ТЕОРІЯ ОПТИМАЛЬНИХ РІШЕНЬ

УДК 519.6

О.М. ХІМІЧ, О.В. ПОПОВ, О.В. ЧИСТЯКОВ

ПРО РОЗРОБКУ ПАРАЛЕЛЬНИХ АЛГОРИТМІВ ДЛЯ НОВІТНІХ ПРОЦЕСОРІВ INTEL XEON PHI

Розглянуто особливості реалізації ефективних алгоритмів для комп'ютерів з новітніми процесорами Intel Xeon Phi. Запропоновано паралельний алгоритм розв'язування часткової узагальненої алгебраїчної проблеми власних значень із стрічковими симетричними матрицями. Приведено результати апробації розробленого алгоритму.

Вступ. Математичне моделювання на сьогодні є актуальним, інноваційним напрямком у різних предметних областях: аеродинаміка, ядерна енергетика, економіка, медицина, будівництво тощо. Безперервне зростання параметрів задач, необхідність комп'ютерних досліджень більш повних моделей об'єктів і процесів є могутнім стимулом зростання продуктивності комп'ютерів. На сьогодні такими комп'ютерами є паралельні комп'ютери різної архітектури, зокрема, суперкомп'ютери з новітніми багатоядерними процесорами Intel Xeon Phi, які використовуються як потужні співпроцесори [1] або як основні процесори [2]. Тобто процесор Intel Xeon Phi в паралельному комп'ютері – це хост-процесор, який розроблений для забезпечення високої ефективності розпаралелення обчислень.

В останньому рейтингу TOP 500 (за листопад 2017 р.) чотири комп'ютери з процесорами Intel Xeon Phi (різного типу) займають 2-у, 7-у – 9-у позиції [3].

Для ефективного розв'язування задач на цих комп'ютерах необхідно створювати алгоритми та програми, які враховують як властивості математичної моделі, так і архітектурні та технологічні особливості комп'ютера.

Особливості створення ефективних паралельних алгоритмів для розв'язування задач на комп'ютерах з хост-процесором Intel Xeon Phi. Розглянемо деякі способи підвищення ефективності паралельних алгоритмів для комп'ютерів з новітніми процесорами Intel Xeon Phi на прикладі чисельного розв'язування часткової узагальненої алгебраїчної проблеми власних значень – одного з важливих класів задач, який виникає при математичному моделюванні процесів різної фізичної природи. Ефективним алгоритмом вважаємо алгоритм, що забезпечує розв'язування задачі з гарантованою точністю результатів при мінімальному використанні обчислювальних ресурсів та часу.

Обчислювальна схема алгоритму методу ітерацій на підпросторі. Розглянемо метод ітерацій на підпросторі для обчислення r мінімальних власних значень і відповідних їм власних векторів задачі

$$Ax = \lambda Bx, \tag{1}$$

де A, B – симетричні стрічкові додатно визначені матриці порядку n.

Цей метод є узагальненням методу обернених ітерацій і полягає у побудові для задачі (1) послідовності підпросторів E_t ($t = 1, 2, ...$), яка збігається до підпростору E_∞, що містить шукані власні вектори [4]. На t-ій ітерації обчислюється ортогональний базис підпростору E_t і, якщо досягнута необхідна точність наближеного розв'язку, визначаються шукані власні пари.

Таким чином, реалізація методу ітерацій на підпросторі задачі (1) із стрічковими матрицями зводиться до виконання для $t = 1, 2, ...$ таких кроків [5, 6]:

– знаходження розв'язку системи лінійних алгебраїчних рівнянь (СЛАР)

$$AX_t = Y_{t-1}; \tag{2}$$

– обчислення проекції матриці A на підпростір E_t

$$A_t = X_t^T Y_{t-1} \equiv X_t^T A X_t; \tag{3}$$

– обчислення прямокутної матриці

$$W_t = BX_t; \tag{4}$$

– обчислення проекції матриці B на підпростір E_t

$$B_t = X_t^T W_t \equiv X_t^T B X_t; \tag{5}$$

– розв'язування повної проблеми власних значень для проекцій

$$A_t Z_t = B_t Z_t \Lambda_t; \tag{6}$$

– обчислення наближення

$$Y_t = W_t Z_t. \tag{7}$$

Якщо після c ітерацій виконуються умови закінчення ітераційного процесу, наприклад, $\left|\frac{\lambda_i^{(c)} - \lambda_i^{(c-1)}}{\lambda_i^{(t)}}\right| \le \varepsilon$, то проводиться додаткова ітерація і наближеними розв'язками задачі (1) вважаються $\lambda_i^* = \lambda_i^{(c+1)}$, $(i = 1, 2, \ldots, r)$ та перші r стовпчиків матриці $X^* = X_{c+1} Z_{c+1}$ (мається на увазі, що власні значення упорядковані за зростанням $0 < \lambda_1 \le \lambda_2 \le \ldots \le \lambda_r \le \ldots$).

З роботи [4] відомо, що ітераційний процес збігається лінійно, причому швидкість збіжності λ_i визначається відношенням λ_q / λ_1, де $q -$ розмір підпростору E_q, що ітерується. В послідовних реалізаціях алгоритму рекомендується вибирати $q = \min(2r, r+8)$.

Аналізуючи схему реалізації методу ітерацій на підпросторі (2) – (7) можна відзначити, що найбільш затратною щодо використання обчислювальних ресурсів та комп'ютерного часу є підзадача розв'язування СЛАР одним із алгоритмів методу Холецького [5], наприклад, на основі LL^T-розвинення. Оскільки на кожній ітерації виконується розв'язування СЛАР (2) з однією і тією ж матрицею A, то LL^T-розвинення виконується до початку ітераційного процесу. Тоді на кожній ітерації розв'язується задача (2) з факторизованою матрицею.

Розглянемо деякі фактори, які можуть впливати на ефективність паралельного алгоритму для хост-прорцесора Intel Xeon Phi.

Схема декомпозиції даних між процесорами. Можуть бути паралельні комп'ютери з декількома Intel Xeon Phi, кожен з яких має окрему пам'ять, а зв'язки між процесорами здійснюються через деякі комунікаційні засоби. В цьому випадку розглядається комп'ютер з розподіленою пам'яттю. Для розпаралелення обчислень та даних задачі між процесорами на такому комп'ютері доцільно використовувати системy MPI [7].

Для розв'язування задачі (1) на паралельному комп'ютері з декількома хост-процесорами Intel Xeon Phi матриця A розділяється на квадратні блоки $A_{I,J}$ порядку s. Елементи головної діагоналі та нижнього або верхнього трикутника (в залежності від використаних схем алгоритмів) ненульових блоків стрічкової симетричної матриці розподіляються між процесорами у відповідності з одновимірною блочно-циклічною схемою [5, 6].

За цією схемою блок $A_{I,J}$ зберігається в процесі з логічним номером $(I + l) \bmod p$ (результат операції $k \bmod j$ – залишок від ділення k на j, $-1 \le l \le p-2$ – зсув, зазвичай $l = -1$).

В результаті виконання LL^T-розвинення матриці A задачі (1) блоки нижньої трикутної матриці L або верхньої трикутної матриці L^T будуть розподілені аналогічно.

Така ж блочно-циклічна схема розподілу використовується для елементів матриці B та прямокутних матриць ітерованих векторів X_t, Y_t, W_t на етапах (3 – 7) обчислювальної схеми. При цьому достатньо розподіляти та зберігати лише ненульові елементи матриці B в такій послідовності: піддіагональні, діагональні та наддіагональні. Це значно спрощує алгоритм перемноження такої матриці на прямокутну матрицю, не суттєво збільшуючи загальний обсяг даних [5].

Урахування архітектурних та технологічних особливостей новітнього процесора Intel Xeon Phi. Багатоядерний процесор Intel Xeon Phi, що розглядається, став першим завантажувальним хост-процесором компанії Intel, який спеціально розроблений для розпаралелювання задач надвеликого розміру, і в якому вперше інтегровані пам'ять і технології комутації. Тому, на відміну від існуючих процесорів, які використовуються як співпроцесори, та GPU-прискорювачів, його функціональні можливості не обмежені комунікаційними затримками.

Завдяки відсутності таких обмежень новітні процесори Intel Xeon Phi забезпечують велику ефективність і масштабованість, здатні працювати в різноманітних конфігураціях. Цей процесор є третім поколінням процесорів на базі архітектури Intel MIC та другим у поколінні Xeon Phi (Intel Xeon Phi x200 Family) під кодовою назвою Knights Landing.

Розглянемо його структуру та архітектуру (рисунок) [2].

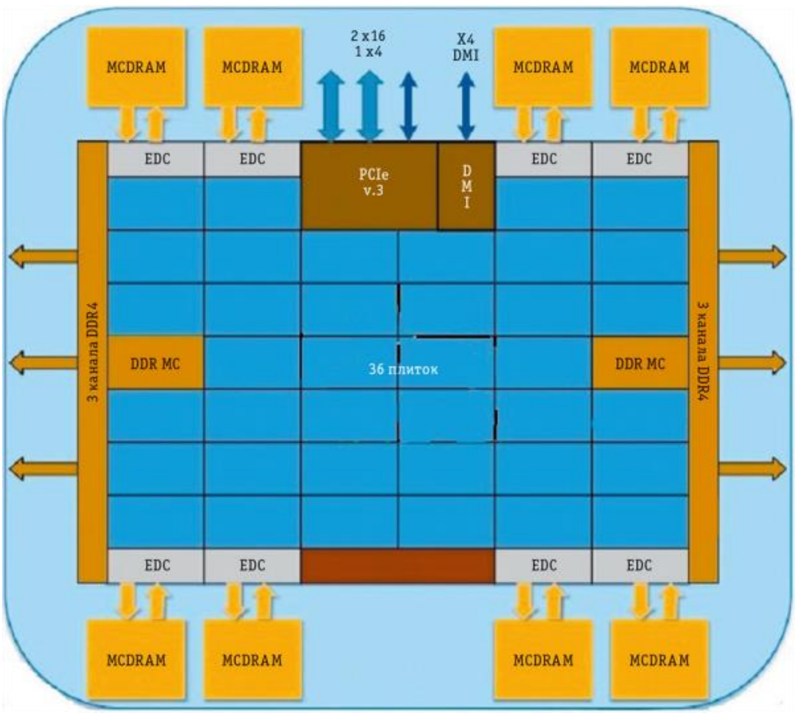

РИСУНОК. Архітектура процесора Knights Landing

Процесор Knights Landing має у своєму складі до 36 так званих плиток (tiles) із топологією зв'язків між ними – «двовимірна решітка». Кожна «плитка» – два ядра Intel Atom Airmont (14 НМ версія Silvermont) з двома VPU (векторними процесорними пристроями) – AVX512 для роботи з числами у форматі з плаваючою комою подвійної точності.

Кожне ядро має кеші команд і даних по 32 Кбайт з додатковим кешем другого рівня місткістю 1 Мбайт, що розділяється між ядрами плитки. У мікросхемі забезпечується когерентність кеша другого рівня для всіх ядер з загальною місткістю до 36 Мбайт. Одне таке ядро припускає одночасне використання чотирьох ниток, або тредів (потоки команд, що виконуються одночасно). До складу процесора входять також вісім модулів «ближньої» пам'яті MCDRAM (Multi-Channel DRAM) загальною місткістю 16 Гбайт та пропускною здатністю більше 400 ГБайт/с, що мають доступ до плитки.

Крім того, є ще можливість для звернення до «далекої» пам'яті DDR4 2400 місткістю до 384 Гбайт і пропускною здатністю більше 90 Гбайт/с. «Ближня» пам'ять може працювати в трьох різних режимах: як кеш «далекої» пам'яті (Cache Mode); у складі єдиного адресного простору з «далекою» пам'яттю (Flat Mode); у комбінованому режимі (Hybrid Mode), коли частина MCDRAM використовується як кеш, а частина – в єдиному адресному просторі з оперативною пам'яттю DDR4.

На основі проведеного аналізу архітектурних можливостей хост-процесора Intel Xeon Phi можна визначити такі способи підвищення ефективності виконання обчислень: розпаралелення обчислень між ядрами процесора, векторизація обчислень, використання кеш-пам'яті різних рівнів.

Як показали дослідження, для розпаралелення обчислень між ядрами із спільною пам'яттю процесора Intel Xeon Phi найбільш придатною є система OpenMP [8].

В архітектурі процесора Intel Xeon Phi другого покоління передбачено використання системи команд AVX512 з підтримкою векторних регістрів. Це дає змогу ефективно реалізувати векторизацію при виконанні скалярних обчислень, що сприяє швидкодії виконання обчислень. В системі OpenMP версії 4.0 [8] є функціональні можливості для векторизації обчислень на цих процесорах. У програмах, які реалізують паралельний алгоритм методу ітерацій на підпросторі, матрично-векторні операції виконуються з векторизацією.

Крім того, для виконання матрично-векторних операцій можна використовувати відповідні функції версії бібліотеки програм Intel MKL (Math Kernel Library) [9], яка адаптована компанією Intel для використання на новітніх процесорах Intel Xeon Phi.

Коротко опишемо як виконуються обчислення на процесорі Intel Xeon Phi за допомогою програм з Intel MKL. При першому виклику якої-небудь функції бібліотеки відбувається перевірка апаратних можливостей комп'ютера та вибирається варіант коду, який дає максимально ефективне використання паралелізму SIMD-команд і регістрів, а також вибирається стратегія роботи з надшвидкою пам'яттю MCDRAM. Причому функції Intel MKL коректно працюють при одночасному виклику з декількох потоків. Таким чином, використання відповідних функцій з бібліотеки програм Intel MKL для реалізації математичних операцій над матрицями та векторами забезпечує ефективне використання пам'яті під час обчислень.

Алгоритмічно-програмне забезпечення, яке розглядається, апробовано на низці тестових та практичних задач, зокрема, для дослідження стійкості композитних матеріалів при експлуатаційних навантаженнях і технологічних процесах виготовлення волокнистих і шаруватих композитних матеріалів [10], що зводиться до розв'язування часткової узагальненої алгебраїчної проблеми власних значень (АПВЗ).

Проведемо порівняльну часову характеристику розв'язування АПВЗ для стрічкових матриць: порядок матриць – 12282; напівширина стрічки матриці A – 6212 та матриці B – 71; обсяг пам'яті – 2 Gb, використовуючи такі комп'ютери: інтелектуальна робоча станція кластерної архітектури Інпарком_256 [11] – 32 вузли, по два Xeon 5606 (4 ядра), 8 Gb RAM на кожен вузол; персональний багатоядерний суперкомп'ютер з графічними процесорами Інпарком_pg [12] – 1 вузол, два Xeon 5606 (4 ядра), 24 Gb RAM, 2 GPU Tesla K40; персональний суперкомп'ютер Інпарком_phi з новітнім хост-процесором Intel Xeon Phi [13] – 1 вузол, Xeon Phi 7210 (64 ядра, 16 Gb MCDRAM), 192 Gb RAM.

Час розв'язування задачі на Інпарком_256: послідовним алгоритмом – 5 хв. 40 сек; паралельним алгоритмом [5], використовуючи 4 вузли (8 процесорів по 4 ядра) час розв'язування – 3 хв. 13 сек; на Інпарком_pg гібридним алгоритмом [6] при використанні одного GPU – 0 хв. 18 сек, при використанні двох GPU – 0 хв. 10 сек; на Інпарком_phi (64 ядра), паралельним алгоритмом – 0 хв. 7 сек.

Отже, у порівнянні з послідовною версією алгоритму отримано такі прискорення: за паралельним алгоритмом на MIMD-комп'ютері – в 7 раз; за гібридним алгоритмом на багатоядерному комп'ютері з графічними процесорами – в 18 і 33 раз, використовуючи один і два GPU відповідно; за розробленим паралельним алгоритмом для комп'ютера з новітнім Intel Xeon Phi процесором – в 45 раз.

Висновки. В роботі розглянуто особливості реалізації ефективних алгоритмів для комп'ютерів з багатоядерними хост-процесорами Intel Xeon Phi на прикладі запропонованого паралельного алгоритму розв'язання часткової узагальненої алгебраїчної проблеми власних значень для симетричних додатно визначених стрічкових матриць. Проведена апробація розробленого алгоритму при розв'язуванні задачі стійкості конструкцій свідчить про перспективність напрямку підвищення продуктивності комп'ютерного моделювання процесів за рахунок використання новітніх процесорів Intel Xeon Phi.

А.Н. Химич, А.В. Попов, А.В. Чистяков

О РАЗРАБОТКЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ НОВЕЙШИХ ПРОЦЕССОРОВ INTEL XEON PHI

Рассмотрены особенности реализации эффективных алгоритмов для компьютеров с новейшими процессорами Intel Xeon Phi. Предложены параллельный алгоритм решения частичной обобщенной алгебраической проблемы собственных значений с ленточными симметричными матрицами. Приведены результаты апробации разработанного алгоритма.

A.N. Khimich, A.V. Popov, A.V. Chistyakov

ABOUT DEVELOPING PARALLEL ALGORITHMS FOR PROCESSORS INTEL XEON PHI

Features of implementing effective algorithms for computers with the latest Intel Xeon Phi processors are considered and a parallel algorithm for solving a partial generalized algebraic eigenvalue problem with band symmetric matrices is proposed. The results of approbation of the developed algorithm are presented.

Список літератури

1. Intel Xeon Phi Coprocessor System Software Developers Guide, revision 2.03. 2012. P. 201.
2. HC27.25.710Knights–Landing–Sodani–Intel_copy.pdf.
3. TOP 500 – 2017(11). http://www.TOP500.org/
4. Парлет Б. Симметричная проблема собственных значений. М.: Мир, 1983. 318 с.
5. Химич А.Н., Молчанов И.Н., Попов А.В., Чистякова Т.В., Яковлев М.Ф. Параллельные алгоритмы решения задач вычислительной математики. Киев: Наук. думка, 2008. 247 с.
6. Химич А.Н., Попов А.В., Чистяков А.В. Гибридные алгоритмы решения алгебраической проблемы собственных значений с разреженными матрицами. *Кибернетика и системный анализ*. 2017. **53**, № 6. С. 132 – 146.
7. Немнюгин С.А. Параллельное программирование для многопроцессорных вычислительных систем. СПб.: БХВ-Петербург. 2002. 400 с.
8. OpenMP. V. 4.0. http://www.openmp.org/mp-documents/OpenMP4.0.pdf/
9. Intel Math Kernel Library (MKL). Reference Manual https://software.intel.com/en-us/articles/mkl-reference-manual
10. Гузь А.Н., Декрет В.А. Модель коротких волокон в теории устойчивости композитов. LAP. 2015. 315 с.
11. Молчанов И.Н., Мова В.И..Интеллектуальные параллельные компьютеры на графических процессорах для решения научно-технических задач. Праці Міжнародної молодіжної математичної школи «*Питання оптимізації обчислень*» (ПОО-XXXVII). Київ. 2011. С. 121 – 122.
12. Хіміч О.М., Молчанов І.М., Попов О.В. та інші. Інтелектуальний персональний суперкомп'ютер для розв'язування науково-технічних задач. *Наука і інновації.* 2016. 12, № 4. С. 17 – 31.
13. Чистяков О.В. Про деякі особливості розв'язання алгебраїчної проблеми власних значень на паралельних комп'ютерах з процесорами Intel Xeon Phi. Праці Міжнародної наукової конференції *«Сучасна інформатика: проблеми, досягнення та перспективи розвитку»*. Київ. 2017. С. 132 – 146.

Одержано 12.03.2018