Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
https://nv.nltu.edu.ua
https://doi.org/10.36930/40300421
Article received 17.09.2020 р.
Article accepted 03.09.2020 р.
UDC 004.424.5

ISSN 1994-7836 (print)
ISSN 2519-2477 (online)

@ Correspondence author
V. Ya. Antoniv
volodya.antoniv@gmail.com

І. Г. Цмоць, В. Я. Антонів

Національний університет "Львівська політехніка", м. Львів, Україна

УДОСКОНАЛЕННЯ ПАРАЛЕЛЬНОГО СОРТУВАННЯ МАСИВІВ ЧИСЕЛ МЕТОДОМ ЗЛИТТЯ

Удосконалено метод сортування злиттям способом просторового розпаралелення процесу сортування, яке зведено до одночасного отримання елементів зростаючого та спадаючого масивів. Визначено та розглянуто такі основні етапи розроблення потокового графу для паралельного сортування масивів даних з використанням удосконаленого методу злиття, як декомпозиція алгоритму сортування масиву із $m \times N$ чисел, проектування комунікацій між функціональними операторами, укрупнення функціональних операторів, планування процесу сортування масиву із $m \times N$ чисел. Розроблено орієнтований на архітектуру графічного процесора конкретизований потоковий граф паралельного сортування масивів даних з використанням удосконаленого методу сортування злиттям. Запропоновано розробку програмних засобів паралельного сортування масивів даних з використанням удосконаленого методу злиття виконувати на основі комплексного підходу, який охоплює: дослідження, розроблення та вдосконалення алгоритмів і методів паралельного сортування масивів даних; потокові графи алгоритмів паралельного сортування; архітектуру графічного процесора GPU та програмну модель CUDA. Показано, що попри необхідність розроблення додаткових програм для внутрішніх операцій, а саме: створення тимчасових масивів для зберігання проміжних результатів, ядра програми, циклів виконання ядра програми у потоках та блоках, бінарного пошуку, присвоєння ключів записав на фінальній стадії порівняння двох масивів і формування вихідного відсортованого масиву, реалізація на графічному процесорі паралельного сортування масивів даних з використанням удосконаленого методу злиття, забезпечує значне зменшення часу сортування порівняно з використанням тільки центрального процесора.

Ключові слова: потоковий граф; паралельне сортування; метод злиття; алгоритми сортування; попарне порівняння; масив даних; графічний процесор.

Вступ

Сучасний етап розвитку інформаційних технологій характеризується нагромадженням великих масивів даних. Під час опрацювання таких масивів найчастіше доводиться використовувати операції сортування, метою якого є пришвидшення пошуку потрібної інформації. Основним способом підвищення швидкодії сортування великих масивів даних є розпаралелювання процесу сортування та використання масово-паралельних обчислюваних засобів із великим обсягом пам'яті. До таких засобів належать графічні процесори GPU (англ. *Graphics Processing Unit*), які є процесорами класу SIMD (англ. *Single Instruction Multiple Data*). Особливістю SIMD процесорів є те, що в них одна операція використовується одночасно для опрацювання множини незалежних даних. Для розроблення програмного забезпечення для сортування інтенсивних масивів даних, частина якого працює на CPU (центральний процесор), а частина на GPU, доцільно використати кросплатформову систему компіляції та виконання програм CUDA.

Програмна реалізація процесу сортування даних на GPU вимагає вдосконалення та розроблення нових паралельних алгоритмів сортування масивів даних. Паралельні алгоритми сортування масивів даних для реалізації на GPU повинні бути:

- добре структурованими з детермінованим переміщенням даних;
- ґрунтуватися на однотипних операціях з локальними та регулярними зв'язками;
- використовувати конвеєризацію та просторовий паралелізм.

З огляду на це особливої актуальності набуває проблема розроблення та вдосконалення паралельних алгоритмів сортування даних і їх орієнтація на графічні процесори.

Об'єкт дослідження – процеси сортування даних в інформаційних технологіях.

Предмет дослідження – паралельні алгоритми, методи і засоби сортування даних, що дають змогу зменшити час сортування великих масивів даних.

Мета дослідження – удосконалення методу сортування масивів даних злиттям та його орієнтація на архітектуру графічного процесора.

Для досягнення зазначеної мети визначено такі основні *завдання дослідження*:

- вдосконалити метод сортування злиттям і орієнтувати його на паралельне сортування масивів даних;

Інформація про авторів:

Цмоць Іван Григорович, д-р техн. наук, професор, кафедра автоматизованих систем управління. **Email:** tsmots@gmail.com; https://orcid.org/0000-0002-4033-8618

Антонів Володимир Ярославович, аспірант, кафедри автоматизованих систем управління. **Email:** volodya.antoniv@gmail.com; https://orcid.org/0000-0002-4544-4612

Цитування за ДСТУ: Цмоць І. Г., Антонів В. Я. Удосконалення паралельного сортування масивів чисел методом злиття. Науковий вісник НЛТУ України. 2020, т. 30, № 4. С. 134–142.

Citation APA: Tsmots, I. G., & Antoniv, V. Ya. (2020). Improvement of parallel sorting by method of merging. *Scientific Bulletin of UNFU*, *30*(4), 134–142. https://doi.org/10.36930/40300422

- розробити потоковий граф для паралельного сортування масивів чисел з використанням вдосконаленого методу злиття;
- розробити для графічного процесора програму паралельного сортування масивів даних з використанням удосконаленого методу злиття.

Наукова новизна отриманих результатів дослідження полягає в тому, що вдосконалений метод сортування двох масивів злиттям, який завдяки одночасному об'єднанню двох масивів, починаючи як з максимальних, так і мінімальних значень, забезпечує розпаралелення процесу сортування та зменшує у два рази кількості тактів сортування порівняно з послідовним злиттям.

Практична значущість результатів дослідження полягає в тому, що розроблено потоковий граф для паралельного сортування масивів чисел, який в разі використання вдосконаленого методу сортування двох масивів злиттям забезпечує зменшення часу сортування внаслідок ефективного просторового розпаралелення процесу сортування масивів даних. Відповідно, розроблена для графічного процесора GPU програма паралельного сортування масивів даних з використанням удосконаленого методу злиття, яка завдяки використанню апаратно-програмної моделі CUDA і потокового графу алгоритму забезпечує зменшення часу сортування.

Аналіз останніх досліджень та публікацій. Задачі розроблення паралельних методів, алгоритмів і програмних засобів для сортування великих масивів даних з використанням графічних процесорів GPU розглянуто у працях [1, 2, 3].

Аналіз відомих методів сортування [1, 5, 6, 7, 9, 15, 17, 18] показав, що вони ґрунтуються на операціях попарного порівняння та перестановки у разі, коли порядок цих значень не відповідає умовам сортування. Методи сортування на основі таких операцій відрізняються один від одного вибором пар значень для порівняння. Ці методи сортування масивів даних можна об'єднати в такі групи: сортування вставкою, обмінне сортування, сортування злиттям і сортування вибором. Більшість цих методів сортування масивів даних орієнтовані на послідовну реалізацію.

Аналіз літератури [4, 10, 11, 12, 13] показує, що для паралельно-конвеєрної реалізації на графічних процесорах найбільше підходять алгоритми сортування даних методом злиття. В основі алгоритмів сортування даних методом злиття знаходиться базова операція об'єднання двох або більше упорядкованих масивів в один упорядкований масив. Особливістю алгоритмів сортування даних методом злиття є те, що вони є добре структурованими і реалізуються на однотипних операціях попарного порівняння та перестановки.

З аналізу публікацій [4, 12] випливає, що основним методом зменшення часу сортування масиву чисел є просторово-часове розпаралелення процесу сортування. Для орієнтації алгоритмів сортування масивів даних на графічні процесори необхідно, щоб такі алгоритми були подані у вигляді потокових графів, які забезпечують знаходження оптимальних просторово-часових рішень.

У публікаціях і дослідженням [8, 12, 14, 17] недостатньо уваги приділено питанням комплексного підходу до розроблення програмних засобів сортування великих масивів даних на графічних процесорах GPU.

Результати дослідження та їх обговорення

1. ***Розроблення потокового графу паралельного сортування масивів даних методом злиття.*** Для сортування масиву із $m \times N$ чисел пропонуємо використати метод сортування злиттям. На початку сортування вхідний масив із $m \times N$ чисел розбивається на m масивів розміром N чисел. Для сортування масиву із N чисел методом злиття кожне число подається як упорядкований масив розміром одиниця. Сортування такого масиву методом злиття ґрунтується на операціях послідовного об'єднання упорядкованих масивів. Унаслідок виконання першого етапу об'єднання формуються $N/2$ впорядкованих масивів довжиною два. Кількість етапів об'єднання для отримання упорядкованих масивів довжиною N визначають так:

$$p = \lceil \log_2 N \rceil, \tag{1}$$

де $\lceil \ \rceil$ – функція, як визначає більше ціле число.

Сортування m упорядкованих масивів довжиною N будемо виконувати паралельним методом злиття, в основі якого знаходиться базова операція об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$ в один упорядкований масив $\{b_{1i}\}_{i=1}^{2N}$. На першому етапі сортування вхідний масив чисел $\{a_j\}_{j=1}^{mN}$ розбивається на $m/2$ пар упорядкованих масивів довжиною N, які попарно об'єднуються. Таке об'єднання реалізується на основі першого типу макрооперації (базової операції). Унаслідок виконання першого етапу формуються $m/4$ впорядкованих масивів довжиною $2N$. Для виконання другого етапу використовується макрооперація другого типу – об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{2N}$ та $\{a_{2i}\}_{i=1}^{2N}$ в один упорядкований масив $\{b_{1i}\}_{i=1}^{4N}$. Така макрооперація виконується на базі трьох макроопераций першого типу.

Кількість етапів, необхідних для сортування масиву із m масивів чисел, визначають за такою формулою:

$$\mathrm{k} = \lceil \log_2 m \rceil. \tag{2}$$

Кожний s-й етап, де $s = 1,\ldots, k$, реалізується на основі макроопераций s-го типу, кожна з яких виконує об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{2^{s-1}N}$ та $\{a_{2i}\}_{i=1}^{2^{s-1}N}$ в один упорядкований масив $\{b_{1i}\}_{i=1}^{2^{s}N}$. Макрооперація s-го типу ґрунтується на трьох макрооперацiях (s-1)-го типу. Операційною основою всіх макрооперацій є базова операція об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$ в один упорядкований масив $\{b_{1i}\}_{i=1}^{2N}$. Кількість базових операцій, необхідних для виконання макрооперації s-го типу, дорівнює 3^{s-1}.

Алгоритми паралельного сортування масиву із $m \times N$ чисел методом злиття пропонуємо відображати у вигляді потокового графу $F=(\Phi, \Gamma)$, де $\Phi=\{\Phi_1, \Phi_2,\ldots, \Phi_n\}$ – множина функціональних операторів, які реалізують операцію сортування масивів довжиною N і базові операції об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$ в один упорядкований масив $\{b_{1i}\}_{i=1}^{2N}$, Γ- закон відображення зв'язків між операторами. Такі потокові графи забезпечують знаходження оптимальних просто-

рово-часових рішень. Графічно потоковий граф алгоритму паралельного сортування методом злиття масиву із $m \times N$ чисел відображається у вигляді вершин, що відповідають операторам алгоритму Φ_i (сортування масивів довжиною N і об'єднання двох упорядкованих масивів довжиною N водин упорядкований масив) та дуг, які відображають зв'язки між операторами. Кожний функціональний оператор об'єднання двох упорядкованих масивів довжиною N в один упорядкований масив має два входи та два виходи. На кожній вхід послідовно число за числом поступають упорядковані масиви розміром N чисел. На першому та другому виходах функціонального оператора об'єднання послідовно число за числом формується відповідно N максимальних і N мінімальних чисел об'єднаного впорядкованого масиву.

Розроблення потокового графу паралельного сортування масиву із $m \times N$ чисел методом злиття складається з таких чотирьох етапів:

1) декомпозиція алгоритму сортування масиву із $m \times N$ чисел методом злиття;
2) проектування комунікацій між функціональними операторами;
3) укрупнення функціональних операторів;
4) планування процесу сортування масиву із $m \times N$ чисел методом злиття.

На етапі декомпозиції алгоритм паралельного сортування масиву із $m \times N$ чисел методом злиття розбивається на функціональні оператори сортування масивів довжиною N і об'єднання двох впорядкованих масивів довжиною N в один упорядкований масив. Функціональні оператори сортування масивів довжиною N мають один вхід і один вихід, а функціональні оператори об'єднання двох упорядкованих масивів в один мають два входи та два виходи. Для декомпозиції алгоритму сортування масиву чисел методом злиття доцільно використовувати метод функціональної декомпозиції, який забезпечує розбиття алгоритму на функціональні оператори та відображає зв'язки між ними.

На етапі проектування комунікацій необхідно здійснити просторово-часове розміщення функціональних операторів Φ_i і їх закріплення за ярусами. Структура з'єднань у потоковому графі між функціональними операторами Φ_i сусідніх ярусів визначається етапами сортування.

Потоковий граф алгоритму паралельного сортування масиву із $m \times N$ чисел методом злиття наведено на рис. 1, де: $a_{1i} - a_{mi}$ – m входи даних, $\Phi_{01} - \Phi_{0m}$ – функціональні оператори сортування масивів чисел довжиною N, $\Phi_{s1} - \Phi_{s(2m/2^s)}$ – макрооперації s-го типу, які реалізують злиття двох упорядкованих масивів довжиною $2^{s-1}N$ чисел в один довжиною 2^sN чисел.

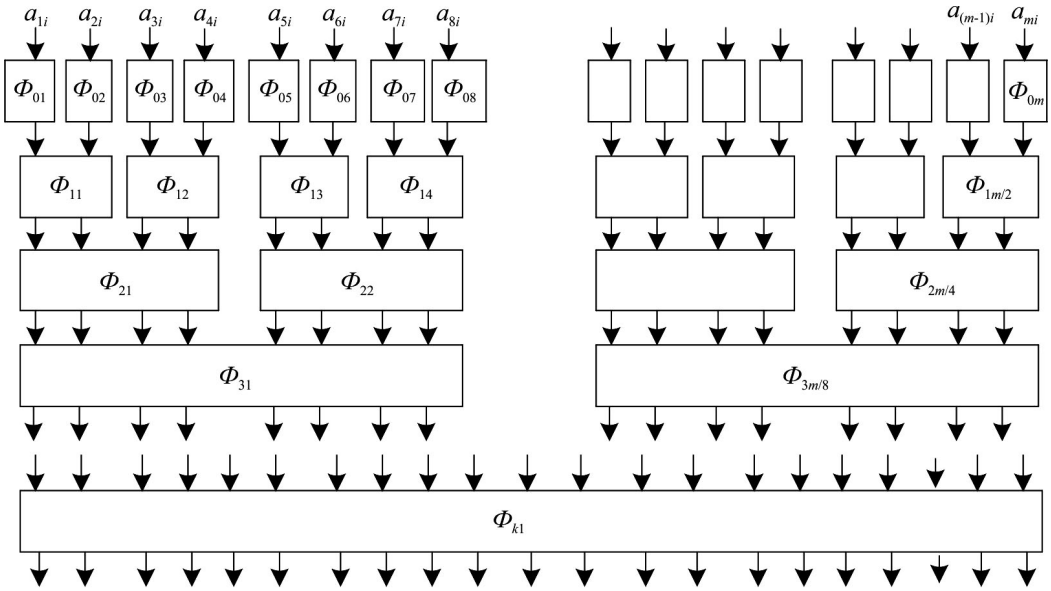

Рис. 1. Структура потокового графу алгоритму паралельного сортування масиву із m×N чисел методом злиття

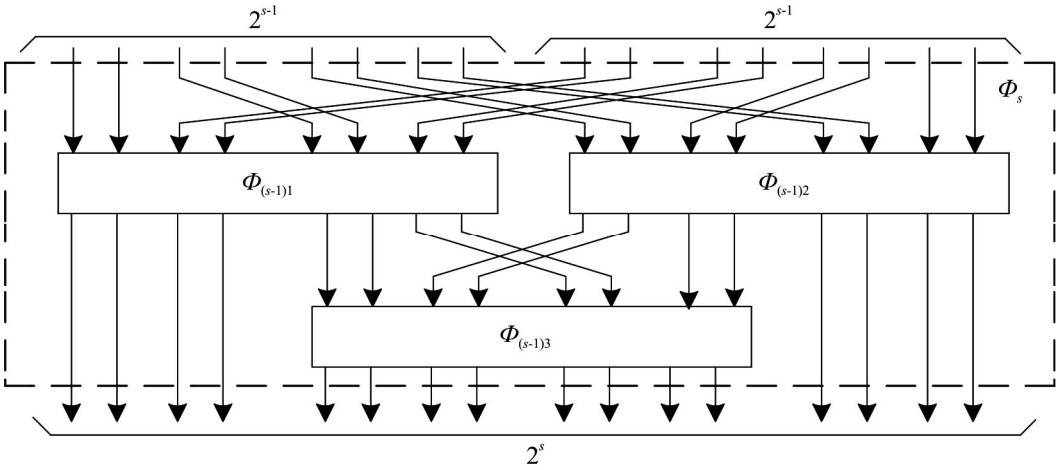

Рис. 2. Структура реалізації макрооперації s-го типу

Потоковий граф алгоритму паралельного сортування масиву із $m \times N$ чисел методом злиття складається із (k + 1) ярусів. У нульовому ярусі виконуються оператори сортування масивів довжиною N чисел, а у s-их ярусах – макрооперація злиття двох упорядкованих масивів довжиною $2^{s-1}N$ чисел в один довжиною 2^sN чисел. У s-их ярусах виконує об'єднання двох упорядкованих масивів довжиною $2^{s-1}N$ чисел в один довжиною 2^sN чисел, яке реалізуються на основні макрооперації s-го типу. Кожна макрооперація s-го типу реалізується на трьох макрооперацiях (s-1)-го типу (рис. 2).

Усі макрооперації s-го типу, які використовуються під час сортування масиву методом злиття, реалізуються на основі базової операції об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$ в один упорядкований масив $\{b_{li}\}_{i=1}^{2N}$. Кількість базових операцій R, яка використовується для реалізації макрооперації s-го типу, визначають так:

$$R = 3^s .$$

Етап проектування полягає в укрупненні операцій за допомогою об'єднання функціональних операторів Φ_{jk} і каналів передачі даних між ярусами. Мета цього етапу полягає в тому, щоб отримати конкретизований потоковий граф, який орієнтований на реалізацію з використанням графічного процесора. Цей етап пов'язаний з плануванням процесу сортування.

На етапі планування процесу сортування відбувається збереження інформації про структуру потокового графа алгоритму паралельного сортування масиву із $m \times N$ чисел методом злиття. На цьому етапі визначаються величини затримок, виконується планування процесу сортування і перестановки даних. Для коректного відображення процесу сортування даних у конкретизований потоковий граф вводяться оператори затримки, управління та перестановки даних. Орієнтований на графічний процесор конкретизований потоковий граф отримуємо шляхом лінійної проекції потокового графу на горизонтальну вісь X. Проекцію конкретизованого графу алгоритму паралельного сортування масиву із $m \times N$ чисел методом злиття наведено на рис. 3, де: Φ_{cl} – функціональний оператор сортування масиву довжиною N чисел методом злиття (l=1,., m), $\Phi_{БОh}$ – функціональний оператор базової операції об'єднання (h=1, ..., m/2) двох упорядкованих масивів довжиною N в один упорядкований масив 2N, $\Phi_{МЗП}$ – макрооператор затримки та перестановки, $\Phi_{МУ}$ – макрооператор управління.

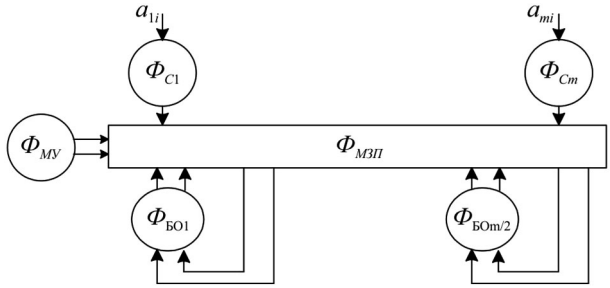

Рис. 3. Лінійна проекція потокового графу алгоритму сортування злиттям на горизонтальну вісь X

Отримана лінійна проекція потокового графу алгоритму сортування злиттям на горизонтальну вісь X орієнтована на архітектуру графічного процесора GPU. З допомогою макрооператора управління $\Phi_{МУ}$ імакрооператора перестановки та затримки $\Phi_{МЗП}$, визначаються величини затримок, перестановки даних і здійснюється планування процесу сортування.

2. *Реалізація оператора послідовного сортування масиву з N чисел методом злиття.* Послідовне сортування вхідного масиву із N чисел методом злиття передбачає розбивається його N упорядкованих масивів довжиною одиниця та виконання операцій послідовного їх об'єднання. Кількість етапів послідовного об'єднання упорядкованих масивів визначається за формулою (2). Операції об'єднання упорядкованих масивів ґрунтуються на елементарних операціях – попарного порівняння та перестановки елементів масиву.

Структуру засобів реалізації послідовного сортування методом злиття вхідного масиву із N чисел наведено на рис. 4, де: a_{li} – вхід даних, P_{1k}, P_{2k} – дані для налаштування відповідно першого та другого лічильників, $ТІ_1$ – перші тактові імпульси, *ПУ* – початкова установка, $ЗпА_1$, $ЗпА_2$ – сигнали запису адреси відповідно у перший та другий лічильники, $У_1$, $У_2$ та $У_3$, $У_4$, – сигнали управління комутаторами відповідно першим і другим, $ТІ_2$ – другі тактові імпульси, $Зп/Чт_1$, $Зп/Чт_2$ – сигнали вибору режиму роботи блоків пам'яті відповідно першого та другого, $ВК_1$, $ВК_2$ –сигнали вибору кристала відповідно першого та другого блоків пам'яті, $У_5$, $У_6$ – сигнали управління відповідно першим та другим шинними формувачами, *РП* – результат порівняння, $У_7$ – сигнали управління третім комутатором, $У_7$, $У_8$ – сигнали управління відповідно третім та четвертим шинними формувачами, БУ – блок управління, ГА – генератор адреси, Рг – регістр, См – суматор, ЛчА – лічильник адреси, Км – комутатор, БП – блок пам'яті, ШФ –шинний формувач, b_{li} – вихід відсортованих чисел.

Перед початком сортування вхідний масив із N чисел необхідно записати в $БП_1$. Для цього комутатори $Км_1$ і $Км_3$ встановлюються відповідно у стан, коли на виходи $Км_1$ надходять дані з виходів лічильника $ЛчА_3$, а на виходи комутатор $Км_3$ надходять дані з першого входу, шинні формувачі $ШФ_1$ і $ШФ_2$ у третій стан, а $ШФ_3$ на передачу даних, а блок $БП_1$ переводиться у режим запису. З БУ надходить сигнал ПУ, який встановлює лічильники $ЛчА_1$-$ЛчА_3$ та регістри $Рг_1$-$Рг_4$ у нуль. Числа масиву послідовно поступають на вхід даних a_{li} і у кожному такті роботи записуються в блок $БП_1$. Після N тактів у $БП_1$ записується масив із N чисел, які необхідно відсортувати.

Для сортування масиву із N чисел необхідно виконати p етапів об'єднання упорядкованих масивів. Кожний v-й етап, де v = 1,…, p, реалізується на основі макрооперації об'єднання v-го типу, яка зводиться до об'єднання двох упорядкованих масивів $\{a_{1i}\}_{i=1}^{2^{v-1}}$ та $\{a_{2i}\}_{i=1}^{2^{v-1}}$ в один упорядкований $\{b_{li}\}_{i=1}^{2^{v}}$. Для v-го етапу об'єднання масивів необхідна кількість таких макрооперацій визначається так:

$$z = \frac{N}{2^v} . \qquad (3)$$

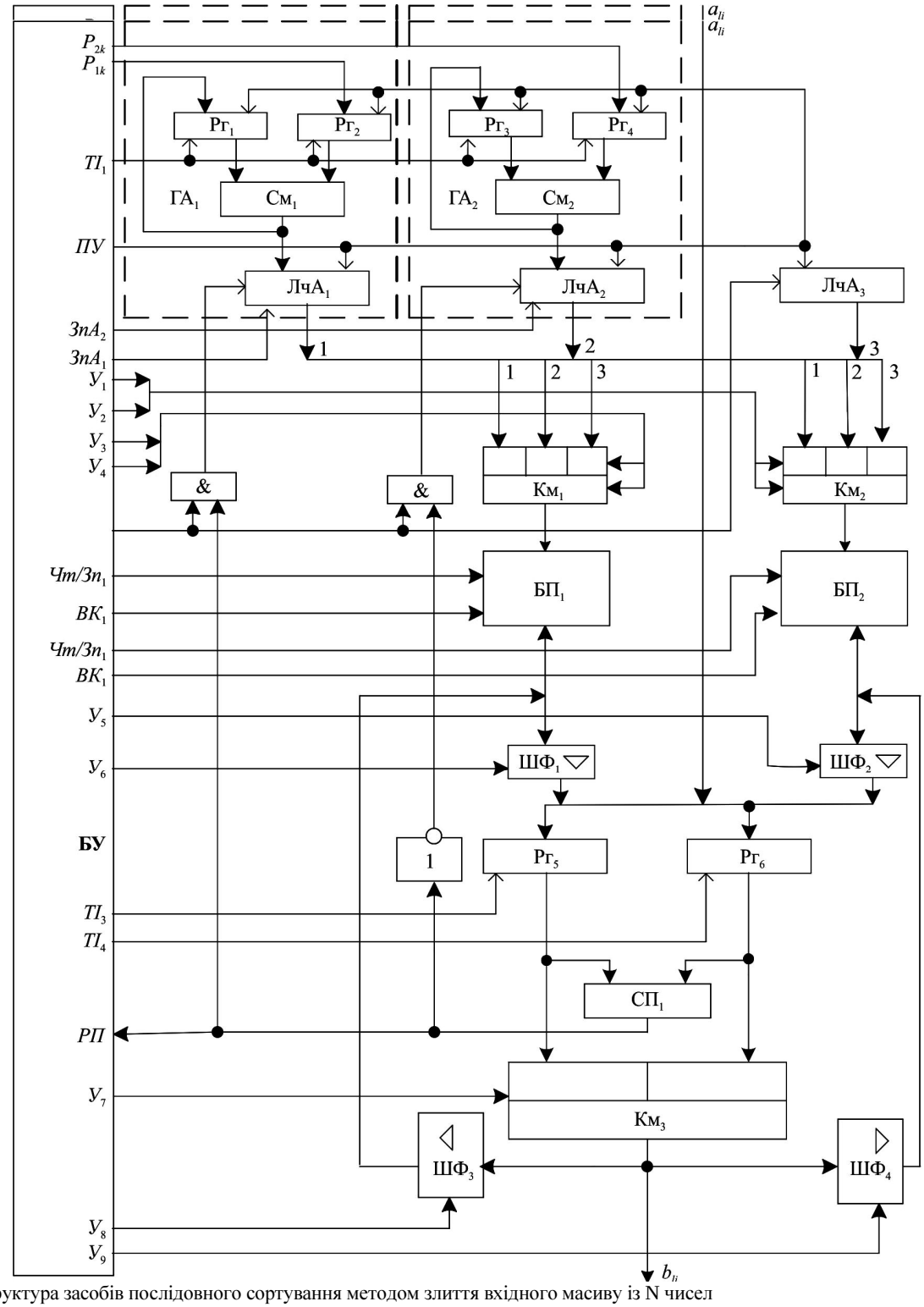

Рис. 4. Структура засобів послідовного сортування методом злиття вхідного масиву із N чисел

Розглянемо алгоритм покрокового виконання *v*-го етапу об'єднання масивів:

1 крок – встановлюємо регістри $Рг_1$-$Рг_4$, лічильники $ЛчА_1$-$ЛчА_3$ у нуль, комутатори $Км_1$ і $Км_2$ на передачу адреси відповідно з перших ($А_1$) і третіх ($А_3$) входів, шинні формувачі $ШФ_1$ і $ШФ_4$ на передачу даних, $ШФ_2$ і $ШФ_3$ у третій стан;

2 крок –записуємо число 2^v в регістри $Рг_2$ і $Рг_4$, а число 2^{v-1} в $Рг_3$ і лічильник $ЛчА_2$;

3 крок – зчитуємо з $БП_1$ найбільше число першого масиву та записуємо в $РГ_5$;

4 крок – переключаємо комутатор $Км_1$ на передачу адреси з других входів ($А_2$), зчитуємо з $БП_1$ найбільше число другого масиву та записуємо в $РГ_6$;

5 крок – порівнюємо числа з регістрів $РГ_5$ і $РГ_6$ і результати порівняння передаємо у БУ, який формує сигнал управління $Км_3$, що забезпечує передачу більшого числа та його запис у $БП_2$;

6 крок – збільшуємо на 1 значення лічильників $ЛчА_3$ та лічильника, за адресом якого було зчитано більше число;

7 крок – встановлюємо комутатор $Км_1$ на передачу адреси з виходів лічильника, за адресом якого було зчитано більше число та зчитуємо за даною адресом з $БП_1$ число, яке записується у регістр, з якого число було записано в $БП_2$;

8 крок і наступні повторюють 5, 6 і 7 кроки до моменту, коли один з лічильників $ЛчА_1$ або $ЛчА_2$ стане дорівню-

вати відповідно (2^{v-1}–1) або (2^v–1), тобто, коли з $БП_1$ повністю зчитаний один із масивів;

Наступні кроки зводяться до читання з $БП_1$ залишку чисел іншого масиву та запис їх $БП_2$.

Далі переходимо до об'єднання наступних двох масивів в один. Таке об'єднання передбачає запис в $ЛчА_1$ числа 2^v а в $ЛчА_2$ числа (2^{v-1}+2^v) та повторення кроків, починаючи 3-го, до запису об'єднаного упорядкованого масиву в $БП_2$. Відсортований масив із N чисел отримуємо при виконанні p-го етапу об'єднання масивів на виході b_{li}.

3. *Удосконалення паралельного сортування методом злиття*. В основі алгоритмів паралельного сортування методом злиття знаходиться базова операція об'єднання двох впорядкованих масивів чисел $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$ в один впорядкований масив $\{a_i\}_{i=1}^{2N}$ чисел. Таке об'єднання виконується за $2N$ тактів. Метою вдосконалення паралельного сортування методом злиття є зменшення кількості тактів, необхідних для отримання відсортованого масиву $\{a_i\}_{i=1}^{2N}$ чисел із двох відсортованих масивів чисел $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$. Для зменшення часу об'єднання двох упорядкованих масивів чисел $\{a_{1i}\}_{i=1}^{N}$ та $\{a_{2i}\}_{i=1}^{N}$ пропонуємо розпаралелити процес об'єднання масивів способом одночасного злиття двох масивів, починаючи як з максимальних, так і мінімальних значень. Відсортований масив із $2N$ чисел отримаємо на двох виходах: на першому виході (N_{1Cmax} найбільших чисел), на другому виході (N_{2Cmin} найменших чисел). Кількість тактів, необхідних для такого об'єднання, дорівнює N, що у два рази менше порівняно з наявними алгоритмами об'єднання.

Базова операція об'єднання двох впорядкованих масивів чисел ґрунтуються на елементарних операціях – попарного порівняння та перестановки елементів масивів. При об'єднанні двох масивів чисел, починаючи з максимальних, використовують такі операції:

$$y_{\max i} = \begin{cases} 0, \text{ коли } a_{1\max i} \le a_{2\max i} \\ 1, \text{ коли } a_{1\max i} > a_{2\max i} \end{cases}, \quad (4)$$

$$a_{C\max i} = \begin{cases} a_{1\max i}, \text{ коли } y_{\max i} = 1 \\ a_{2\max i}, \text{ коли } y_{\max i} = 0 \end{cases}, \quad (5)$$

а при об'єднанні масивів чисел, починаючи з мінімальних, такі:

$$y_{\min i} = \begin{cases} 0, \text{ коли } a_{1\min i} \le a_{2\min i} \\ 1, \text{ коли } a_{1\min i} > a_{2\min i} \end{cases}, \quad (6)$$

$$a_{C\min i} = \begin{cases} a_{1\min i}, \text{ коли } y_{\min i} = 0 \\ a_{2\min i}, \text{ коли } y_{\min i} = 1 \end{cases}, \quad (7)$$

де: $y_{\max i}$, $y_{\min i}$ – результат попарного порівняння відповідно елементів максимальних і мінімальних масивів; $a_{1\max i}$, $a_{2\max i}$ – i-ті елементи порівняння максимальних значень відповідно першого та другого масивів; $a_{1\min i}$, $a_{2\min i}$ – i-ті елементи порівняння мінімальних значень відповідно першого та другого масивів; $a_{C\max i}$, $a_{C\min i}$ – i-ті відсортовані елементи масивів відповідно максимальних чисел і мінімальних чисел.

Структуру засобів реалізації паралельного об'єднання двох масивів чисел наведено на рис. 5, де: $Вх_1$, $Вх_2$ – входи надходження відповідно першого і другого масивів чисел, Чт/Зп – вхід вибору режиму роботи пам'яті читання або запис, ТІ – вхід тактових імпульсів, ЛА – лічильник адреси, Км – комутатор, БП – блок пам'яті, СП – схема порівняння, Рг – регістр.

Алгоритм паралельного об'єднання двох масивів чисел виконується в два етапи. На першому етапі виконується паралельна запис першого N_1 та другого N_2 відсортованих масивів у блоки пам'яті відповідно $БП_1$, $БП_3$ та $БП_2$, $БП_4$. На другому етапі виконується злиття масивів, записаних у $БП_1$ і $БП_3$ за спаданням, а в $БП_2$ і $БП_4$ за зростанням. Кожний етап виконується за N тактів. Тривалість кожного такту визначається формулою:

$$t_{TI} = t_{БП} + t_{СП} + t_{Км} + t_{Рг}, \quad (5)$$

де $t_{БП}$, $t_{СП}$, $t_{Км}$ і $t_{Рг}$ – час, відповідно, звертання до пам'яті, порівняння двох чисел, затримки комутатора і запису в регістр.

Блок-схему алгоритму паралельного об'єднання двох масивів чисел наведено на рис. 6, де: $Лч_ц$ – лічильник циклів, ВхTI – вхід тактових імпульсів.

Для паралельного сортування m упорядкованих масивів чисел довжиною N чисел використаємо базову операцію паралельного об'єднання двох масивів чисел (див. рис. 2). Особливістю виконання такої базової операції є суміщення в часі процесу видачі відсортованих чисел з їх записом у наступні блоки пам'яті. Результатом паралельного сортування m упорядкованих масивів чисел довжиною N чисел є відсортований масив розміром $m \times N$ чисел, найбільші N числа якого послідовно отримуємо на першому вході, а найменші N числа – на m виході.

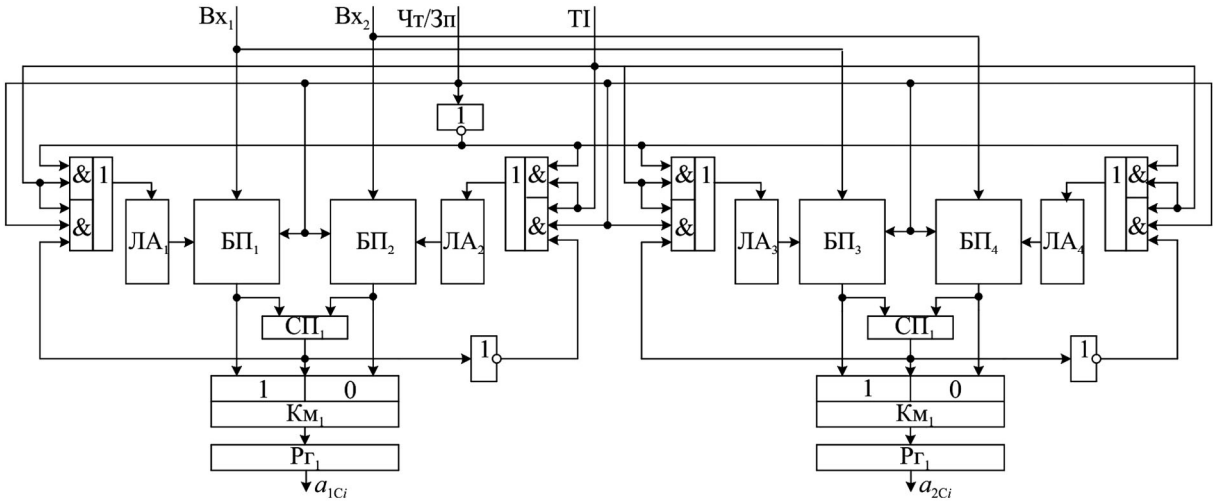

Рис. 5. Структура засобів реалізації паралельного об'єднання двох масивів чисел

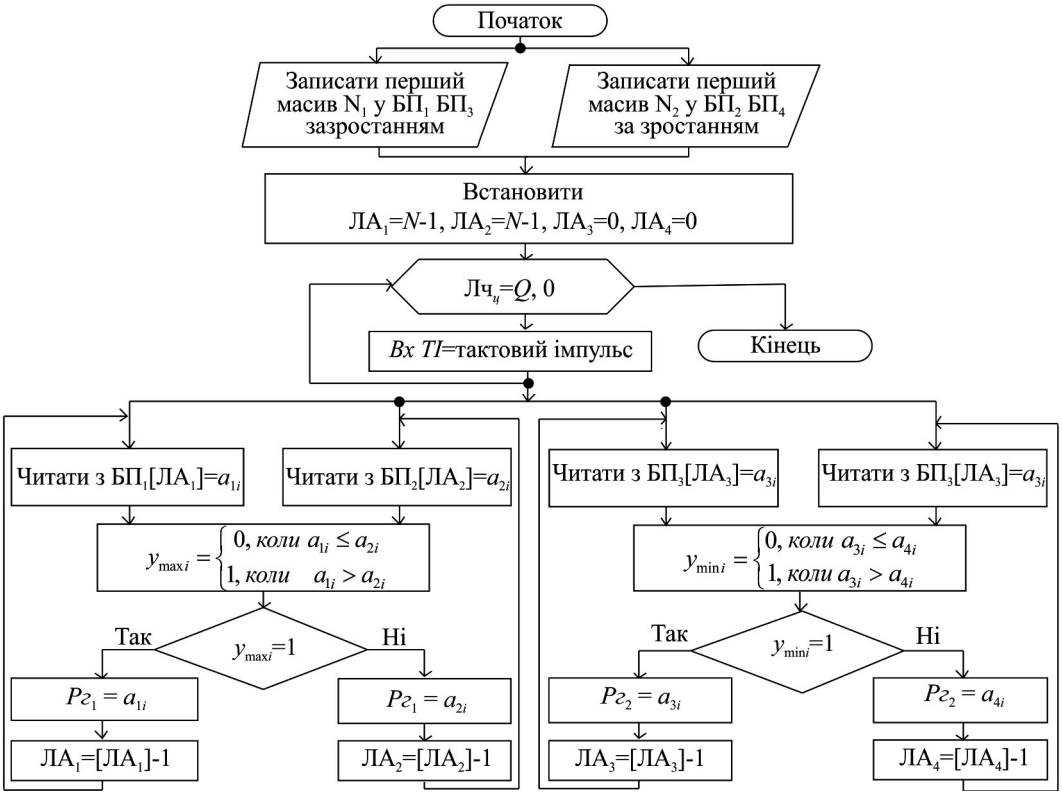

Рис. 6. Блок-схема алгоритму паралельного об'єднання двох масивів чисел

4. ***Розроблення програми паралельного сортування масивів даних з використанням удосконаленого методу злиття для графічного процесора***. Розроблення програмних засобів для паралельного сортування масивів даних з використанням удосконаленого методу злиття базується на використанні комплексного підходу, який охоплює: розробку та вдосконалення алгоритмів і методів паралельного сортування масивів даних; дослідження; потокові графи алгоритмів; апаратно-програмну модель CUDA з використанням особливостей архітектури графічного процесора GPU.

Для реалізації паралельного сортування масивів даних з використанням удосконаленого методу злиття було використано багатоядерний графічний процесор nVidia GeForce RTX 2060, який включає 1920 ядр та підтримує архітектуру і програмну модель CUDA.

Оскільки CUDA – це програмна модель, орієнтована на багатоядерну архітектору, для її ефективного використання необхідно використовувати принципи та підходи паралельного програмування. Основною відмінністю програмної реалізації алгоритмів на CUDA (GPU), порівняно з програмуванням на центральному процесорі, є використання потоків, блоків та сіток. Потік у графічному процесорі – це основний елемент даних, що підлягають обробленню. На відміну від потоків процесора, потоки CUDA надзвичайно "легкі", що відповідно забезпечує невеликі затрати на операції обміну між потоками. Відповідно блоки – це об'єднання від 64 до 512 потоків. Блоки графічного процесора складаються в сітки. Перевага такого групування полягає в тому, що кількість блоків, які одночасно обробляються графічним процесором, тісно пов'язані з апаратними ресурсами, що призводять до високого рівня масштабування програми. Оскільки ядро програми за один виклик використовується для всіх потоків сітки не звертаючи уваги на наявні ресурси графічного процесора. Якщо обладнання має недостатньо ресурсів, воно виконує блоки послідовно; якщо ресурсів достатньо, блоки обробляються паралельно. Це, відповідно, означає, що один і той самий код програми ефективно виконується як на відносно простих, так і на складніших графічних процесорах.

Базуючись на описаних принципах паралельного програмування на графічних процесорах, першим кроком реалізації програми паралельного сортування масивів даних з використанням удосконаленого методу злиття є декомпозиція методу на рівні блоків та на рівні потоків.

Базовою операцією, що виконується на кожному потоці, є об'єднання двох упорядкованих масивів чисел, яка складається з двох кроків – попарного порівняння та перестановки елементів масивів.

Загальну схему алгоритму методу сортування злиття з розподіленими між блоками і потоками базовими операціями для масиву із 16-ти чисел наведено на рис. 7.

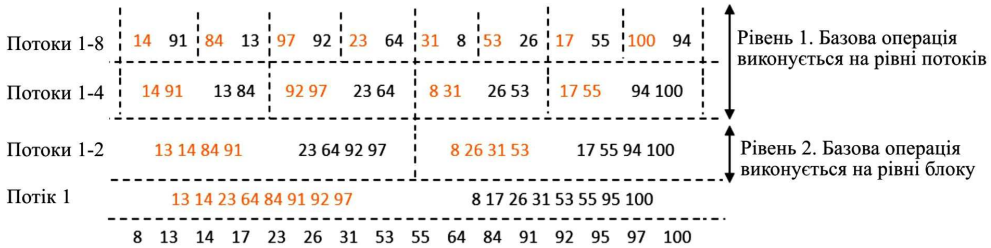

Рис. 7. Схема алгоритму методу сортування злиття на графічному процесорі

Окрім реалізації ядра програми, для виконання базової операції сортування на блоках і потоках графічного процесора необхідне розроблення додаткових програм для внутрішніх операцій: створення тимчасових масивів для зберігання проміжних результатів, циклів виконання програм у потоках та блоках, бінарного пошуку, присвоєння ключів записав на фінальній стадії порівняння двох масивів і формування вихідного відсортованого масиву.

Графіки залежності часу виконання програми паралельного сортування даних удосконаленим методом злиття для GPU та CPU залежно від розмірів масивів наведено на рис. 8.

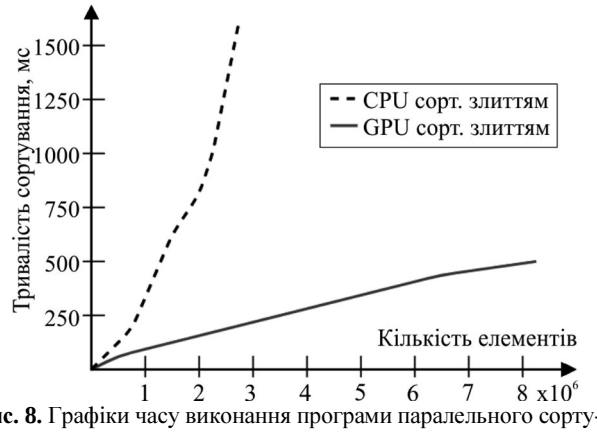

Рис. 8. Графіки часу виконання програми паралельного сортування даних удосконаленим методом злиття для GPU та CPU залежно від розмірів масивів

Незважаючи на велику кількість допоміжних операцій при реалізації паралельного сортування методом злиття на графічному процесорі nVidia GeForce RTX 2060 дає збільшення швидкодії приблизно в 100 разів порівняно з центральним процесором Inter i7.

Висновки

1. Удосконалено метод сортування двох масивів злиттям, який внаслідок одночасного об'єднання двох масивів, починаючи як з максимальних, так і мінімальних значень, забезпечує розпаралелення процесу сортування та зменшення у два рази кількості тактів сортування порівняно з послідовним злиттям.
2. Визначено, що основними етапами розроблення потокового графу паралельного сортування масивів даних з використанням удосконаленого методу злиття, який орієнтований на особливості архітектури графічного процесора, є: декомпозиція алгоритму сортування масивів даних з використанням вдосконаленого методу злиття; проектування комунікацій, а саме проектування локальних і регулярних зав'язків між функціональними операторами; укрупнення функціональних операторів; планування процесу сортування масивів даних з використанням удосконаленого методу злиття.
3. Розроблено потоковий граф паралельного сортування масивів даних, який завдяки використанню вдосконаленого методу сортування двох масивів злиттям, забезпечує ефективне просторове розпаралелення процесу сортування масивів даних і зменшення часу сортування.
4. Розроблено програмний засіб на графічному процесорі GPU паралельного сортування масивів даних з використанням вдосконаленого методу злиття, який внаслідок використання програмної моделі CUDA і конкретизованого потокового графу алгоритму забезпечує зменшення часу сортування приблизно у 100 разів порівняно з використанням тільки центрального процесора CPU.

References

1. Боресков, А. В. и др. (2012). Параллельные вычисления на GPU. *Архитектура и программная модель CUDA*. Москва: Изд-во Московского университета, 336 с.
2. Воеводин, В. В., Воеводин, Вл. В. (2002). *Параллельные вычисления*. СПб.: Изд. дом БХВ-Петербург, 608 с.
3. Гергель, В. П. (2010). *Высокопроизводительные вычисления для многопроцессорных многоядерных систем*. Москва: Изд-во Московского университета, 544 с.
4. Иванов, К. К., Раздобудько, С. А., Ковалев, Р. И. (2017). Параллельные методы сортировки. Текст: непосредственный, электронный. *Молодой ученый*, *7*(141), 15–16.
5. Кнут, Д. (2000). Искусство программирования, том 3: Сортировка и поиск, 2-е изд. Москва: Изд-во Наука, 832 с.
6. Кормен Т., Лейзерсон, Ч., Ривест, Р. (2004). Алгоритмы: построение и анализ: пер. с англ., под ред. А. Шеня. Москва: Изд-во МЦНМО, БИНОМ. Лаборатория знаний, 960 с.
7. Левитин, Ананий. (2006). Алгоритмы: введение в разработку и анализ.: пер. с англ. Москва: Изд. дом "Вильямс", 576 с.
8. Лунин, Д. В., Скворцов, С. В. (2014). Организация параллельных вычислений на платформе CUDA. *Вестник Рязанского государственного радиотехнического университета*, *49*, 77–82.
9. Мельничук, А. С., Луценко, С. П., Громовий, Д. С., Трофимова, К. В. (2013). Аналіз методів сортування масиву чисел. *Технологический аудит и резервы производства*, *4/1*(12), 37–40.
10. Немнюгин, С. А., Стесик, О. Л. (2002). *Параллельное программирование для многопроцессорных систем*. СПб.: Изд. дом БХВ-Петербург, 400 с.
11. Седжвик, Роберт. (2002). Фундаментальные алгоритмы на C++. *Анализ/Структуры данных/Сортировка/Поиск*: пер. с англ. СПб.: ООО "ДиаСофтЮП", 688 с.
12. Цмоць, І. Г, Кісь, Я. П., Антонів, В. Я. (2015). Застосування графічного процесора для підвищення швидкодії процесу сортування великих масивів даних. *Scientific Bulletin of UNFU*, *25*(6), 328–334. Retrieved from: https://nv.nltu.edu.ua/index.php/journal/article/view/973
13. Gryciuk, Y., & Grytsyuk, P. (2016). Implementation details for the cipher key generation Cardano permutation. *Modern Problems of Radio Engineering, Telecommunications and Computer Science. Proceedings of the 13th International Conference on TCSET 2016*, pp. 498–502. https://doi.org/10.1109/TCSET.2016.7452098
14. Hrytsiuk, Yu., & Bilas, O. (2019). Visualization of Software Quality Expert Assessment. *IEEE 2019 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies* (*CSIT 2019*), (Vol. 2, pp. 156–160), 17–20 September, 2019. https://doi.org/10.1109/stc-csit.2019.8929778
15. Hrytsiuk, Yu., Grytsyuk, P., Dyak, T., & Hrynyk, H. (2019). Software Development Risk Modeling. *IEEE 2019 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies* (*CSIT 2019*), (Vol. 2, pp. 134–137), 17–20 September, 2019. https://doi.org/10.1109/stc-csit.2019.8929778
16. Tsmots, I., Rabyk, V., Skorokhoda, O., & Antoniv, V. (2017). FPGA Implementation of Vertically Parallel Minimum and Maximum Values Dtermination in Array of Numbers. *14th International Conference The Experience of Desingning and Application of CAD Systems in Microelectronics*, (CADSM 2017). Proceedings 7916123, pp. 234–236.
17. Tsmots, I., Skorokhoda, O., & Rabyk, V. (2016). Structure Software Model of a Parallel-Vertical Multi-input Adder for FPGA Implementation. *Computer Sciences and Information Technologies. Proceedings of 11th International Scientific and Technical Conference* (CSIT). Lviv, pp. 158-160.
18. Tsmots, I., Skorokhoda, O., Antoniv, V., & Rabyk, V. (2018). Vertically-paraller method and VLSI-structure sorting of one-dimension alarrays. *Proceedings of 13th International Scientific and Technical Conference* (CSIT). Lviv, pp. 112–116.

I. G. Tsmots, V. Ya. Antoniv
Lviv Polytechnic National University, Lviv, Ukraine

IMPROVEMENT OF PARALLEL SORTING BY METHOD OF MERGING

The current stage of information technology is characterized by the accumulation of a large amount of data. For processing such arrays, systems often have to use sorting operations, which occupy about 40 % of the total time of working with data. The main ways to increase the speed of data sorting operations are the development of parallel methods and algorithms focused on mass-parallel computing tools with large shared memory (GPUs, which are SIMD processors) and the implementation on a device with using modern database elements such as very large-scale integration (VLSI). Analysis of existing literature sources showed that the existing sorting methods consist of two main operations. The first operation is a pairwise comparison and the second one is a permutation of elements in the case when the order of these values does not meet the sorting conditions. But most of these sorting methods are focused on sequential implementation and are not suitable for parallel computing. However, merge sorting is an exception. Due to the basic operation and algorithm of the method, merge sorting provides the opportunity of the high scaling on parallel systems since the input array could be separated on a huge amount of subsets that are not related to each other and each subset could be calculated in parallel. Considering the parallel characteristics of merge sorting, the best option for software implementation is GPU with program model CUDA. CUDA is software and hardware architecture for parallel computing from Nvidia, which can significantly increase computing performance because it focuses on a multi-core architect. The use of CUDA and appropriate algorithms reduces sorting time by about 100 times compared to using only the CPU. Thus, the paper considers the approach and steps for improving the parallel method of merge sorting and provides the algorithm for implementing the method of merge sorting on the graphics processing unit with CUDA architecture.

Keywords: flow graph; parallel sorting; merge method; sorting algorithms; pairwise comparison; data array; graphics processing unit.

Ознайомитись з вимогами до авторів статей можна на сайті видання:

https://nv.nltu.edu.ua

Підп. до друку 24.09.20. Формат 60×84/$_8$. Папір офсетний. Друк офсетний.
Ум. др. арк. 16,51. Ум. фарбо-відб. 16,97. Облік.-вид-арк. 16,74. Тираж 250 прим.
Зам. № *22/2020*

Видавець: Редакційно-видавничий центр НЛТУ України
79057, м. Львів, вул. Генерала Чупринки, 103
Тел.: (032) 240-23-50
E–mail: nv@nltu.edu.ua https://nv.nltu.edu.ua

Свідоцтво про внесення суб'єкта видавничої справи до
Державного реєстру видавців, виготовників і розповсюджувачів видавничої продукції
(Серія ДК, № 2062 від 17.01.2005 р.)

Свідоцтво про державну реєстрацію друкованого засобу масової інформації
(Серія КВ, № 11889-760ПР від 26.10.2006 р.)

Згідно з наказом Міністерства освіти і науки України від 17.03.2020 р., № 409,
"**Науковий вісник НЛТУ України**" належить до Переліку наукових фахових видань України (категорія Б),
в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора
і кандидата наук за галузями науки – біологічні, сільськогосподарські, технічні та економічні
й такими спеціальностями: 051 – *економіка*; 091 – *біологія*; 101 – *екологія*; 187 – *деревообробні та меблеві технології*; 205 – *лісове господарство*; 206 – *садово-паркове господарство*, а наказом від 2.07.2020 р., № 886 – такими спеціальностями: 122 – *комп'ютерні науки*; 126 – *інформаційні системи та технології*.