

Технология параллельного расчета нестационарных задач газовой динамики

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ»

Разработаны алгоритм и технология параллельного расчета нестационарной трехмерной задачи движения газа в многосвязной области на основе современных средств из библиотеки параллельных потоков TPL (Threading Parallel Library), входящей в ядро языка Visual C# 4.0. Предложенный алгоритм расчета позволяет организовать параллельные вычисления в отдельных подобластях, на которые разбивается глобальная расчетная область. Используется концепция многозадачности, предполагающая распределение всей расчетной нагрузки по отдельным задачам, которые ответственны за расчет в своей подобласти. Синхронизация расчетов обеспечивается использованием параллельных циклов нового стандартного класса Parallel.

Ключевые слова: алгоритм, параллельные вычисления, газовая динамика, технология удаленного вызова RPC, многоядерные процессоры, метод установления, локальная сеть.

Введение

Существует множество задач, решение которых требует огромных затрат временных ресурсов процессора электронно-вычислительной техники. К числу такого рода задач относятся нестационарные трёхмерные задачи газовой динамики, в частности задачи экологии атмосферы: формирование, дальнейшее рассеяние и горение газовых примесей в приземном слое атмосферы с учетом сложного рельефа местности, возникающие в результате техногенных аварий (пролив сжиженного газа из емкости хранения и его испарение, выброс сжатого газа в результате разрушения баллонов высокого давления, взрыв горючего с образованием токсичного облака и др.). Вследствие того, что процесс расчёта занимает достаточно длительное время, возникает необходимость сократить его каким-либо образом для решения поставленной инженерной задачи за практически приемлемые сроки. Одним из приемов уменьшения временных затрат является организация параллельных вычислений. Цель данной работы – разработка эффективного алгоритма параллельного расчета нестационарных задач газовой динамики и реализация его применительно к математической модели процесса рассеяния примесных газов в атмосфере.

1. Анализ методов распараллеливания расчета

Выбор технологии параллельного расчета во многом определяется имеющимися аппаратными средствами (локальный многопроцессорный компьютер, локальный компьютер с многоядерным процессором, локальная компьютерная сеть одноядерных компьютеров и т.д.). В настоящее время для решения задач большой размерности, которые требуют существенных временных ресурсов, используют локальные компьютерные сети с применением процедурной блокирующей синхронной технологии распределенных вычислений RPC (Remote Procedure Call) [1], согласно которой вызов удаленных программ подобен вызову функций в языке C, а пересылка данных осуществляется на основе транспортных протоколов TCP

(Transmission Control Protocol) или UDP (User Datagram Protocol) в едином формате обмена XDR (XML-Data Reduced) [2].

Однако распределение вычислительной нагрузки на отдельные компьютеры сети требует дополнительных усилий по синхронизации выполняемых расчетов в клиентской части программы и неизбежно связано с проблемами передачи больших объемов информации между серверами и клиентами при разделении общих данных. Этим недостатком лишены подходы, связанные с использованием многопроцессорных компьютеров [3], так как общие данные располагаются в единой оперативной памяти и отсутствует передача по сети. Но такого рода высокопроизводительная вычислительная техника (мультикомпьютеры Silicon Graphics, Cray T3D, Cray T3E, IBM SP2 и др.) все еще является достаточно дорогостоящей и не имеет широкого распространения в инженерной практике большинства научно-технических организаций и учебных заведений.

Наиболее популярны сейчас системы параллельного программирования на основе передачи сообщений MPI (Message Passing Interface) [4], идея которых предполагает представление параллельной программы в виде множества параллельно исполняющихся процессов (программы процессов обычно запрограммированы с использованием последовательных языков С или Фортран), взаимодействующих друг с другом в ходе исполнения с помощью коммуникационных процедур, которые и составляют библиотеку MPI. Однако такого рода библиотеки не входят в ядро языков программирования и являются программным продуктом сторонних разработчиков, что затрудняет создание параллельных программ. Кроме того, сама реализация библиотек MPI довольно сложна и требует искусства для достижения высокой производительности программ.

С выпуском .NET 4.0 в распоряжении разработчика появился целый набор инструментов параллельного программирования (класс Parallel, параллельные конструкции задач, параллельные коллекции), который называется PFX (Parallel Framework) [5]. Класс Parallel и параллельные конструкции составляют библиотеку TPL (Task Parallel Library), которая входит в ядро языка программирования Visual C# 4.0. Общая схема построения параллельной программы состоит в следующем:

- разбиение вычислительной нагрузки на части;
- выполнение этих частей параллельно с помощью потоков;
- объединение результатов после того, как они становятся доступными, с учетом безопасности потоков.

Учитывая требования задачи трехмерного расчета задачи газовой динамики, а также такие преимущества многопоточного программирования на базе инфраструктуры PFX, как встроенность основных инструментов в ядро языка программирования C#, облегченные возможности по синхронизации потоков и разделению данных, возможность реализовать данную технологию на персональных компьютерах с многоядерными процессорами, которые получают все большее распространение, делаем вывод о необходимости применения многозадачной технологии с применением средств библиотеки TPL, как наиболее подходящей для решения поставленной задачи.

2. Математическая модель

Для сравнительного вычислительного эксперимента, позволяющего оценить эффективность разработанной технологии параллельного расчета, использовалась математическая модель формирования и рассеяния газовой воздушной

смеси в приземном слое атмосферы [6]. Предполагается, что основным фактором, влияющим на рассматриваемые физические процессы, является конвективный перенос массы, импульса и энергии. Поэтому достаточно использовать упрощенные уравнения Навье–Стокса, которые получены отбрасыванием вязких членов в уравнениях движения газовой смеси (эйлеров подход с источниковыми членами).

Расчетная область представляет собой параллелепипед, который расположен в правой декартовой системе координат (рис. 1) и разбит на пространственные ячейки, размеры которых определяются масштабом характерных особенностей области (шероховатостью поверхностей, габаритами объектов). Разобьем расчетную область на несколько подобластей, а работу по расчету течения примеси в отдельных подобластях распределим между объектами задач.

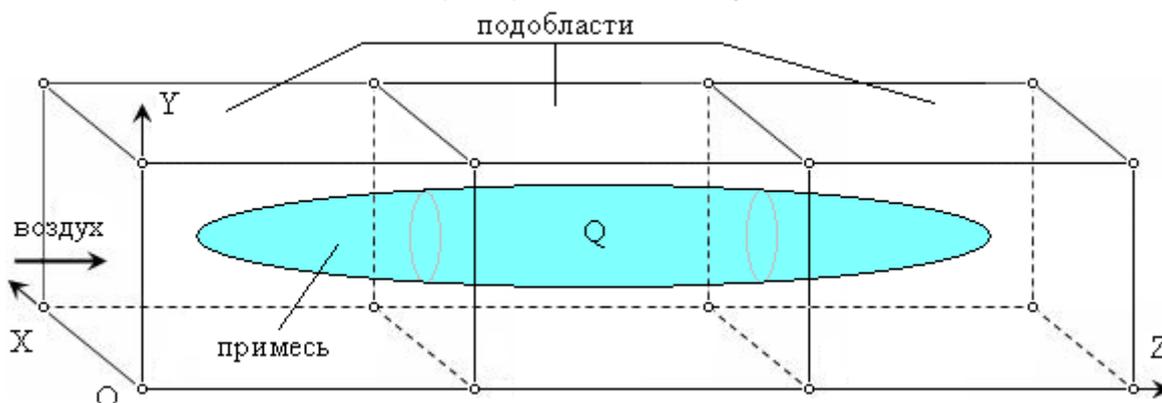


Рис. 1. Компьютерная модель рассеяния облака газовой примеси с концентрацией Q

Компьютерное решение системы фундаментальных уравнений газовой динамики для смеси, дополненной законами сохранения массы примесей в интегральной форме, получено явным методом Годунова [7]. Для аппроксимации уравнений Эйлера применяется конечно-разностная схема первого порядка. Центральные разности второго порядка используются для диффузионных источников членов в уравнениях сохранения примесей. Простая интерполяция давления применяется в вертикальном направлении. Метод Годунова характеризуется робастным алгоритмом, устойчивым к большим возмущениям параметров потока (например, давления).

3. Алгоритм параллельного расчета

Потоки массы, импульса, энергии и концентрации примеси через смежные грани подобластей являются разделяемыми данными чтения-записи, поэтому во избежание проблем с синхронизацией будем вычислять эти данные независимо для каждой подобласти с учетом небольшой потери эффективности алгоритма. В данном случае разделяемыми данными чтения служат газодинамические параметры в ячейках, примыкающих к разделительным границам, и синхронизация для доступа к ним не нужна.

Разделяемым данным между подобластями является также глобальный шаг по времени, который рассчитывается как минимальный по всей области для обеспечения устойчивости нестационарного расчета. Чтобы избежать возможных проблем с синхронизацией доступа чтения-записи к шагу по времени, предлагается в каждой подобласти использовать предварительно рассчитанный глобальный шаг по времени, вычислять локальные для подобластей минимальные шаги по времени, наименьший из которых определяется после объединения вычислительных потоков на данном временном слое.

С учетом допущений и упрощений общий алгоритм нестационарного расчета движения газа в глобальной расчетной области можно представить в виде следующих формальных блоков:

- расчет геометрических параметров глобальной сетки, причем количество ячеек вдоль оси OZ должно быть кратно количеству подобластей KML;
- инициализация газодинамических параметров глобальной сетки;
- создание массива MinTimeSteps минимальных шагов по времени для каждой из KML подобластей (в начальный момент времени глобальный шаг по времени dt равен 0):

```
MinTimeSteps = new double[KML];
```

- запуск основного цикла по времени t до истечения заданного времени окончания расчета TimeStop:

```
while (t < TimeStop) {...};
```

- формирование и запуск задач для параллельного расчета течения в текущий момент времени в каждой nm-й подобласти из имеющихся KML подобластей с помощью параллельного цикла Parallel.For из библиотеки TPL; телом каждой задачи является вызов функции FlowSubDomain(), принимающей дополнительно в качестве параметров шаг по времени dt, текущее время t, диапазон ячеек L1–L2 текущей подобласти вдоль оси OZ:

```
Parallel.For(0, KML, nm =>
{
    MinTimeSteps[nm] = FlowSubDomain(dt, t, L1, L2, nm);
})
```

- определение нового минимального глобального шага по времени (kfl - заданный коэффициент запаса по времени Куранта, FindMinDT() – функция, определяющая глобальное минимальное значение в массиве MinTimeSteps минимальных шагов по времени для каждой из KML подобластей):

```
dt = kfl * FindMinDT();
```

- переход на новый временной слой во всей расчетной области:
t = t + dt;
- завершение основного цикла по времени t (заданное время окончания расчета TimeStop истекло);
- вывод результирующих газодинамических параметров.

4. Апробация разработанной технологии

Предложенный алгоритм расчета был реализован в виде программы расчета на языке C#. Тестирование приложения и анализ эффективности параллельного алгоритма проводилось на четырехядерном процессоре для расчета рассеяния сферического облака водорода радиусом 4 м с координатами центра $x=5$ м, $y=0$ м,

$z=10,5$ м в расчетной области с габаритами $10 \times 10 \times 180$ м и двумя вариантами по количеству ячеек вдоль координатных осей $10 \times 10 \times 180$ и $10 \times 10 \times 300$. Время рассеяния составляло 10 с, а скорость ветра на входе в расчетную область была 15 м/с. Глобальная расчетная область разбивалась соответственно на 1–4 подобластей для каждого варианта вычислительной сетки.

Ускорение S_p , получаемое при использовании параллельного алгоритма для p процессоров (ядер процессора), по сравнению с последовательным вариантом выполнения вычислений определяется так:

$$S_p = \frac{t_1}{t_p}, \quad (1)$$

где t_1 – время решения задачи на ЭВМ с последовательным алгоритмом; t_p – время выполнения параллельного алгоритма. Естественно, что количество процессоров (ядер) определяет максимальный эффект от разбиения глобальной расчетной области на ряд подобластей.

Результаты вычислительного эксперимента представлены на рис. 2, 3.

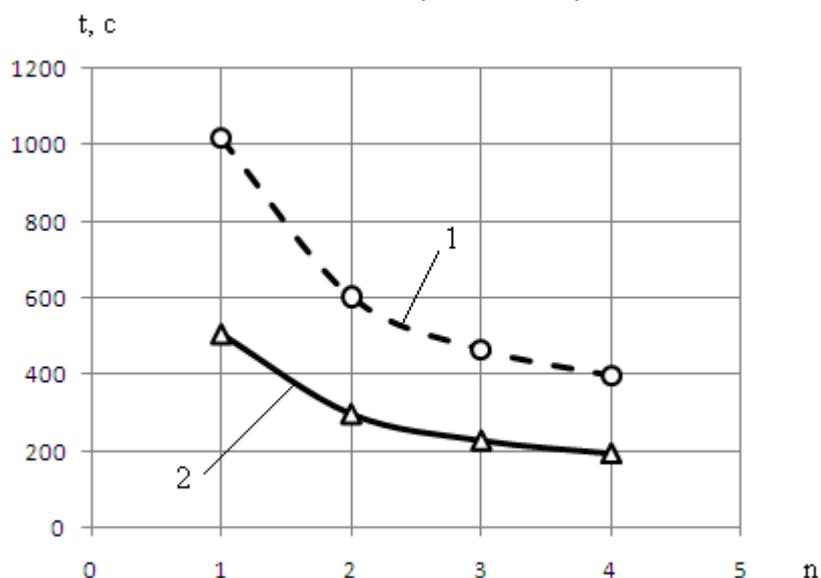


Рис. 2. Зависимость времени расчета t от количества ядер процессора n :
1 – вариант сетки $10 \times 10 \times 300$; 2 – вариант сетки $10 \times 10 \times 180$

Видно, что с увеличением количества разбиений до четырех затраченное время процессора уменьшается, но дальнейшее разбиение не приводит к снижению времени. Это объясняется использованием процессора с четырьмя ядрами. Максимальное физическое время процессора затрачивается в случае отсутствия разбиения глобальной области (рис. 2).

В обоих вариантах разбиения области присутствует выравнивание зависимостей с увеличением количества подобластей – кривые стремятся к асимптотам. Такое поведение можно объяснить увеличением «накладных расходов», связанных с дополнительными затратами времени на создание и удаление все большего количества рабочих потоков на каждом временном слое и необходимой работой по синхронизации их функционирования. Расчет эффективности алгоритма дает идентичные результаты для обоих вариантов вычислительной сетки (рис. 3). Эффективность растет с увеличением количества

подобластей.

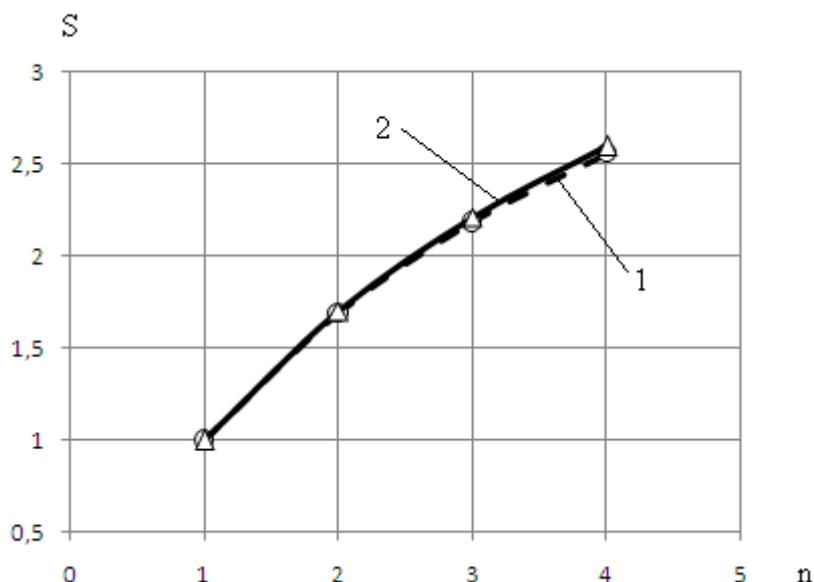


Рис. 3. Зависимость ускорения S от количества ядер процессора n :
1 – для сетки $300 \times 10 \times 10$; 2 – для сетки $180 \times 10 \times 10$

Заключение

Разработанный алгоритм параллельного расчета нестационарной трехмерной задачи движения газа в многосвязной области на основе интегрированной в ядро языка программирования Visual C# 4.0 технологии параллельных потоков позволяет организовать параллельные вычисления в отдельных подобластях, на которые разбивается глобальная расчетная область, и существенно снизить время расчета на компьютерах с многоядерными процессорами. Для реализации алгоритма используется концепция многозадачности, предполагающая распределение всей расчетной нагрузки по отдельным задачам, которые ответственны за расчет в своей подобласти. Синхронизация расчетов в подобластях обеспечивается использованием параллельных циклов нового класса Parallel стандартной библиотеки параллельных потоков TPL языка C#.

Список литературы

1. Разработка алгоритма параллельных вычислений при решении задач газовой динамики [Текст] / Ю. А. Скоб, К.П. Коробчинский, Д. С. Морозов, В. В. Шенцов // Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. Вып. 41. – Х.: Нац. аэрокосм. ун-т «ХАИ», 2009. – С. 109–115.
2. Локальные вычислительные сети [Текст]: справочник / Под ред. С.В. Назарова. – М.: Финансы и статистика, 1994. – 250 с.
3. Гергель, В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем [Текст]: учеб. пособие / В.П. Гергель, Р.Г. Стронгин. – Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2003. – 184 с.

4. Корнеев, В. Д. Параллельное программирование в MPI [Текст] / В. Д. Корнев. – Новосибирск: Изд-во ИВМиМГ СО РАН, 2002. – 215 с.
5. Freeman, A. Pro .NET 4 Parallel Programming in C# [Текст] / A. Freeman – New York: Apress, 2010. – 311 p.
6. Granovskiy, E.A. Numerical Modeling of Hydrogen Release, Mixture и Dispersion in Atmosphere [Электронный ресурс] / E.A. Granovskiy, V.A. Lyfar, Yu.A. Skob, M.L. Ugryumov // 1-st International Conference on Hydrogen Safety. – Pisa (Italy). – 2005. – 10 p. – Режим доступа: <http://conference.ing.unipi.it/ichs2005/Papers/110021.pdf>. – 3.06.2011 г.
7. Численное решение многомерных задач газовой динамики [Текст] / С.К. Годунов, А.В. Забродин, М.Я. Иванов и др. – М.: Наука, Гл. ред. физ.-мат. лит-ры, 1976. – 400 с.

Рецензент: д-р техн. наук, проф., проф. кафедры информатики М.Л. Угрюмов, НАКУ им. Н.Е. Жуковского «ХАИ», Харьков.

Поступила в редакцию 16.06.11

Технологія паралельного розрахунку нестационарних задач газової динаміки

Розроблено алгоритм і технологію паралельного розрахунку нестационарної тривимірної задачі руху газу в багатоз'язній області на основі сучасних засобів із бібліотеки паралельних потоків, що є складовою частиною алгоритмічної мови програмування Visual C# 4.0. Запропонований алгоритм розрахунку дозволяє організувати паралельні обчислення в окремих підобластях, на які розбивається глобальна розрахункова область. Використовується концепція багатозадачності, згідно з якою все розрахункове навантаження розподілено по окремих задачах, кожна з яких відповідає за розрахунок у власній підобласті. Синхронізацію розрахунків забезпечено за допомогою використання паралельних циклів нового стандартного класу Parallel мови C#.

Ключові слова: алгоритм, паралельні обчислення, газова динаміка, технологія віддаленого виклику RPC, багатоядерні процесори, метод установлення, локальна мережа.

A parallel calculation technology of time-dependent gas dynamics problems

An algorithm and a parallel calculation technology of unsteady three-dimensional problem of gas dynamics in a multiply connected domain on the basis of modern means of Threading Parallel Library that is a core member of the programming language Visual C# 4.0. The proposed algorithm allows to organize parallel computations in the selected subdomains of the global computational domain. A concept of multitasking which assumes the distribution of the whole computational load to individual tasks which are responsible for the calculation of its subdomains is used. Synchronization of the calculations is provided by the use of parallel loops from the new standard class Parallel.

Keywords: calculation algorithm, parallel computing, gas dynamics, RPC technology, multi-core processor, time-dependent problem, local network