Теорія оптимальних рішень

УДК 519.6

О.В. ЧИСТЯКОВ

ПРО ЕФЕКТИВНІСТЬ ОБЧИСЛЮВАЛЬНИХ АЛГОРИТМІВ ДЛЯ КОМП'ЮТЕРІВ ГІБРИДНОЇ АРХІТЕКТУРИ

Розглянуто особливості реалізації ефективного паралельного гібридного алгоритму розв'язування часткової узагальненої алгебраїчної проблеми власних значень із стрічковими симетричними матрицями. Представлено оцінку ефективності алгоритму та проведено апробацію на тестових задачах.

Вступ. Математичне моделювання – це актуальний напрямок 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 – могутній стимул зростання продуктивності комп'ютерів. На сьогодні такими комп'ютерами є паралельні комп'ютери різної архітектури, зокрема суперкомп'ютери гібридної архітектури, що поєднують обчислення на багатоядерних комп'ютерах MIMD-ахітектури (CPU) з прискоренням обчислень на графічних процесорах (GPU).

Для ефективного розв'язування обчислювальних задач на цих комп'ютерах необхідно створювати алгоритми та програми, які враховують як властивості математичної моделі, так і архітектурні та технологічні особливості комп'ютера. Наприклад, ефективність алгоритму можна значно покращити за рахунок багатопоточного виконання матрично-векторних операцій з великими обсягами даних на графічних процесорах синхронно з копіюванням масивів даних від CPU до GPU та назад.

Крім того, час розв'язування задач значною мірою залежить від правильності обраних параметрів запуску програм, що реалізують розроблені алгоритми.

Архітектурні особливості гібридного комп'ютера. Розглядаємо гібридний комп'ютер, до складу якого входить багатоядерний та багатовузловий комп'ютер MIMD-архітектури з декількома графічними процесорами

SIMD-архітектури, кожен з яких має окрему пам'ять, а зв'язки між процесорами CPU здійснюються через деякі комунікаційні засоби [1].

Дворівнева організація пам'яті гібридного комп'ютера передбачає дворівневу MPI+CUDA паралельну реалізацію обчислень. На верхньому рівні розпаралелення здійснюється між розподіленою пам'яттю обчислювальних вузлів, використовуючи міжпроцесорні обміни за допомогою системи розпаралелення MPI, а на нижньому рівні − на GPU за допомогою технології CUDA.

На основі проведеного аналізу архітектурних та технологічних особливостей багатоядерних комп'ютерів з графічними прискорювачами можна визначити такі вимоги щодо розробки ефективних гібридних алгоритмів:

- розподіл вихідної задачі на частини (підзадачі), які можуть бути реалізовані в значній мірі незалежно одна від одної;
- визначення які підзадачі доцільніше виконувати на CPU, а які − на GPU та встановлення інформаційних залежностей між ними;
- створення ефективної топології MIMD-комп'ютера з необхідної кількості процесорних ядер CPU, тобто з виконуваних на них MPI-процесів;
- рівномірне завантаження процесів CPU, що використовуються, та синхронізація обмінів між ними;
- виконання підзадач (математичних операцій), які потребують найбільших затрат комп'ютерного часу, на GPU;
- мінімізація обмінів між процесами CPU та між CPU і GPU;
- масштабованість алгоритму – забезпечення можливості ефективно розв'язувати задачі з використанням різної кількості процесів.

Особливості створення алгоритмів для розв'язування обчислювальних задач на гібридних комп'ютерах. Розглянемо деякі способи підвищення ефективності паралельних алгоритмів для комп'ютерів з графічними процесорами на прикладі чисельного розв'язування задач часткової узагальненої алгебраїчної проблеми власних значень методом ітерацій на підпросторі. Задачі цього класу є одними з фундаментальних ресурсномістких задач, які виникають 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 [2]

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

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

Цей метод – узагальнення 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 [2].

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 [3, 4]:

- знаходження розв'язку 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 [2] 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 зазначити, що найбільш затратною щодо використання обчислювальних ресурсів та комп'ютерного часу є підзадача розв'язування СЛАР алгоритмом методу Холецького [3] 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.

Схема декомпозиції даних між процесорами. Для розв'язування задачі (1) на гібридному комп'ютері з декількома GPU 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 одномірною блочно-циклічною схемою [4, 5].

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 у такій послідовності: піддіагональні, діагональні та наддіагональні. Це значно спрощує алгоритм перемноження такої матриці на прямокутну матрицю, не суттєво збільшуючи загальний об'єм даних.

Реалізація гібридного алгоритму методу ітерацій на підпросторі. На основі проведеного аналізу методу ітерацій на підпросторі та з метою ефективного використання архітектурних і технологічних особливостей гібридного комп'ютера етапи розв'язування задачі (1) розділимо на чотири підзадачі різної обчислювальної складності:

1) формування розподіленої між задіяними процесами на CPU матриці Y_0 початкових векторів, що ітеруються, таких, щоб матриця B_t була додатно визначеною. Оскільки підматриці E_t ітерованих векторів розподілені між процесами блоками рядків, то цю операцію можна виконати в кожному з них без обмінів, наприклад, за алгоритмом, який описано в [2];

2) LL^T-факторизація стрічкової симетричної додатно визначеної матриці A, використовуючи, наприклад, гібридний блочний алгоритм з [3];

3) виконання ітераційного процесу (2) – (7), за яким на кожній ітерації ($t=1,2,\ldots$) обчислення виконуються на процесах CPU за наступною схемою:

а) розв'язування гібридними алгоритмами [4] системи лінійних рівнянь (2) з трикутними матрицями, використовуючи отримане на попередньому кроці LL^T-розвинення матриці A;

б) обчислення прямокутної матриці $W_t = BX_t$, тобто добутку прямокутної матриці на стрічкову матрицю (4), що виконується в такому порядку:

– пересилка в кожний процес CPU з процесу, де вони постійно розташовані, елементів (рядків) прямокутної матриці, що використовуються для обчислення чергових kp рядків матриці W_t;

– обчислення процесами, на відповідних процесорах GPU, часткових сум для елементів чергових *kp* рядків матриці W_t;

– мультиобмін чергових *kp* рядків матриці W_t, тобто мультизбирання рядків часткових сум елементів, яке можна поєднати з їх розподілом між процесами CPU з відповідністю із схемою зберігання.

в) обчислення добутків (3) та (5) прямокутних матриць для формування проекцій матриць A та B на підпростір. Виконуються кожним процесом на відповідних GPU, після чого здійснюється збір результуючих матриць проекцій. Причому обчислення добутків прямокутних матриць можна виконувати асинхронно з іншими обчислювальними операціями або обмінами на CPU;

г) розв'язування повної узагальненої АПВЗ (6) методом Якобі, враховуючи порівняно невеликий порядок матриць проекцій підзадачі, виконуються процесами на CPU без обмінів даними;

д) перевірка умов закінчення ітераційного процесу в кожному процесі CPU;

е) обчислення (7) нової матриці ітерованих векторів Y_t (або матриці наближених власних векторів X^*) виконується на процесорах GPU у відповідності з розподілом даних (обчислюється підматриця матриці Y_t або X^*), причому немає необхідності в обмінах даними між процесорними пристроями;

4) аналіз отриманих результатів за технологією з [5].

Дослідження ефективності гібридного алгоритму методу ітерацій на підпросторі. Визначимо коефіцієнт прискорення розробленого алгоритму.

Час розв'язування задачі за розробленим алгоритмом при використанні одного CPU та відповідного йому GPU становить

$$T_1 = O_1 t_C + O_{1G}\, t_G / n_o + O_{oG} t_{oG} + O_{cG} t_{cG},$$

а час розв'язання тієї ж задачі при використанні p CPU та p GPU є

$$T_p = O_p t_C + O_{pG}\, t_G / n_o + O_o t_o + O_{oG} t_{oG} + O_c t_c + O_{cG} t_{cG},$$

де O_1, O_p – кількість операцій, що виконуються на CPU, O_{1G}, O_{pG} – кількість операцій, що виконуються на GPU, відповідно послідовною та паралельною версією алгоритму; t_C, t_G – середній час виконання однієї арифметичної операції з плаваючою комою на CPU та GPU відповідно; n_o – кількість операцій, які можуть бути одночасно виконані на GPU; t_o, t_{oG} – час, необхідний для обміну одним машинним словом між процесами на CPU або між CPU та GPU відповідно; O_o, O_{oG} – обсяг обмінів (кількість машинних слів), що виконуються одним CPU та GPU відповідно; t_c, t_{cG} – час, необхідний для синхронізації двох CPU або CPU та його GPU відповідно; O_c, O_{cG} – кількість синхронізацій, що виконуються одним CPU та GPU відповідно.

При виконанні дослідження та розв'язування часткової узагальненої АПВЗ для симетричних стрічкових матриць частина етапів (формування матриці початкових ітерованих векторів, LL^T-розвинення матриці A, обчислення оцінок наближеного розв'язку) має фіксовану кількість арифметичних операцій, а для ітераційного процесу (2) – (7) кількість арифметичних операцій пропорційна кількості виконаних ітерацій.

Кількість ітерацій, необхідна для знаходження r мінімальних власних значень, як правило, є величина $O(r)$, причому вона чим менше, тим більший розмір q ітерованого підпростору.

Таким чином, $T_p = T_p^{(F)} + c_I T_p^{(1I)}$, а коефіцієнт прискорення запропонованого алгоритму можна представити у вигляді

$$S_p = \frac{T_1}{T_p} = \frac{T_p^{(F)}}{T_p} S_p^{(F)} + \frac{c_I T_p^{(1I)}}{T_p} S_p^{(It)},$$

де c_I – кількість ітерацій, $T_p^{(F)}$, $T_p^{(1I)}$ – відповідно час розв'язування підзадач з фіксованою кількістю арифметичних операцій та час виконання однієї ітерації на архітектурі з використанням p CPU та p GPU, $S_p^{(F)}$ та $S_p^{(1I)}$ – коефіцієнти прискорення алгоритмів відповідних підзадач.

Далі введемо такі позначення: $T_p^{(LLT)}$ – час виконання LL^T-розвинення матриці A; $T_p^{(SS)}(k)$ – час розв'язування СЛАР виду (2) з k правими частинами (використовуючи обчислене раніше LL^T-розвинення матриці); $T_p^{(Fo)}$, $T_p^{(Io)}(k)$ – відповідно час виконання інших операцій при розв'язуванні підзадач з фіксованою кількістю арифметичних операцій та час виконання однієї ітерації.

Тоді $T_p^{(F)}$, $T_p^{(1I)}$ можна записати у вигляді:

$$T_p^{(F)} = T_p^{(LLT)} + T_p^{(SS)}(q) + T_p^{(Fo)}, \quad T_p^{(1I)} = T_p^{(SS)}(q) + T_p^{(Io)}(q). \tag{8}$$

Ми розглядаємо гібридний алгоритм розв'язування часткової проблеми власних значень стрічкових матриць, тому введемо додаткові позначення: m_A, m_B – напівширина стрічки матриць A та B відповідно, η_B – середня кількість ненульових елементів у одному рядку матриці B, $t_C^{(E)}(d) = t_c + dt_o$ та $t_G^{(E)}(d) = t_{cG} + dt_{oG}$ часи обміну масивом з d подвійних слів між двома CPU та CPU + GPU відповідно.

Лема 1. Для розв'язування СЛАР із стрічковою симетричною додатно визначеною матрицею A розміру $n >> q > r$, $m_A > sp$ гібридним алгоритмом на основі LL^T-розвинення на комунікаційній мережі «гіперкуб» при використанні p CPU та p GPU справедливі такі оцінки часових характеристик:

$$T_p^{(LLT)} \approx \frac{n}{p}\left(\frac{s^2}{3}pt_C + \max\left\{m_A^2\frac{t_G}{n_o}, t_C^{(E)}(sm_A)\frac{p}{s}\log_2 p\right\} + 2psm_A\frac{t_G}{n_o} + t_G^{(E)}(sm_A)\frac{p}{s}\right),$$

$$T_p^{(SS)}(k) = \frac{n}{p}\left(kt_C p\log_2 p + (4m_A + 2ps)k\frac{t_G}{n_o} + (t_C^{(E)}(sk)\log_2 p + 2t_G^{(E)}(sk))\frac{2p}{s}\right).$$

Тут s – розмір блоку матриці A, k – кількість шуканих власних значень.

Лема 2. Для гібридного алгоритму методу ітерацій на підпросторі розв'язування АПВЗ для стрічкових симетричних матриць, за умов $n >> q > r$, $m_A \geq m_B$, $1 \leq \eta_B << m_B$, час виконання обчислень з фіксованою кількістю арифметичних операцій – $T_p^{(Fo)}$ та час виконання однієї ітерації – $T_p^{(Io)}(k)$ оцінюються за формулами:

$$T_p^{(Fo)} = \frac{n}{p}\left(\frac{n + 2(n+2)p\log_2 p}{n}qt_C + (2\eta_B + 6)q\frac{t_G}{n_o}\right) + \left(2t_C^{(E)}(sq)\log_2 p + 2t_G^{(E)}(sq)\right)\frac{n}{s},$$

$$T_p^{(Io)}(k) = \frac{n}{p}\left(\frac{(n+2k)\log_2 p + O(k^2)}{n}pkt_C + (2\eta_B + 4k)k\frac{t_G}{n_o}\right) +$$

$$+\left(t_C^{(E)}(sk)\log_2 p + 2t_G^{(E)}(sk)\right)\frac{n}{s}.$$

Теорема. Для гібридного алгоритму методу ітерацій на підпросторі, за умов $n >> q$, $m_A \geq m_B$, $m_A >> q$, $m_A > sp$, $\eta_B << m_B$ та $t_G^{(m)}(s) < t_C^{(E)}(sm_A)$, для коефіцієнта прискорення справедлива наступна оцінка:

$$S_p \approx \frac{1}{T_p^{(SLAE)} + c_I T_p^{(1I)}}\left(\frac{T_p^{(SLAE)}T_1^{(SLAE)} + c_I T_p^{(F)}T_1^{(1I)}}{T_p^{(F)}}\right),$$

де

$$T_p^{(SLAE)} = T_p^{(LLT)} + T_p^{(SS)},$$

$$T_p^{(F)} = T_p^{(LLT)} + T_p^{(SS)} + T_p^{(Fo)}, \qquad T_p^{(1I)} = T_p^{(SS)} + T_p^{(Io)},$$

$$T_p^{(SLAE)} = \frac{n}{p}\left(\frac{t_G m_A(m_A + 2ps + 4k) + 2t_G psk + pn_o((4pk + m_A)t_G^{(E)} + s^2 t_C)}{n_o}\right),$$

$$T_p^{(F)} = \frac{n}{p}\left(\frac{s^2}{3}pt_C + \frac{t_G(m_A^2 + 2psm_A + 4km_A)}{n_o} + t_G^{(E)}p(2nq + m_A) + 2q(t_C + t_C^{(E)}n)\right),$$

$$T_p^{(1I)} = \frac{n}{p}\left(\frac{pk\log_2 p(2nt_C + kt_C + n^2 t_C^{(E)})}{n} + \frac{t_G(2k + 2m_A + ps)2k + 2n_o pkt_G^{(E)}n}{n_o}\right).$$

На рис. 1 показано залежність прискорення гібридного алгоритму від напівширини стрічки матриці та кількості використаних GPU, що отримано при розв'язуванні АПВЗ для стрічкових симетричних матриць порядку 250 000 з різною напівшириною стрічки, побудованих шляхом дискретизації методом скінченних елементів змішаної крайової задачі для оператора Лапласа в прямокутному паралелепіпеді.

Дослідження проведено на інтелектуальній робочій станції гібридної архітектури Інпарком_g такими технічними характеристиками: CPU – серії Intel(R) Xeon(R) E5606; тактова частота 2.13 GHz; швидкість 4,8 GT/s; кеш-пам'ять 8 Mb; у вузлі – 2 CPU по 4 ядра, Max Memory Size 288 Gb; графічні процесори – Nvidia Tesla M2090; пам'ять 6 Gb.

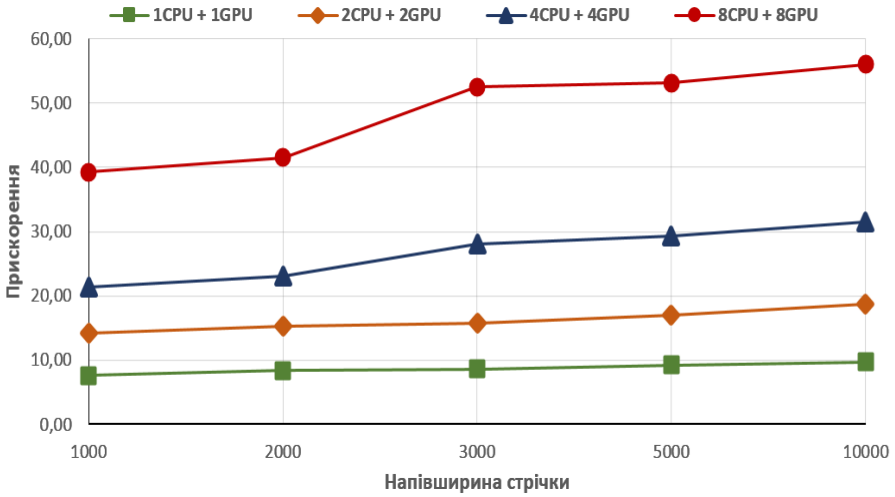

РИС. 1. Залежність прискорення алгоритму від напівширини стрічки матриці

На рисунку ми бачимо, що при збільшенні напівширини стрічки вихідної матриці від 3000 і більше прискорення гібридного алгоритму зростає (кількість використаних GPU – більше двох). При використанні матриці з меншою напівшириною стрічки отримане прискорення набагато менше. Це пов'язано з недостатньою завантаженістю обчислювальних пристроїв та великою кількістю міжпроцесорних обмінів.

На рис. 2 показано результати дослідження алгоритму на Інпарком_g при розв'язуванні АПВЗ для матриці з напівшириною стрічки 10000, використовуючи блоки різного порядку.

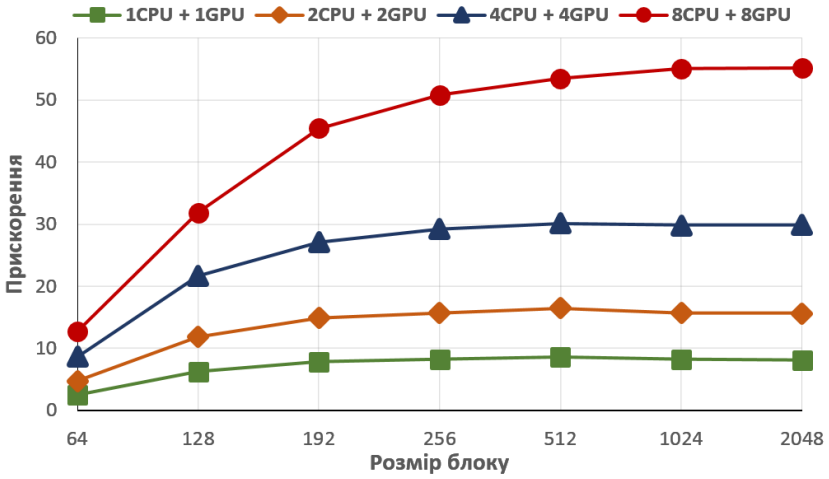

РИС. 2. Залежність прискорення алгоритму від порядку блоків

З представлених графіків на рисунку видно, що при збільшенні розміру блоку до 1024 на гібридному комп'ютері отримано найбільше прискорення. При використанні блоків меншого розміру отримане прискорення набагато менше. Це пов'язано зі збільшенням кількості міжпроцесорних обмінів, неефективним використанням кеш-пам'яті центрального процесора, а також збільшенням кількості викликів програмних функцій матрично-векторних обчислень.

Результати проведених експериментів узгоджуються з теоретичними оцінками коефіцієнтів прискорення та ефективності розробленого гібридного алгоритму.

Висновки. В роботі розглянуто деякі особливості ефективного створення та використання паралельних алгоритмів на комп'ютерах гібридної архітектури, на прикладі запропонованого гібридного алгоритму методу ітерацій, на підпросторі розв'язання часткової узагальненої алгебраїчної проблеми власних значень для симетричних додатно визначених стрічкових матриць.

Проведено теоретичне дослідження ефективності розробленого алгоритму та експериментальна апробація при розв'язуванні тестових задач різного порядку та з різними параметрами запуску. Встановлено залежність результуючої продуктивності від ефективного використання архітектурних особливостей гібридного комп'ютера, схеми декомпозиції даних між процесорами, а також багатопоточного виконання матрично-векторних операцій з великими обсягами даних на графічних процесорах синхронно з копіюванням масивів даних від CPU до GPU та назад.

А.В. Чистяков

ОБ ЭФФЕКТИВНОСТИ ВЫЧИСЛИТЕЛЬНИХ АЛГОРИТМОВ ДЛЯ КОМПЬЮТЕРОВ ГИБРИДНОЙ АРХИТЕКТУРЫ

Рассмотрены особенности реализации эффективных алгоритмов для компьютеров гибридной архитектуры, предложен эффективный гибридный алгоритм решения частичной обобщенной алгебраической проблемы собственных значений с ленточными симметричными матрицами, представлена оценка коефициента ускорения алгоритма. Приведены результаты апробации разработанного алгоритма.

A.V. Chistyakov

ABOUT EFFICIENCY OF COMPUTING ALGORITHMS FOR COMPUTERS OF HYBRID ARCHITECTURE

The features of the implementation of efficient algorithms for computers of hybrid architecture are considered, an efficient hybrid algorithm for solving a partial generalized algebraic eigenvalue problem with tape symmetric matrices is proposed, and an estimate of the acceleration coefficient of the algorithm is presented. The results of testing the developed algorithm are given.

Список літератури

1. Немнюгин С.А. Параллельное программирование для многопроцессорных вычислительных систем. СПб.: БХВ-Петербург. 2002. 400 с.
2. Парлет Б. Симметричная проблема собственных значений. М.: Мир, 1983. 318 с.
3. Хіміч О.М., Попов О.В., Баранов А.Ю., Чистяков О.В. Гібридний алгоритм розв'язування задач на власні значення для стрічкових матриць *Теорія оптимальних рішень*. К: Ін-т кібернетики імені В.М. Глушкова НАН України. 2016. С. 86 – 94.
4. Khimich A.N., Popov A.V., Chistyakov O.V. Hybrid algorithms for solving the algebraic eigenvalue problem with sparse matrix. *Cybernetics and Systems Analysis.* 2017. Vol. 53, N 6. P. 132 – 146.
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.

Одержано 21.02.2019