

ФОРМАЛІЗАЦІЯ ПРОЦЕСУ ПЛАНУВАННЯ ЗА МЕТОДОМ RMRT-EDF З УРАХУВАННЯМ ОБМЕЖЕНИХ РЕСУРСІВ

Анотація: Стаття присвячена формалізації процесу планування задач у системах з обмеженими обчислювальними ресурсами за допомогою гібридного підходу RMRT-EDF (Reactive Multi-Resource Token – Earliest Deadline First). Метою дослідження є розроблення дискретно-часової математичної моделі, що забезпечує гарантований запуск кожної задачі не пізніше ніж через 1 с після надходження та водночас дотримання багатовимірних обмежень CPU, оперативної пам'яті та I/O.

Запропонована модель інтерпретує планувальник як замкнену систему автоматичного керування зі зворотним зв'язком: задачі спершу проходять токен-фільтрацію ресурсів, після чого упорядковуються за правилом EDF; кількість одночасних запусків визначається адаптивним лімітом, керованим Model Predictive Control із ковзним горизонтом оптимізації.

Ключові слова: RMRT-EDF, MPC, планування задач, ресурсні обмеження, EDF, токен-модель, диспетчеризація задач

Вступ

У сучасних умовах цифровізації економіки, зростання складності інформаційно-керуючих систем та підвищення вимог до їхньої швидкодії обумовлює необхідність застосування нових адаптивних методів планування задач. Зокрема, у системах з обмеженими обчислювальними ресурсами (CPU, оперативна пам'ять, підсистема введення/виведення) і критичними часовими вимогами (реального часу) важливим є гарантоване забезпечення запуску задач у межах допустимого інтервалу - зазвичай не більше 1 секунди.

Класичні підходи до диспетчеризації задач, що базуються на статичних або детермінованих евристичних, не здатні адекватно реагувати на динамічні зміни вхідного потоку задач, флуктуації ресурсоемності та непередбачувані системні збурення. Актуальність проблеми підсилюється зростанням кількості веб- та IoT-сервісів, чутливих до затримок, де порушення сервісного рівня (SLA) безпосередньо пов'язане з фінансовими втратами або деградацією якості обслуговування.

У цьому контексті перспективним є застосування гібридної реактивної стратегії RMRT-EDF (Reactive Multi-Resource Token – Earliest Deadline First) [1], яка поєднує дедлайн-орієнтоване планування, контроль ресурсів за допомогою токен-моделі та механізми адаптивного самоналаштування. Запропонований підхід розглядається як

замкнена система автоматичного керування з багатоканальним зворотним зв'язком, що забезпечує стійке функціонування при високих динамічних навантаженнях.

Аналіз літературних джерел

Сучасні дослідження з планування задач у системах із жорсткими часовими вимогами та обмеженими обчислювальними ресурсами можна згрупувати за п'ятьма провідними напрямками: статичні евристичні алгоритми; реактивні алгоритми реального часу; метаетристичні та гібридні підходи; токен-орієнтовані та ресурсо-обмежені моделі; енергоефективні та дедлайн-орієнтовані стратегії.

Методи групи статичних евристичних алгоритмів формують розклади апріорно, спираючись на відомі параметри задач та ресурсів. Класичні дисципліни FCFS, SJF і Round Robin демонструють високу швидкодію та низьку складність реалізації, однак мають обмежену адаптивність у динамічних середовищах. Результати експериментального порівняння алгоритмів [3; 8] підтверджують, що за фіксованого навантаження RR забезпечує найкращий баланс між пропускну здатністю й середнім часом очікування, тоді як FCFS є найпростішим у реалізації. Теоретичні засади дедлайн-орієнтованого планування для жорсткого реального часу закладено у працях [14; 16], де доведено оптимальність EDF для однопроцесорних систем, але за умови відсутності ресурсних обмежень.

Реактивні алгоритми реального часу коригують чергу виконання під дією зовнішніх подій і базуються на постійному зворотному зв'язку з телеметрії. У роботах [9; 10; 17] показано, що Earliest Deadline First та його модифікації здатні забезпечити низьку частоту пропусків дедлайнів навіть за пікового навантаження. Проте відсутність механізму попередньої перевірки наявних ресурсів призводить до конфліктів за CPU та пам'ять і, відповідно, до погіршення загальної пропускну здатності системи.

Для подолання комбінаторної вибуховості та локальних мінімумів активно застосовують еволюційні та роєві методи. Алгоритми Gray Wolf Optimization [2] і Bees Life [6] демонструють високу якість розкладів, а гібридні схеми з машинним навчанням [11; 12; 13] адаптивно підлаштовують пріоритети задач у процесі виконання. Основним недоліком залишається відсутність строгих гарантій дотримання дедлайнів і задоволення багатовимірних ресурсних обмежень..

Токен-моделі репрезентують доступні ресурси у вигляді дискретних «жетонів», що спрощує контроль взаємного виключення. У роботі [7] введено концепцію «кошика» токенів для CPU та RAM, яка дозволяє відфільтровувати задачі ще до їх потрапляння у чергу планувальника. Недоліком підходу є відсутність механізму адаптації до неочікуваних стрибків навантаження та підтримки кількох типів ресурсів одночасно.

З огляду на зростання питомого енергоспоживання центрів обробки даних розвиваються багатокритеріальні алгоритми, що поєднують мінімізацію енергії з дотриманням часових обмежень. Метод EAEFA [5] і гібридні енергозберігальні алгоритми [12] показують помітне зниження споживання енергії, але виконують запуск

задач із затримкою, що перевищує 1с, і тому непридатні для latency-sensitive веб-сервісів.

Наведений аналіз демонструє відсутність універсальної стратегії, здатної гарантовано запускати усі задачі протягом 1 с після надходження та водночас утримувати споживання CPU, RAM та I/O у встановлених межах. Таким чином, залишається актуальною розробка гібридних реактивних підходів, які б об'єднали дедлайн-орієнтоване сортування, попередній допуск за токен-моделлю та адаптивне регулювання пулу воркерів — саме цим вимогам відповідає метод RMRT-EDF, детально розглянутий у наступних розділах.

Постановка задачі дослідження

У сучасних обчислювальних середовищах, що характеризуються високою динамікою надходження задач, непередбачуваністю їх ресурсоємності та жорсткими обмеженнями за часом (реального часу), виникає необхідність у побудові адаптивних систем керування, здатних забезпечити своєчасне виконання задач без перевищення доступних ресурсів. Одним із перспективних підходів до вирішення цієї проблеми є використання гібридної стратегії планування RMRT-EDF, яка поєднує принципи керування за відхиленням, за збуренням та адаптивного самоналаштування.

У рамках теорії автоматичного керування система з RMRT-EDF може бути інтерпретована як замкнена система автоматичного регулювання зі зворотним зв'язком, де:

- об'єкт керування - це пул потоків, яка реалізує запуск задач;
- керуючий елемент - модуль допуску та диспетчер, що приймають рішення про запуск задач;
- виконавчий елемент - потоки з пулу виконання задач;
- вимірювальний елемент – телеметрична підсистема, що оцінює фактичне використання ресурсів;
- збурюючі впливи – непередбачуваний вхідний потік задач зі змінними дедлайнами та профілями споживання ресурсів;
- зворотний зв'язок – телеметричні дані про якість виконання задач, використання ресурсів і частоту порушення дедлайнів.

Метою дослідження є побудова математичної моделі функціонування RMRT-EDF, що реалізує адаптивне регулювання запуску задач із врахуванням багатовимірних обмежень за ресурсами (CPU, RAM, I/O) і часових дедлайнів.

Математична модель

У запропонованій математичній моделі RMRT-EDF система описується як дискретно-часова система керування, що функціонує з фіксованим тактом планування. В таблиці 1 наведено основні параметри та позначення, які використовуються в даному розділі.

Таблиця 1.

Основні позначення параметрів

Позначення	Опис
r_i	$r_i = (c_i, m_i, io_i)$ потреба в CPU, RAM, I/O - токенах
e_i	тривалість
d_i	Дедлайн
$J_i = (r_i, e_i, d_i)$	Профіль i -ої задачі
$R = (C_{max}, M_{max}, L_{max})$	Глобальна місткість токена
P_{max}	Максимальний розмір пулу потоків
$k \in N$	Дискретний крок часу $\Delta t = 1c$
$\hat{r}_i(k), \hat{e}_i(k)$	Прогноз потреби та тривалості задачі на кроці k
A_k	Множина задач, що надійшли від тригерів на кроці k
S_k	Множина задач, стартованих на k
F_k	Множина задач, що завершилися на k
D_k	Допустимі (після токен-фільтра) джоби на k
H	Горизонт оптимізації в receding-horizon (від 5 до 20 кроків)
$\alpha_{1...4}$	Вагові коефіцієнти [13] в цільовій функції (6)

Основними змінними стану системи є: вектор доступних токенів для ресурсів (CPU, RAM, IO), довжина черги очікування задач, кількість активних потоків у пулі виконавців.

$$x_k = (C_k, M_k, I_k, q_k, p_k), \quad (1)$$

де C_k, M_k, I_k – вільні токени CPU, RAM, I/O; q_k – довжина черги; p_k – кількість активних потоків в пулі потоків.

Черга зростає, коли приходять нові задачі, та скорочується, коли їх запускають; пул потоків відбиває поточне завантаження виконавців.

$$\begin{aligned} q_k &= q_{k-1} + |A_k| - |S_k| \\ p_k &= p_{k-1} + |S_k| - |F_k|. \end{aligned} \quad (2)$$

Задачі, що надходять, проходять токен-фільтрацію, тобто допускаються лише за наявності достатньої кількості відповідних ресурсів. Допущені задачі впорядковуються за правилом Earliest Deadline First (EDF). Кількість задач, які запускаються, визначається адаптивним лімітом, що залежить від поточного стану ресурсів.

$$x_{k+1}^{tokens} = (C_k, M_k, I_k) - \sum_{J_i \in S_k} \hat{r}_i + \sum_{J_i \in F_k} \hat{r}_i, \quad (3)$$

де $\hat{r}_i$ прогноз споживання ресурсів задачею i .

При запуску токени віднімаються; після завершення – повертаються. Правило старту задач з допомогою токен-фільтрації визначається за формулою:

$$\hat{r}_i(k) \leq (C_k, M_k, I_k) \rightarrow J_i \in D_k \quad (4)$$

Сортування задач відбувається за правилом EDF: задачі в D_k упорядковуються за зростанням d_i . Перші s_k задач із черги D_k запускаються, якщо $l_i(k) = 1$. Ліміту стартів задач розраховується за формулою (5).

$$s_k = \min(P_{max} - p_k, P_k, |D_k|),$$

$$P_k = \frac{\beta c_k}{c} \quad (5)$$

де β – коефіцієнт чутливості адаптивного ліміту до поточних CPU-токенів; $\bar{c}$ – середній запит CPU-токенів.

Особливість підходу – застосування оптимізаційного механізму Model Predictive Control (MPC) [18] з ковзним горизонтом планування H , що дозволяє на кожному кроці обирати такий набір задач, який мінімізує багатокритеріальну цільову функцію, враховуючи: затримку старту задач (latency), кількість порушень дедлайнів, відхилення від бажаного профілю використання ресурсів.

$$\frac{\min}{l_i(k:k+H-1)} \sum_{j=0}^{H-1} (\alpha_1 miss_{k+j} + \alpha_2 delay_{k+j} + \alpha_3 \|x_{k+j} - \bar{x}\|^2 + \alpha_4 q_{k+j}), \quad (6)$$

$\bar{x}$ – бажаний/плановий профіль використання ресурсів; $miss_{k+j}$ – кількість дедлайн-пропусків, індикатор (0/1) пропуску дедлайну на кроці $k+j$; $delay_{k+j}$ – середня затримка старту.

$$\begin{cases} x_{k+j+1} = f(x_{k+j}, u_{k+j}) \\ \hat{r}_i(k+j) \leq (C_{k+j}, M_{k+j}, I_{k+j}), \\ p_{k+j} \leq P_{max} \\ delay(J_i) \leq 1 \end{cases} \quad (7)$$

де $u_{k+j} = l_i(k+j)$ - бінарні рішення про запуск.

Після розв'язання плану застосовується лише перший крок $l_i(k)$; горизонт зсувається на $k+1$.

Таким чином, запропонована система не лише реагує на поточні умови, а й здійснює прогнозування майбутнього стану, що є характерним для інтелектуальних САК із передбаченням.

Висновки

У роботі представлено функціональний та математичний підхід до побудови автоматизованої системи керування запуском задач із використанням гібридного алгоритму RMRT-EDF. Запропонована система розглядається як дискретно-часова САК з багатоконтурним зворотним зв'язком, у якій елементи диспетчеризації, допуску та телеметрії реалізують функції відповідно до класичної структури автоматичного регулювання. Запропонована математична модель враховує ресурсні обмеження CPU, RAM, I/O, дедлайни задач, а також застосовує адаптивне управління через токен-механізм і метод Model Predictive Control (MPC) з ковзним горизонтом оптимізації.

Проведений аналіз літературних джерел підтвердив, що традиційні підходи не забезпечують гарантованого дотримання як дедлайнів, так і ресурсних обмежень в умовах змішаного динамічного навантаження. У цьому контексті RMRT-EDF виступає як перспективна стратегія, що поєднує переваги реактивних та адаптивних САК, забезпечуючи більш передбачувану і стабільну поведінку обчислювальної системи при високих навантаженнях.

У межах подальших досліджень планується провести імітоване моделювання роботи системи на різних профілях навантаження; дослідити обмеження роботи запропонованого підходу; дослідити стійкість і керованість системи за допомогою методів аналізу ТАК.

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. URL: https://ist.kpi.ua/wp-content/uploads/2025/05/infocom-advanced-solutions-2025_compressed-3.pdf
2. Naseem Adnan AlsamaraiOsman Nuri Uçan, Improved Performance and Cost Algorithm for Scheduling IoT Tasks in Fog–Cloud Environment Using Gray Wolf Optimization Algorithm // Applied Sciences. – 2024. – Т. 14, № 4. – С. 1670. – DOI: 10.3390/app14041670. – Режим доступу: <https://www.mdpi.com/2076-3417/14/4/1670>
3. Abubakar Sani Jato Performance Evaluation of Operating Systems Scheduling Algorithms // ResearchGate. – 2024. – Режим доступу: https://www.researchgate.net/publication/379263571_PERFORMANCE_EVALUATION_OF_OPERATING_SYSTEMS_SCHEDULING_ALGORITHMS.
4. Liang, F., Zhang, Z., Lu, H., Li, C., Leung, V. C. M., Guo, Y., & Hu, S. Resource Allocation and Workload Scheduling for Large-Scale Distributed Deep Learning: A Survey [Electronic resource] / F. Liang, Z. Zhang, H. Lu, C. Li, V. C. M. Leung, Y. Guo, S. Hu // arXiv preprint. – 2024. – June. – arXiv:2406.08115. – Mode of access: https://www.researchgate.net/publication/381372558_Resource_Allocation_and_Workload_Scheduling_for_Large-Scale_Distributed_Deep_Learning_A_Survey.
5. Kumar, M. S., & Karri, G. R. EAEFA: An Efficient Energy-Aware Task Scheduling in Cloud Environment [Електронний ресурс] / M. S. Kumar, G. R. Karri // EAI Endorsed Transactions on Scalable Information Systems. – 2023. – Т. 11, № 3. – Режим доступу: https://www.researchgate.net/publication/374071179_EAEFA_An_Efficient_Energy-Aware_Task_Scheduling_in_Cloud_Environment
6. Bitam, S. Bees Life Algorithm for Job Scheduling in Cloud Computing [Електронний ресурс] / S. Bitam // Proceedings of the 2012 International Conference on Computer Engineering & Systems (ICCES). – 2012. – С. 186–191. – Режим доступу: https://www.researchgate.net/publication/287656786_Bees_life_algorithm_for_job_scheduling_in_cloud_computing
7. Smesseim, A. Dynamic Thread Allocation for Distributed Jobs using Resource Tokens [Електронний ресурс] / A. Smesseim. – 2019. – 59 с. – Магістерська дисертація. – ETH Zurich. – Режим доступу: https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/362308/1/Smesseim_Ali.pdf
8. Al-Haidari, F., & Balharith, T. Enhanced Round-Robin Algorithm in the Cloud Computing Environment for Optimal Task Scheduling [Електронний ресурс] / F. Al-Haidari, T. Balharith // ResearchGate. – 2021. – Режим доступу: <https://www.researchgate.net/>

[publication/351453682_Enhanced_Round-Robin_Algorithm_in_the_Cloud_Computing_Environment_for_Optimal_Task_Scheduling](#)

9. Zhou, J., Liu, B., & Gao, J. A task scheduling algorithm with deadline constraints for distributed clouds in smart cities [Електронний ресурс] / J. Zhou, B. Liu, J. Gao // PeerJ Computer Science. – 2023. – Vol. 9. – Article e1346. – DOI: 10.7717/peerj-cs.1346. – Режим доступу: <https://peerj.com/articles/cs-1346/>

10. Abohamama, A. S., El-Ghamry, A., & Hamouda, E. Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud-Fog Environment [Електронний ресурс] / A. S. Abohamama, A. El-Ghamry, E. Hamouda // Journal of Network and Systems Management. – 2022. – Vol. 30, Article 54. – DOI: 10.1007/s10922-022-09664-6. – Режим доступу: https://www.researchgate.net/publication/361715957_Real-Time_Task_Scheduling_Algorithm_for_IoT-Based_Applications_in_the_Cloud-Fog_Environment

11. Huang Y., Zhang S., Wang B. An Improved Genetic Algorithm with Swarm Intelligence for Security-Aware Task Scheduling in Hybrid Clouds // Electronics. – 2023. – Vol. 12, No. 9. – Article No. 2064. – DOI: <https://doi.org/10.3390/electronics12092064>

12. Garba, A. J., Gital, A. Y., Boukari, S., & Zambuk, F. U. (2023). Hybrid Meta-Heuristics Based Task Scheduling Algorithm for Energy Efficiency in Fog Computing. International Journal of Advanced Scientific Research and Engineering, 9(2), 1–8. <https://doi.org/10.31695/IJASRE.2023.9.2.3>

13. Wang J., Li S., Zhang X., Wu F., Xie C. Deep reinforcement learning task scheduling method based on server real-time performance // PeerJ Computer Science. – 2024. – Vol. 10. – Article e2120. – DOI: <https://doi.org/10.7717/peerj-cs.2120>

14. Liu C. L., Layland J. W. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment // Journal of the ACM. – 1973. – Vol. 20, No. 1. – P. 46–61. – DOI: <https://doi.org/10.1145/321738.321743>.

15. Le Boudec J.-Y., Thiran P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. – Berlin: Springer, 2001. – 274 p. – (Lecture Notes in Computer Science; Vol. 2050). – ISBN 978-3-540-42184-9

16. Buttazzo G. C. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. – 3rd ed. – New York: Springer, 2011. – 524 p. – (Real-Time Systems Series; Vol. 24). – ISBN 978-1-4614-0675-4. – DOI: 10.1007/978-1-4614-0676-1

17. S. A. A. Shah, M. A. Khan, M. A. Khan, and M. A. Khan. Designing and Implementing of Earliest Deadline First Scheduling Algorithm on Standard Linux // 2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS). – 2010. – P. 107–112. – DOI: <https://doi.org/10.1109/CISIS.2010.34>

18. Zak S. H. An Introduction to Model-based Predictive Control (MPC) // ECE 680 – Fall 2017. – 2017. – 22 p. – Available at: <https://www.researchgate.net/profile/Mohamed-Mourad-Lafifi/post/Model-Predictive-Control-examples/attachment/60202ac761fb570001029f61/AS%3A988637009301508%401612720839656/download/An+Introduction+to+Model-based+Predictive+Control+%28MPC%29.pdf>