УДК 004.75

ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ ДЛЯ ЕФЕКТИВНОЇ РЕАЛІЗАЦІЇ ЧИСЕЛЬНОГО ПРОГНОЗУ ПОГОДИ

Ю.М. Тирчак, А.І. Черняк

Інститут пограмних систем НАН України
03187, Київ, проспект Академіка Глушкова, 40/5,
т.н.: +38(044)526-15-38. E-mail: dor@isofts.kiev.ua, d2macster@gmail.com

Розглядається методика створення паралельної програмної реалізації для ефективного чисельного розв'язування одновимірних конвекційно-дифузійних рівнянь, що застосовується для чисельного моделювання прогнозу погоди і забруднення повітря. Методика забезпечує кількісні та якісні вдосконалення чисельних розв'язків. Наводяться результати чисельного експерименту та їх паралельної реалізації.

A technique is considered forefficient numerical solution of one-dimensional convection-diffusion equations that arise in numeric meteoroligic modeling and air pollution. The technique leads to significant quantitive and qualitative improvements in the behaviour of the numerical solutions. The study shows that the first-order upstream scheme is stable and efficient. The results of a numerical experiment and theor parallel implamentation are provided.

Вступ

Одними з головних параметрів якісного прогнозування метеорологічних величин є географічна точність та завчасність. Для досягнення високих показників цих параметрів розробляються спеціальні моделі та чисельні методи регіонального прогнозування погоди [1─4], адже географічна точність сучасних методів з використанням глобальних моделей прогнозу складає близько 150 км. До того ж глобальні методи прогнозу погоди не можуть описати всі процеси атмосфери і не враховують особливості поверхні планети, майже непридатні для сильноградієнтних полів та вимагають величезного обсягу обчислень, що становить серйозу проблему для довгострокових прогнозів. Українським гідрометеорологічним інститутом розробляється принципово новий метод, придатний як для глобального, так і для регіонального прогнозування [5]. Цей метод вимагає суттєво меншого обсягу обчислень, враховує рельєф поверхні, може мати значно дрібнішу сітку та може бути використаний за наявності сильно градієнтних полів та неоднорідних сіток.

Застосування паралельних обчислень є вкрай важливим для одержання оперативного (швидкісного) та високоточного метеорологічного прогнозу, оскільки забезпечує необхідне прискорення обчислень паралельних алгоритмів при їх виконанні на потужних мультипроцесорних системах. Проте розробка ефективних паралельних алгоритмів для мультипроцесорних систем є непростою науково-технічною проблемою і вимагає вирішення питань міждисциплінарних досліджень з використанням знань математичних, архітектурних й програмних моделей паралельних обчислень [6].

У цій роботі виконано аналіз та експериментальну паралельну програмну реалізацію конвекційно-дифузійної обчислювальної моделі метеорологічного прогнозування, розробленої у [5] і спрямованої на подолання вищезазначених недоліків глобальних моделей прогнозування. Викладено основи математичної моделі прогнозування, описано підхід до крупно-блочного розпаралелювання обчислювального методу та наведено результати експериментів для перевірка точності та ефективності програмної реалізації на мультипроцесорній системи кластерної архітектури [7].

1. Математична модель та різницевий метод для чисельного прогнозування

При математичному моделюванні фізичних процесів, що відбуваються у атмосфері Землі, як правило, користуються гідродинамічними моделями, що базуються на рівняннях Нав'є─Стокса, з додаванням у якості замикаючих рівняння стану та тепло- і масопереносу. Вони представляють собою нелінійні диференційні рівняння у частинних похідних. У загальному випадку розв'язку рівнянь гідродинаміки та тепло- і масопереносу не існує. Основним способом їх розв'язку залишаються чисельні методи.

У векторній формі рівняння мають вигляд:

$$\frac{\partial R}{\partial t} + A^{\alpha}\frac{\partial R}{\partial x_{\alpha}} + \frac{\partial}{\partial x_{\alpha}} B^{\alpha}\frac{\partial R}{\partial x_{\alpha}} = F, (\alpha = 1,2,3)\,, \tag{1}$$

де R і F – вектор-функції розміру m, A^{α}, B^{α} - квадратні матриці розміром $m \times m$. Для такого рівняння деференціальний оператор являтиме собою

$$D = A^{\alpha}\frac{\partial}{\partial x_{\alpha}} + \frac{\partial}{\partial x_{\alpha}} B^{\alpha}\frac{\partial}{\partial x_{\alpha}}\ . \tag{2}$$

© A.Ju. Shelestov, N.N. Kussul, S.V. Skakun, 2006

Рівняння (1) може бути перетворене таким чином:

$$\frac{\partial R}{\partial t}+\eta^{\alpha}(t,\tau)D_{\alpha}R=f^{\alpha}, \tag{3}$$

де $\eta^{\alpha}(t,\tau)=m\delta_j^{\alpha}$, $t\in[n\tau+(j-1)\tau/m, n\tau+j\tau/m]$. Фактично, таке представлення означає, що на кожному часовому інтервалі $[n\tau+(j-1)\tau/m, n\tau+j\tau/m]$ буде розв'язуватись система одновимірних диференціальних рівнянь.

При спрощеній постановці задачі для одновимірного рівняння конвективної дифузії у прямокутнику $\overline{Q}=\{x,t:0\le x\le l, 0\le t\le T\}$ розглядається рівняння конвективної дифузії компоненти *q* вектор-функції *R:*

$$\begin{aligned}&\frac{\partial q}{\partial t}+v\frac{\partial q}{\partial x}=\frac{\partial}{\partial x}\left(\mu\frac{\partial q}{\partial x}\right)+bq+f,\ \mu\ge 0,\ 0\le x\le l,\ t\ge 0,\\&q(x,0)=\eta(x), 0\le x\le l,\\&q|_{x=0}=\alpha(t), q|_{x=l}=\beta(t), t>0.\end{aligned} \tag{4}$$

Рівняння (4) включає у собі такі члени, як: нестаціонарний член $\frac{\partial q}{\partial t}$, який описує швидкість зміни величини *q*, конвективний член $v\frac{\partial q}{\partial x}$, який описує конвекцію величини q, що обумовлена рухом оточуючого середовища зі швидкістю *v*, дифузійний (дисипативний) член $\frac{\partial}{\partial x}\left(\mu\frac{\partial q}{\partial x}\right)$, який описує дифузію величини q в оточуючому середовищі вздовж осі *x*, внутрішнє джерело у точці *x* у момент часу *t* (виділення чи поглинання величини *q* всередині області $\overline{Q}$) і густину зовнішніх джерел $f(x,t)$.

Серед існуючих на сьогодні явних схем вирішення задачі (4) стійкими є лише схеми, що апроксимовують конвективний член різницями проти потоку. Вони мають похибку розв'язку рівняння (4) $O[\tau,h]$. Умова стійкості накладає обмеження на часовий крок $\tau\le[(|v|+2\mu/h)/h]^{-1}$. При цьому у системі виникає штучна в'язкість, яка приводить до затухання коливань розв'язку. Але в окремих випадках схема з різницями проти потоку стає настільки неточною, що не може розглядатись, як схема вирішення рівняння (4). З іншого боку, основними недоліками існуючих неявних схем можна відмітити наступні:

при їх реалізації на кожному часовому кроці доводиться розв'язувати систему алгебраїчних рівнянь з трьохдіагональною матрицею,

у випадках розв'язку нелінійного рівняння руху ($q=v, \mu=0$) може з'явитись нестійкість, і доведеться вводити штучну в'язкість,

при моделюванні реальної атмосфери можуть траплятись досить різкі градієнтні фронти, що може викликати труднощі при обчисленнях, які будуть пов'язані з нескінченною швидкістю поширення збурення.

Українським гідрометеорологічним інститутом було запропоновано стійку різницеву схему [5] чисельного розв'язку рівняння (4), де в загальному тривимірного випадку пропонується користуватись методом розщеплення для кожного часового інтервалу.

Запропонований метод є явним. Але при цьому він не вимагає досить жорстких обмежень на часовий крок, він обмежується лише здоровим глуздом. А це є особливо важливим при детальному моделюванні поведінки атмосфери у напрямку вісі *z*, яка перпендикулярна поверхні Землі.

Суть методу зводиться до наступного.

При $v\ge 0$

$$q_j^{n+1}=\frac{p_j q_{j-1}^{n+1}-d_j q_{j+1}^{n}+(1+d_j)q_j^{n}+\tau f_j^{n}}{1+p_j-\tau b}, \tag{5}$$

де

$$p_j=\frac{\tau}{h_{j-1}+h_j}\left(\frac{h_j}{h_{j-1}}v_j^{n+1}+\frac{\mu_j^{n+1}+\mu_{j-1}^{n+1}}{h_{j-1}}\right), \tag{6}$$

$$d_j=\frac{\tau}{h_{j-1}+h_j}\left(\frac{h_{j-1}}{h_j}v_{j-1}^{n}-\frac{\mu_{j+1}^{n}+\mu_j^{n}}{h_j}\right). \tag{7}$$

При $v < 0$

$$q_{j-1}^{n+1} = \frac{\left(1+s_{j-1}\right)q_{j-1}^{n} + s_{j-1}q_{j-2}^{n} - r_{j-1}q_{j}^{n+1} + \tau f_{j}^{n}}{1 - r_{j-1} - \tau b}, \tag{8}$$

де

$$r_{j-1} = \frac{\tau}{h_{j-2}+h_{j-1}}\left(\frac{h_{j-2}}{h_{j-1}}v_{j-1}^{n+1} + \frac{\mu_{j-1}^{n+1}+\mu_{j-1}^{n+1}}{h_{j-1}}\right), \tag{9}$$

$$s_{j-1} = \frac{\tau}{h_{j-2}+h_{j-1}}\left(\frac{h_{j-1}}{h_{j-2}}v_{j-1}^{n} + \frac{\mu_{j-1}^{n}+\mu_{j-2}^{n}}{h_{j-2}}\right). \tag{10}$$

Але можуть виникнути такі умови, за яких швидкість руху середовища зліва від точки зміни знаку буде від'ємною, а справа − додатньою. Тому для проведення розрахунку необхідні значення у наступний момент часу. Ці значення можна одержати за такими формулами:

$$q_{j-1}^{n+1} = \frac{\left(1+p_j\right)\bullet G_j - r_{j-1}J_j}{\left(1+p_j\right)\bullet\left(1-r_j\right) + p_j r_j}, \tag{11}$$

$$q_{j}^{n+1} = \frac{p_j G_j + \left(1-r_{j-1}\right)\bullet J_j}{\left(1+p_j\right)\bullet\left(1-r_j\right) + p_j r_j}, \tag{12}$$

де

$$G_j = \left(1-s_{j-1}\right)\bullet q_{j-1}^{n} + s_{j-1}q_{j-2}^{n}, \tag{13}$$

$$J_j = \left(1+d_j\right)\bullet q_j^{n} - d_j q_{j+1}^{n}. \tag{14}$$

Саме це і забезпечує принципово новий момент запропонованого методу в порівнянні з іншими методами розрахунку, для яких зміна знаку швидкості руху в такий спосіб може призвести до нестійкостей розв'язку або значних витрат обчислювальних потужностей.

2. Постановка обчислювального експерименту

Для програмної реалізації та перевірки моделі було вибрано два основних рівняння, які є залежними одне від одного, та без яких не можуть бути розв'язані інші рівняння, що описують поведінку атмосфери у межах методу, що розглядається.

$$\frac{\partial u}{\partial t} = -\frac{u}{r\cos\varphi}\frac{\partial u}{\partial\lambda} - \frac{v}{r}\frac{\partial u}{\partial\varphi} + \left(2\Omega + \frac{u}{r\cos\varphi}\right)v\sin\varphi - \frac{1}{r\cos\varphi}\frac{1}{\rho}\frac{\partial P}{\partial\lambda} +$$
$$+\frac{1}{r\cos\varphi}\frac{\partial}{\partial\lambda}\left(K_G\frac{\partial u}{\partial\lambda}\right) + \frac{1}{r}\frac{\partial}{\partial\varphi}\left(K_G\frac{\partial u}{\partial\varphi}\right) + \frac{\partial}{\partial z}\left(K_M\frac{\partial u}{\partial z}\right), \tag{15}$$

$$\frac{\partial v}{\partial t} = -\frac{u}{r\cos\varphi}\frac{\partial v}{\partial\lambda} - \frac{v}{r}\frac{\partial v}{\partial\varphi} + \left(2\Omega + \frac{u}{r\cos\varphi}\right)u\sin\varphi - \frac{1}{r}\frac{1}{\rho}\frac{\partial P}{\partial\varphi} +$$
$$+\frac{1}{r\cos\varphi}\frac{\partial}{\partial\lambda}\left(K_G\frac{\partial v}{\partial\lambda}\right) + \frac{1}{r}\frac{\partial}{\partial\varphi}\left(K_G\frac{\partial u}{\partial\varphi}\right) + \frac{\partial}{\partial z}\left(K_M\frac{\partial v}{\partial z}\right), \tag{16}$$

де u – швидкість вітру в напрямку, паралельному до паралелей; v – швидкість вітру у напрямку, параллельному до меридіан; r – радіус Землі; λ - довгота; φ - широта; ρ - густина повітря; P – тиск повітря (mbar); K_G - коефіцієнт горизонтального турбулентного обміну (m^2/c); K_M – коефіцієнт вертикального турбулентного обміну (m^2/c).

Для постановки експериментів з чисельного моделювання і прогнозування прийняті певні спрощення даної моделі щодо запропонованої в [5] оригінальної моделі, що полягають у наступному. По-перше, у неї відсутнє врахування рельєфу. По-друге, відсутня вертикальна складова вітру. По-третє, вважається, що похідна $\frac{\partial P}{\partial\lambda}$ є незмінною з висотою. Такі спрощення дозволяють зберегти основну ідею методу обчислень. Успішне розв'язання такої системи рівнянь після її ускладнення не призводить до суттєвих змін програмної реалізації.

Постановка задачі вимагає не просто розрахунку погодних умов на заданий термін, а проведення цих розрахунків у максимально короткий термін. Тому перед програмною реалізацією була поставлена вимога масштабованості щодо даних, тобто програма має бути здатною виконати прогноз для довільних регіонів планети та використовувати розрахункову сітку довільної щільності. Відповідно, програмна реалізація обчислень має виконуватись на паралельній високопродуктивній архітектурі.

Демонстрація і перевірка роботи програми здійснюється за даними прогнозування погоди для Європи. При цьому було визначено два параметри програми − змінні довготи і широти λ і φ, що змінюються у межах [0, pi/2]. Метеорологічні дані отримуються з одного з світових метеорологічних центрів, що розташований у м. Офенбах (Німеччина) [7]. Вони являють собою початкові умови, тобто набір вимірів, для конкретного моменту часу, а також прогнозовані дані для наступних моментів часу з кроком у 12 годин. Ці дані потрапляють на вхід програми. Після отримання прогнозу за допомогою програми, результат порівнюється з реальними даними цього метеорологічного центру. Головний інтерес для порівняння представляють потужні градієнтні поля, тобто області фронтів.

3. Розпаралелювання обчислень за програмою прогнозування

Спрощена обчислювальна модель описується системою з двох рівнянь, кожне з яких є тривимірним. Для кожного часового кроку кожне з тривимірних рівнянь можна представити системою 3-х одновимірних рівнянь. Кожне таке одновимірне рівняння є незалежним від інших і може бути розв'язане окремо. Після отримання результатів розв'язання 3-х одновимірних рівнянь, на які розпалось тривимірне рівняння, до порахованих значень необхідно застосувати операцію знаходження середнього арифметичного аби отримати початкові умови для наступного часового кроку. Тобто, задача теоретично піддається розпаралелюванню і може бути розв'язана на багатопроцесорній системі з використанням більш ніж одного процесора. Характеристиками експериментального паралельного програмування задачі прогнозування, що представляють науковий інтерес, є такі:

- загальний час виконання у співвідношенні із часом, на який проводиться прогноз;
- коефіцієнт розпаралелювання та прискорення;
- максимальний об'єм даних та розмір сітки.

3.1. Особливості архітектури обчисювального комплексу. Метою експериментів з паралельного програмування було проведення досліджень на мультипроцесорному кластері Інституту кібернетики ім. В.М. Глушкова НАН України [8].

Обчислювальні кластери Інституту кібернетики ім. В.М. Глушкова НАН України СКІТ-1 та СКІТ-2 є мультипроцесорними архітектурами з розподіленою пам'яттю, тобто без можливості прямого доступу до адресного простору сусідніх вузлів. Відповідно, це не дозволяє застосувати просту модель розпаралелювання, коли кожен із задіяних для задачі процесорів розв'язує своє одновимірне рівняння для усього простору, на якому розглядається задача. Просту модель можна реалізувати на кластерній архітектурі, але це вимагає синхронізації для кожного часового кроку усіх даних на всіх вузлах, задіяних для обрахунку. Пересилання даних між вузлами призводить до того, що процесори не виконують корисних розрахунків, а просто очікують синхронізації або виконують операції, пов'язані з прийомом-передачею даних. Така схема ставить досить жорстке обмеження на розмір моделі, оскільки усі параметри і дані задачі мають уміститись у пам'ять кожного з вузлів. При чому, незалежно від кількості таких вузлів, кожному з них доводиться дублювати дані усіх інших вузлів. Така проста схема добре підходила б для реалізації на мультипроцесорній архітектурі зі спільною пам'яттю. А для наявного обчислювального кластеру застосовується крупно-блочне розпаралелювання, при якому увесь простір, для якого виконуються обчислення, розбивається на області, і кожна область підлягає обчисленню на окремому процесорі. При необхідності процеси, що виконують розрахунки над своїми областями, можуть обмінюватись даними.

3.2. Структури даних. Важливим питанням розробки паралельної програми є вибір структур даних задачі, що тісно пов'язане з вибором мови програмування. На кластері встановлено бібліотеки *lam* та *scali*, що дозволяє створювати програми мовами C/C++ та Fortran. У якості мови програмування було обрано C++, оскільки вона дозволяє створити ієрархічну структуру даних у поєднанні з можливістю використання переваг об'єктно-орієнтованого програмування.

Структуру даних можна уявити таким чином (рис. 1). Тут **u_data** та **v_data** – змінні класу **cubicle**, що утримує всю інформацію щодо однієї з величин, які підлягають обчисленню. У цьому класі знаходиться масив **initial_data** початкових даних для усього простору, який припадає для моделювання даним процесом, **next_x, next_y, next_z** – три масиви такого ж розміру для збереження проміжних даних при обрахуванні одновимірних рівнянь на кожному часовому кроці, а також структура **border**, у якій зберігається інформація, необхідна для апроксимації значень на границях області. Структура size утримує інформацію про розміри області по вісі *x*, *y* та *z*. Структура **neighbors** утримує номери процесів, що є сусідніми за даними для конкретного процесу. Структури **f, v** та μ будуть розглянуті пізніше.

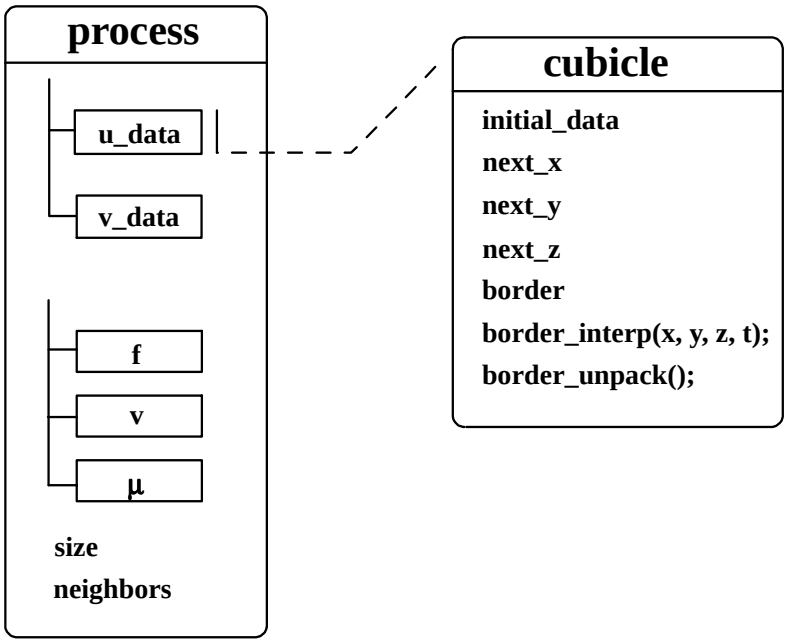

Рис. 1. Структура даних

3.3. Схема паралельних обчислень. Метод крупно-блочного розпаралелювання передбачає таку організацію обчислень, при якій схема обчислень має пройти декілька етапів проектування і виглядає наступним чином:

1) приймається рішення про те, яким чином буде розбита область прогнозування по кожній з осей;
2) створюється відповідна кількість процесів (кількість областей плюс один службовий процес);
3) службовий процес підраховує кількість вузлів сітки, яка припадатиме кожному процесу обчислень, а також визначає ті процеси, що будуть сусідніми за даними, та передає ці дані процесам, що виконуватимуть обчислення;
4) службовий процес розбиває дані за початкових умов у відповідності до областей, які процеси будуть обраховувати, та передає ці дані;
5) службовий процес з'ясовує, які дані знаходяться на границях області, що була вихідною, а також які процеси мають у своєму складі цю область; службовий процес надсилає ті частини границь, які потрібні кожному процесу; службовий процес виконує таку процедуру для початкових умов а також для даних, що є прогнозом світового метеоцентру;
6) процеси, що виконують розрахунки, отримують необхідні дані та проводять обчислення для заданого терміну;
7) службовий процес збирає дані та записує їх у вихідний файл.

У нашому випадку прийнятним способом декомпозиції тривимірної сітки на підобласті є вибір під областей у вигляді кубів (рис. 2), оскільки він забезпечує, по-перше, регулярність розбиття, по-друге – масштабованість паралельної програми та можливість отримання паралельного вирішення задачі незалежно від кількості наявних на кластері обчислювальних вузлів (процесорів).

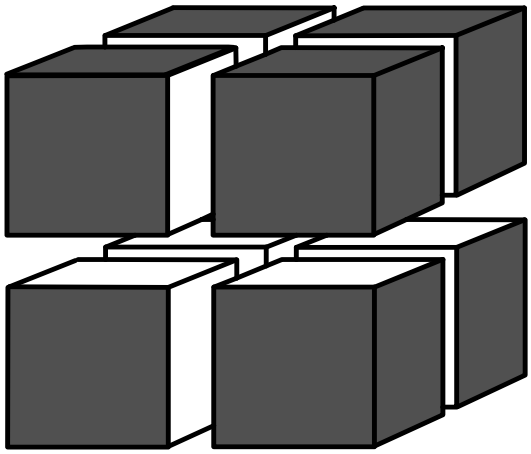

Рис. 2. Розбиття тривимірної сітки на підобласті

3.4. Процес обчислень. У даній моделі кожен з процесів розв'язує рівняння 15 та 16 для своєї підобласті простору (куба). Процес власне обчислень відбувається у такий спосіб, при якому рівняння 16 для кожного часового кроку розпадається на наступну систему одновимірних рівнянь:

$$\frac{\partial u}{\partial t} = -\frac{u}{r\cos\varphi}\frac{\partial u}{\partial \lambda} + \left(2\Omega + \frac{u}{r\cos\varphi}\right)v\sin\varphi - \frac{1}{r\cos\varphi}\frac{1}{\rho}\frac{\partial P}{\partial \lambda} + \frac{1}{r\cos\varphi}\frac{\partial}{\partial \lambda}\left(K_G\frac{\partial u}{\partial \lambda}\right), \quad (17)$$

$$\frac{\partial u}{\partial t} = -\frac{v}{r}\frac{\partial u}{\partial \varphi} + \frac{1}{r}\frac{\partial}{\partial \varphi}\left(K_G\frac{\partial u}{\partial \varphi}\right), \quad (18)$$

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial z}\left(K_M\frac{\partial u}{\partial z}\right), \quad (19)$$

а рівняння 16, у свою чергу, розпадається на систему:

$$\frac{\partial v}{\partial t} = -\frac{u}{r\cos\varphi}\frac{\partial v}{\partial \lambda} + \left(2\Omega + \frac{u}{r\cos\varphi}\right)u\sin\varphi - \frac{1}{r}\frac{1}{\rho}\frac{\partial P}{\partial \varphi} + \frac{1}{r\cos\varphi}\frac{\partial}{\partial \lambda}\left(K_G\frac{\partial v}{\partial \lambda}\right), \quad (20)$$

$$\frac{\partial v}{\partial t} = -\frac{v}{r}\frac{\partial v}{\partial \varphi} + \frac{1}{r}\frac{\partial}{\partial \varphi}\left(K_G\frac{\partial u}{\partial \varphi}\right), \quad (21)$$

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial z}\left(K_M\frac{\partial v}{\partial z}\right). \quad (22)$$

Отже, тепер на кожному часовому кроці кожне рівняння буде розв'язуватись окремо, а потім з одержаних результатів відбуватиметься підготовка даних для наступного часового кроку.

Під час розв'язання одновимірного рівняння спочатку з'ясовується, чи є на відрізку вузли, для яких необхідно застосовувати вирази (11) та (12). Такі вузли можуть розташовуватись на межах області даних процесу, тому необхідно отримати від сусідніх процесів інформацію щодо швидкості у середовищі. Для цього процеси обмінюються пограничними даними про координатні швидкості у середовищі. Далі загальна організація обчислень влаштовується за методом фронту хвилі [6, 9].

Маючи інформацію про вузли, у яких змінюється напрямок швидкості, відрізок розбивається на області, для яких обчислення проводяться за виразами (5) та (8). Якщо процес має у своєму складі вузли з пограничного шару початкової області, і ці вузли належать відрізку, для якого проводиться моделювання, то значення в цих вузлах обчислюється. Після цього виконується перевірка, чи достатньо даних для початку обчислень на підобластях відрізка. Якщо знайдуться такі підобласті, для них проводиться моделювання. Після цього перевіряється, чи потрібно переслати обраховані дані з граничних підобластей процесам, що є сусідами та чекають на дані. Якщо знайдеться хоча б одна область, для якої не вистачає даних для початку роботи, процес порахує усі інші області, а потім буде чекати надходження даних.

Для того, аби мінімізувати час очікувань, перед початком розв'язування задачі для одновимірного рівняння для усієї області, що припадає одному процессу, відбувається підрахунок величин v та μ, виділення пограничних шарів величин v, μ та q і проводиться їх пересилання потрібним процесам за один раз. Тому під час моделювання можлива передача лише щойно порахованих даних. При цьому усі пересилання є неблокуючими [10], що забезпечує асинхронність виконання операцій обміну між процесами і мінімізацію часу непродуктивних очікувань.

Процес моделювання потребує службових масивів для зберігання тимчасових даних. Таки масиви створюються перед процесом моделювання, а в його ході відповідні функції отримують лише посилання на нього. Це суттєво зменшує час обчислень за рахунок того, що на кожній ітерації процеси уникають непотрібних процедур виділення, а потім звільнення досить великих обсягів пам'яті.

Далі у таблиці та на діаграмі (рис. 3) показано експериментальні дані щодо часу виконання паралельних обчислень на мультипроцесорному кластері СКІТ-2.

Таблиця

Кількість процесорів, задіяних для обчислень	Час виконання прогнозу погоди на 48 годин, с	Розбиття області у напрямках x, y, z
1	104	1,1,1
2	74.5	2,1,1
3	54	3,1,1
4	42.6	4,1,1
5	36	5,1,1
6	34	6,1,1
2	63.5	1,2,1
3	49.6	1,3,1

4	39.1	1,4,1
5	31.1	1,5,1
6	29.5	1,6,1
2	48	1,1,2
3	37	1,1,3
4	78.7	2,2,1
4	45.7	2,1,2
8	57.5	2,2,2
12	66.5	3,2,2
12	45.3	2,3,2
12	49.6	2,2,3
18	48	3,3,2
18	40.3	2,3,3

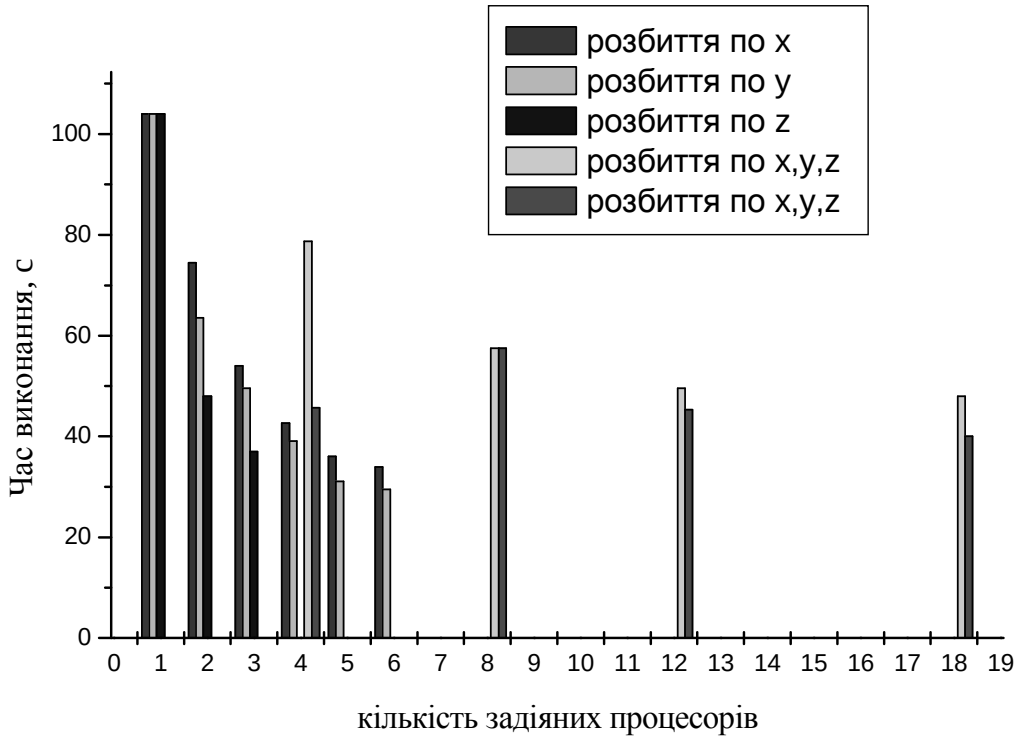

Рис. 3. Діаграма залежності часу виконання прогнозування від кількості процесорів

Отже, з даних таблиці та діаграми видно, що розбиття області лише по одній осі є вигіднішим, ніж проводити розбиття одночасно по двом чи навіть по трьом осям. Зокрема, це пояснюється лише тим, що об'єм даних задачі є малим аби скористатись перевагами крупно-блочного розпаралелювання.

Висновки

У роботі розглягуті методика ефективного чисельного розв'язування одновимірних конвекційно-дифузійних рівнянь, що застосовується для чисельного моделювання прогнозу погоди і забруднення повітря, та її паралельна програмна реалізація на мультипроцесорній систпемі кластерної архітетктури. В основі методики лежить принципово новий метод, придатний як для глобального, так і для регіонального прогнозування метеорологічних величин, рохроблений Українським гідрометеорологічним інститутом [5]. Цей метод вимагає суттєво меншого обсягу обчислень, враховує рельєф поверхні, може мати значно дрібнішу сітку і може бути використаний за наявності сильно градієнтних полів та неоднорідних сіток.

Разом з тим ефективність обчислень забезпечується не тільки кількісними та якісними вдосконаленнями чисельних розв'язків, а й методикою ефективної паралельної реалізації обчислень на мультипроцесорній архітектурі. Наведені в роботі результати чисельних експериментів та паралельних обчислень свідчать про добру придатність запропонованих алгоритмів до паралельної реалізації та їх значний потенціал для створення на їх основі прикладних комп'ютерних систем.

1. *Прусов В.А., Дорошенко А.Е., Приходько С.В., Тырчак Ю.М., Черныш Р.И.* Методы эффективного решения задач моделирования и прогнозирования региональных атмосферных процессов // Проблемы программирования. - 2004. - № 2-3. - С. 556-559.
2. *Прусов В.А., Черниш Р.І.* „Ефективні методи чисельного розв'язання диференціальних рівнянь моделі циркуляції атмосфери", "Теоретичні т прикладні аспекти побудови програмних систем", НаУКМА.-К. 5-8 жовнтя. 2004. − С. 125─129.
3. *Дорошенко А.Ю., Тирчак Ю.М.* Ефективне розпаралелювання обчислень для задач фізики атмосфери. // Праці Міжнародної конф."Теоретичні т прикладні аспекти побудови програмних систем", НаУКМА.─К. 5-8 жовнтя. ─ 2004. − С. 325─329.
4. *Doroshenko A.Yu., Prusov V.A.* Methods of efficient modeling and forecasting regional atmospheric processes, in I. Faragó, Á.Havasi, K. Georgiev (editors) „Advances in Air Pollution Modeling for Environmental Security", NATO Science Series, 54, Springer Verlag, 406 p. – 2005. ─ P. 143─152.
5. *Prusov V.A., Doroshenko A.Yu.* On efficient numerical solution of one-dimensional convection-diffusion equations in modeling atmospheric processes // International J. of Environment and Pollution. - 2006, (прийнята до публікації).
6. *Дорошенко А.Е.* Математические модели и методы организации высокопроизводительных параллельных вычислений. Алгебродинамический подход. – Киев: Наук. думка, 2000. - 177 с.
7. http://www.dwd.de, веб-сайт метеорологічного центру в м. Офенбах
8. http://cluster.icyb.kiev.ua – сайт, присвячений архітектурі кластерів Інституту кібернетики імені В.М. Глушкова СКІТ-1 та СКІТ-2.
9. Программирование на параллельных вычислительных системах / Под ред. Р. Бэбба II. - М.: Мир, 1991. - 372 с.
10. *Немнюгин С.А. Стесик О.Л.* Параллельное программирование для многопроцессорных вычислительных систем.– СПб: БХВ - Петербург, 2002. – 400 с.