

Корнага Ярослав Ігорович*Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ*

ORCID 0000-0001-9768-2615

Лемешко В'ячеслав Анатолійович*Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ*

ORCID 0009-0008-1495-715X

ПОРІВНЯННЯ АЛГОРИТМІВ MODIFIED FIRST-COME FIRST-SERVED TA REACTIVE MULTI-RESOURCE TOKEN – EARLIEST DEADLINE FIRST ЗА УМОВ ОБМЕЖЕНОГО ДОСТУПУ ДО ОПЕРАТИВНОЇ ПАМ'ЯТІ

Анотація. У статті представлено порівняльне експериментальне дослідження ефективності двох алгоритмів планування задач в умовах обмеженого доступу до оперативної пам'яті: модифікованого алгоритму Modified First-Come First-Served (MFCFS) та гібридної стратегії Reactive Multi-Resource Token – Earliest Deadline First (RMRT-EDF). У межах дослідження запропоновано вдосконалення класичного алгоритму FCFS, яке передбачає перевірку доступності оперативної пам'яті до моменту передачі задачі на виконання. Такий підхід дозволяє зменшити ймовірність переповнення системи і збоїв у роботі планувальника. Проте, у ситуаціях, коли перша у черзі задача не може бути виконана через нестачу ресурсів, навіть за наявності менш вимогливих задач у черзі, виникає ефект блокування, що негативно впливає на загальну продуктивність.

На противагу, алгоритм RMRT-EDF реалізує адаптивну політику планування, яка враховує як дедлайни задач, так і актуальний стан системних ресурсів. Завдяки концепції ресурсних токенів здійснюється відбір задач, що можуть бути оброблені в межах доступного ресурсу, з пріоритетом на найменший дедлайн. Крім того, в реалізації використано модуль моніторингу системних ресурсів з періодом оновлення 500 мс, що забезпечує актуальні дані про використання пам'яті в реальному часі та дозволяє алгоритму адаптуватися до змін навантаження.

Для оцінки ефективності було розроблено тестове середовище, яке імітує сценарії високої інтенсивності навантаження з понад 10 000 задач, що відрізняються за ресурсними характеристиками. Експерименти проведено в двох режимах: із застосуванням обмежень на пам'ять і без них. Результати показали, що в умовах дефіциту оперативної пам'яті алгоритм RMRT-EDF значно перевершує MFCFS за кількістю виконаних задач без порушення SLA, зберігаючи при цьому порівняльний рівень загальної продуктивності.

Отримані результати підтверджують доцільність застосування реактивних адаптивних стратегій планування в умовах обмежених обчислювальних ресурсів та слугують підґрунтям для подальших досліджень у напрямку оптимізації диспетчеризації у реальному часі.

Ключові слова: диспетчеризація задач, обмежені ресурси, FCFS, RMRT-EDF, дедлайн, SLA-порушення, бізнес-сервіси.

Kornaga Yaroslav*National Technical University of Ukraine 'Igor Sikorsky Kyiv Polytechnic Institute', Kyiv*

ORCID 0000-0001-9768-2615

Lemeshko Vyacheslav*National Technical University of Ukraine 'Igor Sikorsky Kyiv Polytechnic Institute', Kyiv*

ORCID 0009-0008-1495-715X

COMPARATIVE ANALYSIS OF MODIFIED FIRST-COME FIRST-SERVED AND REACTIVE MULTI-RESOURCE TOKEN – EARLIEST DEADLINE FIRST ALGORITHMS UNDER LIMITED ACCESS TO MAIN MEMORY

Abstract. This paper presents a comparative experimental study of the efficiency of two task scheduling algorithms under conditions of limited access to main memory: the Modified First-Come First-Served (MFCFS) algorithm and the hybrid strategy known as Reactive Multi-Resource Token – Earliest Deadline First (RMRT-EDF). As part of the study, an enhancement to the classical FCFS algorithm is proposed, incorporating a pre-execution check of memory availability. This modification reduces the likelihood of system overloads and scheduler failures. However, in cases where the first task in the queue cannot be executed due to insufficient memory—despite the presence of less resource-intensive tasks—the system experiences a blocking effect, which negatively impacts overall performance.

In contrast, the RMRT-EDF algorithm implements an adaptive scheduling policy that takes into account both task deadlines and the current state of system resources. Based on the concept of resource tokens, this approach prioritizes tasks with the earliest deadlines among those that fit within the available resources. Additionally, a resource

monitoring module with a 500 ms update interval is employed, providing real-time memory usage data and enabling the algorithm to dynamically adapt to workload fluctuations.

To evaluate the algorithms' effectiveness, a test environment was developed to simulate high-intensity workloads involving over 10,000 tasks with varying resource demands. The experiments were conducted in two modes: with and without memory constraints. The results demonstrated that under memory-limited conditions, RMRT-EDF significantly outperforms MFCFS in the number of successfully completed tasks without SLA violations, while maintaining a comparable level of overall performance.

The findings confirm the viability of using reactive, adaptive scheduling strategies in resource-constrained environments and lay the groundwork for further research into real-time dispatching optimization.

Keywords: task scheduling, resource constraints, FCFS, RMRT-EDF, deadline, SLA violation, business services.

Постановка проблеми

У сучасних інформаційних системах одним із ключових викликів є забезпечення стабільного та передбачуваного виконання задач в умовах обмежених обчислювальних ресурсів. Це стосується не лише великих інфраструктурних провайдерів, але й локальних інстанцій, де кожен екземпляр платформи автоматизації бізнес-сервісів працює у відокремленому екземплярі з фіксованими квотами на оперативну пам'ять, процесорний час, I/O та інші ресурси. У таких системах важливо забезпечити виконання задач відповідно до заданих дедлайнів, з мінімальними порушеннями угод про рівень обслуговування (SLA), навіть у випадках різкої зміни інтенсивності навантаження або обмеженої доступності ресурсів.

Незважаючи на значний прогрес у галузі досліджень алгоритмів диспетчеризації задач, більшість класичних методів, таких як FCFS (First-Come First-Served), орієнтовані передусім на збереження черговості обробки, нехтуючи при цьому контекстом поточного ресурсного стану системи або часовими вимогами задач. Це призводить до появи блокувань, надмірного споживання ресурсів, зниження пропускну здатності та збільшення кількості порушених дедлайнів. Проблема ще більше ускладнюється у випадках, коли ресурси розподіляються між задачами нерівномірно, а їх споживання є важко передбачуваним.

Сучасні тенденції вказують на необхідність переходу до дедлайн-орієнтованих та ресурсно-адаптивних стратегій планування, які можуть гнучко реагувати на зміни стану системи в режимі реального часу. Однак такі підходи часто базуються на складних математичних моделях або машинному навчанні, що ускладнює їх застосування в середовищах із обмеженою обчислювальною потужністю або там, де вимагається простота реалізації. Таким чином, виникає потреба у створенні легковагових, реактивних алгоритмів, здатних:

- враховувати поточну доступність ресурсів під час планування задач;
- забезпечувати пріоритезацію на основі дедлайнів;
- ефективно адаптуватися до змін навантаження без залучення складних обчислень;
- бути придатними для використання в умовах ізольованих обчислювальних середовищ з фіксованими квотами.

З урахуванням зазначеного, актуальність дослідження полягає у тому, щоб порівняти ефективність класичного алгоритму FCFS, доповненого механізмом перевірки ресурсів, із гібридним підходом RMRT-EDF (Reactive Multi-Resource Token - Earliest Deadline First) [1]. Останній базується на концепції токенів ресурсного навантаження, що дозволяє динамічно фільтрувати задачі, виходячи з доступних ресурсів, та обирати серед них ту, яка має найменший дедлайн. Водночас модифікований FCFS пропонує мінімальне втручання в логіку черговості, забезпечуючи перевірку ресурсної відповідності першої задачі у черзі.

Практична значущість полягає у створенні адаптивного симуляційного середовища, що дозволяє емпірично оцінити поведінку обох алгоритмів за різних рівнів навантаження та в умовах обмеження оперативної пам'яті. Це дозволяє не лише верифікувати теоретичні припущення щодо переваг дедлайн-орієнтованого планування, але й сформулювати практичні рекомендації для систем, де SLA має критичне значення, а надмірне ресурсне споживання неприпустиме.

Отже, поставлена проблема є важливою як з точки зору розвитку теорії планування задач у багаторесурсних середовищах, так і з позицій практичної реалізації гнучких та надійних систем диспетчеризації в умовах обмежених ресурсів. Подальші дослідження у цьому напрямку сприятимуть формуванню нових ефективних підходів до управління навантаженням у критичних і комерційно важливих інформаційних системах.

Аналіз останніх досліджень і публікацій

Забезпечення стабільного та передбачуваного виконання завдань у реальному часі в умовах обмежених обчислювальних ресурсів залишається одним із ключових викликів як для вбудованих

систем, так і для сучасних хмарних архітектур. Значна кількість досліджень присвячена вдосконаленню алгоритмів диспетчеризації з урахуванням дедлайнів, доступних ресурсів, енергоспоживання та особливостей апаратної архітектури.

У роботі [2] окреслено перспективи адаптивних алгоритмів, проте не подано детального аналізу або експериментального обґрунтування реактивних рішень, які враховують динамічний стан системи та багатовимірні обмеження ресурсів.

Представлений метод планування завдань [3] у розподілених системах за умов обмеженої інформації про ресурсні потреби, що ґрунтується на пошуку найближчих сусідів із використанням локалізованого хешування та алгебри скінченних предикатів. Хоча запропонований підхід демонструє ефективність при високій інтенсивності завдань, автори вказують на недоліки у стабільності при середньому та низькому навантаженні, а також на обмежену здатність до забезпечення гарантій виконання в рамках SLA-дедлайнів. Відсутність пріоритетної обробки задач та реактивної адаптації до системного стану зумовлює потребу в експериментальній перевірці гібридних рішень, з поєднанням пріоритетів дедлайнів та контролем доступних ресурсів у режимі реального часу.

У дослідженні [4] запропоновано метод моделювання часових характеристик задач у системах реального часу з використанням сіток Петрі та програмної симуляції роботи планувальників типу EDF і Non-Priority. Хоча результати моделювання демонструють переваги EDF із витисканням, зокрема менший час порушення дедлайну порівняно з безпріоритетною стратегією, дослідження не враховує динамічну змінність обчислювального середовища чи обмеження ресурсів ЦП/оперативної пам'яті. Відсутній аналіз реактивного реагування на зміну навантаження, що робить актуальним подальше експериментальне дослідження.

У роботі [5] проаналізовано часові характеристики підзадач DAG-задач (Directed Acyclic Graph tasks) і показано, що глобальне планування на основі лише загальних параметрів задач є надто песимістичним. Автор обґрунтував доцільність переходу до підхідного планування на рівні підзадач із врахуванням локальних параметрів, таких як джиттер (максимально можлива затримка у старті підзадачі після її активації) та локальні дедлайни. Це підсилює аргументацію щодо необхідності експериментальної перевірки реактивних алгоритмів.

Запропонований нелінійно-модифікований алгоритм EDF [6], що враховує не лише часові характеристики задач, але й динамічно розраховані дедлайни з урахуванням параметрів α , β та γ . Проведене моделювання показало зниження середнього часу очікування та порушень дедлайнів у порівнянні з FCFS, SJF та базовим EDF. Однак у роботі не розглянуто вплив локальних перевантажень, зміни доступності ресурсів у реальному часі, а також SLA-ризиків при різкій зміні інтенсивності навантаження.

У роботі [7] запропоновано уніфікований тест забезпечення виконуваності для широкого класу алгоритмів планування типу EDF-Like, зокрема для задач із самопризупиненням і довільними дедлайнами в однопроцесорних системах. Незважаючи на те, що отримані результати значно розширюють область застосування класичних тестів, дослідження не охоплює реактивні сценарії з непередбачуваними затримками чи змінами ресурсних обмежень у реальному часі.

Огляд класичних алгоритмів диспетчеризації задач здійснений [8], зокрема FCFS, SJF (Shortest Job First), RR (Round Robin), HRRN (Highest Response Ratio Next) та EDF, із фокусом на їхню продуктивність у різних середовищах, таких як time-sharing та вбудовані системи. Автори відзначають, що хоча SJF і HRRN забезпечують кращі показники часу очікування, значна частина досліджень зосереджена на оптимізації алгоритму Round Robin, особливо для систем із розподілом за часовими квантами. Проте, навіть попри наявність покращених варіацій EDF, наприклад, IEDF (Improved Earliest Deadline First) чи DVFS (Dynamic Voltage and Frequency Scaling)-адаптивних підходів, залишається відкритим питання ефективного розподілу задач з жорсткими дедлайнами при обмежених ресурсах.

У дослідженні [9] розглядається проблема мінімізації використання кешу при диспетчеризації задач з фіксованими та динамічними пріоритетами (FP та EDF) в умовах обмежених обчислювальних ресурсів. Автори доводять, що існуючі методи кеш-партиціювання здебільшого орієнтовані на забезпечення розкладу, нехтуючи оптимальним використанням кеш-пам'яті. Запропоновані ними евристичні та оптимізаційні моделі дозволяють досягти зменшення кеш-використання на понад 39%, зберігаючи гарантії виконання задач у реальному часі. Однак ці методи вимагають або апріорного знання про поведінку задач, або значних обчислювальних витрат. Таким чином, все ще актуальним є пошук гібридних стратегій.

У роботі [10] запропоновано модифікований алгоритм оптимізації рою частинок (randomized PSO) для розв'язання задачі диспетчеризації у хмарному середовищі з урахуванням вимог SLA. Автори акцентують увагу на мінімізації makespan та максимізації прибутку провайдера, демонструючи

перевагу над класичними підходами, такими як FCFS, SJF та GA. Однак, попри ефективність представленого метавристичного рішення у симульованому середовищі CloudSim, у роботі не розглянуто виконання задач з фіксованими дедлайнами SLA або реакцію алгоритму на зміну доступності ресурсів у реальному часі.

Отже, незважаючи на суттєвий прогрес у галузі ресурсно-обмеженого планування, залишаються нерозв'язаними такі ключові аспекти: забезпечення стабільного виконання задач за умов флуктуацій у ресурсному середовищі; дотримання SLA без надмірного резервування; адаптація до змін навантаження в реальному часі; інтеграція дедлайнів із контролем ресурсних обмежень.

Мета і задачі дослідження

Метою цього дослідження є підвищення ефективності планування задач у середовищах з обмеженими обчислювальними ресурсами шляхом експериментального порівняння традиційного алгоритму FCFS у його модифікованій версії (з контролем доступної оперативної пам'яті) та гібридного алгоритму RMRT-EDF, що враховує одночасно дедлайни та наявність критичних ресурсів. Робота спрямована на уточнення закономірностей впливу ресурсних обмежень на продуктивність системи та рівень дотримання SLA, а також на введення до наукового обігу нових експериментальних даних щодо порівняння реактивних та класичних методів диспетчеризації задач.

Основна ідея публікації полягає у доведенні того, що поєднання дедлайн-орієнтованого планування та реактивного контролю ресурсів дозволяє суттєво знизити кількість SLA-порушень та забезпечити стабільніше використання системних ресурсів у порівнянні з класичним методом. Дослідження вводить нові факти та висновки щодо поведінки алгоритмів у сценаріях пікового навантаження, демонструє закономірності блокувань задач у традиційних чергах без ресурсної адаптації та пропонує практичні рекомендації щодо використання RMRT-EDF у сервісно-орієнтованих і хмарних системах.

Для досягнення поставленої мети було визначено такі задачі дослідження:

- сформувати модель експериментального середовища, яке відображає роботу бізнес-сервісів з обмеженою кількістю ресурсів (оперативної пам'яті);
- реалізувати модифікований алгоритм MFCFS (Modified FCFS), що включає механізм перевірки доступної пам'яті перед передачею задачі на виконання;
- розробити та налаштувати RMRT-EDF із підтримкою багатресурсного токен-планування та пріоритезацією задач за дедлайнами⁴
- провести серію експериментів для оцінки ефективності обох алгоритмів у сценаріях із різним рівнем навантаження та за умов обмежень на використання оперативної пам'яті⁴
- виконати порівняльний аналіз отриманих результатів за показниками продуктивності, частоти SLA-порушень, стабільності використання ЦП та оперативної пам'яті.

Реалізація поставлених завдань дозволила отримати низку нових висновків, що уточнюють недостатньо вивчені аспекти взаємодії черговості задач, дедлайнів та динамічного розподілу обчислювальних ресурсів у реальних інформаційних системах.

Результати дослідження

У межах дослідження було реалізовано симуляційне середовище для порівняння ефективності двох стратегій планування задач: модифікованого MFCFS та гібридного алгоритму RMRT-EDF (позначено як RMRT-EDF). Обидві реалізації були розроблені з урахуванням потреби в адаптації до обмежених ресурсів, зокрема оперативної пам'яті, та підтримкою сценаріїв високого навантаження. Вихідним пунктом стала необхідність забезпечити стабільну роботу системи у середовищі з фіксованими квотами ресурсів на один екземпляр додатку.

Застосування немодифікованого алгоритму FCFS, який передає задачі на виконання без попередньої перевірки доступної оперативної пам'яті, є неприйнятним у режимі обмеження ресурсів. Таке виконання призводить до перевищення ліміту пам'яті й аварійного завершення роботи додатку. Саме тому для проведення експериментів у контрольованих умовах був реалізований модифікований варіант FCFS, який передбачає перевірку поточного обсягу доступної пам'яті перед розміщенням задачі в пулі виконання. У разі, якщо ресурсу недостатньо, задача залишається у черзі до наступної перевірки.

Псевдокод алгоритму MFCFS:

1. **while** (task queue is not empty):
2. updateAvailableResources()
3. task = task queue.peek()
4. **if** task.memoryRequirement $\leq$ availableMemory:
5. dequeue(task)

6. lockResourcesFor(task)
7. execute(task)
8. releaseResourcesAfterExecution(selectedTask)
9. **else:**
10. wait until resources become available or a task finishes

Пояснення до наведеного алгоритму:

1. while (task queue is not empty):
Основний цикл працює, поки черга задач не порожня.
2. updateAvailableMemory()
Оновлюється інформація про поточну кількість вільної оперативної пам'яті.
3. task = task queue.peek()
Зчитується перша задача з черги (без її видалення).
4. if task.memoryRequirement $\leq$ availableMemory:
Перевіряється, чи потребує задача не більше пам'яті, ніж наразі доступно.
5. dequeue(task)
Якщо пам'яті достатньо - задача видаляється з черги.
6. lockResourcesFor(task)
Резервуються ресурси (оперативна пам'ять) для обробки задачі.
7. execute(task)
Задача запускається на виконання.
8. releaseResourcesAfterExecution(selectedTask)
Після завершення задача звільняє зарезервовані ресурси.
9. else:
Якщо пам'яті недостатньо...
10. wait until resources become available or a task finishes
...алгоритм призупиняє виконання і очікує на звільнення ресурсів.

На відміну від класичного алгоритму FCFS, який намагається виконати задачі без перевірки ресурсів, що може призвести до перевищення ліміту пам'яті та аварійного завершення додатку, модифікований варіант MFCFS дозволяє забезпечити стабільну роботу в середовищах з обмеженою пам'яттю, уникаючи критичних ситуацій завдяки попередній перевірці доступних ресурсів.

Алгоритм RMRT-EDF реалізує більш гнучкий підхід: задачі фільтруються на основі їхніх ресурсних вимог, а з-поміж допустимих вибирається та, що має найменший дедлайн. Цей алгоритм демонструє адаптивну поведінку в обмеженому середовищі завдяки інтеграції з модулем моніторингу та контролю ресурсів, що оновлює метрики з періодичністю 500 мс.

Проведено ряд експериментів на робочій станції з такими характеристиками: процесор з тактовою частотою 3.2 GHz (4 фізичних ядра), 16 ГБ оперативної пам'яті, середовище виконання - .NET 8. Для політик розподілу задач на програмному рівні встановлено обмеження використання оперативної пам'яті в 550 МБ, що відображає одну із умов ізольованих екземплярів платформ виконання бізнес-сервісів.

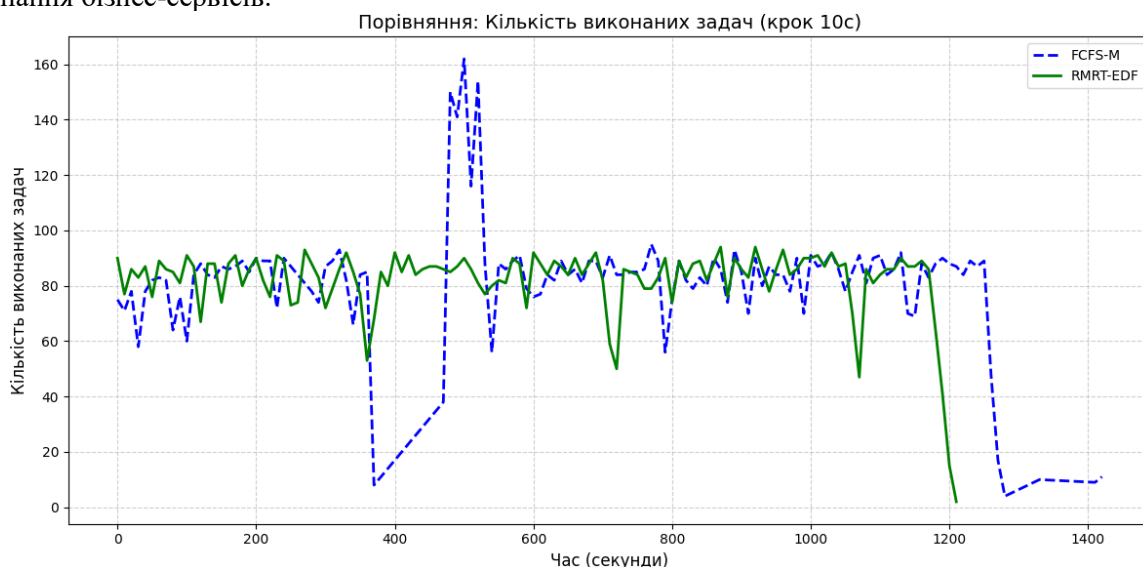


Рис. 1. Продуктивності двох алгоритмів планування задач (MFCFS та RMRT-EDF)

Для перевірки алгоритмів було згенеровано тестові дані у вигляді 1339 сценаріїв, що репрезентують моделі бізнес-процесів із послідовною структурою етапів; у сукупності вони охоплюють 10 000 завдань із варіативними параметрами споживання ресурсів. Для кожного завдання встановлено жорстке обмеження часу виконання тривалістю 1 с. Для моделювання пікових навантажень упродовж експерименту тричі здійснювалося штучне додавання до черги завдань із вимогою виділення 128 МБ оперативної пам'яті, що дозволило оцінити реакцію системи на тимчасове зменшення доступних обчислювальних ресурсів.

Графік (Рис. 1) демонструє, що алгоритм RMRT-EDF забезпечує стабільну та передбачувану кількість виконаних задач протягом усього періоду спостереження, тоді як MFCFS характеризується різкими коливаннями: від піків продуктивності до повних провалів виконання. Це свідчить про те, що RMRT-EDF краще адаптується до змін навантаження та ефективніше використовує обмежені ресурси, тоді як MFCFS, попри можливі короткочасні сплески, є менш надійним у середовищах із жорсткими ресурсними обмеженнями.



Рис. 2. Використання ресурсів під час розподілу задач для (MFCFS та RMRT-EDF) з обмеженням на ресурси

Використання ЦП RMRT-EDF (Рис. 2, верхня частина графіку) демонструє стабільне і рівномірне завантаження процесора в діапазоні 60–80%, без значних провалів. Натомість MFCFS характеризується значною фрагментацією навантаження: присутні як високі піки, так і періоди з майже нульовим завантаженням. Це свідчить про ефективніше розподілення обчислювальних задач RMRT-EDF, особливо в умовах обмежених ресурсів.

Обидва алгоритми працюють у межах доступного обсягу пам'яті (Рис. 2, нижня частина графіку), однак MFCFS демонструє стрибкоподібну зміну рівня споживання, що вказує на періодичне очікування або звільнення ресурсів. RMRT-EDF забезпечує більш плавну динаміку використання пам'яті, з помірною амплітудою коливань, що вказує на кращу адаптацію до доступного обсягу ресурсів.

RMRT-EDF забезпечує стабільніше використання як процесорного часу, так і оперативної пам'яті, що є критичним для систем з обмеженими ресурсами. Модифікований MFCFS, хоча й уникає аварійних збоїв, має менш передбачувану поведінку, що може знижувати загальну ефективність при високому навантаженні.

Висновки і перспективи подальших досліджень

У межах проведеного дослідження було запропоновано та реалізовано модифікацію алгоритму FCFS з урахуванням контролю доступної оперативної пам'яті, що дало змогу уникнути аварійного завершення процесу планування у випадках дефіциту ресурсу. Також було реалізовано та випробувано

гібридний алгоритм RMRT-EDF, який поєднує дедлайн-орієнтовану стратегію з механізмом реактивного фільтрування задач за ресурсними обмеженнями.

Для об'єктивного порівняння було проведено серію експериментів із симуляцією понад 10000 задач у середовищі з програмно встановленим обмеженням оперативної пам'яті (550 МБ). Основні результати зіставлення ефективності двох алгоритмів наведено в таблиці 1.

Кількість SLA-порушень зменшено у 11,7 разів (з 831 до 71), що свідчить про значно вищу надійність обслуговування критичних задач, що становить зниження на 91.4%. Середнє використання оперативної пам'яті знижено на 10.1%, що вказує на ефективніше управління ресурсами без перевищення межі. Центральний процесор у RMRT-EDF утилізується краще - середній рівень завантаження збільшився з 50.86% до 61.96%, що на 21.8% більше, забезпечуючи рівномірніший розподіл навантаження. Максимальна кількість задач за секунду у MFCFS вища, однак це пов'язано зі стрибкоподібною та нестійкою обробкою, яка в умовах обмежень призводить до перевантаження та збоїв. Методологія дослідження, що включає моніторинг ресурсів з інтервалом 500 мс та модуль перевірки доступності, дозволила виявити закономірності блокування черги у MFCFS та стабільної обробки задач у RMRT-EDF. Інструментальна реалізація, забезпечує масштабованість підходу, а також можливість адаптації для інших типів ресурсів, таких як I/O або GPU, та потенційну інтеграцію прогнозних моделей у режимі реального часу.

Таблиця 1

Результати експериментального порівняння MFCFS і RMRT-EDF у середовищі з обмеженням оперативної пам'яті

Показник	MFCFS	RMRT-EDF
Середня кількість (задач/с)	8	8
Максимальна кількість (задач/с)	24	14
Кількість SLA-порушень	831	71
Середнє використання ЦП (%)	50.86	61.96
Середнє використання ОП (МБ)	330.81	297.49
95-й перцентиль використання ОП (МБ)	509.69	453.47
95-й перцентиль використання ЦП (%)	87.1	85.94

До перспективних напрямів подальших досліджень можна віднести: удосконалення алгоритму RMRT-EDF шляхом модифікації модулів алокації та моніторингу ресурсів із можливістю врахування короткотермінового прогнозу їхнього використання в межах фіксованого горизонту планування (feasible horizon); розробку та експериментальну перевірку прогнозного модуля короткострокового навантаження, що дозволить здійснювати адаптивне резервування обчислювальних ресурсів відповідно до очікуваної інтенсивності вхідного потоку задач; моделювання поведінки планувальних стратегій у середовищах із варіативною тривалістю задач, що є характерним для реальних сервісно-орієнтованих систем з непередбачуваною структурою виконання; проведення порівняльного аналізу RMRT-EDF із нейромережевими підходами до планування задач, зокрема тими, що базуються на алгоритмах навчання з підкріпленням (reinforcement learning), з метою визначення граничної ефективності в умовах SLA-критичних навантажень.

Таким чином, отримані результати підтверджують доцільність впровадження RMRT-EDF як адаптивної, ресурсно-ефективної стратегії планування задач у системах з обмеженим ресурсним середовищем і високими вимогами до стабільності обслуговування.

Список використаних джерел

1. Корнага Я. І., Лемешко В. А. RMRT-EDF: реактивний мультиресурсний токен-планувальник для гарантованого запуску задач у реальному часі // InfoCom Advanced Solutions 2025. – Київ, 2025. – С. 35–38. – Режим доступу: https://ist.kpi.ua/wp-content/uploads/2025/05/infocom-advanced-solutions-2025_compressed-3.pdf.
2. Пасічник М. Ю., Зайцев В. Г. Методи диспетчеризації завдань у системах реального часу // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2024. – № 56. – С. 27–35. – DOI: <https://doi.org/10.36910/6775-2524-0560-2024-56-04>
3. Козирєв А., Шубін І. Метод планування завдань оброблення даних у розподілених системах з обмеженою інформацією про доступні ресурси // Сучасний стан наукових досліджень та технологій в промисловості. – 2023. – № 3(25). – С. 27–38. – DOI: <https://doi.org/10.30837/ITSSI.2023.25.027>.
4. Зайцев В. Г., Цибаєв Є. І. Оцінка часових характеристик у комп'ютерних системах реального часу з використанням сіток Петрі // Управління розвитком складних систем. – 2023. – № 54. – С. 48–62. – DOI: <https://doi.org/10.32347/2412-9933.2023.54.48-62>

5. Скільков Н. В. Дослідження часових характеристик підзадач в багатопроцесорних системах реального часу / Н. В. Скільков // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. — 2023. — № 52. — С. 111–118. — DOI: <https://doi.org/10.36910/6775-2524-0560-2023-52-14>.
6. Nutulapati, N. P., Bandarupalli, S. B., Babu, B. S. K., Subbulakshmi, R. Nonlinear Analysis of EDF Scheduling Framework for Resource Distribution in Cloud Computing // Communications on Applied Nonlinear Analysis. — 2025. — Vol. 32, No. 1. — P. 372–384. — ISSN 1074-133X.
7. Günzel M., Chen K.-H., Chen J.-J. EDF-Like Scheduling for Self-Suspending Real-Time Tasks [Electronic resource] // arXiv preprint. — 2021. — arXiv:2111.09725. — Mode of access: <https://arxiv.org/abs/2111.09725>.
8. Li L. Advances in CPU Scheduling Algorithms in Operating Systems // Highlights in Science, Engineering and Technology. Proceedings of the AMCCE 2024 Conference. — 2024. — Vol. 120. — P. 8–13.
9. Sun B., Kloda T., Arribas Garcia S., Gracioli G., Caccamo M. Minimizing cache usage with fixed-priority and earliest deadline first scheduling // Real-Time Systems. — 2024. — Vol. 60. — P. 625–664. — DOI: <https://doi.org/10.1007/s11241-024-09423-7>.
10. Lipsa S., Dash R. K. SLA-based task scheduling in cloud computing using randomized PSO algorithm // Proceedings of the Symposium on Computing and Intelligent Systems (SCI 2024), May 10–11, 2024, New Delhi, India. — CEUR Workshop Proceedings. — 2024. — [Electronic resource]. — Access mode: <https://www.researchgate.net/publication/380664615>.