УДК 681.3.07

Розпаралелення процесу розв'язування задачі Коші у комп'ютерних мережах

Богдан Квятковський[1], Іван Хвищун[2], Любомира Кіт[3]

[1] Львівський національний університет імені Івана Франка, вул. ген. Тарнавського, 107, Львів, 79000
e-mail: MyBohdan@ukr.net
[2] к. т. н., доцент, Львівський національний університет імені Івана Франка, вул. ген. Тарнавського, 107, Львів
[3] Центр математичного моделювання ІППММ ім. Я. С. Підстригача НАН України, вул. Дж. Дудаєва, 15, Львів

Досліджується проблема розпаралелення розв'язування задачі Коші. Розроблено алгоритм і програму багатопроцесорної обчислювальної системи з розподіленою пам'яттю. Сформульовано критерій, на основі якого встановлено доцільність динамічного перерозподілу підзадач між обчислювальними вузлами. Запропоновано алгоритм взаємодії персональних комп'ютерів (ПК), об'єднаних у локальну мережу, у процесі міграції задач. Проведено аналіз результатів роботи розробленої програми.

Ключові слова: розпаралелення, декомпозиція, міграція задач, задача Коші.

Вступ. Покращення ефективності методів розв'язування складних рівнянь, які використовують для опису довільних процесів у математичних моделях, призводить до пошуку нових підходів [1, 2]. Одним із способів підвищення швидкодії процесу розв'язування рівнянь є розпаралелення обчислень, ефективність якого залежить від вибору архітектури системи для розпаралелення процесу розрахунків і математичного алгоритму розв'язування задачі. Для цього доцільно використати розповсюджену багатопроцесорну систему з розподіленою пам'яттю. З огляду на часові затрати на комунікаційні операції між вузлами таких систем потрібно використовувати алгоритми, які дозволяють зменшити кількість шляхів передачі інформації мережею.

1. Постановка задачі

Для дослідження ефективності алгоритму розпаралелення процесу розв'язування задач використовується локальна мережа, вузлами якої є ПК на основі процесорів Intel Pentium 4, об'єднані комутатором FastEthernet. У процесі реалізації алгоритму розпаралелення процесу розв'язування задач використано бібліотеку MPICH [3]. Оцінку ефективності роботи програми можна зробити за результатами, отриманими при розв'язуванні систем звичайних диференціальних рівнянь із відповідними крайовими умовами

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},t), \quad \mathbf{x}(t_0) = \mathbf{x_0}, \quad t \in [t_0, t_k], \tag{1}$$

де $\mathbf{x}$ — вектор невідомих змінних, а $\mathbf{x}_0$ — його початкове значення, $\mathbf{f}(\mathbf{x},t)$ — вектор-функція, періодична по часовій змінній t; t_0, t_k — початкове та кінцеве значення часового інтервалу.

Алгоритм розв'язування системи (1) базується на методі роздільного інтегрування [4, 5]. Цей метод дає можливість незалежно розв'язувати на заданому інтервалі підзадачі, які є складовою частиною математичної моделі. При побудові алгоритму систему звичайних диференціальних рівнянь доцільно подати сукупністю підсистем

$$\dot{\mathbf{x}}_i = \mathbf{F}_i(\mathbf{x}_i, t, \mathbf{v}), \quad i = \overline{1,n}, \tag{2}$$

які пов'язані додатковим рівнянням, записаним відносно змінних зв'язку $\mathbf{v}$

$$\mathbf{G}(\mathbf{x}, \mathbf{v}) = 0. \tag{3}$$

Ефективність реалізованої за таким алгоритмом програми залежить від збалансованості та взаємодії між обчислювальними вузлами. Тому важливо встановити оптимальні алгоритми декомпозиції системи та міграції задач. Міграція полягає у застосуванні вузлів, у яких обчислення на даному часовому інтервалі уже завершені, для розв'язування підзадач з інших обчислювальних вузлів.

2. Алгоритм розпаралелення процесу розв'язування задачі Коші

Програма для розв'язування задачі Коші з використанням багатопроцесорних обчислювальних систем із розподіленою пам'яттю побудована на основі алгоритму роздільного інтегрування [4, 5].

Першочергово в алгоритмі здійснюється тестування мережі, за результатами якого формується таблиця, у якій відображені часові параметри обміну даними мережею. Потреба у таких даних зумовлена виникненням додаткових комунікаційних операцій під час перерозподілу підсистем задачі. Ця інформація також використовується під час міграції задач. Оскільки можуть реалізуватися ситуації, за яких часові затримки, спричинені обміном даними, переважатимуть виграш від міграції, то алгоритм повинен передбачати перевірку доцільності таких дій. Динамічне тестування параметрів мережі можна замінити зчитуванням готових даних із файла, якщо параметри системи залишаються незмінними.

Декомпозиція системи звичайних диференціальних рівнянь здійснюється на кожному часовому інтервалі. Невідомі групуються у підсистеми відповідно до швидкості зміни з часом.

Під час розпаралелення процесу розв'язування задачі на кожному вузлі розв'язується одна або декілька підсистем сформульованої задачі. Після досягнення точки узгодження змінними зв'язку подається запит на міграцію. Процес, який отримує такий запит, перевіряє доцільність міграції. Перерозподіл підзадач між процесами відбувається за виконання такої умови

$$\sum_i N_{S_i} N_{N_i} t_{N_i} > t_M + \sum_j N_{S_j} N_{N_j} t_{N_j}, \tag{4}$$

де N_{S_i} — передбачена кількість кроків числового інтегрування до точки узгодження змінними зв'язку *i*-ої підсистеми; N_{N_i} — передбачена кількість ньютонівських ітерацій на кожному кроці числового інтегрування *i*-ої підсистеми; t_{N_i} — час, передбачений на виконання однієї ньютонівської ітерації в *i*-ій підсистемі; t_M — час, який затрачається на обмін даними мережею, встановлений на основі величин, отриманих під час тестування мережі.

Умова (4) містить час, необхідний для виконання підзадач у процесі, що отримав запит без міграції, сумарний час, потрібний для виконання частини підзадач у вузлі, що послав запит, і час, затрачений на обмін інформацією між вузлами. Передбачити кількість кроків інтегрування для забезпечення умови (3) можна з допомогою співвідношення для адаптивної зміни кроку

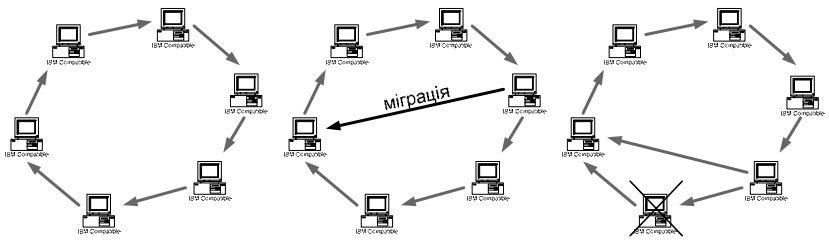
$$N_S = \ln\left(\frac{T_{cor}}{h} + 1\right) \Big/ \ln 2\,, \tag{5}$$

де T_{cor} — відстань до точки узгодження змінними зв'язку; h — поточна величина кроку інтегрування.

Кількість ньютонівських ітерацій на один крок і час виконання однієї ньютонівської ітерації визначаються як середнє значення цих величин на кількох останніх кроках, упродовж яких склад підзадачі не змінювався.

Для підвищення ефективності взаємодії паралельних процесів під час міграції задач запит може поширюватися лише вздовж кільця (рис. 1а). Якщо процес отримав запит на перерозподіл підзадач і згідно з умовою (4) встановив недоцільність такої дії, то запит передається далі вздовж кільця наступному процесові.

Таким чином, просуваючись від вузла до вузла логічно вздовж кільця, запит «шукає» ті підзадачі, для розв'язування яких потрібна допомога. Якщо процес «дає згоду» на міграцію, то передача даних відбувається у напрямку «процес відправник – процес одержувач» (рис. 1б). Якщо ж позитивної відповіді не знайдено та запит повернувся до свого відправника, то процес переходить у режим очікування і вибуває з логічного кільця (рис. 1в).



а) б) в)

Рис. 2. Схема взаємодії обчислювальних вузлів

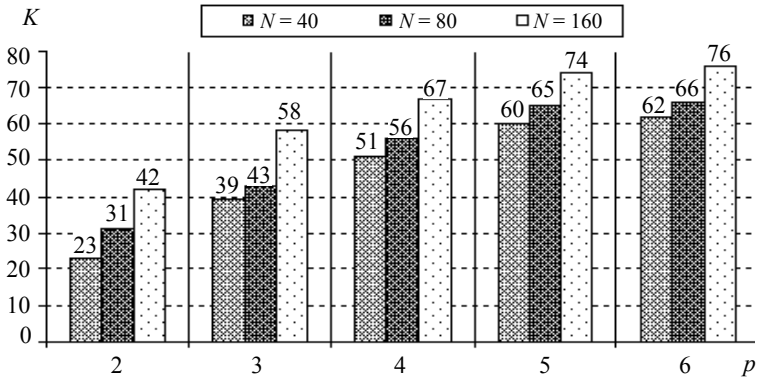


Рис. 2. Коефіцієнт ефективності розпаралелення без застосування міграції задач

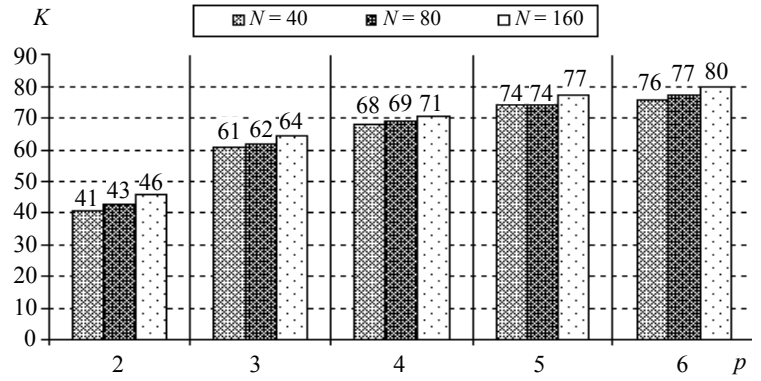


Рис. 3. Коефіцієнт ефективності розпаралелення з застосуванням міграції задач

3. Результати досліджень

Для оцінки ефективності процесу розпаралелення під час розв'язування задачі Коші використовується таке співвідношення

$$K = \frac{T - T_p}{T} 100\%, \tag{6}$$

де T — час виконання програми на одному ПК; T_p — час виконання програми на p ПК.

Ефективність запропонованого алгоритму при розв'язуванні системи звичайних диференціальних рівнянь розмірності N ілюструють діаграми на рис. 2 та 3.

Висновки. Отримані результати показують, що застосування запропонованого алгоритму підвищує ефективність процесу розв'язування задачі Коші. Міграція задач дозволяє досягнути рівномірнішого завантаження обчислювальних вузлів, тому характеристики підвищення швидкодії наближаються до максимально можливих.

Література

[1] *Skelboe S.* Adaptive partitioning techniques for ordinary differential equations // BIT Numerical Mathematics. — 2006. — Vol. 46, № 3. — P. 617-629.

[2] *Petcu D.* Parallelism in solving ordinary differential equations. — Mathematical Monographs 64, Tipografia Universitatii din Timisoara, 1998. — 232 p.

[3] *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. — СПб.: БХВ-Петербург, 2004. — 599 с.

[4] *Stakhiv P.,Rendzinyak S.* A parallel multirate algorithm for the numerical integration of system of nonlinear differential equations // Computing. — 2005. — Vol. 4, Issue 1. — P. 34-41.

[5] *Verhoeven A., Guennouni A. El., Maten E. J. W., Mattheij R. M. M.* A general compound multirate method for circuit simulation problems // Scientific computing in electrical engeneering, series mathematics in industry, ECMI. — 2006. — Vol. 9. — P. 143-150.

Parallelization of the Cauchy problem solving process in computer network

Bogdan Kviatkovsky, Ivan Khvyshchun, Lyubomyra Kit

The problem of the Cauchy problem solving process has been researched. An algorithm and corresponding software for multiprocessor computing system with distributed memory has been developed. A criterion for dynamic spreading of problems between computing nodes of a multiprocessor system is proposed. Algorithm for communication between the nodes of the system during the process of tasks migration is highlighted. An analysis of results obtained with use of developed software is presented.

Распараллеливание процесса решения задачи Коши в компьютерных сетях

Богдан Квятковский, Иван Хвыщун, Любомыра Кит

Рассмотрена проблема параллельного решения задачи Коши. Разработан алгоритм и программа на многопроцессорной вычислительной системе с распределённой памятью. Сформулировано критерии, на основании которых установлена целесообразность динамического перераспределения подзадач по вычислительным узлам. Предложен алгоритм взаимодействия персональных компьютеров (ПК), объединённых в локальную сеть в процессе миграции задач. Проведено анализ полученных результатов.

Отримано 31.10.07