убедились, что современные технологии позволяют создать вычислительный кластер при минимальном навыке выполнения хозяйственных работ, а также принимая во внимание присущий большинству молодых исследователей уровень знаний системного и пользовательского ПО.

Вычислительный кластер, структурная схема которого представлена на рис. 1, является инструментальной средой, предназначенной для приобретения навыков управления и программирования многопроцессорных систем. Заметим, что вследствие применения Linux, стандартных систем программирования и управления ресурсами, аналогичным профессиональных кластерных систем, процесс переноса наработанного программного обеспечения на большие кластерные системы максимально прост и удобен. Кроме того, мощные вычислительные системы не отвлекаются на выполнение рутинных (отладочных операций).

Хотя заметим, что системное программное обеспечение вычислительного кластера должно давать возможность работы как в режиме Windows (подготовка текстового материала, обмен сообщений по E-Mail, привычный режим работы в INTERNET и т.д.) и одновременно управлять работой вычислительного кластера в среде ОС Linux. Проблемы функционирования Windows – кластера, особенности совместной работы Linux – кластера и операционной системы Windows авторы надеются осветить в ближайших публикациях.

Заметим, что компьютерные вычислительные кластеры дополнительно стимулировали развитие новой области знаний – технологии параллельных вычислений

(ТПВ), основные особенности которой для рассматриваемого класса задач освещаются в следующем разделе статьи.

Некоторые принципы конструирования параллельных алгоритмов

Очевидно, что при использовании многопроцессорных систем камнем преткновения становится проблема так называемого «распараллеливания алгоритмов» для программирования на многопроцессорной структуре. Последовательные алгоритмы, используемые при работе на однопроцессорных компьютерах, как правило, не приспособлены для такого распараллеливания. Это объясняется тем, что при практическом применении технологии параллельных вычислений возникают проблемы приобретения опыта в разработке параллельных алгоритмов и программ, их отладке, управления заданиями при работе кластера, интерпретации и представления результатов работы кластерной системы и т.д.

Вообще отметим, переход от парадигмы последовательной алгоритмизации к конструированию алгоритмов параллельной обработки данных требует определенного переосмысления технологии программирования. В этой связи вопросы алгоритмизации ТПВ на сегодняшний день, по мнению авторов, являются наиболее важными и актуальными. Именно по этой причине курсы лекций по технологии параллельных вычислений представлены во многих технических высших учебных заведениях. Однако, к сожалению, на сегодняшний день можно наблюдать чисто теоретический (а порой и схоластический) подход к овладению знаний по параллельной обработке данных. Число часов для овладения практических навыков технологии параллельного программирования выделяется порой не достаточно.

Основные принципы конструирования алгоритмов ТПВ для рассматриваемого класса задач освещены в [5, 7-11]. Рассмотрим некоторые принципиальные особенности конструирования алгоритмов для параллельного процессора.

При этом проиллюстрируем основные алгоритмические идеи формирования параллельного процессора на фоне решения следующей задачи.

Рассмотрим решение краевой задачи Дирихле для одномерного уравнения теплопроводности

$$\frac{\partial Y}{\partial t} = \alpha \frac{\partial^2 Y}{\partial x^2}, \quad x \in [x_0, x_L], \quad t \in [t_0, T] \quad (1)$$

с начальным

$$Y|_{t=t_0} = \varphi(x) \quad (2)$$

и граничными условиями

$$Y|_{x=x_0} = YW(t), \quad Y|_{x=x_L} = YL(t). \quad (3)$$

Уравнение (1) при помощи неявной схемы по времени и центральные разности по координате x сводится к СЛАУ

$$C_p Y_{p+1,1} - Y_{p,1} + D_p Y_{p-1,1} = f_{p,1}, \quad (5)$$

$$p = \overline{1, 2m-1},$$

где

$$\left. \begin{aligned} C_p = D_p = \frac{A}{(1+2A)}, \quad A = \frac{\alpha}{Dx1^2} Dt1 \\ f_{p,1} = -\frac{YOp,1}{(1+2A)} \end{aligned} \right\} \quad (6)$$

Для численного моделирования уравнения (5) на параллельном процессоре можно использовать несколько подходов. Рассмотрим их.

Распараллеливание СЛАУ (5) при помощи перестановок [7]. В основу метода распараллеливания одной и той же математической задачи – системы линейных алгебраических уравнений в форме (5) с функциональным наполнением (6) полагается алгоритм «нечетно-четной» редукции [7]. Идея этого метода состоит в исключении некоторых коэффициентов системы уравнений (5) при помощи элементарных преобразований строк.

Множеству операций алгоритма «нечетно-четной» редукции СЛАУ (5) к параллельной форме поставим во взаимно однозначное соответствие некоторое множество точек, зависящее от четности сеточных узлов на множестве $p = 1, 2m - 1$. На первом этапе редукции, понизив и повысив в СЛАУ (5) индекс p на единицу, найдем:

$$\left. \begin{aligned} Y_{p-1,1} &= -f_{p-1,1} + C_{p-1}Y_{p,1} + D_{p-1}Y_{p-2,1} \\ Y_{p+1,1} &= -f_{p+1,1} + C_{p+1}Y_{p+2,1} + D_{p+1}Y_{p,1} \end{aligned} \right\} (7)$$

Соотношение (7) является достаточно удобным для реализации вычислений на параллельном процессоре. После замещения в (5) крайних слагаемых по формулам (7) получают уравнение трехдиагональной структуры относительно $\{Y_{2,1}, Y_{4,1}, \dots, Y_{2m-2,1}\}$. Повторное применение этого процесса редукции приводит к уравнению относительно $\{Y_{4,1}, Y_{8,1}, \dots, Y_{2m-4,1}\}$ и т.д. до тех пор, пока не приходят к одному уравнению относительно элемента $Y_{m,1}$, соответствующего центральному узлу. При этом необходимо принять число узлов $m = 2^k$, ($k \in Z$), где k является степенью двойки. Применение изложенного алгоритма для решения трехдиагональных систем на параллельных вычислительных процессорах весьма перспективно [7].

Распараллеливание СЛАУ (5) численно-аналитическим методом прямых [8]. Рассмотрим вариант распараллеливания СЛАУ (5) для случая, когда ее функциональное наполнение представлено в виде гиперболических функций, которые являются линейными комбинациями базисного решения двухточечной краевой задачи. Для реализации распараллеливания СЛАУ (5) предлагается использовать метод прогонок.

Известно, что в методе прогонок процедура прямой прогонки реализуется рекуррентно формированием двух последовательностей E_p, G_p , ($p = 1, 2m - 1$) по формулам:

$$E_p = \frac{C_p}{1 - D_p E_{p-1}}, \quad G_p = \frac{D_p G_{p-1} - f_{p,1}}{1 - D_p E_{p-1}}, \quad (8)$$

где старт для данной задачи обеспечивается входными параметрами

$$E_0 = 0, \quad G_0 = Y_{0,1} = fW(t_j). \quad (9)$$

После завершения процедуры прямой прогонки в направлении возрастания индекса p вплоть до $p=2m-1$, процедура обратной прогонки выполняется по рекуррентной формуле:

$$Y_{p,1} = E_p Y_{p+1,1} + G_p \quad (10)$$

в направлении убывания индекса p от $p=2m-1$ до $p=1$. Старт этой процедуры обеспечивается выполнением условия $Y_{2m,1} = fL(t_j)$, реализующего ввод в алгоритм

прогонок правого граничного условия (3).

Вычисление $\{E_p, G_p\}$ при таком подходе реализуется в явном виде. Процедура обратной прогонки по формуле (10) позволяет найти решение в замкнутой форме относительно любого узла.

Соотношение (10) и положено в основу разработки алгоритма распараллеливания. Такой алгоритм абсолютно устойчив для любых входных данных, имеет максимальную параллельную форму и, следовательно, минимально возможное время его реализации на параллельных вычислительных устройствах.

Если можно назначить один процессор на один узел расчета, то становится возможным проведение расчетов во всех узлах сеточной области параллельно и одновременно.

О проблеме латентности вычислительных кластеров MPP архитектуры

В данной статье авторы освещают опыт работы с вычислительным кластером MPP архитектуры. Такая архитектура обладает рядом бесспорных преимуществ. Однако отмечают один ее недостаток, который объясняется проблемой латентности. При этом заметим, что каждый процессор вычислительного кластера работает в своем адресном пространстве и никаких общих переменных с другими процессорами не имеет. В тоже время основным способом взаимодействия между процессорами кластера является посылка сообщений при помощи коммуникационной технологии MPI. Такое взаимодействие осуществляется в MPI посредством явной пересылки сообщений. Вполне очевидно, что межпроцессорное взаимодействие идет намного медленнее, чем происходит локальная обработка данных самими процессорами. Например, если на каком-либо процессоре потребовались данные, расположенные в памяти другого процессора, то на передачу этих данных необходимо время, как правило, значительно превосходящее время обращения к своей локальной памяти.

Вообще заметим, что кроме проблемы латентности здесь могут возникать *тупиковые ситуации*. Суть которых состоит в следующем. Предположим, два процессора вычислительного кластера должны обменяться данными. Тогда для гарантии корректного использования буфера применяется схема, при которой посылающий процесс ждет начала приема. И здесь может возникнуть классический тупик. Идея которого состоит в следующем. Первый процесс не может вернуться из процедуры отправки, поскольку второй не начал приема сообщения. В тоже время второй процесс не может начать прием сообщения, поскольку сам по схожей причине застрял на соответствующей отсылке.

Принимая во внимание указанное, можно отметить, что основное интуитивно понятное правило получения эффективных программ вычислительного кластера MPP архитектуры состоит в том, чтобы выделить побольше примерно одинаковых по вычислительной сложности независимых фрагментов, которые поменьше обменивались данными во время работы кластера по соответствующей программе. С точки зрения анализа и преобразования структуры программ это правило можно разделить на две задачи:

- выявление потенциального параллелизма фрагмента кода, выполняемого на некотором процессоре;

- нахождение множества возможных распределений по процессорам, согласованных с найденным ресурсом параллелизма.

Вторая задача предполагает поиск таких способов распределения данных по процессорам вычислительной системы, при которых обмены данными между процессорами были бы минимальными.

В [12] предложено несколько способов разрешения тупиковых ситуаций, основанных на применении соответствующим образом процедур (MPI_SEND, MPI_RECV, MPI_IRECV, MPI_SENDRECV и др.) коммуникационной технологии MPI. Авторы статьи предлагают свой подход к решению проблемы латентности и разрешения тупиковых ситуаций. Идея такого подхода состоит в решении указанных проблем алгоритмически, принимая во внимание то обстоятельство, что самая благоприятная ситуация возникает в том случае, когда удастся распределить процесс вычислений по пользовательской программе так, что обмены данными между процессорами отсутствуют вовсе. Проиллюстрируем указанный подход для исследуемого класса задач.

При решении нестационарных задач рассматриваемого класса задач явными или неявными методами почти всегда расчеты ведутся по временным слоям, последовательно слой за слоем. Поэтому счет по времени особенно эффективен на параллельных вычислительных системах с общей памятью. Как показали исследования, проведенные в [4], возможность реализации алгоритма по макровершинам оказывается реальной и для широкого круга алгоритмов и в нашем случае. В частности, эта возможность широко используется при решении задач линейной алгебры блочными методами [13]. На наш взгляд целесообразно использовать аналитические методы математической физики. В частности, применив к (1) – (3) интегральное преобразование Лапласа получим решение этой задачи в конечном виде

$$Y_{1+\varepsilon_{k,1}}(\tau, \varepsilon_x) = \left[1 - \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{(2n-1)} \cos(\gamma_n \varepsilon_x) e^{-\gamma_n^2 \tau} \right] J, \quad (11)$$

где

$$\begin{cases} \varepsilon_x = \frac{x-x_1}{x_2-x_1} \in [-1,1], \\ \tau = \frac{a}{L^2} t \in [0, t_k]. \end{cases} \quad (12)$$

В (11), (12) L – характерный линейный размер неограниченной пластины ($x_1=L/2, x_2=L=2$), а в качестве входных условий (2), (3) было принято

$$\varphi(x) = 0, \quad YW(t) = YL(t) \equiv 1. \quad (13)$$

Математическая модель (11) допускает неограниченное распараллеливание алгоритма, как по временной переменной τ , так и по координате ε_x . Действительно, переход от функции (11) к ее сеточному представлению приводит к задаче отображения аналитической функции конечной совокупностью чисел. Если можно назначить один процессор на один узел такого расчета, то становится возможным проведение расчетов во всех узлах сеточной области параллельно и одновременно, минуя замену дифференциального оператора разностным. Но самое

главное заключается в том, что данные при этом можно распределять так, чтобы не было пересылок вовсе.

Многомерные задачи. Напомним, что при построении неявных разностных схем для численного решения многомерных эволюционных задач используются различные подходы, основанные на представлении разностной схемы в виде последовательности более простых разностных аналогов. Это методы расщепления – одно из важных средств при расчете решений многомерных нестационарных задач математической физики [14].

Рассмотрим дифференциальную задачу вида

$$\begin{cases} \frac{\partial u}{\partial \tau} = Au, \quad 0 < \tau < T, \\ u|_{\tau=0} = \text{задано}, \end{cases} \quad (14)$$

где A – некоторый оператор по пространственным переменным, например:

$$Au \equiv \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right). \quad (15)$$

Тогда правая часть уравнения (14) принимает вид

$$Au = A_1 u + A_2 u. \quad (16)$$

Уравнение (14), представленное в виде

$$\frac{\partial u}{\partial \tau} = A_1 u + A_2 u,$$

расщепляется на следующие два

$$\frac{\partial u}{\partial \tau} = A_1 u + A_2 u, \quad (17)$$

$$\begin{cases} \frac{\partial \vartheta}{\partial \tau} = A_1 \vartheta, \quad \tau_{J-1} \leq \tau \leq \tau_J, \\ \vartheta(x, y, \tau_{J-1}) = u(x, y, \tau_{J-1}), \\ \frac{\partial W}{\partial \tau} = A_2 \vartheta, \quad \tau_{J-1} \leq \tau \leq \tau_J, \\ W(x, y, \tau_{J-1}) = \vartheta(x, y, \tau_{J-1}). \end{cases} \quad (18)$$

Заметим,

$$W(x, y, \tau_J) = u(x, y, \tau_J) + O(D\tau^2), \quad (19)$$

что и дает основание на каждом интервале времени $\tau_{J-1} \leq \tau \leq \tau_J$ вместо задачи (14) последовательно решать задачи (17) и (18).

Для фактического решения уравнений (17) и (18) формально аппроксимируем эти уравнения какими-либо разностными. Тогда возникает разностная схема расщепления, подробно освещенная в [15] и получившая название локально-одномерного расщепления. Как показано выше, если же при решении задач (17), (18) воспользоваться аналитическими методами, то получим *аналитическое расщепление*. При аналитическом расщеплении возможно решение задачи аналитическими методами, минуя разностные аппроксимации одномерных операторов Лапласа и временных операторов. В отличие от разностных методов расщепления аналитические схемы позволяют сначала по известным значениям

$$u|_{\tau=0} = \varphi(x, y) = 0, \quad (20)$$

вычислить вспомогательные функции (17) $\vartheta(x, y, \tau_J)$ одновременно на пакете временных слоев

$\{J=1,2,3,\dots,J_n\}$, а потом из аналитического решения задачи (18) одновременно вычислить $u(x,y,\tau_J)$ для всех J одновременно, исключая пересылку данных.

Пример. Проиллюстрируем алгоритм на примере решения задачи о температурном поле для пластины заданного размера ($L=2\tau$) при $\varphi(x,y)=0$, имеющего в стационарном режиме точное аналитическое решение (ф. 6.18, стр. 162, [17]) при следующих граничных условиях первого рода

$$\begin{cases} u|_{y=0} = u|_{y=l} \equiv 1, \\ u|_{x=0} = u|_{x=2l} \equiv 0, \end{cases} \quad (21)$$

где $l=1$ – характерный линейный параметр.

Построение алгоритма начнем с решения задачи (18). При этом для такой задачи точное аналитическое решение в поле изображений имеет следующий вид

$$\bar{W}_{1+\varepsilon_y,1}(x,\varepsilon_y,S) = \frac{1}{S} \frac{ch(\sqrt{\frac{S}{A_y}} \cdot \varepsilon_y)}{ch(\sqrt{\frac{S}{A_y}})}, \quad (22)$$

где

$$\begin{cases} \varepsilon_y = \frac{y-y_1}{y_2-y_1} \in [-1,1], y_1=0.5, y_2=1, \\ A_y = \frac{D\tau 1_y}{(y_2-y_1)^2}, \quad D\tau 1_J = \tau_J - \tau_0. \end{cases} \quad (23)$$

Функция (22) является мероморфной и ее особыми точками являются:

1. $S=0$ – простой полюс,
2. $\sqrt{\frac{S_n}{A_y}} = i\gamma_n, \quad \gamma_n = \frac{\pi}{2}(2n-1)$ – бесконечное

множество простых полюсов, корней уравнения

$$\cos \gamma_n = 0, \quad n = 1,2,3,\dots \quad (24)$$

Воспользовавшись табличными функциями оригинал изображения (22) представляется зэта-функцией [16]

$$\begin{aligned} W_{1+\varepsilon_y,1}(x,\varepsilon_y,\varepsilon_t) = \\ = [1 - \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{(2n-1)} \cos(\gamma_n \varepsilon_y) e^{-\gamma_n^2 A_y \varepsilon_t}], \end{aligned} \quad (25)$$

где индекс 1 обозначает срединную плоскость по координате Y .

Для задачи (17) решение конструируется на основе решения уравнения второго порядка

$$\begin{aligned} \bar{g}_{1+\varepsilon_x,1}''(\varepsilon_x, Y, S) - \frac{S}{A_y} \bar{g}_{1+\varepsilon_x,1}'(\varepsilon_x, Y, S) = \\ = -\frac{1}{A_y} g_{0_{1+\varepsilon_x,1}}(\varepsilon_x) \end{aligned} \quad (26)$$

где: $g_{0_{1+\varepsilon_x,1}}(\varepsilon_x)$ – известная начальная функция,

$$A_y = \frac{D\tau 1}{Dx 1^2}, \quad Dx 1 = 1 \text{ - срединная поверхность области.}$$

Решение представляется в следующем виде

$$\begin{aligned} \bar{g}_{1+\varepsilon_x,1}(\varepsilon_x, Y, S) = \\ = \bar{\mu}_{1+\varepsilon_x,1}(\varepsilon_x, S) + C_1 e^{\beta x \varepsilon_x} + D_1 e^{-\beta x \varepsilon_x}, \end{aligned} \quad (27)$$

где: $\bar{\mu}_{1+\varepsilon_x,1}(\varepsilon_x, S)$ – частное решение неоднородного уравнения (26), C_1, D_1 – константы интегрирования, определяемые из условий (21)

$$g_{1+\varepsilon_x,1}|_{\varepsilon_x=\pm 1} \equiv 0; \quad (28)$$

$$\beta = \sqrt{\frac{S_n}{A_y}} \text{ - корни характеристического уравнения}$$

$$\beta^2 - \frac{S}{A_x} = 0. \quad (29)$$

Допустим, что функция $g_{0_{1+\varepsilon_x,1}}(\varepsilon_x)$ в (26)

имеет квадратичную по ε_x зависимость

$$g_{0_{1+\varepsilon_x,1}}(\varepsilon_x) = g_{0_{1,1}} + \varepsilon_x g_{0_{1,2}} + \varepsilon_x^2 g_{0_{1,3}}, \quad (30)$$

где

$$\begin{cases} g_{0_{1,2}} = \frac{1}{2}(g_{0_{2,1}} - g_{0_{0,1}}), \\ g_{0_{1,3}} = \frac{1}{2}(g_{0_{2,1}} + g_{0_{0,1}} - 2g_{0_{1,1}}), \end{cases} \quad (31)$$

центрированные разности первого и второго порядков.

Легко убедиться, что функция

$$\bar{\mu}_{1+\varepsilon_x,1}(\varepsilon_x, S) = \frac{1}{S} g_{0_{1+\varepsilon_x,1}}(\varepsilon_x) + 2 \frac{A_x}{S^2} g_{0_{1,3}} \quad (32)$$

удовлетворяет уравнению (26) и, следовательно, является его частным решением. Тогда с учетом (28) найдем

$$\begin{aligned} \bar{g}_{1+\varepsilon_x,1}(\varepsilon_x, Y, S) = \\ = \frac{g_{0_{1,1}}}{S} \left\{ (1 - \varepsilon_x^2) + 2 \frac{A_x}{S ch(\sqrt{\frac{S}{A_x}})} [ch(\sqrt{\frac{S}{A_x}} \varepsilon_x) - \right. \\ \left. - ch(\sqrt{\frac{S}{A_x}})] \right\} \end{aligned} \quad (33)$$

так как $g_{0_{0,1}} = g_{0_{2,1}} \equiv 0$.

Функция (33) мероморфна и имеет особые точки:

1. $S=0$ – простой полюс,
2. $\sqrt{\frac{S_n}{A_x}} = i\gamma_n, \quad \gamma_n = \frac{\pi}{2}(2n-1)$ – бесконечное

множество простых полюсов, корней уравнения (24).

Заметим, что второе слагаемое в (33) при $S=0$ имеет устранимую особую точку. Следовательно, оригиналом изображения (33) будет следующая зэта-функция

$$\begin{aligned} g_{1+\varepsilon_y,1}(\varepsilon_x, Y, \varepsilon_t) = \\ = g_{0_{1,1}} \left\{ \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{(2n-1)} \Psi_n(\varepsilon_x) e^{-\gamma_n^2 A_x \varepsilon_t} \right\}, \end{aligned} \quad (34)$$

где

$$\left\{ \begin{aligned} \Psi_n(\varepsilon_x) &= \left\{ \sum_{k=0}^{\infty} (-1)^k \frac{[\frac{\gamma_n}{2}(1+\varepsilon_x)]^{2k}}{(2k+1)!} \sum_{k=0}^{\infty} (-1)^k \frac{[\frac{\gamma_n}{2}(1+\varepsilon_x)]^{2k}}{(2k+1)!} \right\} \\ \gamma_n &= \frac{\pi}{2}(2n-1), \quad A_x = \frac{D\tau l_y}{Dx^2}, \quad \varepsilon_\tau = \frac{\tau_J - \tau_0}{D\tau l_y} \in [0,1], \\ \mathcal{G}_{0,1,1} &= \mathcal{H}_{1+\varepsilon_y,1}(0,0,1). \end{aligned} \right. \quad (35)$$

В соответствии с аналитической схемой расщепления сначала по известным начальным условиям (13) вычисляется вспомогательная функция (25). Далее по формуле (34) определяется искомое решение задачи (14)-(16) в разные моменты времени $D\tau l_J$. Результаты моделирования сведены в табл. 2, где сравниваются с точным решением $u(1,1/2)=0,8867$ в стационарном режиме по изменению установлением температуры в центральном узле.

Таблица 2.

$D\tau l$	$u(1,1/2)$	$\varepsilon \%$
0,985	0,89196	-0,59
1,0	0,88799	-0,13

Таким образом, при аналитическом расщеплении удается выявить такую информацию о структуре программы, которая задается совокупностью покрывающих функций, аналитических по времени и координатному направлению расщепления. Описание функций в виде самостоятельных программных единиц (процедур, функций), позволяет сократить объем и улучшить структуру параллельной программы, а также распределить массивы так, что обмены данными между процессорами отсутствуют вовсе.

Вычислительные эксперименты

Эффективность предложенного подхода подтверждается решением задач нестационарной теплопроводности, некоторыми особенностями моделирования обратных задач исследования теплофизических свойств материалов, задач прогноза экологических систем под влиянием естественных и антропогенных факторов. Особенности моделирования указанного класса задач авторы планируют осветить в ближайших публикациях.

Выводы и перспективы дальнейших исследований

1. Суперкомпьютеры в настоящее время малодоступны вследствие огромной стоимости и цены обслуживания. В этой связи реальной альтернативой является применение вычислительных систем кластерного типа, один из вариантов, которой освещается в данной статье.
2. Являясь достаточно новой технологией, параллельные вычислительные системы кластерного типа эффективны при решении широкого класса нестационарных многомерных задач, позволяя при этом повысить производительность и качество вычислений.
3. Применение операционной системы LINUX позволяет максимально просто перейти на мощные многопроцессорные системы, не отвлекаясь на выполнение рутинных отладочных операций.

4. Предложен эффективный подход к решению проблемы латентности и разрешения тупиковых ситуаций для вычислительных кластеров архитектуры MPP.
5. Разработанные программные средства могут применяться для решения широкого класса многомерных нестационарных задач. Авторы надеются ознакомить читателей с особенностями моделирования на вычислительных кластерах в ближайших публикациях.

ЛИТЕРАТУРА

1. Роуч П. Вычислительная гидродинамика / Пер. с англ. – М.: Мир, 1980. – 616с.
2. Коздоба Л. А. Вычислительная теплофизика. – Киев: Наук. Думка, 1992. – 224с.
3. Миленин А., Дья Х., Мускальски З., Пилярчик Я. Моделирование с помощью метода конечных элементов процесса волочения проволоки в роликовых волокнах // Метизы 2(12), 2006. С. 30-33.
4. Воеводин В.В., Воеводин В.В. Параллельные вычисления. – СПб.: ВHV-Петербург, 2002, 608 с.
5. Иващенко В.П., Швачич Г.Г., Шмукин А.А. Некоторые аспекты проблемы математического моделирования задач металлургической теплофизики на основе применения параллельных вычислительных систем кластерного типа // Сучасні проблеми металургії. Наукові праці. Том 7. Дніпропетровськ: «Системні технології», 2005. С. 23-30.
6. Ладис А.О. Как построить и использовать суперкомпьютер. – М.: Бестселлер, 2003, 240 с.
7. Швачич Г. Г., Шмукин А. А. Особенности конструирования параллельных вычислительных алгоритмов для ПЭВМ в задачах тепло- и массообмена // Восточно Европейский журнал передовых технологий. 2(8) 2004. С. 42-47.
8. Швачич Г. Г., Шмукин А. А. О концепции неограниченного параллелизма в задачах теплопроводности // Восточно Европейский журнал передовых технологий. 3(9) 2004. С. 81-84.
9. Швачич Г. Г., Шмукин А. А. Особенности векторизации вычислений при моделировании процессов тепло – и массообмена // Математическое моделирование. 1(13) 2005. С. 23-28.
10. Иващенко В.П., Швачич Г.Г., Шмукин А.А. Некоторые математические аспекты конструирования параллельных вычислительных алгоритмов в задачах моделирования сложных систем // II International Conference “Strategy of Quality in Industry and Education” (June, 2-9 2006, Varna, Bulgaria): Proceeding, Volume 1. P. 196- 201.
11. Швачич Г.Г. О проблеме конструирования параллельных вычислительных систем максимально параллельной формы // Материалы I Международной научно-практической конференции «Наука и технологии: шаг в будущее – ‘2006». Том 12. – Белгород. Руснаучкнига, 2006. С. 26-29.
12. Антонов А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. – М.: Изд.- во МГУ, 2004. – 71 с.
13. Воеводин В.В. Математические основы параллельных вычислений.- М.: МГУ, 1991. – 345 с.
14. Годунов С.К., Рябенький В.С. Разностные схемы (введение в теорию). – М.: Наука, 1973. – 400 с.

-
15. Самарский А.А. Теория разностных систем. – М.: Наука, 1977. – 653 с.
16. Лыков А.В. Теория теплопроводности. – М.: Высшая школа, 1967. – 600 с.
17. Шнайдер П. Инженерные проблемы теплопроводности. – М.: Мир, 1960, 478 с.

пост. 13.09.07.