УДК 004.424

Прусов В.А.[1], д. ф.-м. н., проф.,
Дорошенко А.Ю.[2], д. ф.-м. н., проф.,
Кацалова Л.М.[3], к. ф.-м. н.,
Бекетов О.Г.[4], аспірант

Паралельні обчислення двовимірної задачі конвективної дифузії на відеокарті

Основу сучасних метеорологічних моделей складають нелінійні тривимірні рівняння конвективної дифузії. Задача реалізації цих рівнянь є обчислювально складною. Застосування паралельних обчислень при розв'язанні таких задач є поширеною практикою.

В даній роботі пропонується застосування розпаралелювання на відеографічних процесорах при реалізації метеорологічних моделей. Представлено результати розв'язання тестової задачі за допомогою технології CUDA та графічного прискорювача. Проведено аналіз отриманих результатів.

Ключові слова: метеорологія, конвективна дифузія, паралельні обчислення, графічний прискорювач

[1,3] Український науково-дослідний гідрометеорологічний інститут, 03028, м. Київ, просп. Науки, 37

[2,4] Інститут програмних систем НАНУ, 03680, Київ-187, просп. Академіка Глушкова, 40

V.A.Prusov[1], D.Sci(Phys-Math.), Prof.,
A.Y.Doroshenko[2], D.Sci(Phys-Math.), Prof.,
V.M.Katsalova[3], Ph.D. (Phys-Math.),
O.G.Beketov[4], Postgrad. Stud.

Parallel computing of two-dimensional convective diffusion problem using graphics processing unit

The basis of modern meteorological models are nonlinear three-dimensional convection diffusion equations. The problem of realization of these equations is computationally complex. Usage of parallel computing for solving such problems is a common practice.

In this paper the application of parallelization with graphics processors for meteorological models implementation is proposed. The results of solving the test problem using CUDA technology with graphics processing unit are presented. An analysis of the obtained results provided.

Key words: meteorology, convective diffusion, parallel computing, graphics processing unit

[1,3] Ukrainian Hydrometeorological Institute, 03028, Kyiv, 37 Nauky Av.

[2,4] Institute of Software Systems NAS Ukraine, 03680, Kyiv-187, 40 Glushkova Av.

E-mails: [1] vitaliy.prusov@gmail.com, [2] dor@isofts.kiev.ua, [3] lessiagook@bigmir.net, [4] beketov.oleksii@gmail.com

Статтю представив д. т. н., проф. Гаращенко Ф.Г.

Вступ

В сучасній науці екологічні, кліматичні та синоптичні прогнози тісно пов’язані з математичним моделюванням циркуляції атмосфери. Рівняння Навьє-Стокса та тепло-, масо переносу, що складають основу сучасних моделей циркуляції [1], є нелінійними тривимірними рівняннями конвективної дифузії. Як правило, модель складають більше десятка таких рівнянь. Саме тому задача реалізації моделі циркуляції атмосфери має значну обчислювальну складність, а також, обмеження на строк отримання розв’язку. На сьогодні, в розвинених країнах розв’язання таких задач проводиться із застосуванням паралельного програмування та використанням багатопроцесорних суперкомп’ютерів. Проте ще досі, для України, можливість використання дорогої високопродуктивної обчислювальної техніки залишається проблемою.

Альтернативною платформою для проведення паралельних обчислень є відеографічні процесори, що вирізняються порівняною дешевизною, компактністю та економністю. Цей новий напрямок компьютерних обчислень є надзвичайно актуальним для вітчизняної науки.

В даній роботі приведено приклад застосування технології паралельних обчислень на відеографічних процесорах при розв'язанні двовимірної задачі конвективної дифузії з метою показати ефективність такого підходу при реалізації моделей циркуляції атмосфери.

Огляд GPGPU-технології CUDA

Специфіка роботи графічного прискорювача полягає у тому, що він має одночасно обробляти велику кількість пікселів візуалізуючого пристрою. Тому архітектура графічного прискорювача побудована таким чином, щоб уможливити одночасне виконання операцій з певним набором даних (SIMT-архітектура). В результаті розвитку GPU технологій з'явилося нове спрямування в обчислювальній техніці - GPGPU (*General Purpose computing on Graphics Processing Units*) – використання графічних процесорів для обчислювальних задач.

В 2006 році розробник графічних прискорювачів NVIDIA презентував GPGPU технологію CUDA (*Compute Unified Device Architecture*), що дозволяє проводити обчислення використовуючи графічні прискорювачі. Маючи виробничу продуктивність порядку сотень гігафлопс, графічні прискорювачі надають змогу проводити об'ємні обчислення навіть на звичайному ПК.

CUDA – програмно-апаратна архітектура, що дозволяє проводити обчислення за допомогою графічних процесорів NVIDIA. Графічний прискорювач (GPU) розглядається як спеціальний пристрій, що є масивно-паралельним сопроцесором центрального пристрою (CPU), має власну пам'ять та здатен одночасно виконувати велику кількість підпрограм – тредів. При виконанні програма на CUDA використовує як центральний пристрій, так і графічний. Типова схема виконання програми наступна:

1. Виділення області пам'яті на GPU та копіювання даних з CPU у виділену область пам'яті GPU.
2. Запуск ядра – паралельної частини програми, що виконується на GPU. Запуск виконує та керує ним CPU.
3. Копіювання отриманих результатів з пам'яті GPU до CPU та очищення виділеної пам'яті.

Основний процес CUDA виконується на головному пристрої. CPU-код ініціалізує GPU, розподіляє пам'ять відеокарти та системну пам'ять, копіює вихідні дані в пам'ять відеокарти, здійснює запуск ядер, копіює отримані результати з відеопам'яті, звільняє пам'ять і завершує роботу.

Апаратно графічні прискорювачі NVIDIA, що підтримують технологію CUDA, складаються з набору CUDA-ядер, кожне з яких здатне одночасно виконувати певну кількість тредів. Усі треди підпорядковуються наступній ієрархії. Верхній рівень ієрархії – сітка – підпорядковує усі треди, що виконують ядро. Сітка являє собою одно- або двовимірний масив блоків. Кожен блок – це одновимірний або двовимірний масив тредів, причому всі блоки, що утворюють сітку, мають однакові розмірність та розмір. Звертання до окремих тредів відбувається за допомогою індексів: кожен блок у сітці має адресу (індекс блоку у сітці), аналогічно кожен тред у блоці має свій власний індекс всередині блоку; таким чином, кожний тред має унікальний ідентифікатор. Треди можуть взаємодіяти між собою лише всередині одного блоку; під взаємодією розуміється використання окремої для кожного блоку так званої спільної пам'яті, а також синхронізація тредів, що може бути здійснена між тредами окремого блоку, проте не може бути здійснена на всьому GPU. Програма GPU (ядро) виконується над сіткою блоків потоків.

Таким чином, розділяючи основну задачу на сукупність підзадач, що можуть виконуватись незалежно одна від одної, і розв'язуючи ці підзадачі, використовуючи одночасно виконувані треди, досягається паралелізм виконання алгоритму.

Розв'язання графічних задач не потребує високої точності обчислень, тому звичайні відеокарти до останнього часу не підтримували 64-розрядний тип змінних з плаваючою крапкою. Проте спеціально розроблені відеокарти для розрахункових задач підтримують низку додаткових можливостей, таких як подвоєна точність, тривимірні сітки, глобальна синхронізація. Використовуючи їх можливості можна ефективно розв'язувати обчислювально громіздкі математичні задачі, до яких відноситься реалізація прогностичних метеорологічних моделей.

Чисельний експеримент

Рівняння Навьє-Стокса та тепло-, масо переносу, що складають основу сучасних метеорологічних моделей, є нелінійними тривимірними рівняннями конвективної дифузії:

$$\frac{\partial u}{\partial t} + \Lambda u = f\ ,\ (x_1, x_2, x_3) \in \Omega / \Gamma\ , \qquad t > 0\ , \quad (1)$$

$$u(0,x_1,x_2,x_3)=u^0(x_1,x_2,x_3),\quad (x_1,x_2,x_3)\in\Omega, \quad (2)$$

$$u(t,x_1,x_2,x_3)=0,\quad (x_1,x_2,x_3)\in\Gamma,\ t>0, \quad (3)$$

де $\Omega=[0,\ell_1]\times[0,\ell_2]\times[0,\ell_3]$ – просторова область визначення задачі, Γ – границя області Ω, $u=u(t,x_1,x_2,x_3)$ – залежна функція, $f=f(t,x_1,x_2,x_3)$ – вільний член рівняння, $\Lambda=\sum_{\alpha=1}^{3}\Lambda_\alpha$ – просторовий диференціальний оператор, що подається через суму простіших операторів:

$$\Lambda_\alpha=v_\alpha\frac{\partial}{\partial x_\alpha}-\frac{\partial}{\partial x_\alpha}\left(\mu_\alpha\frac{\partial}{\partial x_\alpha}\right),$$

$$v_\alpha=v_\alpha(x_1,x_2,x_3),\ \mu_\alpha=\mu_\alpha(x_1,x_2,x_3)>0.$$

Для проведення чисельного експерименту було розлянуто частковий випадок задачі (1)-(3) з відомим розв'язком:

$$\frac{\partial u}{\partial t}+v_1\frac{\partial u}{\partial x_1}+v_2\frac{\partial u}{\partial x_2}= =\frac{\partial}{\partial x_1}\left(\mu_1\frac{\partial u}{\partial x_1}\right)+\frac{\partial}{\partial x_2}\left(\mu_2\frac{\partial u}{\partial x_2}\right)+f \quad (4)$$

при $(x_1,x_2)\in[0;1]^2$, $t\in[0;10]$,

$$u(0,x_1,x_2)=\sin(x_1+x_2) \quad (5)$$

при $(x_1,x_2)\in[0;1]^2$, $t=0$,

$$u(t,x_1,x_2)=u_A(t,x_1,x_2) \quad (6)$$

при $(x_1,x_2)\in\partial[0;1]^2$, $t\in[0;10]$,

де

$$v_k=\sin(x_k),\mu^{(k)}=0.001+0.1*\sin^2(x_k)>0,$$

$$f(t,x_1,x_2)= =(v_1+v_2-(1+0.1*(\sin(2x_1)+\sin(2x_2))))\times \times\cos(x_1+x_2-t)+(\mu_1+\mu_2)*\sin(x_1+x_2-t).$$

Аналітичний розв'язок задачі (4)-(6) має вигляд

$$u_A(t,x_1,x_2)=\sin(x_1+x_2-t). \quad (7)$$

Розв'язання задачі (4)-(6) проводилось за допомогою адитивно-усередненого методу розщеплення [2] та методу явного рахунку [3]. Застосовано алгоритм трирівневого паралелізму (модифікований адитивно-усереднений метод (МАУМ)), представлений в [4]. Було розроблено реалізацію наведеного підходу для архітектури відеографічного прискорювача [5] засобами CUDA та OpenMP.

Створена реалізація МАУМ використовує рівномірну декомпозицію області Ω сіткою з кроком h розбиття часового проміжку з кроком τ. Розв'язок отримується в кожній точці сітки, і порівнюється із точним роз'язком в кінцевий момент часу.

У табл. 1 наведено часові витрати на виконання двох програм, що використовують метод МАУМ, розроблених для різних обчислювальних платформ - багатопроцесорної системи зі спільною пам'яттю (OpenMP API) та графічного прискорювача (CUDA API). Розрахунки проводились із використанням графічного прискорювача *NVIDIA GeForce GTX 650 Ti (768 CUDA-ядер, базова частота 928MHz, об'єм глобальної пам'яті 1024Mb)* та процесора *Intel Core i5-3570 (4 ядра, базова частота 3.40GHz, 64-бітний набір інструкцій)* у 64-бітному форматі представлення чисел з плаваючою крапкою.

Таблиця 1.

Час розв'язання задачі (4)-(6) з використанням CPU та GPU за допомогою МАУМ при різних значеннях просторового та часового кроків (h - просторовий крок, τ - крок за часом, T_{CPU} та T_{GPU} - час розв'язання задачі на CPU та GPU платформах відповідно, $err_{\max}$ - найбільше відхилення від точного розв'язку в точках сітки, $acc=T_{CPU}/T_{GPU}$ - відносне прискорення).

$1/h$	τ	T_{CPU}	T_{GPU}	$err_{\max}$	acc
128	h	1,37	1,52	0,0072	0,90
256	$h/2$	25,81	14,26	0,0048	1,81
512	$h/4$	683,63	176,44	0,0027	3,87
1024	$h/8$	10966	2687,32	0,0015	4,08

Для оптимального навантаження GPU параметр розбиття області $1/h$ обирався кратним 128. Із поданої таблиці видно, що порівняно невелике збільшення точності потребує суттєвого збільшення витрат часу на проведення обчислень.

При збільшенні кількості задіяних при розв'язанні повністю розпаралелюваної задачі процесорів удвічі, час, витрачений на розв'язання задачі, не може скоротитися більш, ніж у два рази. На практиці через витрати часу на передачу даних коефіцієнт прискорення при подвоєнні кількості процесорів ніколи не досягає двох. Як видно із таблиці, щоб досягти результатів продуктивності, які показує досліджувана відеокарта при значенні переметрів $1/h=256$ та $\tau=h/2$, знадобилося б два процесори. Щоб

наблизитися до результатів прискорювача при $1/h=1024$ та $\tau=h/8$, необхідно задіяти не менше чотирьох процесорів. З огляду на співвідношення вартості графічного прискорювача та відеокарти, використання останньої є економічно вигідним для розв'язання розглянутого типу задач.

На наступному графіку зображена залежність часу виконання від розміру просторової сітки зі сталим часовим кроком для GPU та одного ядра CPU.

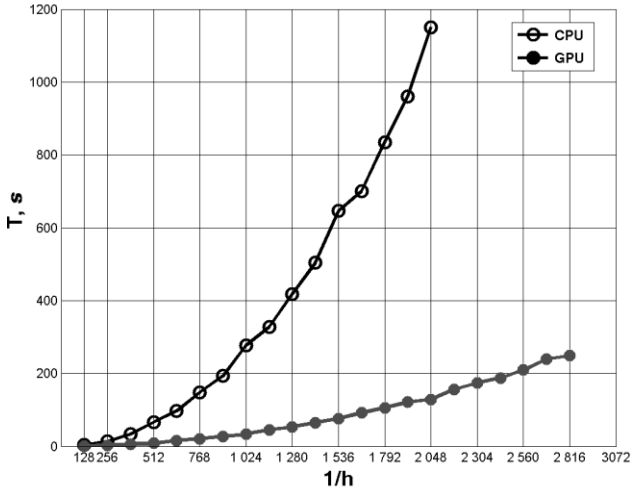

Рисунок 1. *Час розв'язання задачі (4)-(6) з використанням GPU та одного ядра CPU в залежності від розміру розбиття області при фіксованому* $\tau=0.01$.

Як ілюструє отриманий графік, при подрібненні області час, витрачений на обчислення відеокартою, зростає повільніше, ніж час, витрачений процесором. Графічний прискорювач, хоча й володіє меншою тактовою частотою, в силу особливостей своєї архітектури порається із задачами, які потребують великої кількості однотипних операцій (що властиво для задач на сітках) краще, ніж процесорне ядро, яке може виконувати операції лише з послідовним потоком даних.

Недоліком відеографічного прискорювача є обмежена оперативна пам'ять. У сучасних користувацьких відеокарт її об'єм не перевищує 4Gb, на відміну від оперативної пам'яті центрального пристрою, об'єм якої може бути значно більшим. Проте таке обмеження не поширюється на GPU, спеціалізованих саме на високопродуктивних обчисленнях.

Висновки

В даній роботі запропоновано використовувати відеографічні процесори при розв'язанні складних задач чисельного прогнозу, як альтернативу багатопроцесорним машинам.

Для обгрунтування доцільності такого підходу було розв'язано тестову задачу (4)-(6) із застосуванням паралельних обчислень на графічних прискорювачах. Розроблено реалізацію алгоритму розв'язку тестової задачі для архітектури відеографічного прискорювача засобами CUDA.

Представлено результати чисельного експерименту й проведено їх аналіз.

Згідно отриманих даних можна зробити висновок, що запропонований підхід ефективний при розв'язанні тестової задачі. Застосування розпаралелювання на відеокарті дає відчутне зменшення часу розв'язання задачі, дозволяє зменшувати часові й просторові кроки, тим самим покращувати точність розв'язку.

Так як алгоритм розпаралелювання для тестової задачі аналогічний алгоритму розв'язку складних рівнянь, що складають прогностичні метеорологічні моделі, можна стверджувати, що запропонований підхід ефективний для реалізації останніх. Тобто, паралельні обчислення на відеографічних процесорах й засоби CUDA доцільно застосовувати при реалізації математичних моделей циркуляції атмосфери.

Список використаних джерел

1. V.A.Prusov, A.Y.Doroshenko. Modelling of natural and anthropogenic processes in the atmosphere. – Kyiv: Naukova Dumka, 2006. – 542 p. (in ukr.)
2. Gordeziani D.G., Meladze G.V. Simulation of the third boundary value problem for multidimensional parabolic equations in an arbitrary domain by one-dimensional equations. – Zn vychisl. Mat. Mat. Fiz. – 1974. - №1. – P. 246- 250 (in rus.)
3. Prusov V.A., Doroshenko A.Y. Chernish R.I., Huk L.M. Efficient difference scheme numerical solution of the convective diffusion problem. – Cybernetics and Systems Analysis. - 2007. - №3. - P. 64 – 74 (in rus.).
4. Chernysh R.I. Modified additive-averaged splitting algorithm, its parallel realization and application to meteorological problems. – Manuscript. Taras Shevchenko National University of Kyiv, Kyiv, 2010 (in ukr.)
5. NVIDIA. NVIDIA CUDA C Programming Guide 4.1, November 2011.

Надійшла до редколегії 08.07.2013