ПЕДАГОГІЧНА АКАДЕМІЯ:
НАУКОВІ ЗАПИСКИ

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ

УДК 004.272.4:519.6:37.091.33

DOI https://doi.org/10.5281/zenodo.14247733

Методологія вибору та візуалізації паралельних алгоритмів для інтерактивного навчання

Сіциліцин Юрій Олександрович

PhD, старший викладач кафедри інформатики і кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького, 69000, м. Запоріжжя, вул. Наукового містечка, 59, Україна, https://orcid.org/0000-0002-3888-5575

Ібрагімова Людмила Анатоліївна

PhD, старший викладач кафедри інформатики і кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького, 69000, м. Запоріжжя, вул. Наукового містечка, 59, Україна, https://orcid.org/0000-0002-1194-5128

Прийнято: 19.11.2024|Опубліковано: 29.11.2024

Анотація. *Метою статті є розробка методології вибору та візуалізації паралельних алгоритмів, яка спрямована на підвищення ефективності викладання паралельного програмування. Особлива увага приділяється алгоритмам, що здатні наочно ілюструвати принципи багатопоточності, синхронізації потоків та оптимізації ресурсів, а також підвищувати розуміння*

студентами складних концепцій завдяки інтерактивній демонстрації. У дослідженні застосовано комплексний підхід, що включає аналіз літератури, оцінку алгоритмів за розробленими критеріями та синтез практичних рекомендацій для їх візуалізації. Критерії оцінки охоплюють демонстрацію принципів паралельного програмування, інтуїтивність візуалізації, її інтерактивність, можливість поступового ускладнення та відповідність навчальним цілям. На основі цих критеріїв проведено класифікацію алгоритмів і визначено їх придатність для інтеграції у навчальний процес. У ході дослідження було обрано чотири паралельні алгоритми: сортування злиттям, обчислення добутку матриць, пошук у графі (BFS) та генетичне програмування. Для кожного з них розроблено індивідуальні рекомендації щодо інтерактивної візуалізації, включаючи приклади анімацій і графічних представлень. Запропоновані підходи демонструють, як розподіл задач між потоками, синхронізація та оптимізація ресурсів можуть бути ефективно візуалізовані для навчальних цілей. Розроблені методи інтерактивного навчання сприяють розвитку критичного мислення у студентів, допомагаючи краще розуміти складні концепції. Запропонована методологія підвищує якість підготовки майбутніх інженерів-програмістів, формуючи у них "паралельне мислення". Візуалізації допомагають студентам ефективно опановувати складні концепції паралельного програмування, експериментувати з різними алгоритмами та спостерігати за їх виконанням у реальному часі. Впровадження таких підходів у навчальні програми сприятиме адаптації студентів до викликів сучасного ринку праці, забезпечуючи їх необхідними знаннями та практичними навичками. Основні результати дослідження мають практичне значення для розробки віртуальних лабораторій, які можуть бути адаптовані до потреб освітніх закладів і забезпечити високий рівень інтерактивності навчання. Це відкриває

перспективи для подальших досліджень у галузі інтерактивного навчання та розробки нових освітніх технологій.

Ключові слова: *паралельні алгоритми, інтерактивне навчання, візуалізація, паралельне програмування, освітні технології.*

Methodology for selecting and visualizing parallel algorithms for interactive learning

Yurii Sitsylitsyn

PhD, Senior Lecturer at the Department of Informatics and Cybernetics Bogdan Khmelnitsky Melitopol State Pedagogical University, 69000, Zaporizhzhia, Naukovogo Mistechka St., 59, Ukraine, https://orcid.org/0000-0002-3888-5575

Liudmyla Ibragimova

PhD, Senior Lecturer at the Department of Informatics and Cybernetics Bogdan Khmelnitsky Melitopol State Pedagogical University, 69000, Zaporizhzhia, Naukovogo Mistechka St., 59, Ukraine, https://orcid.org/0000-0002-1194-5128

Abstract. *The aim of this article is to develop a methodology for selecting and visualizing parallel algorithms to enhance the effectiveness of teaching parallel programming. Special attention is given to algorithms that can visually demonstrate the principles of multithreading, thread synchronization, and resource optimization, while improving students' understanding of complex concepts through interactive demonstrations. The study employs a comprehensive approach that includes literature analysis, evaluation of algorithms based on developed criteria, and synthesis of practical recommendations for their visualization. The evaluation criteria encompass the demonstration of parallel programming principles, visualization intuitiveness,*

interactivity, scalability, and alignment with educational objectives. Based on these criteria, the algorithms were classified and their suitability for integration into the educational process was determined. The study identified four parallel algorithms: merge sort, matrix multiplication, graph traversal (BFS), and genetic programming. Tailored recommendations for the interactive visualization of each algorithm were developed, including examples of animations and graphical representations. The proposed approaches illustrate how task distribution among threads, synchronization, and resource optimization can be effectively visualized for educational purposes. These interactive learning methods foster critical thinking in students and help them better comprehend complex concepts. The proposed methodology enhances the training quality of future software engineers by fostering "parallel thinking." Visualizations enable students to effectively grasp complex concepts of parallel programming, experiment with various algorithms, and observe their execution in real-time. Integrating such approaches into educational programs will help students adapt to modern job market challenges by providing them with the necessary knowledge and practical skills. The key findings have practical significance for the development of virtual laboratories that can be tailored to the needs of educational institutions, ensuring a high level of interactivity in learning. This opens opportunities for further research in interactive learning and the development of new educational technologies.

Keywords: *parallel algorithms, interactive learning, visualization, parallel programming, educational technologies.*

Постановка проблеми у загальному вигляді та її зв'язок з важливими науковими чи практичними завданнями. Паралельні обчислення займають важливе місце у сучасному програмуванні, забезпечуючи можливість обробляти великі обсяги даних швидше та ефективніше. У світі, де обсяги інформації постійно зростають, а завдання стають все складнішими, паралельні обчислення

дозволяють оптимізувати роботу програмних систем та покращити їх продуктивність. Вони використовуються в різних галузях, від наукових досліджень до комерційних додатків, забезпечуючи необхідну потужність для виконання складних обчислювальних завдань.

Однак, вивчення паралельних обчислень є складним завданням, особливо для студентів, які вже звикли до послідовного програмування. Після тривалого вивчення та практики роботи з послідовними алгоритмами, перехід до паралельного програмування може бути досить складним. Ця складність зумовлена необхідністю зміни підходів до мислення та розуміння процесів, які відбуваються під час виконання паралельних задач. Студенти повинні навчитися працювати з одночасним виконанням декількох процесів, синхронізацією потоків та ефективним розподілом ресурсів.

У цьому контексті виникає потреба в розвитку у студентів здатності до "паралельного мислення". Цей термін означає уміння розуміти та застосовувати принципи паралельного програмування, бачити процеси в їхньому взаємозв'язку та передбачати можливі конфлікти і шляхи їх вирішення. Паралельне мислення дозволяє програмістам ефективно розробляти та оптимізувати паралельні алгоритми, що є критично важливим у сучасному програмуванні.

Одним із перспективних підходів до навчання студентів паралельному програмуванню є використання візуалізацій. Візуалізація паралельних алгоритмів може стати потужним інструментом для ілюстрації їхньої роботи та допомогти студентам краще зрозуміти принципи паралельного програмування. Через візуалізацію студенти можуть бачити, як різні процеси взаємодіють, як відбувається розподіл задач та які проблеми можуть виникати під час паралельного виконання. Це дозволяє наочно демонструвати складні концепції, роблячи їх більш зрозумілими та доступними.

Використання візуалізацій у навчальному процесі може значно підвищити ефективність засвоєння матеріалу. Вони можуть бути інтерактивними, дозволяючи студентам експериментувати з різними алгоритмами, бачити результати своїх дій в реальному часі та краще розуміти наслідки певних рішень. Такий підхід не лише сприяє кращому розумінню паралельних обчислень, але й стимулює інтерес до цієї важливої галузі програмування.

Отже, вибір паралельних алгоритмів для їх візуалізації є важливим кроком у підготовці майбутніх фахівців у сфері програмування. Це дозволяє не лише спростити процес навчання, але й підвищити якість підготовки студентів, допомагаючи їм опанувати ключові навички, необхідні для успішної роботи у сучасному світі технологій.

Аналіз останніх досліджень і публікацій. Питання вибору алгоритмів та їх візуалізації для освітніх цілей розглядалося на різних рівнях, включно з міжнародними. Наукова спільнота вивчала цю тему через призму програмування, педагогіки та технологічних інновацій. Сучасні дослідження базуються на напрацюваннях останніх років, коли технології багатопоточності та розподілених обчислень досягли значного рівня розвитку. Дослідження Patrick Diehl, Steven R. Brandt та Hartmut Kaiser [1] є ключовим для розуміння ролі сучасного C++ у навчанні паралельного програмування. Автори досліджують переваги використання візуалізацій для демонстрації багатопоточності. Праця Chilei Wang та співавторів [2] висвітлює аспекти масово паралельних алгоритмів для графів. Автори вказують на важливість інтерактивного представлення складних процесів, таких як обчислення найкоротших шляхів у динамічних графах. Згідно з навчальним посібником «Паралельні та розподілені обчислення» (2021) [3], особливу увагу приділено викликам масштабування у паралельних системах. Автори розглядають практичні підходи до створення алгоритмів, що забезпечують ефективне

використання ресурсів, і пропонують методи, які можуть бути адаптовані для інтерактивного навчання студентів.. Luís Cunha та його колеги [4] зосереджуються на паралелізмі дерев та графів, підкреслюючи необхідність інтуїтивних інструментів для вивчення алгоритмів. Праця Subodh Kumar [5] є фундаментальною для вивчення базових концепцій паралельного програмування та пропонує освітні методи для студентів, які лише розпочинають вивчати цю тему. Колектив авторів [6] досліджує застосування алгоритму пошуку в ширину (BFS) у реальних сценаріях, зокрема при обробці великих графів. У роботі акцентується на важливості оптимізації алгоритму для досягнення високої продуктивності, а також на можливостях його апаратного прискорення, що забезпечує значне підвищення ефективності обчислень. Автори також підкреслюють потенціал використання таких підходів для інтерактивного навчання завдяки наочності та практичній значущості BFS. Tianyi Yu, Wei Li [7] проаналізували паралельні алгоритми сортування, що мають високу освітню цінність завдяки можливості поступового ускладнення, що робить їх придатними для навчальних цілей. Значний внесок у дослідження можливостей метаевристичних алгоритмів зроблено у роботі Saman M. Almufti [8]. Автори аналізують роль візуалізації в оптимізації алгоритмів, що застосовуються у складних обчисленнях. У статті [9] розглядаються алгоритми, орієнтовані на графічне представлення обчислювальних процесів, із застосуванням інноваційних підходів до їх інтерактивної демонстрації. Основна увага приділяється створенню навчальної системи, що забезпечує динамічну візуалізацію роботи алгоритмів, покроковий опис їх виконання та подання теоретичної інформації. У документі Computer Science Curricula 2023 [10] акцентується на необхідності інтерактивних навчальних інструментів, які дозволяють студентам експериментувати з паралельними алгоритмами.

Виділення невирішених раніше частин загальної проблеми. Таким чином, попри значні наукові досягнення, залишаються відкритими питання вибору алгоритмів для візуалізації з урахуванням їх освітньої цінності, створення інтерактивних інструментів для демонстрації складних процесів та формування «паралельного мислення» у студентів. Запропонована стаття спрямована на вирішення цих питань та внесення нового підходу до інтерактивного навчання паралельного програмування.

Формулювання цілей статті

Ця стаття присвячена аналізу та вибору оптимальних паралельних алгоритмів для їх візуалізації у контексті навчання паралельного програмування. Відповідно до мети визначені наступні завдання:

- Визначити, які алгоритми найкраще ілюструють основні принципи паралельного програмування, такі як розподіл задач, синхронізація потоків та оптимізація використання ресурсів.
- Сформулювати об’єктивні критерії оцінки алгоритмів з урахуванням їхньої освітньої цінності, інтерактивності та можливостей візуалізації.
- На основі розроблених критеріїв обрати алгоритми, які мають максимальний навчальний потенціал та відповідають сучасним вимогам паралельного програмування.
- Показати, як використання візуалізацій може полегшити розуміння студентами складних концепцій та сприяти розвитку навичок паралельного мислення.
- Запропонувати способи інтерактивної демонстрації алгоритмів, які можуть бути впроваджені у навчальний процес, включаючи анімації, графічні представлення та інші методи.

Виклад основного матеріалу дослідження з повним обґрунтуванням здобутих наукових результатів. Аналіз та вибір оптимальних паралельних

алгоритмів для візуалізації потребує чіткого визначення критеріїв, які дозволять об'єктивно оцінити їхню придатність для навчальних цілей. У розглянемо критерії, що дозволять визначити найбільш ефективні алгоритми для візуалізації та інтерактивного навчання студентів.

Перше, на що ми звертаємо увагу при виборі паралельних алгоритмів для візуалізації, - це їхня освітня цінність. Під освітньою цінністю розуміється здатність алгоритму наочно демонструвати основні принципи та концепції паралельного програмування. Алгоритми повинні бути достатньо простими для розуміння студентами, але водночас мати достатню складність для демонстрації важливих аспектів паралельних обчислень. За цим критерієм можна сформувати наступні вимоги до алгоритмів:

- Вибрані алгоритми повинні показувати такі концепції, як розподіл задач, синхронізація потоків, управління ресурсами та уникнення конфліктів.
- Алгоритми повинні дозволяти поступове введення нових елементів та концепцій, починаючи з простих прикладів і поступово переходячи до більш складних задач.
- Алгоритми повинні відповідати темам, що входять до навчальної програми з паралельного програмування, та бути актуальними для сучасних викликів у галузі.

Другим важливим критерієм є можливості візуалізації алгоритмів. Візуалізація повинна бути зрозумілою, інтерактивною та здатною наочно демонструвати роботу алгоритму. Вона повинна допомагати студентам бачити, як відбувається виконання паралельних задач, як взаємодіють потоки, та які проблеми можуть виникати. За цим критерієм можна сформувати наступні вимоги до алгоритмів:

- Візуалізації повинні бути інтуїтивно зрозумілими та легко сприйматися студентами. Використання графічних елементів та анімацій може значно покращити розуміння складних процесів.
- Студенти повинні мати можливість взаємодіяти з візуалізацією, змінювати параметри алгоритму, спостерігати за його поведінкою у різних умовах та експериментувати з різними сценаріями.
- Візуалізація повинна підтримувати як прості, так і складні сценарії, дозволяючи студентам бачити роботу алгоритмів на різних рівнях складності.

Третім критерієм є аспекти багатопоточності та оптимізації. Вибрані алгоритми повинні не лише демонструвати базові принципи паралельного програмування, але й дозволяти студентам ознайомитися з питаннями оптимізації та ефективного використання ресурсів. За цим критерієм можна сформувати наступні вимоги до алгоритмів:

- Алгоритми повинні демонструвати, як ефективно розподіляти задачі між потоками, уникати нерівномірного завантаження та максимізувати продуктивність.
- Вибрані алгоритми повинні показувати різні методи синхронізації потоків, управління критичними секціями та уникнення станів гонитви.
- Алгоритми повинні допомагати студентам зрозуміти, як оптимізувати використання процесорних ресурсів та пам'яті для досягнення максимальної ефективності.

Після визначення критеріїв для оцінки паралельних алгоритмів ми можемо перейти до детального аналізу кожного алгоритму з точки зору їх відповідності цим критеріям. Для формування оцінок по алгоритмам ми звернулись до наступних джерел: огляд алгоритмів паралелізму спільної пам’яті в сучасному C++ і ВПО [1], огляд паралельного алгоритму пошуку найкоротшого шляху [2], огляд паралелізму графів та дерев [3], грунтовний твір по основам пералельного

програмування від С.Кумар [5], огляд метаевристичних паралельних алгоритмів [7] та Computer Science Curricula 2023 [10]. Будемо оцінювати відповідність алгоритму вимогам за десятибальною шкалою де 1 – найнижча відповідність, а 10 – найвища. Для спрощення представлення в таблиці пронумеруємо вимоги:

Вимога 1: Демонстрація основних принципів паралельного програмування

Вимога 2: Можливість поступового ускладнення

Вимога 3: Відповідність темам навчальної програми

Вимога 4: Інтуїтивна зрозумілість візуалізації

Вимога 5: Інтерактивність візуалізації

Вимога 6: Можливість масштабування

Вимога 7: Розподіл навантаження

Вимога 8: Синхронізація та управління потоками

Вимога 9: Оптимізація ресурсів

Нижче представлено таблицю 1, що містить оцінки кожного алгоритму за всіма вимогами які ми визначили у трьох критеріях.

Таблиця 1

Оцінки паралельних алгоритмів

Вид паралельного алгоритму	**Вимога 1**	**Вимога 2**	**Вимога 3**	**Вимога 4**	**Вимога 5**	**Вимога 6**	**Вимога 7**	**Вимога 8**	**Вимога 9**	**Загальна оцінка**
Паралельний алгоритм сортування злиттям	7	4	10	5	7	10	3	7	8	61
Паралельний алгоритм сортування Quicksort	5	4	8	8	3	6	5	2	8	49
Паралельний пошук	6	2	5	1	10	6	9	1	10	50
Паралельний алгоритм обчислення добутку матриць	5	7	10	9	3	5	3	7	5	54
Паралельний пошук у графі (BFS)	9	7	8	4	9	2	10	9	10	68
Паралельне знаходження мінімального остовного дерева (Kruskal's Algorithm)	5	2	8	7	8	3	1	4	2	40
Алгоритм паралельного пошуку шаблону (Rabin-Karp)	8	4	2	6	6	10	4	6	2	48
Паралельний алгоритм обробки зображень (Edge Detection)	7	2	2	4	8	7	9	8	5	52
Паралельне генетичне програмування	2	5	8	10	9	9	1	9	7	60
Алгоритм паралельного інтегрування методом Монте-Карло	9	8	1	8	8	3	1	8	3	49

Джерело: власна розробка авторів

Розглянемо результати аналізу. Демонстрація основних принципів паралельного програмування оцінює, наскільки добре алгоритм ілюструє ключові аспекти паралельного програмування. Найвищі оцінки отримали алгоритми паралельний пошук у графі (BFS) (9) та паралельного інтегрування методом Монте-Карло (9), що підкреслює їхню ефективність у цьому контексті. Можливість поступового ускладнення визначає, наскільки легко можна збільшувати складність алгоритму. Алгоритми обчислення добутку матриць (7), паралельний пошук у графі (BFS) (7) та інтегрування методом Монте-Карло (8) отримали найвищі оцінки, вказуючи на їхню гнучкість у навчанні. Відповідність темам навчальної програми оцінює, як добре алгоритм інтегрується в навчальну програму. Високі оцінки отримали алгоритми обчислення добутку матриць (10) та сортування злиттям (10). Інтуїтивна зрозумілість візуалізації оцінює, наскільки зрозумілою є візуалізація роботи алгоритму. Найвищі оцінки отримали алгоритми паралельного обчислення добутку матриць (9) та генетичного програмування (10). Інтерактивність візуалізації оцінює здатність алгоритму до інтерактивного навчання. Паралельний пошук (10) та генетичне програмування (9) мають високі оцінки. Можливість масштабування оцінює, наскільки алгоритм здатен обробляти збільшені обсяги даних. Високі оцінки мають алгоритми сортування злиттям (10) та паралельного пошуку шаблону (10). Розподіл навантаження оцінює ефективність розподілу обчислювального навантаження між потоками. Найвищі оцінки отримали BFS (10) та Паралельний алгоритм обробки зображень (9). Синхронізація та управління потоками оцінює, як добре алгоритм управляє потоками та їхньою синхронізацією. Найвищі оцінки мають алгоритми BFS (9) та генетичне програмування (9). Оптимізація ресурсів оцінює здатність алгоритму ефективно використовувати ресурси. Найвищі оцінки отримали BFS (10) та паралельний пошук (10).

Для подальшого розгляду візьмемо алгоритми загальна оцінка яких вище середнього значення загальної оцінки. В результаті отримуємо такі алгоритми:

1. Паралельний алгоритм сортування злиттям (61)
2. Паралельний алгоритм обчислення добутку матриць (54)
3. Паралельний пошук у графі (BFS) (68)
4. Паралельне генетичне програмування (60)

Розглянемо ці алгоритми та можливості їх анімації докладніше.

Паралельне сортування злиттям є однією з ефективних технік для обробки великих наборів даних, оптимізуючи використання многопоточності та зменшуючи загальний час сортування. Цей алгоритм базується на класичному алгоритмі сортування злиттям, але адаптований для виконання на декількох процесорах або ядрах одночасно [11]. Основна ідея паралельного сортування злиттям полягає у розподілі початкового набору даних на декілька підмножин, які можуть бути відсортовані незалежно в різних потоках або процесах. Після сортування окремих підмножин результати зливаються в один відсортований набір. Алгоритм можна розділити на дві основні фази: розподіл і злиття.

На фазі розподіл початковий набір даних ділиться на менші підмножини, причому кількість підмножин зазвичай відповідає кількості доступних обчислювальних ядер. Кожна підмножина сортується окремо, що може виконуватись паралельно без взаємодії з іншими підмножинами.

На фазі злиття сортовані підмножини послідовно зливаються в більші сортовані множини. Цей процес продовжується, доки всі окремі підмножини не будуть об'єднані в один відсортований набір даних.

Паралелізація процесу злиття також можлива і включає більш складні механізми для ефективного злиття сортованих підмножин. Злиття може бути організовано так, щоб кожен потік або процес займався злиттям двох підмножин, а результат злиття знову був переданий до злиття з наступною підмножиною в

іншому потоці. Такий підхід може значно зменшити загальний час сортування, особливо при великих обсягах даних [12].

Паралельне сортування злиттям є потужним інструментом в сучасному програмуванні, зокрема в областях, де потрібна швидка обробка великих обсягів даних, як у фінансових аналізах, великих базах даних та наукових дослідженнях.

Для демонстрації паралельного алгоритму сортування злиттям у вигляді анімації, користувач спочатку буде вводити кількість ядер віртуальної системи та розмір масиву, який потрібно відсортувати. Це дозволить адаптувати анімацію під ресурси та потреби користувача.

Після введення необхідних параметрів, користувач натискатиме кнопку "Старт", що ініціює візуалізацію процесу сортування. Анімація яскраво покаже, як масив спочатку розділяється на менші частини, кожна з яких відсортовується в окремому потоці, а потім відбувається їх поступове злиття у відсортований масив. Кожен етап сортування буде відображений за допомогою різних кольорів або індикаторів, що робить процес інтуїтивно зрозумілим і навчальним.

Додатково, інтерфейс може включати налаштування швидкості анімації, що дозволить користувачам спостерігати за кожним кроком детальніше або отримати швидкий огляд процесу. Також можливе введення варіантів візуалізації: від простого зображення барів, що представляють елементи масиву, до більш складних графічних представлень, які підкреслюють структуру даних і алгоритмічні операції. Це зробить інструмент корисним як для новачків, так і для досвідчених розробників, що досліджують паралельне програмування.

Візуалізація паралельного алгоритму сортування злиттям дасть студентам можливість розуміти розподіл задач шляхом демонстрації розділення масиву на підмножини для обробки в різних потоках. Можливість побачити паралельне виконання завдань при одночасній обробці підмножин, підкреслюючи ефективність багатопоточності. Побачити синхронізацію потоків шляхом

демонстрації узгодження роботи потоків під час сортування. В результаті - така демонстрація полегшує розуміння складних алгоритмічних концепцій через візуальні підказки.

Паралельний алгоритм обчислення добутку матриць є ефективною технікою для обробки великих матричних множин, оптимізуючи використання багатопоточності та зменшуючи загальний час обчислень. Цей алгоритм базується на класичному методі множення матриць, але адаптований для виконання на декількох процесорах або ядрах одночасно [13]. Основна ідея полягає в розподілі вихідної задачі на підзадачі, які можуть виконуватись незалежно в різних потоках або процесах.

Алгоритм можна розділити на дві основні фази: розподіл даних і обчислення добутків.

На фазі розподілу матриці розбиваються на менші блоки, причому кількість блоків зазвичай відповідає кількості доступних обчислювальних ядер. Кожен блок обробляється окремо, що може виконуватись паралельно без взаємодії з іншими блоками.

На фазі обчислення кожен потік виконує множення блоків і зберігає результати у відповідних позиціях вихідної матриці. Цей процес продовжується, доки всі блоки не будуть оброблені.

Паралелізація процесу обчислення включає ефективний розподіл даних і управління пам’яттю для мінімізації конфліктів доступу. Також можна організувати обмін даними між потоками для оптимізації розподілу ресурсів. Такий підхід значно зменшує загальний час обчислень, особливо при роботі з великими матрицями.

Паралельний алгоритм обчислення добутку матриць є потужним інструментом у сучасному програмуванні, зокрема в областях, де потрібна

швидка обробка великих обсягів даних, як у наукових дослідженнях, машинному навчанні та комп'ютерній графіці.

Для демонстрації паралельного алгоритму у вигляді анімації, користувач вводить розмір матриць і кількість ядер системи. Це дозволяє адаптувати анімацію під ресурси та потреби користувача.

Після введення необхідних параметрів, користувач натискає кнопку "Старт", що ініціює візуалізацію процесу обчислень. Анімація яскраво показує, як матриці розбиваються на блоки, кожен з яких обробляється в окремому потоці, а потім результати об'єднуються у вихідну матрицю. Кожен етап буде відображений за допомогою різних кольорів або індикаторів, що робить процес зрозумілим і навчальним.

Додатково, інтерфейс може включати налаштування швидкості анімації, що дозволяє користувачам спостерігати за кожним кроком детальніше або отримати швидкий огляд процесу. Це робить інструмент корисним для навчання паралельного програмування.

Візуалізація паралельного алгоритму обчислення добутку матриць дасть студентам можливість розуміти розподіл навантаження шляхом демонстрації розподілення роботи між потоками, можливість побачити одночасну обробку різних частин матриць, що дозволить студентам у подальшому підвищити продуктивність своїх програм. Візуалізація ілюструє процес оптимізації ресурсів шляхом ефективного використання процесорів. Показує управління спільними ресурсами, запобігання конфліктам. Допомагає краще засвоїти концепції паралельного програмування через візуальні підказки.

Паралельний пошук у графі (BFS) є ефективною технікою для аналізу великих графових структур, оптимізуючи використання багатопоточності та зменшуючи загальний час обробки. Цей алгоритм базується на класичному пошуку в ширину, але адаптований для виконання на декількох процесорах або

ядрах одночасно [14]. Основна ідея паралельного BFS полягає у розподілі роботи між потоками, які одночасно досліджують різні частини графа. Це досягається шляхом поділу вузлів на окремі підмножини, які обробляються незалежно.

Алгоритм складається з двох основних фаз: розподілу та обробки. На фазі розподілу початкові вузли, з яких починається пошук, розподіляються між доступними потоками. Кожен потік досліджує свої вузли і додає сусідні вузли до черги для подальшої обробки.

На фазі обробки всі вузли, які були додані до черги, паралельно обробляються потоками. Паралелізація включає в себе координацію доступу до спільних структур даних, щоб уникнути конфліктів і забезпечити коректність алгоритму.

Алгоритм паралельного пошуку у графі на псевдокоді може виглядати наступним чином:

Паралельний BFS є потужним інструментом для аналізу великих мереж, таких як соціальні мережі, веб-графи та інші складні структури. Завдяки використанню багатопоточності, він дозволяє одночасно обробляти великі обсяги даних, розподіляючи навантаження між кількома ядрами процесора.

Цей підхід значно зменшує час обробки, оскільки кожен потік досліджує окрему частину графа незалежно. У соціальних мережах це може бути корисним для аналізу зв’язків між користувачами, пошуку впливових вузлів або виявлення спільнот. У веб-графах BFS допомагає в аналізі структур посилань, що важливо для алгоритмів ранжування сторінок.

Загалом, паралельний BFS ефективний для сучасних застосувань, де швидкість і масштабованість мають критичне значення, забезпечуючи швидку обробку та аналіз великих масивів даних.

Візуалізація алгоритму паралельного BFS допоможе студентам зрозуміти розподіл роботи між потоками, підкреслюючи ефективність паралельної

обробки. Через паралельне виконання видно, як вузли обробляються одночасно, що демонструє приріст швидкості. Візуалізація показує управління доступом до спільних структур даних, колірна схема дозволяє легко відслідковувати відвідані вузли, що допомагає зрозуміти алгоритм у дії. Інтерактивна візуалізація сприяє глибшому розумінню концепцій паралельного програмування.

Паралельне генетичне програмування є ефективною технікою для оптимізації складних задач, використовуючи багатопоточність для прискорення еволюційного процесу. Цей алгоритм базується на класичному генетичному програмуванні, адаптованому для паралельного виконання на декількох процесорах або ядрах.

Основна ідея полягає в тому, щоб розподілити обчислювальне навантаження між кількома потоками, кожен з яких обробляє частину популяції, виконуючи операції відбору, схрещування та мутації незалежно. Після кожного покоління результати об'єднуються для створення нової популяції.

Алгоритм можна розділити на дві основні фази: паралельна обробка популяцій та об'єднання результатів.

На фазі паралельної обробки популяція ділиться на підмножини, які обробляються незалежно в різних потоках. Кожен потік виконує еволюційні операції, такі як відбір, схрещування та мутація, для створення нового покоління.

На фазі об'єднання результати з різних потоків комбінуються для створення єдиної нової популяції, яка буде використана в наступному поколінні. Цей підхід значно прискорює процес еволюції, особливо при великій популяції та складних задачах.

Паралельне генетичне програмування є потужним інструментом у сучасному програмуванні, зокрема в областях штучного інтелекту, оптимізації та машинного навчання [15].

Для демонстрації паралельного алгоритму у вигляді анімації, користувач вводить параметри, такі як розмір популяції, кількість поколінь та кількість потоків. Після цього натискання кнопки "Старт" ініціює візуалізацію процесу, яка покаже, як різні потоки паралельно обробляють підмножини популяції та як результати об'єднуються. Кожен етап буде відображений за допомогою різних кольорів або індикаторів, що робить процес зрозумілим і навчальним.

Інтерфейс може включати налаштування швидкості анімації, що дозволяє користувачам детально спостерігати за кожним кроком або отримати швидкий огляд процесу.

Алгоритм паралельне генетичне програмування на псевдокоді може виглядати наступним чином:

Візуалізація паралельного алгоритму генетичного програмування дозволить студентам побачити ключові етапи генетичного програмування, такі як відбір, схрещування та мутація. Дасть уявлення про розподіл задач шляхом демонстрації, як популяція розподіляється на підмножини для паралельної обробки, підкреслюючи ефективність розподілу навантаження між потоками. Ілюструє одночасну обробку підмножин популяції різними потоками, що демонструє переваги багатопоточності. Показує процес об'єднання результатів від різних потоків для формування нової популяції, підкреслюючи важливість синхронізації в паралельних обчисленнях. Полегшує засвоєння складних концепцій через візуальні підказки та анімацію. Дає можливість спостерігати за покращенням продуктивності при використанні паралельного виконання порівняно з послідовним.

Застосування в реальних сценаріях: Підкреслює практичну цінність генетичного програмування для вирішення складних задач оптимізації.

Висновки. Проведений аналіз вказує на перспективи використання вибраних паралельних алгоритмів для подальшого їх використання при

створенні «Віртуальної лабораторії паралельних обчислень» на уроках паралельного програмування у студентів університету напряму «комп'ютерні науки». Використання візуалізацій паралельних алгоритмів дозволяє покращити розуміння складних концепцій паралельного програмування, що є критично важливим у сучасних умовах швидкого розвитку технологій та зростання обсягів даних. Основні перспективи використання віртуальної лабораторії ми бачимо у допомозі студентам краще зрозуміти процеси, що відбуваються під час виконання паралельних задач, завдяки наочній демонстрації розподілу задач, синхронізації потоків та оптимізації ресурсів. Студенти набувають вміння працювати з одночасним виконанням кількох процесів, що сприяє розвитку критичного мислення та здатності передбачати можливі конфлікти та шляхи їх вирішення. Віртуальна лабораторія дозволяє студентам експериментувати з різними алгоритмами, змінювати параметри та спостерігати за результатами в реальному часі. Це сприяє кращому засвоєнню матеріалу та підвищує інтерес до паралельного програмування. Використання актуальних паралельних алгоритмів, таких як сортування злиттям, обчислення добутку матриць та пошук у графах, забезпечує студентам знання та навички, необхідні для роботи у сучасній технологічній сфері. Віртуальна лабораторія може бути налаштована під індивідуальні потреби студентів та ресурси університету, що дозволяє ефективно використовувати доступні обчислювальні потужності та забезпечити високий рівень підготовки.

Впровадження «Віртуальної лабораторії паралельних обчислень» у навчальний процес забезпечить студентам глибоке розуміння принципів паралельного програмування, підвищить якість їхньої підготовки та допоможе успішно адаптуватися до вимог сучасного ринку праці. Це не тільки сприятиме підвищенню освітнього рівня, але й стимулюватиме інтерес до подальших досліджень та розробок у галузі паралельних обчислень.

Список використаних джерел

1. Diehl P., Brandt S. R., Kaiser H. Shared memory parallelism in Modern C++ and HPX. *arXiv:2302.07191*. 2023. URL: https://doi.org/10.48550/arXiv.2302.07191 (date of access: 10.10.2024)

2. Wang, C., Hua, QS., Jin, H. et al. Massively parallel algorithms for fully dynamic all-pairs shortest paths. *Front. Comput. Sci.* 2024. Vol. 18. URL: https://doi.org/10.1007/s11704-024-3452-2 (date of access: 10.10.2024)

3. Минайленко Р.М. Паралельні та розподілені обчислення: навч. посіб. Кропивницький: Видавець Лисенко В. Ф., 2021. 153 с. URL: https://dspace.kntu.kr.ua/server/api/core/bitstreams/396e02d2-725b-47b5-a1c0-ae07a9bec326/content (date of access: 10.10.2024)

4. Cunha L., Marciano E., Moraes A., Santiago L., Santos C. New parallelism and heuristic approaches for generating tree-spanners in graphs. *Concurrency and Computation: Practice and Experience.* 2024. Vol. 36, Issue 15. URL: https://doi.org/10.1002/cpe.8106 (date of access: 10.10.2024)

5. Kumar S. Introduction to Parallel Programming. Cambridge: Cambridge University Press, 2022. 280 p.

6. Li K. at al. ScalaBFS: A Scalable BFS Accelerator on HBM-Enhanced FPGAs. arXiv:2105.11754. 2021. URL: https://doi.org/10.48550/arXiv.2105.11754 (date of access: 10.10.2024)

7. Yu T., Li W. A Creativity Survey of Parallel Sorting Algorithm. *arXiv:2202.08463*. 2022. URL: https://doi.org/10.48550/arXiv.2202.08463 (date of access: 10.10.2024)

8. M. Almufti, S., Ahmad Shaban, A., Arif Ali, Z., Ismael Ali, R. and A. Dela Fuente, J. 2023. Overview of Metaheuristic Algorithms. *Polaris Global Journal of Scholarly Research and Trends.* 2023. Vol. 2, Issue 2. p. 10–32. URL: https://doi.org/10.58429/pgjsrt.v2n2a144 (date of access: 10.10.2024)

9. Козак, Д. Р., Коротєєва, Т. О. Проєктування навчальної системи візуалізації роботи алгоритмів. *Scientific Bulletin of UNFU*. 2022. № 32(5), с. 80-86. URL: https://doi.org/10.36930/40320511 (date of access: 10.10.2024).

10. Computer Science Curricula 2023. URL: https://csed.acm.org/wp-content/uploads/2023/03/Version-Beta-v2.pdf (date of access: 10.10.2024).

11. Цмоць, І. Г., Антонів, В. Я. Удосконалення паралельного сортування масивів чисел методом злиття. *Scientific Bulletin of UNFU*. 2020. №30(4), с. 134-142. URL: https://doi.org/10.36930/40300422 (date of access: 10.10.2024).

12. Мартинюк Т. Б., Круківський Б. І. Класифікаційний аналіз методів сортування. *Вісник Вінницького політехнічного інституту*. 2023. № 3 [13]/ URL: https://ir.lib.vntu.edu.ua//handle/123456789/42805 (date of access: 10.10.2024).

13. Коцовський В. М. Теорія паралельних обчислень: навчальний посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/38994/Навчальний%20посібник.pdf (date of access: 10.10.2024).

14. Черниш А. В. Моделі, методи та засоби побудови паралельних алгоритмів : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 123 Комп'ютерна інженерія. Харків: ХНУРЕ. 2024. 72 с. URL: https://openarchive.nure.ua/handle/document/27559 (date of access: 10.10.2024).

15. Мойсеєнко О.В, Гарасимів Т.Г. Використання генетичних алгоритмів при оптимізації процесу формування тест-планів програмного забезпечення комп'ютерних систем. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2023. Том 34 (73) № 3. URL: https://tech.vernadskyjournals.in.ua/journals/2023/3_2023/part_1/30.pdf (date of access: 10.10.2024).