

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ
СЕМЕНА КУЗНЕЦЯ МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ
СЕМЕНА КУЗНЕЦЯ МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

ЄНГАЛИЧЕВ СЕРГІЙ ОЛЕКСАНДРОВИЧ

УДК 004.03(043.3)

ДИСЕРТАЦІЯ

МЕТОД ПЛАНУВАННЯ ЗАДАЧ У РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ
СИСТЕМАХ ЗА НЕВИЗНАЧЕНОСТІ МЕТАДАНИХ

Спеціальність 122 «Комп'ютерні науки»

Галузь знань 12 «Інформаційні технології»

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ С. О. Єнгалічев

(підпис, ініціали та прізвище здобувача)

Науковий керівник: Семенов Сергій Геннадійович, доктор технічних наук,
професор

Дисертація є ідентичною іншим примірникам дисертації

Голова разової спеціалізованої вченої ради ДФ 64.055. _____

д.т.н., професор _____

Харків – 2026

АНОТАЦІЯ

Єнгаличев С. О. Метод планування задач у розподілених інформаційних системах за невизначеності метаданих. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки», галузь знань 12 «Інформаційні технології». Харківський національний економічний університет імені Семена Кузнеця Міністерства освіти і науки України. – Харків, 2026.

Дисертація є теоретико-методичним дослідженням, присвяченим удосконаленню методів планування задач у розподілених інформаційних системах в умовах невизначеності метаданих. Робота спрямована на розв'язання проблем ефективного розподілу обчислювальних ресурсів у гетерогенних та динамічних середовищах з урахуванням неповноти, неточності та змінності інформації про характеристики задач, ресурсів і середовища виконання. Провідна ідея дослідження полягає у розробленні нового методу планування задач, що забезпечує підвищення ефективності функціонування розподілених систем шляхом інтелектуальної обробки метаданих та адаптації процесу планування за умов їх невизначеності.

Актуальність теми обумовлена стрімким розвитком розподілених обчислювальних середовищ, зокрема хмарних, туманних та крайових систем, зростанням складності обчислювальних задач і необхідністю забезпечення високої продуктивності, масштабованості та надійності їх виконання. Аналіз сучасних наукових досліджень і практичних рішень показав, що більшість існуючих методів планування задач ґрунтуються на припущенні повної та точної інформації про параметри системи, що не відповідає реальним умовам функціонування розподілених інформаційних систем. Це призводить до зниження ефективності планування, перевитрат ресурсів та погіршення показників якості обслуговування, що зумовлює потребу в науковому

обґрунтуванні та розробленні нових методів планування задач за умов невизначеності метаданих.

Об'єктом дослідження є процес планування та розподілу задач в гетерогенних розподілених інформаційних системах.

Предметом дослідження є метод планування задач у розподілених інформаційних системах з урахуванням невизначеності метаданих.

У першому розділі дисертаційної роботи виконано системний аналіз процесу планування задач у розподілених комп'ютерних та інформаційних системах. Сформульовано та проаналізовано вимоги до процесу планування з урахуванням гетерогенності обчислювальних ресурсів, динамічності навантаження, обмежень за продуктивністю та необхідності ефективного використання обчислювальних потужностей.

Розглянуто та узагальнено основні підходи до підвищення продуктивності розподілених інформаційних систем, зокрема шляхом застосування паралельних обчислень, оптимізації структури обчислювальних процесів і вдосконалення методів управління ресурсами. Проведено аналіз систем автоматизованого проєктування цифрових пристроїв та методів синтезу функціональних структур спеціалізованих процесорів, що дозволило визначити ефективні принципи апаратно-алгоритмічної реалізації обчислювальних задач. Обґрунтовано вибір показників ефективності паралельної реалізації алгоритмів, які застосовуються для кількісного оцінювання результатів планування та продуктивності виконання задач.

У завершальній частині розділу здійснено аналіз відомих планувальників завдань і систем управління ресурсами, визначено їх функціональні можливості, переваги та обмеження. Виявлено невирішені проблеми, пов'язані з використанням існуючих методів планування в умовах неповноти та невизначеності метаданих, що обґрунтовує доцільність подальших досліджень та визначає напрями розроблення нового методу планування задач у розподілених інформаційних системах.

У другому розділі дисертаційної роботи розроблено та досліджено моделі опрацювання даних і логічного розмежування доступу в інформаційних системах з урахуванням різноманітності сутностей та особливостей їх взаємодії. На основі формалізованої постановки завдання проаналізовано підходи до організації доступу до даних у складних розподілених середовищах, що характеризуються неоднорідністю типів даних, суб'єктів і ресурсів.

Розглянуто методи об'єднання простіших рольових моделей і виконано синтез ієрархічних рольових моделей, що дозволяє забезпечити масштабованість і гнучкість політик доступу. Запропоновано моделі примусової типізації на основі відносин довіри, які забезпечують підвищення рівня узгодженості та безпеки при обробленні даних у багатокомпонентних інформаційних системах. Сформульовано основні положення процесу об'єднання політик безпеки, що враховують відмінності між сутностями та обмеження, зумовлені їх функціональними ролями.

Проведено порівняльні дослідження моделей опрацювання та логічного розмежування доступу до даних, за результатами яких визначено їх переваги, недоліки та області доцільного застосування в умовах гетерогенності інформаційних систем. Отримані результати використано для побудови моделі процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями.

У межах розділу сформульовано постановку задачі планування, розроблено узагальнену модель планування завдань і сутностей із залежностями, а також побудовано математичну модель планування завдань розповсюдження даних. Наведено результати дослідження математичної моделі планування задач розповсюдження окремого n -го набору типів даних, які підтверджують коректність запропонованого підходу та слугують теоретичною основою для подальшого розвитку методів планування задач у розподілених інформаційних системах.

У третьому розділі дисертаційної роботи розроблено інтелектуальний метод планування задач у розподілених інформаційних системах, призначений

для функціонування в умовах гетерогенності обчислювальних ресурсів, динамічності навантаження та невизначеності метаданих. На основі формалізованої постановки задачі визначено вимоги до інтелектуального процесу планування, критерії оптимальності та показники ефективності, що враховують як продуктивність виконання задач, так і адаптивність системи до змін її стану.

Виконано формалізацію моделі планування задач у гетерогенних розподілених комп'ютерних системах (HDCS), яка відображає залежності між задачами, обчислювальними ресурсами та потоками даних. Запропонована модель дозволяє враховувати невизначеність метаданих шляхом використання прогнозних оцінок параметрів виконання задач і стану системи, що створює передумови для застосування інтелектуальних механізмів прийняття рішень.

Розроблено інтелектуальний метод планування задач у розподілених інформаційних системах та визначено його загальну архітектуру. Метод базується на поєднанні механізмів прогнозування критичного шляху виконання задач і навантаження на систему з процедурами адаптивного керування та оптимального призначення процесорів задачам. Інтелектуальна складова методу реалізує аналіз поточного та прогнозованого стану системи, що дозволяє динамічно коригувати плани виконання задач, зменшувати вплив невизначеності метаданих та підвищувати ефективність використання обчислювальних ресурсів.

У завершальній частині розділу визначено підхід до оцінювання ефективності запропонованого інтелектуального методу, що ґрунтується на експериментальному дослідженні його характеристик у порівнянні з відомими прототипами. Сформульовані в розділі модельні припущення, алгоритмічні рішення та критерії оцінювання слугують теоретичною та методичною основою для проведення експериментальних досліджень методу планування задач у розподілених інформаційних системах з урахуванням невизначеності метаданих, представлених у четвертому розділі дисертаційної роботи.

В останньому розділі дисертаційної роботи наведено результати експериментальних досліджень інтелектуального методу планування задач у розподілених інформаційних системах з урахуванням невизначеності метаданих. Сформульовано дослідницькі гіпотези, визначено дизайн експерименту та обґрунтовано систему показників ефективності, що дозволяє комплексно оцінювати продуктивність, адаптивність і робастність запропонованого методу в умовах гетерогенності обчислювальних ресурсів і динамічності середовища виконання.

Описано архітектуру експериментального прототипу планувальника, яка реалізує розроблені в попередніх розділах модельні та алгоритмічні рішення. Архітектура забезпечує можливість відтворюваного проведення експериментів, збору статистичних даних і гнучкого налаштування параметрів планування, що є необхідним для коректної оцінки ефективності інтелектуального методу.

Розглянуто методи порівняльного аналізу, абляційного дослідження та протокол навчання інтелектуальної складової планувальника, які дозволяють оцінити внесок окремих компонентів методу в загальний результат. Запропоновано підхід до моделювання невизначеності метаданих і визначено протокол перевірки робастності, спрямований на дослідження стійкості методу до похибок, неповноти та варіативності вхідної інформації.

Наведено експериментальні результати, виконано порівняльний аналіз із відомими прототипами та здійснено їх інтерпретацію. Обговорено отримані результати з точки зору підтвердження висунутих гіпотез і практичної доцільності застосування запропонованого інтелектуального методу планування задач у розподілених інформаційних системах за умов невизначеності метаданих.

Практична цінність результатів дисертаційної роботи полягає у можливості безпосереднього використання розробленого інтелектуального методу планування задач у реальних розподілених інформаційних системах, що функціонують в умовах гетерогенності обчислювальних ресурсів, динамічності навантаження та невизначеності метаданих. Запропоновані моделі, алгоритми

та програмні рішення можуть бути застосовані під час проєктування та модернізації систем управління обчислювальними ресурсами в хмарних, туманних і крайових обчислювальних середовищах.

Цінність окремих практичних результатів підтверджуються їх впровадженням у діяльність компанії «LineUp» (акт впровадження від 17.04.2025 р.) та LLC «Arshilon» (акт впровадження від 23.12.2025 р.), інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах, який базується на поєднанні DAG- та GERT-графів із алгоритмами штучного інтелекту: Graph Attention Network, Proximal Policy Optimization та Bayesian Optimization. Це дозволило динамічно адаптувати розподіл задач між обчислювальними ресурсами в умовах змінного навантаження. Отримані результати використано при створенні модуля автоматизованого керування навантаженням у хмарному середовищі LineUp, що дало змогу покращити збалансування ресурсів, зменшити час затримок і підвищити стійкість системи до змін середовища виконання. Крім того, у діяльності LLC «Arshilon» впроваджено інтелектуальний метод планування задач у розподілених інформаційних системах із застосуванням ML для прийняття рішень щодо призначення задач на ресурси та корекції розкладу при зміні стану системи. Практичні рекомендації використано для організації процесу планування в реальному часі, урахування політик доступу в прикладних інформаційних системах та вдосконалення алгоритмів управління процесом розподілу задач між структурними підрозділами, що сприяло покращенню балансування ресурсів.

Результати дослідження забезпечують підвищення ефективності використання обчислювальних ресурсів, зменшення часу виконання задач і підвищення стійкості процесу планування до неповноти та неточності вхідної інформації. Розроблений метод може бути інтегрований у наявні планувальники та системи управління ресурсами без суттєвої зміни їх архітектури, що спрощує впровадження та знижує витрати на адаптацію.

Окремі положення дисертаційної роботи можуть бути використані в науково-дослідній (довідки №81/2025 від 26.12.2025 р. та №82/2025 від 26.12.2025 р.) та навчальній діяльності під час викладання дисциплін (довідка №26/86-29-22 від 02.03.2026 р.), пов'язаних із розподіленими обчисленнями, інтелектуальними системами та управлінням ресурсами, а також у подальших прикладних дослідженнях, спрямованих на розвиток методів інтелектуального планування задач у складних обчислювальних середовищах.

Ключові слова: розподілені інформаційні системи, планування задач, інтелектуальне планування, гетерогенні обчислювальні середовища, невизначеність метаданих, управління ресурсами, управління навантаженням, серверні кластери, хмарні та граничні обчислення, графові моделі, оптимізація параметрів, математичне моделювання, нейромережеві моделі, масштабованість, робастність.

ABSTRACT

Yengalychev S. O. Task scheduling method in distributed information systems under metadata uncertainty. – Qualifying scientific work as a manuscript.

The thesis for obtaining a Doctor of Philosophy degree in specialty 122 «Computer Science», field of knowledge 12 «Information technologies». Simon Kuznets Kharkiv National University of Economics of the Ministry of Education and Science of Ukraine. Kharkiv, 2026.

The dissertation is a theoretical and methodological study dedicated to improving task scheduling methods in distributed information systems under metadata uncertainty. The work is aimed at solving the problems of effective distribution of computing resources in heterogeneous and dynamic environments, taking into account the incompleteness, inaccuracy, and variability of information about the characteristics of tasks, resources, and the execution environment. The leading idea of the research is to develop a new task scheduling method that increases

the efficiency of distributed systems through intelligent metadata processing and adaptation of the scheduling process under uncertainty.

The relevance of the topic is driven by the rapid development of distributed computing environments, including cloud, fog, and edge systems, the growing complexity of computing tasks, and the need to ensure high performance, scalability, and reliability of their execution. Analysis of modern scientific research and practical solutions showed that most existing task scheduling methods are based on the assumption of complete and accurate information about system parameters, which does not correspond to the real conditions of distributed information systems. This leads to a decrease in scheduling efficiency, resource overruns, and deterioration of quality of service indicators, which necessitates scientific justification and development of new task scheduling methods under metadata uncertainty.

The object of the study is the process of planning and distributing tasks in heterogeneous distributed information systems.

The subject of the study is the task scheduling method in distributed information systems, considering metadata uncertainty.

In the first chapter of the dissertation, a systems analysis of the task scheduling process in distributed computer and information systems is performed. Requirements for the scheduling process are formulated and analyzed, taking into account the heterogeneity of computing resources, dynamic loading, performance constraints, and the need for efficient use of computing power. The main approaches to increasing the performance of distributed information systems are reviewed and generalized, in particular through the use of parallel computing, optimization of the structure of computing processes, and improvement of resource management methods. An analysis of computer-aided design systems for digital devices and methods for synthesizing functional structures of specialized processors was conducted, which allowed identifying effective principles of hardware-algorithmic implementation of computing tasks. The choice of performance indicators for the parallel implementation of algorithms used for quantitative evaluation of scheduling results and task performance is justified. In the final part of the chapter, an analysis of

known task schedulers and resource management systems is carried out, identifying their functional capabilities, advantages, and limitations. Unresolved problems related to the use of existing scheduling methods under incompleteness and uncertainty of metadata are identified, which justifies the feasibility of further research and determines the directions for developing a new task scheduling method in distributed information systems.

In the second chapter of the dissertation, data processing models and logical access control in information systems are developed and investigated, taking into account the heterogeneity of entities and the peculiarities of their interaction. Based on a formalized task statement, approaches to organizing data access in complex distributed environments characterized by heterogeneity of data types, subjects, and resources are analyzed. Methods for merging simpler role-based models are considered, and hierarchical role models are synthesized, ensuring scalability and flexibility of access policies. Proposed are mandatory typing models based on trust relationships, which ensure increased consistency and security during data processing in multi-component information systems. The main provisions of the security policy merging process are formulated, taking into account differences between entities and constraints determined by their functional roles. Comparative studies of data processing models and logical access control were conducted, determining their advantages, disadvantages, and areas of feasible application in heterogeneous information systems. The results obtained were used to build a model for the data distribution task scheduling process, considering differences between entities. Within the chapter, the scheduling problem statement is formulated, a generalized model for scheduling tasks and entities with dependencies is developed, and a mathematical model for scheduling data distribution tasks is built. Results of the study of the mathematical model for scheduling tasks of distributing a specific n -th set of data types are presented, confirming the correctness of the proposed approach and serving as a theoretical basis for further development of task scheduling methods in distributed information systems.

In the third chapter of the dissertation, an intelligent task scheduling method in distributed information systems is developed, designed for functioning under heterogeneity of computing resources, dynamic loading, and metadata uncertainty. Based on the formalized task statement, requirements for the intelligent scheduling process, optimality criteria, and performance indicators are determined, taking into account both task execution performance and system adaptability to changes in its state. Formalization of the task scheduling model in heterogeneous distributed computer systems (HDCS) is performed, reflecting the dependencies between tasks, computing resources, and data flows. The proposed model allows accounting for metadata uncertainty by using predictive estimates of task execution parameters and system state, creating prerequisites for the application of intelligent decision-making mechanisms. An intelligent task scheduling method in distributed information systems is developed, and its general architecture is defined. The method is based on a combination of critical path prediction mechanisms for task execution and system load with adaptive control procedures and optimal processor assignment to tasks. The intelligent component of the method implements analysis of the current and predicted state of the system, allowing dynamic correction of task execution plans, reducing the impact of metadata uncertainty, and increasing the efficiency of using computing resources. In the final part of the chapter, an approach to evaluating the effectiveness of the proposed intelligent method is defined, based on experimental study of its characteristics in comparison with known prototypes. The model assumptions, algorithmic solutions, and evaluation criteria formulated in the chapter serve as the theoretical and methodological basis for conducting experimental research on the task scheduling method in distributed information systems considering metadata uncertainty, presented in the fourth chapter of the dissertation.

In the final chapter of the dissertation, results of experimental studies of the intelligent task scheduling method in distributed information systems considering metadata uncertainty are presented. Research hypotheses are formulated, the experiment design is defined, and a system of performance indicators is justified, allowing for a comprehensive assessment of the productivity, adaptability, and

robustness of the proposed method under heterogeneity of computing resources and dynamic execution environments. The architecture of the experimental scheduler prototype is described, implementing the model and algorithmic solutions developed in previous chapters. The architecture provides the capability for reproducible experiments, statistical data collection, and flexible configuration of scheduling parameters, which is necessary for a correct evaluation of the intelligent method's effectiveness. Methods of comparative analysis, ablation study, and the training protocol for the intelligent component of the scheduler are considered, allowing for the evaluation of the contribution of individual method components to the overall result. An approach to modeling metadata uncertainty is proposed, and a robustness verification protocol is defined, aimed at studying the method's stability against errors, incompleteness, and variability of input information. Experimental results are presented, a comparative analysis with known prototypes is performed, and their interpretation is carried out. The results obtained are discussed from the perspective of confirming the hypotheses and the practical feasibility of using the proposed intelligent task scheduling method in distributed information systems under metadata uncertainty.

The practical value of the dissertation results lies in the possibility of direct use of the developed intelligent task scheduling method in real distributed information systems operating under conditions of computing resource heterogeneity, dynamic loading, and metadata uncertainty. The proposed models, algorithms, and software solutions can be applied during the design and modernization of resource management systems in cloud, fog, and edge computing environments.

The value of certain practical results is confirmed by their implementation in the activities of “LineUp” (act of implementation dated 17.04.2025) and LLC “Arshilon” (act of implementation dated 23.12.2025). The intelligent method for task scheduling in heterogeneous distributed information systems, based on the combination of DAG and GERT graphs with artificial intelligence algorithms, namely Graph Attention Network, Proximal Policy Optimization, and Bayesian Optimization, enabled dynamic adaptation of task allocation among computing

resources under variable workload conditions. The obtained results were used in the development of an automated workload management module in the “LineUp” cloud environment, which made it possible to improve resource balancing, reduce delays, and increase the system’s resilience to changes in the execution environment. In addition, LLC “Arshilon” implemented the intelligent task scheduling method in distributed information systems using ML to support decision-making on assigning tasks to resources and adjusting the schedule when the system state changes. The practical recommendations were used to organize real-time scheduling processes, consider access policies in applied information systems, and improve algorithms for managing the task distribution process among structural units, which contributed to better resource balancing.

The research results ensure increased efficiency of computing resource utilization, reduced task execution time, and increased stability of the scheduling process against incompleteness and inaccuracy of input information. The developed method can be integrated into existing schedulers and resource management systems without significant changes to their architecture, simplifying implementation and reducing adaptation costs. Certain provisions of the dissertation can be used in research (certificate №81/2025 dated 26.12.2025 and №82/2025 dated 26.12.2025) and educational (certificate №26/86-29-22 dated 02.03.2026) activities during the teaching of disciplines related to distributed computing, intelligent systems, and resource management, as well as in further applied research aimed at developing intelligent task scheduling methods in complex computing environments.

Keywords: distributed information systems, task scheduling, intelligent scheduling, heterogeneous computing environments, metadata uncertainty, resource management, load management, server clusters, cloud and edge computing, graph models, parameter optimization, mathematical modelling, neural network models, scalability, robustness.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

У вітчизняних та закордонних виданнях, що включені до міжнародних наукометричних баз SCOPUS та Web of Science

1. Semenov S., Zhang M., Yenhalychev S., & Smidovych L. Generalized model of the ADS-B unmanned aerial vehicle data transmission process in a steganographic system. *Сучасний стан наукових досліджень та технологій в промисловості (Innovative technologies and scientific solutions for industries)*. 2022. № 4(22). Рр. 14–19.

Особистий внесок здобувача: здобувачем запропоновано підхід до моделювання передавання даних у телекомунікаційній системі, проведено формалізацію моделі та інтерпретацію результатів моделювання.

DOI: <https://doi.org/10.30837/ITSSI.2022.22.014>

URL: <https://journals.uran.ua/itssi/article/view/271061>

2. Yenhalychev S. O., Mozhaev, O.O., Gnusov Y.V., Manzhai O.V., Strukov V.M., Nosov V.V., Radchenko V.V. Steganographic method of acoustic information protection in critical applications systems. *Innovative Technologies and Scientific Solutions for Industries*, 2023. No. 3 (25). P. 52–63.

Особистий внесок здобувача: здобувачем визначено вимоги до вхідних даних задач і ресурсів та обґрунтовано підхід до врахування їх неповноти та неточності під час планування.

DOI: <https://doi.org/10.30837/ITSSI.2023.25.052>

URL: <https://journals.uran.ua/itssi/article/view/290889>

3. Semenov S., Yenhalychev S., Pochebut M., Sitnikova O. Models of data processing and logical access segregation considering the heterogeneity of entities in information systems. *Innovative Technologies and Scientific Solutions for Industries*. 2024. No. 2 (28). P. 143–152.

Особистий внесок здобувача: здобувачем запропоновано формалізацію моделей обробки даних та сегрегації логічного доступу з урахуванням

гетерогенності сутностей, сформовано постановку задачі розмежування доступу, виконано аналіз результатів і висновки.

DOI: <https://doi.org/10.30837/2522-9818.2024.28.143>

URL: <https://journals.uran.ua/itssi/article/view/308876>

4. Yenhalychev S., Leunencko O., Davydov V. Development an intelligent task scheduling method in heterogeneous distributed information systems. *Eastern-European Journal of Enterprise Technologies. Information and controlling system*. 2025. Vol. 3, No. 9 (135). P. 6–18.

Особистий внесок здобувача: здобувачем розроблено інтелектуальний метод планування задач у гетерогенних розподілених інформаційних системах, виконано постановку задачі на графах типу Directed Acyclic Graph (DAG), спрямований ациклічний граф, обґрунтовано застосування Graph Attention Network (GAT), графової мережі уваги та Reinforcement Learning (RL), навчання з підкріпленням, визначено критерії оцінювання, проведено експериментальну перевірку та сформульовано висновки.

DOI: <https://doi.org/10.15587/1729-4061.2025.329263>

URL: <https://journals.uran.ua/eejet/article/view/329263>

5. Yenhalychev S.O., Semenov S.G.. Experimental studies of the task planning method in distributed information systems taking into account metadata uncertainty. *Автоматизовані системи управління та прилади автоматики*. 2025 No. 4 (187), P. 180–188.

Особистий внесок здобувача: здобувачем розроблено дизайн експериментальних досліджень методу планування задач за невизначеності метаданих, реалізовано процедури моделювання та оцінювання, виконано аналіз результатів і підготовлено висновки щодо ефективності підходу.

DOI: <https://doi.org/10.30837/0135-1710.2025.187.063>

URL: <https://asu-pa.nure.ua/article/view/348532>

Праці апробаційного характеру

6. Semenov S., Lymarenko V., Yenhalyche, S., Gavrilenko S. The Data Dissemination Planning Tasks Process Model Into Account the Entities Differentity. DESSERT'2022: The 12th IEEE International Conference on Dependable Systems, Services and Technologies (Athens, Greece, December 9-11, 2022). 2022. P. 317-322.

Особистий внесок здобувача: здобувачем побудовано процесну модель задач планування розповсюдження даних з урахуванням відмінностей сутностей системи, виконано формалізацію схеми на основі GERT, графічної техніки оцінювання та перегляду, визначено ключові параметри моделі та інтерпретовано результати.

DOI: <https://doi.org/10.1109/DESSERT58054.2022.10018695>

URL: <https://repository.hneu.edu.ua/handle/123456789/38400>

7. Yenhalychev S. O., Muchacki M., Sitnikova O., Model of the Dynamics of the State of Educational Content Recommender System. DESSERT'2023: The 13th IEEE International Conference on Dependable Systems, Services and Technologies. (Athens, Greece, October 13-15, 2023). 2023. P. 60 – 66.

Особистий внесок здобувача: здобувачем розроблено модель динаміки стану рекомендаційної системи освітнього контенту, визначено змінні стану та параметри впливу, проведено аналіз сценаріїв зміни стану системи та підготовлено відповідні висновки.

DOI: <https://doi.org/10.1109/DESSERT61349.2023.10416450>

URL: <https://repository.hneu.edu.ua/handle/123456789/38399>

8. Єнґаличев С. Вибір показників ефективності паралельної реалізації алгоритмів. *Problems of scientific, technical and legal support for cybersecurity in the modern world*. (Krakow, 2024). Krakow: Wydawnictwo Naukowe UKEN, 2024. P. 174-179.

Особистий внесок здобувача: у одноосібній праці, здобувачем повністю виконано обґрунтування вибору показників ефективності паралельної

реалізації алгоритмів, проведено аналіз та сформульовано висновки і практичні рекомендації.

URL: <https://repository.hneu.edu.ua/handle/123456789/34912>

9. Yenhalychev S.O., Leunenko O.V., Semenov S.G.. Known planners assignments and systems management resources analysis. *Free and open source software: Матеріали XIV міжнародної наук.-прак. конф.* (м. Харків, 14 - 16 Лютого, 2023 р.). Харків: FOSS, 2023. С. 10-14.

Особистий внесок здобувача: здобувачем виконано огляд і порівняльний аналіз відомих планувальників задач та підходів до керування ресурсами, сформовано критерії порівняння й обмеження застосовності, підготовлено аналітичну частину матеріалів.

URL: <http://repository.hneu.edu.ua/handle/123456789/29041>

10. Єнгалічев С. О., Семенов С. Г. Аналіз можливостей використання машинного глибокого навчання для розпізнавання сутностей. *Science and Business in Modern Environment: Summer Debates: матеріали V міжнародної наук.-прак. інтернет конф.* (м. Дніпро, 3 - 4 серпня 2023 р.). С. 205-206.

Особистий внесок здобувача: здобувачем проведено аналіз можливостей застосування глибокого навчання для розпізнавання сутностей, сформовано постановку задачі та вимоги до даних, підготовлено узагальнення результатів і висновки.

URL: <https://repository.hneu.edu.ua/handle/123456789/38369>

11. Yenhalychev S. O., Semenov S. G., Leunenko O. V. An overview of the popular free resources for task planning and system management. *Free and open source software: матеріали XV міжнародної наук.-прак. конф.* (м. Харків, 14 - 16 Лютого, 2024 р.). Харків: FOSS, 2024. С. 70-72.

Особистий внесок здобувача: здобувачем здійснено систематизацію та оцінювання популярних відкритих інструментів для планування задач і керування системами, визначено критерії відбору та порівняння, підготовлено практичні рекомендації.

URL: <http://repository.hneu.edu.ua/handle/123456789/31827>

12. Єнгалічев С. О., Леуненко О. В. Імітаційне моделювання та дослідження алгоритмів планування пакетних потоків даних із різними пріоритетами. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024) (м. Харків, 25 квітня, 2024 р.). Молодіжний економічний вісник. 2024. №1. С. 46-50.

Особистий внесок здобувача: здобувачем виконано імітаційне моделювання та експериментальний аналіз алгоритмів планування пакетних потоків даних із різними пріоритетами, сформовано метрики оцінювання та інтерпретовано результати.

URL: <https://repository.hneu.edu.ua/handle/123456789/38208>

13. Леуненко О. В., Єнгалічев С. О. Моделювання та захист передачі даних ресурсоємних задач у гетерогенних хмарних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024) (м. Харків, 25 квітня, 2024 р.). Молодіжний економічний вісник. 2024. №1. С. 59-63.

Особистий внесок здобувача: здобувачем виконано постановку задачі моделювання та захисту передавання даних ресурсоємних задач у гетерогенних хмарних системах, визначено показники ефективності та ризики, підготовлено висновки щодо практичного застосування.

URL: <https://repository.hneu.edu.ua/handle/123456789/38209>

14. Семенов С. Г., Єнгалічев С. О. Метод планування задач в розподілених інформаційних системах з врахуванням невизначеності метаданих. Проблеми інформатизації: матеріали XII міжнародної наук.-техн. конференції (м. Харків, 21- 22 листопада, 2024 р.). Харків: Харківський національний університет радіоелектроніки, 2024. С. 24.

Особистий внесок здобувача: здобувачем сформульовано постановку задачі планування у розподілених інформаційних системах за невизначеності метаданих, запропоновано основні елементи методу та напрями оцінювання його ефективності.

URL: <https://repository.hneu.edu.ua/handle/123456789/38235>

15. Yenhalychev S.O., Leunenko O.V.. Comparative analysis of open source infrastructure as code (IaC) tools for managing heterogeneous cloud resources. *Free and open source software: матеріали XVI міжнародної наук.-прак. конф.* (м. Харків, 13 - 14 Лютого, 2025 р.). Харків: FOSS, 2025. С. 105-107.

Особистий внесок здобувача: здобувачем проведено порівняльний аналіз інструментів Infrastructure as Code (IaC), інфраструктури як коду, для керування гетерогенними хмарними ресурсами, визначено критерії оцінювання (функціональність, відтворюваність, масштабованість, безпека), підготовлено узагальнення і рекомендації.

URL: <https://repository.hneu.edu.ua/handle/123456789/35624>

16. Yenhalychev S. O., Leunenko O. V. Development of Methods for Dynamic Load Balancing During the Transfer of Large Volumes of Data: A Comparative Study of Apache Kafka and RabbitMQ. *Free and open source software: матеріали XVI міжнародної наук.-прак. конф.* (м. Харків, 13 - 14 Лютого, 2025 р.). Харків: FOSS, 2025. С. 108-109.

Особистий внесок здобувача: здобувачем виконано порівняльне дослідження засобів потокової передачі даних Apache Kafka та RabbitMQ, сформовано сценарії навантаження і критерії оцінювання, проведено аналіз обмежень для задач динамічного балансування.

URL: <https://repository.hneu.edu.ua/handle/123456789/35624>

17. Semenov Serhii, Yenhalychev Serhii. The guidelines on secure coding and frameworks to enhance autonomy and mitigate cybersecurity risks. *Сучасні інформаційні системи та технології в цифровому суспільстві: матеріали міжнародної наук.-практ. конференції* (м. Харків, 10 - 11 квітня, 2025 р.). Харків: ХНЕУ імені Семена Кузнеця, 2025. С. 71.

Особистий внесок здобувача: здобувачем сформовано рекомендації з безпечного добору фреймворків для підвищення автономності систем і зниження кіберризиків, структуровано практичні вимоги та типові контролю безпеки.

URL: <https://repository.hneu.edu.ua/handle/123456789/36030>

18. Семенов С. Г., Єнгалічев С. О., Пересічанський В. М. Обговорення результатів дослідження інтелектуального методу планування задач у розподілених інформаційних системах. *Безпека у кіберсфері: матеріали Міжнародної науково-практичної конференції* (м. Кам'янець-Подільський, 28 травня, 2025 р.). Кам'янець-Подільський: ХНУВС, 2025. С. 186-187.

Особистий внесок здобувача: здобувачем підготовлено узагальнення та інтерпретацію результатів дослідження інтелектуального методу планування задач у розподілених інформаційних системах, сформульовано висновки щодо практичної значущості.

URL: <https://repository.hneu.edu.ua/handle/123456789/36711>

19. Семенов С. Г., Єнгалічев С. О. Адаптивне планування задач у розподілених системах на основі алгоритму оптимізації політик із проксимальним оновленням. *Цифрова економіка: Матеріали міжнародної науково-практичної конференції* (м. Київ, 5 - 6 червня, 2025 р.). Київ: КНЕУ, 2025. С. 698-700.

Особистий внесок здобувача: здобувачем обґрунтовано адаптивний підхід до планування задач на основі Proximal Policy Optimization (PPO), оптимізації політик із проксимальним оновленням, визначено постановку задачі для навчання з підкріпленням, описано процедури навчання та вибору політики та критерії оцінювання.

URL: <https://repository.hneu.edu.ua/handle/123456789/38234>

20. Єнгалічев С. О., Леуненко О. В.. Обговорення результатів інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025) (м. Харків, 16 жовтня, 2025 р.). *Молодіжний економічний вісник*. 2025. №3. С. 122-124.

Особистий внесок здобувача: здобувачем виконано аналіз і узагальнення результатів інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах, підготовлено висновки щодо умов застосовності та ефекту від використання методу.

URL: <https://repository.hneu.edu.ua/handle/123456789/38210>

21. Леуненко О. В., Єнгаличев С. О. Математичне моделювання передачі даних ресурсомістких задач у гетерогенних хмарних системах: показники ефективності та методологічні підходи. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025). (м Харків, 16 жовтня, 2025 р.). Молодіжний економічний вісник». 2025. №3. С. 199-202.

Особистий внесок здобувача: здобувачем запропоновано підходи до математичного моделювання передавання даних ресурсомістких задач у гетерогенних хмарних системах, визначено показники ефективності та методологічні принципи оцінювання.

URL: <https://repository.hneu.edu.ua/handle/123456789/38211>

ЗМІСТ

ВСТУП.....	24
РОЗДІЛ 1 АНАЛІЗ ТА ПОРІВНЯЛЬНІ ДОСЛІДЖЕННЯ МЕТОДІВ ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ З ВРАХУВАННЯМ НЕВИЗНАЧЕНОСТІ МЕТАДАНИХ	34
1.1. Аналіз вимог щодо процесу планування задач в розподілених комп'ютерних системах	34
1.2. Шляхи підвищення продуктивності розподілених інформаційних систем ..	39
1.3. Аналіз відомих планувальників завдань і систем управління ресурсами....	57
Висновки до розділу 1	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 1.....	69
РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ.....	77
2.1. Моделі опрацювання та логічного розмежування доступу до даних з урахуванням різномірності сутностей в інформаційних системах	77
2.2. Порівняльні дослідження моделей опрацювання та логічного розмежування доступу до даних.....	93
2.3. Модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями	96
Висновки до розділу 2	108
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 2.....	112
РОЗДІЛ 3 ІНТЕЛЕКТУАЛЬНИЙ МЕТОД ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ.....	118
3.1. Постановка завдання.....	118
3.2. Особливості використання метаданих у розподілених системах планування задач	123
3.3. Формалізація моделі планування задач у HDCS	130

3.4. Метод інтелектуального планування задач у розподілених інформаційних системах.....	140
3.5. Дослідження ефективності запропонованого методу шляхом тестування порівняння продуктивності нового підходу з прототипами.....	160
Висновки до розділу 3	163
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 3.....	167
РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МЕТОДУ ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ З ВРАХУВАННЯМ НЕВИЗНАЧЕНОСТІ МЕТАДАНИХ	171
4.1. Постановка дослідницьких гіпотез, дизайн експерименту та система показників ефективності.....	171
4.2. Архітектура прототипу планувальника експерименту	179
4.3. Методи порівняння, абляційний аналіз і протокол навчання	183
4.4. Моделювання невизначеності метаданих і протокол перевірки робастності.....	189
4.5. Експериментальні результати, порівняльний аналіз і обговорення	193
Висновки до розділу 4	203
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 4.....	208
ВИСНОВКИ.....	212
ДОДАТКИ.....	215

ВСТУП

Актуальність теми.

Розподілені інформаційні системи відіграють ключову роль у забезпеченні ефективних обчислень у сучасних технологіях, зокрема у високопродуктивних обчисленнях, хмарних платформах, системах Інтернету речей та фінансових обчисленнях. Успішна експлуатація таких систем безпосередньо залежить від здатності підтримувати високий рівень доступності сервісів і задовольняти вимоги масштабованості, що зумовлює підвищені вимоги до процесів керування ресурсами та планування виконання задач.

У межах даної роботи під «розподіленими інформаційними системами» розуміються насамперед керовані (orchestrated) гетерогенні обчислювальні середовища з централізованим контуром планування, у яких існує єдина логіка прийняття рішень щодо призначення задач на ресурси (централізований або логічно централізований планувальник у кластері/хмарі/edge-кластері). Децентралізовані (повністю peer-to-peer) системи з розподіленим прийняттям рішень не є основним об'єктом дослідження; вони можуть розглядатися лише як окремий випадок, що потребує іншої організації керування та протоколів узгодження.

Однією з центральних проблем, що постає перед розподіленими системами, є оптимальне планування задач, яке має забезпечити балансування навантаження, мінімізацію затримок та ефективне використання ресурсів. Зазначена проблема істотно ускладнюється гетерогенністю середовищ (різні типи обчислювальних вузлів, мережеві з'єднання та сховища даних), динамічністю умов функціонування, а також невизначеністю метаданих, що погіршує якість прийняття рішень під час побудови та корекції розкладів.

У роботі під метаданими задачі розуміється структурований запис (metadata record), який надходить до планувальника разом із задачею та

використовується для прогнозу тривалості та вартості виконання і перевірки обмежень.

Такі прояви невизначеності на практиці призводять до зростання ризику порушення часових обмежень і недотримання цільових рівнів сервісу SLO (service-level objective), що є критичним для високонавантажених систем реального часу. У роботі вводиться цільова метрика дедлайн-надійності (deadline reliability) – частка задач, завершених до свого дедлайну (або в межах заданого SLO) у межах певного вікна спостереження запусків. Метрика може обчислюватися як загалом по потоку задач, так і по підкласах задач та режимах навантаження.

Попри значну кількість наявних підходів до планування, абсолютна більшість методів і засобів не враховують невизначеність вхідних даних та гетерогенність актуальних процесів і послуг, а тому не забезпечують стабільної ефективності в умовах неповноти інформації про ресурсні вимоги задач. Типовими є й обмеження практичних планувальників, що вимагають апріорного знання ресурсних вимог, попереднього класування задач і визначення політик, водночас не маючи інтелектуальних механізмів обліку невизначеності.

У результаті формується суперечність між зростаючими вимогами до дедлайн-надійності та керованості накладних витрат планування в реальному часі, з одного боку, і обмеженою адаптивністю традиційних детерміністичних та евристичних схем з іншого.

У цьому контексті актуальним науковим завданням є розробка нових та удосконалення існуючих моделей і методу інтелектуального планування задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих, які забезпечують зменшення загального часу виконання задач та балансування навантаження за прийнятних обчислювальних витрат планування.

Зв'язок роботи з науковими програмами, планами, темами.

Дослідження, результати яких викладені в дисертаційній роботі, виконані на кафедрі кібербезпеки та інформаційних технологій Навчально-наукового інституту інформаційних технологій Харківського національного економічного університету імені Семена Кузнеця (ХНЕУ ім. С. Кузнеця) відповідно до планів науково-дослідної роботи кафедри та інституту:

1) «Розробка інформаційної системи з елементами захисту інформації від несанкціонованого поширення» (номер державної реєстрації 0124U004238) – досліджено та реалізовано підходи до побудови систем захисту інформації; на основі теоретичного аналізу, критичного аналізу, теоретичного синтезу, описового методу та опрацювання експериментальних даних обґрунтовано можливість використання створеного програмного продукту для обробки і зберігання інформації з метою підвищення рівня її безпеки та зниження ризиків несанкціонованого поширення даних (довідка №82/2025 від 26.12.2025);

2) «Методична підтримка сталого функціонування ІТ підприємств» (номер державної реєстрації 0125U002650) – здійснено аналіз та порівняльне дослідження методів планування задач у розподілених інформаційних системах з урахуванням невизначеності метаданих; визначено ключові вимоги до таких систем, зокрема масштабованість, надійність, ефективність і безпеку; проаналізовано шляхи підвищення продуктивності розподілених інформаційних систем, а також можливості й обмеження сучасних планувальників завдань та систем управління ресурсами; обґрунтовано необхідність розроблення нових методів планування задач в умовах неповноти інформації про ресурсні вимоги для скорочення часових витрат на виконання прикладних задач обробки даних (довідка №81/2025 від 26.12.2025).

Тематика дисертаційного дослідження узгоджується з пріоритетними напрямками наукової діяльності кафедри, що охоплюють, зокрема, захист даних у хмарних інфраструктурах, методи виявлення загроз у розподілених системах, а також застосування штучного інтелекту в задачах кібербезпеки та розробки інформаційних систем.

Мета і завдання дослідження.

Метою дисертаційної роботи є підвищення ефективності процесу планування та розподілу задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих шляхом розроблення інтелектуального методу планування, що забезпечує зменшення загального часу виконання комплексу задач, покращення балансування навантаження та збереження прийнятних обчислювальних витрат планування.

Для досягнення поставленої мети необхідно розв'язати такі наукові завдання:

1. Провести аналіз сучасних підходів до планування та розподілу задач у гетерогенних розподілених інформаційних системах, систематизувати критерії ефективності та визначити чинники, що зумовлюють деградацію якості розкладу в умовах невизначеності метаданих.

2. Розробити узагальнену математичну модель процесу планування та розподілу задач у гетерогенних розподілених інформаційних системах, яка враховує залежності між задачами, обмеження ресурсів, комунікаційні затримки та ризики порушення дедлайнів за неповної/неточної метаінформації.

3. Побудувати модель графа задач для формалізації залежностей і паралелізму, обґрунтувавши використання орієнтованого ациклічного графа для детермінованих залежностей та GERT для опису стохастичних переходів і невизначеності виконання.

4. Розробити інтелектуальний метод планування задач в розподілених інформаційних системах, із застосуванням підкріплювального навчання для прийняття рішень щодо призначення задач на ресурси та корекції розкладу при зміні стану системи.

5. Розробити спосіб налаштування параметрів вибору політик планування з використанням баєсівської оптимізації для забезпечення компромісу між якістю розкладу та накладними витратами планування залежно від класу навантаження і характеристик середовища.

6. Провести експериментальне дослідження ефективності розробленого методу на репрезентативних сценаріях і порівняти результати з базовими підходами, виконати абляційний аналіз внеску окремих компонентів і оцінити робастність методу до похибок метаданих.

7. Сформулювати практичні рекомендації щодо впровадження у розподілених інформаційних системах.

Об'єкт дослідження: процес планування та розподілу задач в гетерогенних розподілених інформаційних системах.

Предмет дослідження: метод планування задач у гетерогенних розподілених інформаційних системах з урахуванням невизначеності метаданих.

Методи дослідження.

У дисертаційній роботі застосовано системний аналіз і порівняльне узагальнення підходів до планування задач у гетерогенних розподілених інформаційних системах та визначення впливу невизначеності метаданих на якість розкладів. Формалізацію залежностей між задачами виконано засобами теорії графів із використанням DAG (орієнтований ациклічний граф) і ймовірнісного моделювання на основі GERT для опису стохастичних переходів і оцінювання ризику порушення дедлайнів. Задачу планування сформульовано як задачу оптимізації з обмеженнями та багатокритеріальними показниками (makespan, балансування навантаження, накладні витрати планування). Для побудови інтелектуального методу використано методи машинного навчання та штучного інтелекту: GAT для роботи з графом задач і прогнозування параметрів виконання, PPO для адаптивного призначення задач і перепланування, Bayesian Optimization для налаштування параметрів і вибору конфігурацій. Ефективність методу перевірено обчислювальними експериментами й імітаційним моделюванням із порівнянням із базовими алгоритмами та статистичною обробкою результатів.

Наукова новизна отриманих результатів:

1. Уперше розроблено метод інтелектуального планування задач у розподілених інформаційних системах в умовах невизначеності метаданих, який інтегрує прогнозування критичного шляху на графі задач, адаптивне перепланування та автоматизоване налаштування параметрів. Відмінною особливістю методу є синтез залежностей задач у вигляді орієнтованого ациклічного графа та стохастичного графового подання альтернативних переходів із використанням графових нейромережових моделей, алгоритмів навчання з підкріпленням і баєсівського налаштування параметрів. Практична реалізація цього підходу виконана у вигляді прототипу планувальника та експериментального конвеєра оцінювання (формування графа задач, прогнозування критичного шляху, онлайн-перепланування та автоматизоване налаштування параметрів). Експериментальна перевірка на репрезентативних сценаріях підтвердила зменшення загального часу виконання комплексу задач до 31% відносно базових підходів.

2. Удосконалено модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями, яка формалізує неоднорідність сутностей і залежності типу «обчислення-передавання-споживання даних» у межах єдиного процесу планування. Запропоноване розширення моделі використано у практичній процедурі побудови розкладу для гетерогенних сценаріїв із різномірними потоками даних, що забезпечило відтворювану оцінку впливу комунікаційних витрат на балансування навантаження. Це дозволило зменшити стандартне відхилення навантаження до 53%.

3. Набули подальшого розвитку моделі опрацювання та логічного розмежування доступу до даних з урахуванням різномірності сутностей в інформаційних системах. Відмінною особливістю є те, що обмеження доступу та операції опрацювання даних формалізуються на рівні сутностей і використовуються як обмеження під час планування, а не як постфактум-фільтр уже побудованого розкладу, що зменшує кількість непридатних призначень і

повторних перепланувань. Це дозволило підвищити масштабованість до 35%, практичність реалізації – на 25%, а гнучкість налаштувань порівняно з рольовим керуванням доступом – на 40%, забезпечивши сумісність моделей доступу з інтелектуальним плануванням без погіршення показників ефективності.

Практична цінність цього результату полягає в тому, що модель розмежування доступу безпосередньо вбудовується у крок прийняття рішення планувальника (фільтрація недопустимих призначень до формування розкладу), зменшуючи кількість відкатів і повторних перепланувань. Це забезпечує узгодженість заявленого підвищення експлуатаційних показників із реальним процесом планування в прикладних системах.

Узгодженість наукової новизни з практичною складовою забезпечено тим, що кожен заявлений елемент новизни реалізовано у складі прототипу та перевірено експериментально. Гібридне подання залежностей DAG+GERT та механізми адаптивного перепланування реалізовано у прототипному планувальнику й оцінено за критерієм makespan. Модель планування з урахуванням комунікаційних витрат застосовано у сценаріях розповсюдження даних для оцінювання балансування. Модель логічного розмежування доступу інтегровано як обмеження під час прийняття рішень планувальником для усунення недопустимих призначень на ранньому етапі.

Практична значимість отриманих результатів полягає в можливості безпосереднього використання запропонованих моделей та методу під час проектування й удосконалення підсистем планування і розподілу задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих, а також під час побудови механізмів опрацювання даних і логічного розмежування доступу з урахуванням різноманітності сутностей.

Результати дисертаційного дослідження впроваджено у практичну діяльність ІТ-підприємств. Зокрема, у діяльності LLC «Arshilon» (акт впровадження від 23.12.2025 р.) упроваджено інтелектуальний метод планування задач у розподілених інформаційних системах із застосуванням

штучного інтелекту для прийняття рішень щодо призначення задач на ресурси та корекції розкладу в разі зміни стану системи. Практичні рекомендації використано під час організації процесу планування задач у реальному часі та формування політик доступу в прикладних інформаційних системах, що дало змогу покращити балансування ресурсів між структурними підрозділами підприємства. У діяльності компанії «LineUp» (акт впровадження від 17.04.2025 р.) упроваджено інтелектуальний метод планування задач у гетерогенних розподілених інформаційних системах, що базується на поєднанні DAG- та GERT-графів із алгоритмами штучного інтелекту: Graph Attention Network, Proximal Policy Optimization та Bayesian Optimization. Отримані результати використано під час створення модуля автоматизованого керування навантаженням у хмарному середовищі LineUp, що сприяло покращенню балансування ресурсів, зменшенню часу затримок і підвищенню стійкості системи до змін середовища виконання.

Результати дисертаційного дослідження впроваджено в освітній процес Харківського національного економічного університету імені Семена Кузнеця (довідка №26/86-29-22 від 02.03.2026 р.). Зокрема, теоретичні та практичні положення дисертаційної роботи використано під час проходження педагогічної практики та викладання навчальних дисциплін «Безпека програм та даних» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 121 «Інженерія програмного забезпечення», «Комплексний курсовий проект» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 125 «Кібербезпека», а також «Стандартизація та сертифікація захисту інформації» для здобувачів другого (магістерського) рівня вищої освіти за спеціальністю 125 «Кібербезпека та захист інформації». У навчальному процесі використано результати щодо моделювання метаданих у розподілених інформаційних системах, аналізу їх впливу на ефективність оброблення задач, оцінювання навантаження в розподілених обчислювальних середовищах, інтелектуального планування задач та адаптивного розподілу обчислювальних ресурсів, що сприяло підвищенню практичної спрямованості підготовки

здобувачів і формуванню компетентностей з розроблення та аналізу розподілених інформаційних систем.

Отримані покращення мають пряму економічну інтерпретацію. Скорочення makespan та зменшення дисбалансу навантаження підвищують утилізацію ресурсів і знижують витрати на обчислювальні години в надлишковій потужності. Підвищення дедлайн-надійності зменшує кількість порушень SLO та пов'язані з ними штрафи, а контроль планувальника обмежує додаткові витрати на супровід і адміністрування.

Практичну придатність запропонованих рішень підтверджено експериментально такими результатами:

- скорочення загального часу виконання комплексу задач до 31% (статистичну значущість отриманого скорочення підтверджено непараметричним критерієм Манна–Вітні ($p < 0,01$) та оцінкою розміру ефекту δ Кліффа (0,33–0,54) згідно з визначеним протоколом (див. табл. 3.3, 35,2 с проти 51,5 с));

- забезпечення прийнятних накладних витрат планування (у середньому $2,1 \pm 0,4$ мс в офлайн-режимі та $3,0 \pm 0,6$ мс в онлайн-режимі на задачу), що уможливорює застосування в режимі, наближеному до реального часу;

- покращення експлуатаційних властивостей моделей розмежування доступу, зокрема підвищення масштабованості до 35%, практичності реалізації – на 25%, та гнучкості налаштувань – на 40% порівняно з рольовою моделлю керування доступом.

Отже, у дисертаційній роботі отримано цілісний комплекс наукових і практичних результатів. Систематизовано вимоги до планування задач у гетерогенних розподілених інформаційних системах, удосконалено модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями, розроблено інтелектуальний метод планування задач за умов невизначеності метаданих та експериментально підтверджено його ефективність за показниками makespan, балансування навантаження, накладних витрат і робастності.

Апробація результатів дослідження.

Основні положення дисертаційної роботи доповідались та обговорювались на міжнародних науково-практичних конференціях і у фахових наукових заходах, зокрема: IEEE DESSERT (2022, 2023), Free and Open Source Software (2023–2025), CED (2024, 2025), «Проблеми інформатизації» (2024), «Сучасні інформаційні системи та технології в цифровому суспільстві» (2025), «Безпека у кіберсфері» (2025), «Цифрова економіка» (2025), а також у рамках наукових семінарів кафедри кібербезпеки та інформаційних технологій Харківського національного економічного університету імені Семена Кузнеця за тематикою розподілених інформаційних систем та їх безпеки.

Публікації.

За темою дисертації опубліковано 5 наукових праць, з них 5 статей (4 статті у фахових наукових виданнях, що індексуються в Scopus, та 1 стаття у фаховому виданні (категорія Б)). Окрім того є 16 публікацій апробаційного характеру, 2 праці опубліковано у матеріалах міжнародних конференцій IEEE (індексація Scopus), 1 одноосібну публікацію у збірній монографії по результатам конференції та 13 матеріалів доповідей апробаційного характеру на міжнародних конференціях.

Структура і обсяг роботи.

Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Роботу викладено на 226 сторінках машинописного тексту (9,4 авт. арк.). Дисертаційна робота містить 26 рисунків (з них 2 займають 1 повну сторінку), 11 таблиць, 2 додатки – на 12 сторінках, списки використаних джерел до кожного розділу окремо (1 розділ – 61 джерело на 8 сторінках, 2 розділ – 37 джерел на 6 сторінках, 3 розділ – 26 джерел на 4 сторінках, 4 розділ – 28 джерел на 4 сторінках). Обсяг основного тексту дисертації становить 171 сторінку (7,1 авт. арк.).

РОЗДІЛ 1

АНАЛІЗ ТА ПОРІВНЯЛЬНІ ДОСЛІДЖЕННЯ МЕТОДІВ ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ З ВРАХУВАННЯМ НЕВИЗНАЧЕНОСТІ МЕТАДАНИХ

1.1. Аналіз вимог щодо процесу планування задач в розподілених комп'ютерних системах

Розподілені комп'ютерні системи (РКС) визначають сучасний підхід до обчислень та обробки даних, забезпечуючи ефективне використання ресурсів та підтримуючи високий рівень доступності сервісів. Успішна експлуатація розподілених систем вимагає детального аналізу вимог, які ставляться до цих систем.

У сучасних дослідженнях планування задач у розподілених системах розглядається як багатокритеріальна задача керування ресурсами, де масштабованість, пропускна здатність, час відгуку, балансування навантаження та робастність залежать від структури потоку задач, точності ресурсних метаданих і політики призначення робіт на обчислювальні вузли [1–5].

Основними вимогами до розподілених комп'ютерних систем вважаються наступні.

1. Масштабованість (scalability). Розподілені системи повинні бути здатні масштабуватися відповідно до зміни розмірів оброблюваної інформації та об'єму завдань. Тобто масштабованість в розподілених комп'ютерних системах визначає їхню здатність ефективно збільшувати ресурси чи обсяг обробки даних при зростанні навантаження. Цей аспект є ключовим для забезпечення оптимальної продуктивності та відповіді системи на змінні умови використання. Масштабованість можна описати числовими характеристиками.

Горизонтальне масштабування (Scale Out):

кількість вузлів N : Показник кількості вузлів (серверів чи комп'ютерів) може служити метрикою горизонтального масштабування. Чим більше вузлів може бути додано до системи, тим краще;

розділення завдань E : Іншим аспектом є здатність системи розподіляти завдання між різними вузлами, що дозволяє ефективно використовувати ресурси.

Вертикальне масштабування (Scale Up):

ресурси на вузол R : Підвищення продуктивності одного вузла шляхом додавання більш потужних ресурсів, таких як процесори, пам'ять чи сховище;

пропускна здатність PRV : Збільшення пропускної здатності системи шляхом покращення оброблювальної здатності кожного вузла.

Лінійна масштабованість:

пропускна здатність PRL : Визначення того, наскільки зростання ресурсів або навантаження прямує зі збільшенням продуктивності лінійно.

Масштабованість додаванням модулів (Additive Scalability):

можливість додавання нових компонентів Q : Вимірюється можливість системи приймати нові модулі, наприклад, нові вузли чи сервіси, без перерв у роботі.

2. Надійність. Надійність в комп'ютерних системах визначає їхню здатність забезпечити безперебійну та стійку роботу протягом тривалого періоду. Для оцінки надійності можна використовувати різні числові характеристики:

- середній час між відмовами (MTBF – Mean Time Between Failures);
- середній час відновлення (MTTR – Mean Time To Recovery);
- середній час безвідмовної роботи (MTTF – Mean Time To Failure);
- коефіцієнт готовності (Availability);
- фактор надійності (Reliability Factor) – визначає ймовірність того, що система буде працювати без відмови протягом певного періоду.

3. Ефективність. Використання ресурсів повинно бути оптимізованим. Ефективне розподілення завдань між вузлами системи та оптимальне

використання мережевого зв'язку грають важливу роль. Ефективність в комп'ютерних системах може бути виміряна різними числовими характеристиками, які відображають продуктивність та ефективність використання ресурсів системи. Ось деякі з них:

- продуктивність (Performance) – визначає швидкість та продуктивність системи у виконанні завдань;
- коефіцієнт використання ресурсів (Resource Utilization) – показник використання ресурсів системи, таких як процесор, пам'ять, сховище та мережа;
- пропускна здатність (Throughput) – кількість оброблених одиниць інформації або завдань за одиницю часу;
- відгук системи (System Response Time) – час, який потрібен системі для відповіді на запит користувача;
- ймовірність виконання завдань (Task Completion Probability) – ймовірність успішного виконання завдань системою без відмови;
- коефіцієнт ефективності роботи (Efficiency Ratio) – визначає, наскільки ефективно працюють вузли системи або окремі компоненти.

Тому ефективність доцільно оцінювати не лише за продуктивністю окремого вузла, а й за сукупністю показників розкладу: makespan, resource utilization, throughput, latency, cost/energy, fairness та стійкість до помилок у попередніх оцінках тривалості задач [1–5, 8–10, 24, 33, 34].

4. Безпека. Забезпечення конфіденційності, цілісності та доступності даних є критичним. Розподілені системи повинні враховувати сучасні методи захисту від кіберзагроз та несанкціонованого доступу. Оцінка безпеки комп'ютерних систем може бути здійснена за допомогою числових характеристик, що відображають рівень захищеності та вразливості системи до потенційних загроз. Нижче наведені деякі з таких характеристик:

- рівень вразливості (Vulnerability Level) – оцінка загальної вразливості системи;
- інциденти безпеки (Security Incidents) – кількість виявлених інцидентів безпеки протягом певного періоду;

- порушення безпеки (Security Breaches) – кількість фактичних або спроб порушення безпеки;
- рівень шифрування (Encryption Strength) – сила використовуваних шифрувальних алгоритмів;
- час виявлення порушень (Time to Detect) – час, який потрібен для виявлення потенційних загроз після їх виникнення;
- час відновлення (Recovery Time) – час, який потрібен для відновлення системи після інциденту;
- ймовірність виявлення (Detection Probability) – ймовірність виявлення та відсіювання потенційних загроз;
- рівень відповідності стандартам безпеки (Compliance Level) – оцінка того, наскільки система відповідає вимогам та стандартам безпеки.

5. Сумісність. Системи повинні бути сумісними з різноманітним програмним та апаратним забезпеченням, щоб забезпечити легкість інтеграції та обмін даними. Оцінка сумісності комп'ютерних систем може бути виконана за допомогою числових характеристик, які вказують на здатність системи взаємодіяти з іншими програмними та апаратними засобами. Ось деякі ключові характеристики:

- стандарти сумісності (Compatibility Standards) – оцінка відповідності системних компонентів до встановлених стандартів;
- протоколи взаємодії (Interaction Protocols) – кількість та різноманітність протоколів, які підтримуються системою;
- версійна сумісність (Version Compatibility) – здатність системи взаємодіяти з різними версіями програмного чи апаратного забезпечення;
- підтримка форматів даних (Data Format Support) – розмаїття форматів даних, які може опрацьовувати система;
- мовна сумісність (Language Compatibility) – здатність системи працювати з різними програмними мовами та мовами програмування;
- сумісність з операційними системами (Operating System Compatibility) – здатність системи взаємодіяти з різними операційними системами;

- сумісність інтерфейсів (Interface Compatibility) – здатність взаємодіяти з різними інтерфейсами та стандартами вводу/виводу;

- сумісність з мережевими стандартами (Network Compatibility) – здатність системи взаємодіяти з різними мережевими пристроями та стандартами.

Усі зазначені вимоги безперервно пов'язані з методами їх безпосереднього аналізу. Серед цих методів можливо виділити наступні:

- вивчення бізнес-вимог: ретельне вивчення потреб користувачів та бізнес-вимог, що передбачаються від системи, дозволяє побудувати ефективну архітектуру;

- моделювання працездатності: використання моделей для аналізу та прогнозування працездатності системи при різних навантаженнях;

- створення критеріїв надійності: визначення критеріїв та параметрів, які характеризують надійність системи, і розробка методів їх оцінки;

- аналіз безпеки: оцінка вразливостей системи та розробка стратегій для їх запобігання та виявлення;

- визначення масштабованості: встановлення граничних значень обробки даних та можливостей масштабування системи.

Таким чином, аналіз вимог є ключовим етапом у розробці та експлуатації розподілених комп'ютерних систем. Зрозуміння потреб користувачів та бізнес-вимог, а також врахування аспектів масштабованості, надійності, ефективності та безпеки дозволяють побудувати високоефективні та надійні системи, які відповідають сучасним вимогам інформаційного суспільства.

Проведені дослідження показали, що одними з основних технічних засобів, що забезпечує підвищення ефективності планування та обробки задач в розподілених інформаційних системах, є спеціалізовані обчислювальні засоби. Тому виникає завдання підвищення продуктивності таких засобів, шляхом вдосконалення відповідних методів планування задач в розподілених інформаційних системах.

Отже, підвищення продуктивності РІС потребує не тільки спеціалізованих обчислювальних засобів, а й формалізованих методів планування, здатних узгоджувати залежності між задачами, часові обмеження, неоднорідність ресурсів і неповноту початкових метаданих [1–5, 14, 15, 24].

1.2. Шляхи підвищення продуктивності розподілених інформаційних систем

Розв'язання завдань обробки інформації у розподілених інформаційних системах базується на використанні обчислювальних ресурсів. Один з ключових показників, що характеризують ці засоби і впливають на пропускну здатність розподілених інформаційних систем, - це їх продуктивність [1, 2]. Під продуктивністю обчислювального засобу розуміється кількість елементарних арифметичних операцій, виконуваних за одиницю часу [3]. Значення цього показника, передусім, визначається тактовою частотою та конфігурацією процесора.

Один із методів для підвищення продуктивності розподілених інформаційних систем - це створення нової елементної бази з підвищеною тактовою частотою. Однак недоліками цього шляху є висока трудомісткість, вартість та значні часові затрати на розробку нових інструментів, технологій розробки та виробництва цифрових пристроїв [2, 3].

Інший метод для підвищення продуктивності розподілених інформаційних систем ґрунтується на застосуванні методів та засобів паралельної та адаптивної реалізації алгоритмів обробки інформації, включаючи використання існуючої елементної бази. Це передбачає вирішення задачі створення (синтезу) раціональної конфігурації обчислювальної системи і пов'язано з розробкою нових технологій синтезу паралельних цифрових пристроїв та програм на основі наявних комп'ютерних систем, і не вимагає

великих витрат на дорогі технології виробництва. Зараз використання методів паралельної та адаптивної обробки розглядається як найбільш перспективний шлях для підвищення пропускної спроможності [4, 5].

З позицій сучасного планування робочих процесів (workflow scheduling) такі задачі доцільно подавати у вигляді графів залежностей, насамперед спрямованих ациклічних графів (DAG), що дає змогу формалізувати порядок виконання операцій, критичний шлях, комунікаційні витрати та час завершення розкладу (makespan) як цільову характеристику [1, 3, 5, 24, 32, 34].

Практична реалізація паралельної обробки включає вирішення декількох основних завдань. По-перше, серед них важливим є вибір методу паралельної обробки інформації, де на практиці найбільш поширені метод глобального паралелізму, конвеєрна та декомпозиційна обробка. По-друге, при організації паралельних обчислень велика увага повинна бути приділена вибору алгоритмів, які ефективно розпаралелюють вирішення різноманітних функціональних завдань. До таких алгоритмів належать, наприклад, "систолічні алгоритми" (такі як цифрова фільтрація і матричні обчислення), які володіють загальними корисними властивостями регулярності, рекурсивності і локалізованості передачі даних [6, 7]. Для цих алгоритмів вирішення завдання розробки (синтезу) цифрових структур, які реалізують їх, зазвичай є досить простим. Однак на практиці, у рамках комплексу функціональних завдань в технічних системах, такі "регулярні" алгоритми зустрічаються рідко, тому завдання синтезу паралельних цифрових структур стає нетривіальним і вимагає автоматизації.

За результатами аналізу видно, що в літературі широко розглядаються як системи автоматизованого проєктування, так і методи синтезу функціональних структур спеціалізованих процесорів.

При цьому сучасний розвиток розподілених середовищ переносить акцент із суто апаратної реалізації паралелізму на комплексне планування потоків робіт, у якому враховуються взаємозалежності задач, профілі ресурсів, змінність навантаження та віддаленість обчислювальних вузлів [2–5, 28–32].

Аналіз систем автоматизованого проєктування цифрових пристроїв та методів синтезу функціональних структур спецпроцесорів.

Існує два основні підходи до розробки цифрових пристроїв: застосування принципу схемної логіки або принципу програмованої логіки [6]. У першому випадку проєктування ґрунтується на виборі необхідного для втілення алгоритму (або комплексу алгоритмів) набору функціональних модулів та визначенні відповідної схеми їх з'єднань. Пристрої, спроектовані за принципом схемної логіки, забезпечують високу швидкодію та надійність в рамках заданого типу технології елементів. Недоліком цього підходу є обмежена можливість "перебудови" структури пристроїв та систем у разі необхідності змін або розширення їх функціональних можливостей [7, 50].

Інший підхід до проєктування цифрових пристроїв базується на використанні принципу програмованої логіки. Цей метод передбачає створення універсального пристрою за допомогою однієї або кількох великих інтегральних схем (BIC), де необхідне функціонування забезпечується введенням певної програми в пам'ять пристрою. Однак пристрої, створені за цим принципом, характеризуються меншою швидкістю та надійністю порівняно із апаратно реалізованими спеціалізованими процесорами.

Класифікацію розробки цифрових пристроїв можна умовно провести на традиційні системи автоматизованого проєктування (САПР), нові САПР та системи автоматичного проєктування (САП) для мультипаралельних цифрових пристроїв. Фактично, порівняльний аналіз цих підходів виконується на основі основних критеріїв, до яких відносяться [6, 7]:

- функціональне призначення системи проєктування;
- структура рівнів проєктування;
- склад основних етапів проєктування;
- мови специфікації об'єктів проєктування;
- математичний апарат для опису алгоритмів функціонування пристроїв;
- спектр підтримуваних методів паралельної обробки даних;

- ступінь охоплення формалізацією основних етапів проєктування;
- архітектура проєктованих пристроїв: не перебудовується, перебудовується;
- формат даних при специфікації об'єктів проєктування: текстовий/числовий;
- ступінь автоматизації основних етапів проєктування та характер автоматизації: автоматизований/автоматичний;
- кількісні характеристики проєкт, що реалізуються: гранична складність проєктованих об'єктів, тимчасові витрати на проєктування;
- роль людини у системі проєктування;
- потенційні можливості для вирішення проблем System-on-Chip (SoC) та Time-to-Market (T&M) [6, 11].

Під System-on-Chip (SoC) у цьому підрозділі розуміється проєктування високорівнево інтегрованих цифрових систем, у яких різко зростають розміри проєктів, кількість взаємозалежних блоків та вимоги до паралельного виконання етапів розробки. Під Time-to-Market (T&M) розуміється сумарний час від постановки задачі до отримання верифікованого продукту та його готовності до впровадження/випуску на ринок. Зазначений критерій використано як індикатор того, чи здатна система автоматизації (САПР/САП) масштабуватися зі зростанням складності проєктів і скорочувати цикл розробки завдяки кращому розподілу робіт, ресурсів та обмежень у часі.

Аналогічна постановка виникає у гетерогенних розподілених інформаційних системах. Зростання складності графів робіт, обмежені ресурси та часові вимоги призводять до потреби інтелектуального планування задач. Тому огляд САПР/САП і критерії SoC/T&M у цьому розділі розглядаються як мотиваційний приклад класу задач, де ефективний розподіл робіт і ресурсів (у тому числі за умов неповних або неточних описів) є критичним, а запропонований у дисертації метод спрямований на розв'язання цієї проблеми вже в контексті розподілених обчислювальних середовищ.

У термінах розподіленого планування такі конвеєри можуть бути представлені як моделі робочих процесів та спрямованих ациклічних графів (workflow/DAG-моделі), для яких сучасні підходи враховують обмеження передування, гетерогенність обчислювальних вузлів, точність оцінок часу виконання та вплив критичного шляху на підсумковий час завершення [1, 3, 5, 20, 33, 34]. Суть наведених вище проблем полягає відповідно до неможливості створювати за участю людини у прийнятні тимчасові терміни проекти зі складністю 40...60 млн. вентилів (верхня межа за участю людини – 10 млн. вентилів) та у нездатності САПР суттєво скоротити терміни проєктування та видачі продукту на ринок [6, 11, 12].

Зміст основних етапів традиційного автоматизованого синтезу цифрових пристроїв, їх характеру (творчий /формалізований), а також ролі людини-фахівця у процесі синтезу представлені у табл. 1.1.

Таблиця 1.1

Основні етапи автоматизованого синтезу цифрових пристроїв

№ етапу	Зміст етапу синтезу	Характер етапу	Виконавець етапу
1	Формулювання завдання на проєктування: тип пристрою, склад розв'язуваних завдань, апаратний базис, що використовується, вимоги та обмеження.	Творчий	Людина
2	Розробка RTL -схеми пристрою та її формалізованого опису мовами VHDL , Verilog.	Творчий	Людина (за допомогою КС)
3	Розробка вентиляної схеми пристрою та її формалізованого опису.	Творчий	Людина (за допомогою КС)
4	Функціональне моделювання (при однакових всім функціональних модулів тимчасових затримках) і верифікація схеми.	Формальний	КС
5	Тимчасове моделювання вентиляної схеми (з урахуванням реальних часових затримок функціональних модулів) та її корекція.	Формальний	КС та людина
6	Компонування (розбивка вентиляної схеми на блоки, що конфігуруються) і трасування міжкомпонентних зв'язків.	Формальний	КС
7	Тимчасове моделювання вентиляної схеми (з урахуванням реальних часових затримок).	Формальний	КС
8	Верифікація та корекція вентиляної схеми (з урахуванням результатів тимчасового моделювання).	Формальний	КС
9	Формування вихідної документації проєктування.	Формальний	КС

З таблиці 1.1 видно, що на етапах постановки завдання на проектування та розробку RTL – схеми пристрою, а також її формалізованого опису мовами проектування VHDL або Verilog активна роль відводиться людині.

При цьому системний та функціональний рівні проектування не формалізовані та якість їх виконання значною мірою визначається кваліфікацією спеціаліста-проектувальника.

Як приклад домену інженерних конвеєрів розглянемо традиційні САПР, що найчастіше використовуються на практиці, та їх інструментальних засобів (P-CAD, DesignLab, OrCAD та ін) міститься в роботах [1, 2, 11, 13]. Їх перелік та основні характеристики представлені у табл. 1.2.

Зазначені САПР (P-CAD, DesignLab, OrCAD та ін.) у даному розділі розглядаються не як “платформа-основа”, у яку вбудовується новий алгоритм планування, а як представники прикладного домену, де на практиці виникають великі конвеєри взаємозалежних робіт (синтез, моделювання, верифікація, генерація артефактів, перевірки правил, тестування тощо). Такі конвеєри природно формалізуються як граф задач і виконуються у гетерогенних розподілених середовищах (локальні кластери/хмара/CI-інфраструктура) з часовими обмеженнями та неповними метаданими.

Інтеграція з плануванням у роботі виконується на рівні інфраструктури, а не “всередині САПР”. Запропонований метод призначений для планування та диспетчеризації зовнішніх обчислювальних робіт (jobs/tasks), які породжуються конвеєром проектування і запускаються в пакетному (batch) або потоковому (online) режимі. На практиці це реалізується через стандартні механізми запуску задач. Пакетні режим, командний рядок, скрипти, файлові інтерфейси та керовані пайплайни (наприклад, у CI), тобто без необхідності модифікувати внутрішні модулі конкретної САПР і без вимоги наявності спеціального API планування у P-CAD/OrCAD.

Для хмарних варіантів OrCAD та подібних інструментів справді типовим є фокус на колаборації та доступі до бібліотек та даних, тоді як продуктивність обчислень забезпечується зовнішньою інфраструктурою (сервери, кластери,

сховища, ліцензійні сервіси). Саме цей рівень (інфраструктурне планування робіт конвеєра) і є предметом запропонованого в дисертації методу.

Типовий стенд виконання інженерного конвеєра включає робочу станцію інженера (IDE/САПР-клієнт), сервер спільних даних проєкту (спільне сховище артефактів), ліцензійний сервер (за потреби), керуючий вузол планувальника та пул обчислювальних вузлів (CPU/GPU) у кластері або хмарі. Обчислювальні роботи конвеєра передаються планувальнику як job-запити з метаданими (ресурсні вимоги, оцінка тривалості, дедлайн, залежності), після чого планувальник виконує призначення на ресурси та збір телеметрії.

Така постановка узгоджується з підходами ресурсного забезпечення (resource provisioning) і планування робочих процесів (workflow scheduling) у хмарних, граничних та туманних обчисленнях, де якість розкладу залежить від вибору ресурсів, локальності даних, мережових затримок, обмежень якості обслуговування (Quality of Service, QoS) і прогнозованої поведінки потоку задач [28–32, 44].

Таблиця 1.2

Основні характеристики традиційних САПР

№ з /п	Характеристика	Зміст характеристики
1	Функціональне призначення	Проектування PLC, ASIC, FPGA, друкованих плат
2	Рівні проектування	Функціональний (F /RTL), логічний (L) та вентильний (G) рівні
3	Етапи проектування	Створення F-схеми, верифікація / корекція, функціональне (F) та тимчасове (T) моделювання для RTL, логічних та вентильних схем
4	Мови специфікації об'єктів проектування	Мови опису схем: VHDL , Verilog , HDL
5	Математичний апарат опису алгоритмів функціонування пристроїв (компонентів)	Таблиці станів, таблиці переходів, графи переходів, граф-схеми алгоритмів, логічні схеми алгоритмів, кінцеві автомати, булеві рівняння
6	Підтримувані методи паралельної обробки	Поеднання незалежних операцій, конвеєрна обробка
7	Ступінь формалізації етапів проектування	Немає формалізації створення F-і RTL-схем
8	Підтримка/відсутність реконфігурації	Підтримка проектування реконфігурованих ПЛІС (FPG A)
9	рівні реконфігурації	Рівень логічних/вентильних елементів

Продовження таблиці 1.2

№ з /п	Характеристика	Зміст характеристики
10	Формат даних, що застосовується	Текстовий
11	Характер проектування	Автоматизований
12	Складність проектів/час проектування	До 10 млн. еквівалентних вентилів / багато місяців та роки
13	Роль людини	Визначальна роль творчості людини
14	Можливість вирішення проблем SoC та T &M	відсутні

З таблиці 1.2 видно (див. пункт 7), що у традиційних САПР не формалізовано синтез функціональних структур, що є найбільш трудомістким завданням розробки цифрових пристроїв. Крім того, в умовах їх застосування не можна вирішити проблеми SoC і T &M.

Тому останнім часом суттєво активізувалися дослідження у напрямку пошуку нових підходів до проектування цифрових пристроїв, удосконалення науково-методичного апарату їх проектування та розробки нового інструментарію. До основних компаній-розробників нових САПР для автоматизованого проектування замовних надвеликих інтегральних схем (ASIC) і програмованих логічних інтегральних мікросхем (ПЛІС) (FPGA, CPLD) вважаються Synopsys, Cadence, Aldec, CoWare, Mentor Graphics, Synplicity, Simucad, Altera, Xilinx, Actel, Cypress, Quicklogic та ін. Аналіз публікацій [1, 2, 11, 13] показує, що розширення можливостей та підвищення ефективності САПР може бути забезпечене:

- розробкою інструментарію проектування на системному рівні;
- повторним використанням розроблених раніше компонентів (блоків інтелектуальної власності (IP)), які були спроектовані для багаторазового застосування у різних проектах [6];
- спільним проектуванням апаратно-програмних засобів, заснованим на виборі співвідношення між апаратними та програмними компонентами на

основі моделювання та подальшого поєднання у часі процесів їх створення [6, 7, 14].

Основні характеристики нових САПР представлені в табл. 1.3.

Таблиця 1.3

Основні характеристики та нові САПР

№ з /п	Характеристика	Зміст характеристики
1	Функціональне призначення	Проектування CPU, ASIC, FPGA, CPLD, IP Reuse
2	Рівні проектування	Системний (S), функціональний (F/RTL), логічний (L) та вентиляний (G) рівні
3	Етапи проектування	Створення F-опису, поділ апаратних засобів та програмного забезпечення, створення та моделювання RTL-, L-, G-схем, верифікація
4	Мови специфікації об'єктів проектування	Мови опису схем: VHDL, Verilog, HDL. Мови SystemC (версії 1.0, 2.0), Superlog
5	Математичний апарат опису алгоритмів функціонування пристроїв (компонентів)	Таблиці станів і переходів, графи переходів, граф-схеми та логічні схеми алгоритмів, кінцеві автомати, булеві рівняння
6	Підтримувані методи паралельної обробки	Поєднання незалежних операцій, конвеєрна обробка
7	Ступінь формалізації етапів проектування	Формалізація об'єктів S-, F- та RTL-рівнів на основі мов VHDL, Verilog, HDL, SystemC
8	Підтримка/відсутність реконфігурації	Підтримується проектування реконфігурованих на рівні логічних елементів ПЛІС (FPGA, CPLD)
9	Рівні реконфігурації	Рівень логічних елементів ПЛІС
10	Формат даних, що застосовується	Текстовий
11	Характер проектування	Автоматизований
12	Складність проектів/час проектування	Системний рівень проектування: понад 1 млн. вентилів на місяць на одного проектувальника, складність проектів – 100 000...1 000 000 вентилів. Схемотехнічний рівень: близько 100 000 вентилів на місяць на проектувальника, складність проектів – 10 000 вентилів. Зменшення часу на 30...50% проти традиційними САПР
13	Роль людини	Значна роль творчості людини
14	Можливість вирішення проблем SoC та T & M	В даний час відсутня

З таблиці 1.3 видно, що у нових САПР частково формалізований процес проектування F – і RTL – рівнів з допомогою мов VHDL , Verilog , SystemC . Їх застосування дозволило значно спростити процес синтезу цифрових пристроїв (до 1 млн. вентилів) та скоротити часові витрати на їх проектування приблизно на 30...50%. Однак, як і раніше, серед нерозв'язних залишилися проблеми SoC і T&M.

Крім того, і традиційні, і нові САПР не дозволяють використовувати при проектуванні цифрових засобів всю сукупність відомих методів паралельної обробки або їх практично доцільну комбінацію.

Ряд фахівців проводили дослідження, спрямовані на обґрунтування та створення науково-методичного апарату та програмних засобів систем автоматичного проектування паралельних цифрових пристроїв. До таких нових математичних засобів синтезу та аналізу мультипаралельних цифрових систем належать [14, 15, 16]:

- апарат операторних рекурсивних функцій утворюючих полів;
- алгебра кодових матриць;
- біматрична алгебра;
- алгебра тимчасових паралельних граф-схем алгоритмів.

Розроблено методи формального перетворення вихідних алгоритмів, представлених, наприклад, у вигляді Сі-програм, відповідні графи, а також методи синтезу паралельних тимчасових моделей алгоритмів [14, 17, 18].

Крім того, [14, 15, 19] розроблені методи формального функціонального проектування паралельних цифрових пристроїв. Об'єктом функціонального проектування є спецпроцесор з жорсткою (не перебудовується) структурою [6, 7, 20], а результатом - або його функціональна схема, що задовольняє заданим вимогам і обмеженням, або обґрунтований висновок про неможливість вирішення завдання синтезу в умовах наявних вихідних даних.

Перелік основних характеристик систем автоматичного проектування паралельних цифрових пристроїв наведено в табл. 1.4.

Основні характеристики та САП паралельних цифрових пристроїв

№ з /п	Характеристика	Зміст характеристики
1	Функціональне призначення	Проектування замовних апаратних мультипаралельних спецпроцесорів із жорсткою архітектурою (типу ASIC, VLSI)
2	Рівні проектування	Системний (S), функціональний (F/RTL), логічний (L) та вентильний (G) рівні
3	Етапи проектування	Синтез графа Сі – програм, синтез оптимальної паралельної моделі завдання, синтез функціональної / RTL, логічної та вентильної схем пристрою, синтез, верифікація / корекція, функціональне (F) та тимчасове (T) моделювання RTL – , логічної та вентильної схем
4	Мови специфікації об'єктів проектування	Мова Сі (специфікації вихідних завдань)
5	Математичний апарат опису паралельних цифрових пристроїв (компонентів)	Апарат рекурсивних функцій утворюючих полів, кодових матриць алгебри, біматрична алгебра, алгебра тимчасових паралельних граф – схем алгоритмів
6	Підтримувані методи паралельної обробки	Поєднання незалежних операцій, конвеєрна обробка, декомпозиційний метод, кодово-матричний метод
7	Ступінь формалізації етапів проектування	Формалізація процесів на S, F /RTL, L – та G – рівнях. Формалізація створення паралельних тимчасових моделей S – , F / RTL, L – та G – рівнів
8	Підтримка / відсутність реконфігурації	Підтримка проектування пристроїв, що реконфігуруються, відсутня
9	Рівні реконфігурації	відсутні
10	Застосовуваний формат даних	Числовий формат сполучено-зовнішніх множин (CBM) на всіх рівнях проектування
11	Характер проектування	Автоматичне
12	Складність проектів / час проектування	Попередні оцінки [37]: багаторазове збільшення допустимої складності проектів / зниження витрат часу на проектування та видачі продукту на ринок від 50 до 100 і більше разів
13	Роль людини	Завдання вихідних даних: Сі – програм завдань, вимог (обмежень), бібліотек компонентів (елементів)
14	Можливість вирішення проблем SoC та T &M	Є концептуально

З таблиці 1.4 видно (див. пункт 12), що застосування САП паралельних цифрових пристроїв дозволяє багаторазово порівняно з САПР збільшити допустиму складність проектів та скоротити час проектування. У цих системах концептуально закладено можливість вирішення проблем SoC та T &M. Крім

того, розширено перелік застосовуваних при синтезі цифрових пристроїв методів паралельної обробки інформації.

Однак, як і раніше, відсутня можливість одночасного застосування сукупності методів паралельної обробки (або практично доцільної їх комбінації) і проектування спецпроцесорів, що перебудовуються.

Аналіз традиційних (див. табл. 1.2) та нових САПР (див. табл. 1.3), а також перспективних розробок САП (табл. 1.4) показує, що:

- концепцією побудови відомих САПР є виконання людиною найскладніших, творчих, зазвичай, неформалізованих етапів проектування цифрових пристроїв;

- визначальна роль системах автоматизованого проектування спеціаліста – проектувальника призвела до “кризи” сучасних САПР, який проявляється у проблемах SoC, T&M і неможливості використовувати всю сукупність методів паралельної обробки інформації при автоматичному виборі їх раціонального співвідношення для різних галузей застосування, вирішуваних функціональних завдань, а також вимог та обмежень замовника;

- використання «інерційного» текстового формату специфікації об'єктів на всіх рівнях та етапах проектування обмежує можливості скорочення часу проектування та граничну складність проектів.

Під реальним часом в інформатиці розуміється режим роботи обчислювальної системи, у якому час «відгуку» на «подію» вбирається у допустимої величини.

Можлива суттєва нерівномірність інтенсивності вхідних даних вимагає від обчислювальних засобів забезпечення обробки інформації в умовах максимально можливої інтенсивності, тобто повинна бути забезпечена необхідна пропускна спроможність системи. При цьому в умовах обстановки, що динамічно змінюється, доцільно оперативну зміну пріоритетів відповідних функціональних завдань аж до припинення виконання одних і надання необхідних обчислювальних ресурсів для інших. Тому обчислювальні засоби повинні мати властивість «перебудови». Це дозволить залежно від повітряної

обстановки, що складається, вирішувати або весь комплекс покладених на них функціональних завдань, або їх деяку сукупність, і тим самим, забезпечити необхідну пропускну здатність. Про актуальність робіт, вкладених у розробку перебудовуваних спецпроцесорів свідчить велика кількість література [6, 7, 14, 16, 21]. В даний час подібні дослідження по праву відносяться до центральних напрямів створення архітектур паралельних, у тому числі матричних процесорів, що реалізуються у вигляді НВІС.

Таким чином, наукову і прикладну актуальність набуває розробка методики функціонального синтезу паралельних спецпроцесорів, що перебудовуються, із заданими технічними характеристиками для КС.

Перш за все, необхідно провести вибір показників ефективності паралельної реалізації алгоритмів.

Обчислювальні засоби (спецпроцесори) мають характерні риси складної системи. Якість функціонування складної системи зазвичай оцінюють за допомогою показників ефективності, які служать для визначення ступеня її відповідності висунутим вимогам та поставленим перед нею завданням [16]. Для складної системи важко обґрунтувати єдиний (узагальнений) показник ефективності, тому на практиці застосовується сукупність про приватних показників, вибір яких залежить від цільового призначення, тактико-технічних і техніко-експлуатаційних параметрів системи. Як правило, серед них домінуючими є один або кілька показників, які повинні відповідати таким вимогам [7]:

- чисельно характеризувати ступінь виконання найважливішої цільової функції системи;
- забезпечувати можливість оцінки витрат, необхідних для досягнення різних значень, та ступеня впливу різних зовнішніх факторів;
- бути по можливості простими за змістом, які легко вимірюються і мати малу дисперсію, тобто слабо залежати від безлічі неконтрольованих факторів.

Вважатимемо, що спецпроцесор має модульну структуру, а число типів модулів відповідає числу різних операцій, виконуваних у ньому.

Позначимо через:

ρ - Кількість різних типів модулів, що виконують конкретну операцію над операндами заданої розрядності;

$\Theta = \{\theta_i\}_{i=1}^{\rho}$ - безліч різних типів θ_i модулів;

$N = \{n_i\}_{i=1}^r$ ($r \leq \rho$) - вектор (множина) значень n_i кількості модулів типу θ_i ;

$K = \{k_{i\xi}\}_{i=1}^r$ - вектор (множина) значень $k_{i\xi}$ кількості операторів $P_j \in P(\xi)$ ξ -ї гілки алгоритму, що виконуються модулем типу θ_i ;

$T = \{t_i\}_{i=1}^r$ - вектор (множина) значень t_i часу виконання відповідного оператора модулем типу θ_i ;

$Q = \{q_i\}_{i=1}^r$ - вектор (множина) значень q_i еквівалентної вентильної складності модулів типу θ_i .

Аналіз літератури [6, 7, 16, 22] показав, що для оцінки ефективності і паралельних тимчасових моделей алгоритму (комплексу алгоритмів), і функціональних структур паралельних перебудовуються спецпроцесорів (СПП) їх реалізують, використовується, як правило, одна і та сама сукупність відомих показників. Їх умовно можна поділити на основні та додаткові. До основних показників ефективності паралельної реалізації алгоритму (або комплексу алгоритмів) в СПП можуть бути віднесені такі.

По-перше, час виконання алгоритму

$$t = \max_{\xi} (T_{\xi}), \quad (1.1)$$

де T_{ξ} - час реалізації ξ -ї гілки паралельного алгоритму ($\xi \in 1, W$, W - число гілок алгоритму), що визначається за формулою

$$T_{\xi} = \max_{P_j \in P(\xi)} (t_j^H + t_j), \quad (1.2)$$

де $P(\xi)$ - безліч операторів ξ -ї гілки;

t_j^H - Момент початку оператора $P_j \in P(\xi)$;

t_j - Відносна " тимчасова глибина " оператора $P_j \in P(\xi)$.

По-друге, для обчислювальних засобів (спецпроцесорів) систем управління та обробки інформації реального часу одним із найважливіших є вимога щодо забезпечення введення даних та (або) видачі результатів обробки із заданим періодом T або темпом (частотою) $F = 1 / T$ їх оновлення. Значення параметрів T або F , зазвичай, визначаються швидкістю протікання реальних фізичних процесів.

Мінімальне значення періоду оновлення вхідних даних $T_{\text{про min}}$, що забезпечується застосуванням різних методів паралельної обробки (а) – поєднання незалежних операцій; б) – конвеєрна обробка; в) – декомпозиційна обробка), визначається із співвідношення [6, 16]:

$$T \geq T_{o \min} = \begin{cases} t, \\ \max_i (t_i + 2 \cdot t_{\phi}^0), \\ \max_i (t_i + 2 \cdot t_{\phi}^0) / \Delta, \end{cases} \quad (1.3)$$

де t – час виконання алгоритму під час використання поєднання незалежних операцій (обмеження на обладнання відсутні);

t_i – час виконання операції модулем типу θ_i ;

t_{ϕ}^0 – величина циклу зчитування даних з модулів пам'яті, що виконують роль «фіксаторів» проміжних результатів та використовуються в СПП для організації конвеєрної та (або) декомпозиційної обробки;

$\Delta = \left\lceil \frac{\max(t_i)}{T} \right\rceil$ - коефіцієнт декомпозиції (тут через $[x]$ позначено найближче більше ціле x).

По-третє, важливе прикладне значення мають показники, що характеризують апаратну складність цифрових систем реального часу та ефективність використання компонентів (модулів), що входять до їх складу. До таких показників можна віднести еквівалентну вентильну складність і коефіцієнт використання (завантаження) апаратних засобів при паралельній реалізації алгоритму (комплексу алгоритмів).

Еквівалентна вентильна складність СПП визначається співвідношенням [6, 7, 16].

$$Q = \sum_{i=1}^{\rho} q_i \cdot n_i \quad (1.4)$$

Коефіцієнт середнього завантаження модулів типу θ_i розраховується за формулою [7, 16]

$$S_i = \frac{\sum_{\xi=1}^W (t_i \cdot k_{i\xi} \cdot P_{\xi})}{n_i \cdot t}, \quad (1.5)$$

де W - Число гілок в алгоритмі;

P_{ξ} - Імовірність реалізації ξ -ї гілки.

До додаткових показників ефективності можна віднести середній час виконання алгоритму, коефіцієнт зниження часових витрат, загальний коефіцієнт завантаження обладнання і коефіцієнт ефективності розпаралелювання.

У задачах планування для хмарних і кластерних середовищ цей перелік зазвичай розширюється метриками середнього часу завершення робіт, часу очікування в черзі, справедливості розподілу ресурсів між користувачами, енергоспоживання, а також чутливості розкладу до похибок у метаданих [3–5, 24, 33, 47, 50].

Середній час виконання алгоритму визначається із співвідношення [7, 16]

$$t_{cp} = \sum_{\xi=1}^W P_{\xi} t_{\xi}, \quad (1.6)$$

де t_{ξ} - час реалізації ξ -ї гілки паралельної тимчасової моделі алгоритму.

Практичний інтерес представляє кількісна оцінка зниження часових витрат виконання алгоритму (комплексу алгоритмів) за рахунок переходу до паралельної реалізації алгоритму. Відповідний коефіцієнт обчислюється за співвідношенням

$$\delta t = \frac{t_{\text{посл}}}{t_{\text{ср}}}, \quad (1.7)$$

де $t_{\text{посл}}$ - Середній час послідовної реалізації алгоритму.

Ефективність використання обладнання, наприклад, типових модулів спецпроцесора необхідного переліку та складу, можна оцінити за значенням загального коефіцієнта завантаження обладнання виду [16, 23]

$$S = \frac{\sum_{i=1}^{\rho} S_i}{\sum_{i=1}^{\rho} n_i}. \quad (1.8)$$

Для КС, що працюють в умовах жорсткого ліміту часу, заданого циклу управління, основними обмеженнями, що накладаються на обчислювальні засоби, є часові. Однак, вирішуючи завдання функціонального синтезу паралельних спецпроцесорів, що перебудовуються, важливо вміти оцінити “питому вартість” підвищення продуктивності або зниження часових витрат на реалізацію комплексу функціональних алгоритмів за рахунок використання можливостей щодо їх розпаралелювання. Узагальненим показником, що характеризує ці оцінки, є коефіцієнт ефективності розпаралелювання виду

$$R = \delta t \cdot S. \quad (1.9)$$

Вочевидь, що за умов першочергового виконання «часових» вимог «економічно» доцільним виявляється підвищення продуктивності обчислювальних засобів КС реального часу, у якому максимізується значення коефіцієнта R .

Слід звернути увагу на те, що в залежності від сукупності вимог і особливостей функціональних завдань, що вирішуються в КС, деякі додаткові показники, наприклад, коефіцієнт ефективності розпаралелювання, можуть бути віднесені до основних і навпаки.

Саме тому вибір показників ефективності повинен відповідати не тільки фізичній природі обчислювального засобу, а й типу середовища виконання: НРС-кластер, хмарна інфраструктура, edge/fog-рівень або змішана гетерогенна система [4, 28–31, 46, 47].

1.3. Аналіз відомих планувальників завдань і систем управління ресурсами

Для практичної реалізації таких систем розробники використовують різноманітні програмні засоби. Серед такого програмного забезпечення можна виділити: HTCondor, DIET, Slurm і Torque та ін.

У сучасних кластерних, грид-, хмарних і контейнеризованих середовищах ці засоби утворюють окремий клас систем керування ресурсами та пакетного запуску завдань; їх еволюція демонструє перехід від статичних черг і ручних політик до розширюваних механізмів моделювання, прогнозування та адаптивного вибору розкладу [12, 24, 27–29, 46, 47].

Аналіз та порівняльні дослідження поширених планувальників завдань і систем управління ресурсами.

1. Програмне забезпечення HTCondor.

Програмне забезпечення HTCondor створювалося з метою вирішити існуюче завдання завантаження всіх доступних ресурсів, шляхом створення середовища про роботу даних з високою пропускною здатністю (High-Throughput Computing) з індивідуальних робітників станцій і обчислювальних кластерів. У загалом, HTCondor є спеціалізованою пакетною системою для управління величезною кількістю завдань. Як більшість пакетних систем, HTCondor надає механізм черг, політики планування, схему пріоритетів та класифікацію ресурсів.

HTCondor може планувати виконання завдань на виділених робочих станціях, як і більшість пакетних систем, але на відміну від традиційних пакетних систем, також здатний виконувати ефективне завантаження машин для виконання завдань [11, 13, 24, 27].

HTCondor має наступними можливостями:

1. Знаходити доступні машини для надійшли завдань;
2. Підтримувати зберігання проміжного стану виконання завдань;

3. Підтримувати міграцію виконуваних завдань між машинами;
4. Здійснювати управління доступними ресурсами шляхом відображення по споживачів ресурсів на джерела, що надають ресурси, за допомогою системи повідомлень;
5. Підтримувати впорядковане виконання групи взаємозалежних задач, описаною спрямованим ациклічним графом.

Позитивним моментом використання системи HTCondor є те, що вона поєднує в собі функції системи управління ресурсами і системи планування завдань, маючи гнучку систему налаштування бізнес-правил планування.

Недоліком такого рішення є підвищені вимоги до кваліфікації адміністратора: потрібно вручну описувати стратегії планування на всіх рівнях функціонування системи, оскільки ні адаптивності системи до змінним умов. Крім цього відсутність інтелектуальних засобів оцінки невизначеності вхідних даних знижує практичну цінність даної програмної платформи.

Для HTCondor характерною перевагою є підтримка високопродуктивних пакетних обчислень і використання простою робочих станцій, однак ефективність такого підходу значною мірою залежить від коректності опису політик, класів задач та очікуваної поведінки середовища виконання [24, 27, 33].

2. Система DIET.

DIET є проміжним програмним забезпеченням, яке представляє собою рівень між операційною системою та прикладними додатками, і призначений для організації ґрид -обчислень на основі різних типів розподілених ресурсів (робочих станцій, кластерів, ґрид -систем, хмарних систем).

Ієрархія компонентів системи DIET представлена на рис. 1.1.

Програмна архітектура DIET складається з наступних типів компонентів [11, 24, 25, 26, 32]:

1. Клієнт (client) - додаток, яке використовує DIET для рішення задач.
2. Керуючий агент (master agent, MA) - отримує обчислювальний запит від клієнта, потім збирає інформацію про доступних ресурсах із серверів,

обирає найбільш підходящий сервер і відправляє посилання на обраний сервер клієнту.

3. Місцевий агент (local agent , LA) - передає запити і інформацію між керуючими агентами і серверами, виконуючи часткове планування в рамках піддерева серверів.

4. Серверна служба (server daemon , SeD) - служба, яка виконується на кінцевих обчислювальних вузлах і служить для надання інформації про можливі типи розв'язуваних завдань, локально доступні дані та статичної та динамічної інформації про кількість та завантаження ресурсів сов.

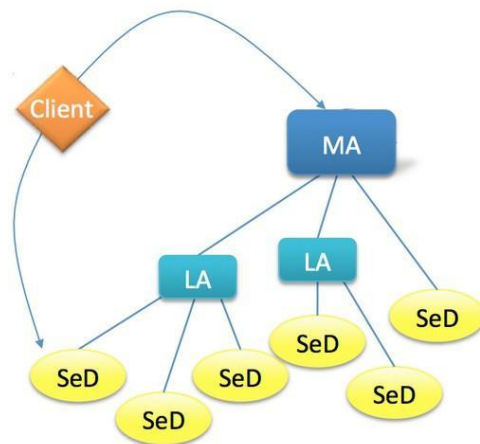


Рис. 1.1. Ієрархія компонентів системи DIET

Підсистема планування в DIET визначає план виконання завдань розподіленим способом: для кожної задачі, що надійшла на обробку, серверні служби надають оцінку продуктивності – набір даних, що відображають властивості ресурсів кожного сервера на момент запиту.

Система DIET надає просту схему автоматичного вибору ресурсів на основі моніторингу стану ресурсів (вузлів системи) та не вимагає завдання інформації про ресурсні вимоги завдань. Це є безперечним гідністю системи DIET.

Крім того, система DIET дозволяє розширювати підсистему планування за допомогою плагінів. Дана функціональність дає можливість розширити

безліч метрик, які використовуються для оцінки продуктивності, метриками прикладної області і задати критерій, відповідно до якого ці дані використовуються на етапі сортування обчислювальних вузлів кандидатів на виконання завдання, що надійшло. В результаті, при розробці плагіна, проблема планування перекладається на розробників кінцевих прикладних додатків.

З недоліків даного рішення варто відзначити необхідність програмної реалізації модуля планування для кожного прикладного завдання, що пов'язано зі суттєвими витратами часу та вимогами до високої кваліфікації персоналу підтримки. Крім того, зневага факторами невизначеності вхідних даних та гетерогенним характером процесів обробки значно знижують ефективність використання цього засобу.

Це підтверджує загальне обмеження проміжного програмного забезпечення для ґрид-обчислень: навіть за наявності моніторингу ресурсів і розширюваної архітектури планувальника залишається проблема адаптації до прикладних метрик, змінної продуктивності вузлів і невизначеності вхідних даних [1–5, 24, 33].

3. Системи управління ресурсами Slurm та Torque

Slurm є відкритою системою управління навантаженням, призначеною для кластерів робочих станції на базі ОС Linux. Надає механізми ексклюзивного і спільного доступу до ресурсів, а також механізми запуску і моніторингу завдань [28, 29, 46, 47].

Стандартна конфігурація системи Slurm передбачає обробку завдань відповідно до пріоритету FIFO, але є можливість додавання власного методу планування через механізм програмних розширень (плагінів).

У випадку Slurm важливою перевагою є масштабованість і модульність менеджера робіт, однак практичне досягнення високої якості планування потребує налаштування політик, оцінювання поведінки черг, моделювання роботи планувальника та врахування особливостей конкретного НРС-навантаження [28, 29, 46, 47].

Аналогічно системі Slurm, система Torque використовується виключно для управління ресурсами, обробляючи запити від планувальника і відаючи низькорівневими аспектами запуску, зупинки і моніторингу завдань. Стандартний планувальник pbs_sched, який поширюється разом із системою Torque, має лише базові можливості та забезпечує неефективне використання ресурсів кластера [2, 11, 13].

Оскільки вибір планувальника в кластерних системах істотно впливає на доступність ресурсів, їхнє завантаження та ефективне використання, розробники системи Torque рекомендують використовувати сторонні системи планування Moab і Maui.

4. Планувальник Moab.

Планувальник Moab має на меті забезпечення рівноправного та ефективного використання ресурсів кластера, визначаючи, де і за яких умов виконувати завдання.

Планувальник Moab підтримує інтеграцію із системами управління ресурсами SLURM і TORQUE [11].

Вузол-обробник у більшості випадків є стандартною робочою станцією, але підтримується також і можливість роботи з віртуальними машинами, наданими гіпервізором.

Планувальник Moab функціонує як у режимі опитування системи управління ресурсами, так і в режимі подій. Як події, що активують наступний цикл планування можуть виступати:

1. Надходження нового завдання обробки даних;
2. Завершення виконання раніше запущеного завдання;
3. Додавання нового вузла-обробника;
4. Внесення змін до політики (налаштування) системи керування ресурсами.

На кожній ітерації планування планувальник запитує систему управління ресурсами актуальну інформацію про стан ресурсів, робоче навантаження і конфігурації політик планування.

Потім Moab виконує наступні дії:

1. Отримує перелік завдань, для яких умови виконання задоволені.

Впорядковує завдання в відповідно до їхнього відносного пріоритету в списку. Для кожного завдання пріоритет визначається відповідно до таких атрибутів, як власник завдання, її розмір, тривалість простою даної задачі в черзі та ін. Як правило, для пріоритезації має значення група різнорідних та незалежних цілей, таких як максимізація завантаження ресурсів, особливий пріоритет користувачів в окремих проектах, виняток надмірного простою задач. Підхід, використовуваний в Moab має на увазі призначення ваг для безлічі незалежних цілей і отримання сумарного значення пріоритету для кожного завдання.

2. Визначає доступні ресурси. Для кожного завдання Moab намагається знайти вузол-обробник, який відповідає вимогам до ресурсів, зазначених в описі задачі. Вузол-обробник вважається придатним лише в том випадку, якщо задоволені ВСІ ресурсні вимоги, вказані в описі завдання.

3. Виділяє ресурси для виконання завдання. Якщо відповідні ресурси є в пулі ресурсів, планувальник застосовує метод вибору вузла (node allocation policy) для вибору найбільш відповідного безлічі ресурсів.

4. Запускає завдання. Після того, як вузол-обробник обраний, планувальник зв'язується із системою управління ресурсами та вказує де і як запускати завдання обробки даних.

Процес виділення ресурсів (вузла-оброблювача) для виконання завдання, як вибір найбільш підходящих ресурсів зі списку доступних, особливо актуальний в середовищах, що характеризуються наступними властивостями [6, 7, 11, 13]:

1. Гетерогенність ресурсів (якісні і кількісні відмінності між вузлами).
2. Наявність поділюваних вузлів (такий вузол може бути використаний для одночасного виконання більше чим для одного завдання).
3. При використанні резервування ресурсів для надання гарантій якості обслуговування (Quality of Service, QoS).

4. Неоднорідний топології мережі (швидкість передачі даних між різними сегментами може суттєво відрізнятись).

Основними цілями методу вибору вузла є підвищення продуктивності при виконанні цього завдання (локальна оптимізація) і надання максимальної свободи для планувальника при плануванні наступних завдань (Глобальна оптимізація).

Метод вибору вузла дозволяє ставити критерії вибору: швидкість вузла, тип резервування (розділений або ексклюзивний доступ) і кількість надлишкових ресурсів і ін. Крім того, можливо налаштувати політики вибору вузлів як на глобальному (загальносистемному) рівні, так і стосовно окремих завданням.

Таким чином, в планувальнику Moab проводиться чітке поділ між методами планування стосовно до завданням і ресурсів:

1. Метод вибору завдання (job selection policy), на основі схеми пріоритетів та інформації про виконання попередніх умов завдання, послідовно витягує з черги завдання на обробку.

2. Метод вибору вузла-обробника (node allocation policy), на основі ресурсних вимог задачі та інформації про завантаження вузлів, для кожної окремої завдання підбирає найбільш підходящий вузол. Пріоритет кожного вузла стосовно завдання, що надійшло, оцінюється за різними зваженим параметрів.

Перевагою підходу, що використовується в планувальнику Moab, можна вважати гнучкість налаштування процесу планування за рахунок використання модульного підходу до рішення проблем планування. Необхідність ручного завдання ресурсних вимог для кожного завдання варто віднести до недоліків системи.

Варто також наголосити, що планування здійснює послідовно для найбільш пріоритетних завдань, що дозволяє отримати локальний оптимум, але може привести до неефективного планування в загалом.

У протигага такому підходу існують методи, які здійснюють планування для групи завдань, що дозволяє оцінювати спільну ефективність запуску і знаходити глобальний оптимум.

Однак, слід зазначити, що планування для груп завдань варто застосовувати лише тоді, коли існує відповідне рівномірне навантаження. Для інтерактивних проблем доцільно виконувати планування стосовно до окремих завданням.

5. Планувальник Maui.

Розробники системи планування Maui виділяють три основні області, для яких здійснюється прийняття рішень з використанням політик [11, 13]:

1. Управління трафіком. Необхідно не виключати ситуації та змагання задачі за одні і ті ж ресурси, оскільки це, як правило, знижує загальну продуктивність кластера, збільшує час виконання таких завдань, і може наводити до відмов.

2. Політики цільового використання. Кластери та інші високопродуктивні платформи створюються до виконання певних прикладних завдань. З метою ефективності планування планувальник повинен дозволяти враховувати бізнес-правила щодо використання системи в процесі планування.

3. Будь-які оптимізації, які можна застосувати у процесі планування для підвищення продуктивності в обробці задач.

Maui відстежує використання класів як споживаного ресурсу, що дає можливість проектувальникам системи створювати політики використання в залежності від типів задач.

Приклад політики виконання завдань: - не більше 8 завдань будь-якого типу можна запускати одночасно - не більше 4 завдань від групи адміністративних користувачів можуть одночасно виконуватися в системі.

Однією з переваг системи для кінцевого користувача є підтримка резервування ресурсів з вказівкою інтервалів часу і списку доступу, що особливо важливо для надання гарантій якості обслуговування.

Головні недоліки використовуваного в планувальнику Maui підходу до планування:

1. Необхідно знати ресурсні вимоги завдання апіорі, що не завжди можливо.
2. Як правило, потрібно поставити умови вибору вузла-обробника.
3. Завдання необхідно заздалегідь розбити на класи в залежності від значень їх атрибутів, параметрів контролю доступу і ін.
4. На основі заданих класів необхідно заздалегідь визначити політики вибору черги запуску.
5. Відсутність інтелектуальних засобів обліку невизначеності вхідних даних.

Подібні обмеження характерні для багатьох політико-орієнтованих планувальників: вони ефективні за наявності повних і коректних описів задач, але втрачають точність у разі неповних оцінок тривалості, змінної пропускну здатності вузлів або стохастичного надходження потоків робіт [33, 34, 51, 52].

У результаті виконаного порівняльного аналізу систематизовано обмеження відомих планувальників завдань і систем управління ресурсами. Встановлено, що ключовими невирішеними проблемами залишаються залежність від апіорно заданих ресурсних вимог задач, недостатнє врахування невизначеності метаданих, слабка адаптивність до зміни стану середовища та відокремлений розгляд вибору задачі й вибору обчислювального ресурсу. Саме ці обмеження визначають необхідність переходу від традиційних евристичних схем до інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах.

Отримані висновки узгоджуються з сучасними тенденціями розвитку інтелектуального планування: для підвищення адаптивності розподілених систем застосовуються методи навчання з підкріпленням, графові нейромережеві моделі, механізми уваги, оптимізація параметрів та стохастичні графові моделі, що дають змогу враховувати структуру задач і поточний стан середовища [31–45, 47–56].

Висновки до розділу 1

У першому розділі дисертаційної роботи проведено аналіз теоретичних і прикладних основ планування задач у розподілених інформаційних системах з урахуванням гетерогенності обчислювального середовища, динамічності навантаження та невизначеності метаданих. Розглянуто вимоги до процесу планування задач, шляхи підвищення продуктивності розподілених інформаційних систем, а також функціональні можливості й обмеження відомих планувальників завдань і систем управління ресурсами.

У результаті аналізу встановлено, що ефективність функціонування розподілених інформаційних систем значною мірою визначається не лише продуктивністю окремих обчислювальних вузлів, а й якістю прийняття рішень щодо розподілу задач між ресурсами. Ключовими вимогами до таких систем є масштабованість, надійність, ефективне використання ресурсів, безпека, сумісність, підтримка динамічної зміни навантаження та здатність працювати в умовах неповної або неточної інформації про параметри задач і ресурсів. Саме тому задача планування в сучасних гетерогенних середовищах не може розглядатися лише як статичне призначення задач на ресурси, а потребує адаптивного підходу, орієнтованого на поточний і прогнозований стан системи.

Показано, що традиційні шляхи підвищення продуктивності, пов'язані переважно з нарощуванням апаратних ресурсів або застосуванням паралельної обробки, мають обмежену ефективність без належного механізму планування. Застосування паралельних, конвеєрних і декомпозиційних методів обробки дозволяє підвищити пропускну здатність системи, однак практичний ефект таких рішень залежить від того, наскільки коректно враховано залежності між задачами, обмеження ресурсів, часові характеристики виконання та можливість зміни умов функціонування системи. Отже, підвищення продуктивності розподіленої інформаційної системи потребує поєднання засобів паралельної обробки з удосконаленими методами планування задач.

У процесі аналізу систем автоматизованого проєктування цифрових пристроїв, методів синтезу функціональних структур спеціалізованих процесорів і показників ефективності паралельної реалізації алгоритмів обґрунтовано необхідність використання кількісних критеріїв для оцінювання результатів планування. До таких критеріїв належать час виконання задач, період оновлення даних, пропускна здатність, завантаження ресурсів, складність реалізації та ефективність розпаралелювання. Встановлено, що ці показники можуть бути використані як основа для формування системи критеріїв оцінювання методів планування у гетерогенних розподілених інформаційних системах.

Порівняльний аналіз відомих планувальників завдань і систем управління ресурсами, зокрема HTCondor, DIET, Slurm, Torque, Moab і Maui, показав, що наявні програмні засоби забезпечують важливі функції керування чергами, призначення задач, моніторингу ресурсів, підтримки пріоритетів, резервування та запуску завдань у кластерних і ґрид-середовищах. Разом з тим встановлено низку суттєвих обмежень, які знижують ефективність їх застосування в умовах невизначеності метаданих. Значна частина таких систем потребує попереднього задання ресурсних вимог задач, ручного налаштування політик планування або заздалегідь визначеної класифікації задач. У разі відсутності достовірної інформації про ресурсні вимоги ці підходи втрачають точність, що призводить до неефективного використання обчислювальних ресурсів, збільшення часу виконання задач і зниження якості обслуговування.

Встановлено, що методи планування, які не потребують детальної інформації про ресурсні вимоги окремих задач, також мають обмеження, оскільки зазвичай не враховують спільні вимоги груп взаємопов'язаних задач, залежності між ними, комунікаційні витрати та загальний вплив локальних рішень на глобальну ефективність розкладу. Це особливо важливо для розподілених інформаційних систем, у яких задачі формують складні графи залежностей, а їх виконання супроводжується передаванням даних між різними

вузлами. У таких умовах локально оптимальний вибір окремого вузла або окремої задачі не гарантує оптимальності плану виконання всієї групи задач.

Узагальнення результатів аналізу дозволило визначити основну науково-прикладну суперечність: з одного боку, сучасні розподілені інформаційні системи потребують адаптивного та ефективного планування задач за умов гетерогенності ресурсів, динамічного навантаження й невизначеності метаданих; з іншого боку, більшість існуючих методів і програмних засобів планування орієнтовані на повну або достатньо точну інформацію про задачі та ресурси, а також не забезпечують належного врахування невизначеності, залежностей між задачами та прогнозованого стану середовища виконання.

Таким чином, за результатами першого розділу обґрунтовано необхідність розроблення нового методу планування задач у розподілених інформаційних системах, який має враховувати гетерогенність обчислювальних ресурсів, залежності між задачами, неповноту й неточність метаданих, динамічну зміну навантаження та потребу в адаптивному переплануванні. Отримані результати аналізу визначають теоретичні передумови для подальшої формалізації задачі планування, побудови математичних моделей розподілу задач і даних та розроблення інтелектуального методу планування, що становить зміст наступних розділів дисертаційної роботи.

З огляду на це запропонований у роботі напрям доцільно розглядати як поєднання класичних моделей робочих процесів та спрямованих ациклічних графів (workflow/DAG-моделей), стохастичного моделювання невизначеності та сучасних інтелектуальних компонентів прогнозування й адаптивного прийняття рішень; додатково такий підхід корелює з дослідженнями суміжних задач управління навантаженням, робастності, справедливості та гетерогенності даних у сучасних інформаційних системах.

Основні результати розділу опубліковано в працях автора: [13, 57–61].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 1

1. Murad S. A., Muzahid A. J. M., Azmi Z. R. M., Hoque M. I., Kowsher M. A. review on job scheduling technique in cloud computing and priority rule based intelligent framework. *Journal of King Saud University - Computer and Information Sciences*. 2022. Vol. 34, No. 6. P. 2309–2331. DOI: 10.1016/j.jksuci.2022.03.027.
2. Khallouli W., Huang J. Cluster resource scheduling in cloud computing: literature review and research challenges. *The Journal of Supercomputing*. 2022. Vol. 78. P. 6898–6943. DOI: 10.1007/s11227-021-04138-z.
3. Khaledian N., Völp M., Azizi S., Shirvani M. H. AI-based & heuristic workflow scheduling in cloud and fog computing: a systematic review. *Cluster Computing*. 2024. Vol. 27. P. 10265–10298. DOI: 10.1007/s10586-024-04442-2.
4. Devi N., Dalal S., Solanki K., Dalal S., Lilhore U. K., Simaiya S., Nuristani N. A systematic literature review for load balancing and task scheduling techniques in cloud computing. *Artificial Intelligence Review*. 2024. Vol. 57. Article 276. DOI: 10.1007/s10462-024-10925-w.
5. Bouabdallah R., Fakhfakh F. Workflow scheduling in cloud-fog computing environments: a systematic literature review. *Concurrency and Computation: Practice and Experience*. 2024. Vol. 36, No. 28. Article e8304. DOI: 10.1002/cpe.8304.
6. Miao Z., Shao C., Li H., Tang Z. Review of Task-Scheduling Methods for Heterogeneous Chips. *Electronics*. 2025. Vol. 14. Article 1191. DOI: 10.3390/electronics14061191.
7. Ahmad W., Gautam G., Alam B. et al. An analytical review and performance measures of state-of-art scheduling algorithms in heterogeneous computing environment. *Archives of Computational Methods in Engineering*. 2024. Vol. 31. P. 3091–3113. DOI: 10.1007/s11831-024-10069-8.
8. Zhang L., Ai M., Tan R., Man J., Deng X., Li K. Efficient Prediction of Makespan Matrix Workflow Scheduling Algorithm for Heterogeneous Cloud

Environments. *Journal of Grid Computing*. 2023. Vol. 21. Article 75. DOI: 10.1007/s10723-023-09711-9.

9. Pabitha P., Nivitha K., Gunavathi C., Panjavarnam B. A chameleon and remora search optimization algorithm for handling task scheduling uncertainty problem in cloud computing. *Sustainable Computing: Informatics and Systems*. 2024. Vol. 41. Article 100944. DOI: 10.1016/j.suscom.2023.100944.

10. Mahdizadeh M., Montazerolghaem A., Jamshidi K. Task scheduling and load balancing in SDN-based cloud computing: a review of relevant research. *Journal of Engineering Research*. 2024. Article in press. DOI: 10.1016/j.jer.2024.11.002.

11. Yenhalychev S. O., Leunenko O. V., Semenov S. G. Known planners assignments and systems management resources analysis. *Free and open source software: матеріали XIV Міжнародної науково-практичної конференції*, Харків, 14–16 лютого 2023 р. Харків: FOSS, 2023. С. 10–14.

12. Wen S., Han R., Qiu K., Ma X., Li Z., Deng H., Liu C. H. K8sSim: A Simulation Tool for Kubernetes Schedulers and Its Applications in Scheduling Algorithm Optimization. *Micromachines*. 2023. Vol. 14, No. 3. Article 651. DOI: 10.3390/mi14030651.

13. Yenhalychev S. O., Semenov S. G., Leunenko O. V. An overview of the popular free resources for task planning and system management. *Free and open source software: матеріали XV Міжнародної науково-практичної конференції* (м. Харків, 14–16 лютого 2024 р.). Харків: FOSS, 2024. С. 70–72.

14. Yenhalychev S., Leunenko O., Davydov V. Development an intelligent task scheduling method in heterogeneous distributed information systems. *Eastern-European Journal of Enterprise Technologies*. 2025. Vol. 3, No. 9 (135). P. 6–18. DOI: 10.15587/1729-4061.2025.329263.

15. Semenov S., Yenhalychev S., Pochebut M., Sitnikova O. Models of data processing and logical access segregation considering the heterogeneity of entities in information systems. *Innovative Technologies and Scientific Solutions for Industries*. 2024. No. 2 (28). P. 143–152. DOI: 10.30837/2522-9818.2024.28.143.

16. Yenhalychev S. O. Вибір показників ефективності паралельної реалізації алгоритмів. *Problems of scientific, technical and legal support for cybersecurity in the modern world*. Krakow: Wydawnictwo Naukowe UKEN, 2024. P. 174–179.
17. Song X., Liu L., Fu J., Zhang X., Feng J., Pei Q. A Reinforcement Learning-based DAG Tasks Scheduling in Edge-Cloud Collaboration Systems. *IEEE GLOBECOM* 2023. 2023. P. 1771–1776. DOI: 10.1109/GLOBECOM54140.2023.10436802.
18. Song W., Chen X., Li Q., Cao Z. Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics*. 2023. Vol. 19, No. 2. P. 1600–1610. DOI: 10.1109/TII.2022.3189725.
19. Semenov S., Zhang L., Cao W., Bulba S., Babenko V., Davydov V. Development of a fuzzy GERT-model for investigating common software vulnerabilities. *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 6, No. 2 (114). P. 6–18. DOI: 10.15587/1729-4061.2021.243715.
20. NoorianTalouki R., Hosseini Shirvani M., Motameni H. A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms. *Journal of King Saud University - Computer and Information Sciences*. 2022. Vol. 34, No. 8, Part A. P. 4902–4913. DOI: 10.1016/j.jksuci.2021.05.011.
21. Семенов С. Г., Єнгаличев С. О. Адаптивне планування задач у розподілених системах на основі алгоритму оптимізації політик із проксимальним оновленням. *Цифрова економіка: матеріали Міжнародної науково-практичної конференції* (м. Київ, 5–6 червня 2025 р.). Київ, 2025. С. 698–700.
22. Verma P., Maurya A. K., Yadav R. A survey on energy-efficient workflow scheduling algorithms in cloud computing. *Software: Practice and Experience*. 2023. Vol. 54. P. 637–682. DOI: 10.1002/spe.3292.
23. Li P., Wang H., Tian G., Fan Z. Towards Sustainable Cloud Computing: Load Balancing with Nature-Inspired Meta-Heuristic Algorithms. *Electronics*. 2024. Vol. 13, No. 13. Article 2578. DOI: 10.3390/electronics13132578.

24. Golmohammadi A., Kamel Tabbakh S. R., Ghaemi R. A review on workflow scheduling and resource allocation algorithms in distributed mobile clouds. *Transactions on Emerging Telecommunications Technologies*. 2023. Vol. 34, No. 8. Article e4811. DOI: 10.1002/ett.4811.
25. Kjorveziroski V., Filiposka S., Trajkovic V. IoT Serverless Computing at the Edge: A Systematic Mapping Review. *Computers*. 2021. Vol. 10, No. 10. Article 130. DOI: 10.3390/computers10100130.
26. Ashraf M., Shiraz M., Abbasi A., Alqahtani O., Badshah G., Lasisi A. Microservice Application Scheduling in Multi-Tiered Fog-Computing-Enabled IoT. *Sensors*. 2023. Vol. 23, No. 16. Article 7142. DOI: 10.3390/s23167142.
27. Erickson R. A., Fienen M. N., McCalla S. G., Weiser E. L., Bower M. L., Knudson J. M., Thain G. Wrangling distributed computing for high-throughput environmental science: An introduction to HTCondor. *PLOS Computational Biology*. 2018. Vol. 14, No. 10. Article e1006468. DOI: 10.1371/journal.pcbi.1006468.
28. Jette M. A., Wickberg T. Architecture of the Slurm Workload Manager. Job Scheduling Strategies for Parallel Processing. *Lecture Notes in Computer Science*. 2023. Vol. 14283. P. 3–23. DOI: 10.1007/978-3-031-43943-8_1.
29. Jokanovic A., D’Amico M., Corbalan J. Evaluating SLURM simulator with real-machine SLURM and vice versa. *IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems*. 2018. P. 72–82. DOI: 10.1109/PMBS.2018.8641556.
30. Goudarzi M., Palaniswami M., Buyya R. Scheduling IoT applications in edge and fog computing environments: a taxonomy and future directions. *ACM Computing Surveys*. 2022. Vol. 55, No. 7. Article 140. DOI: 10.1145/3544836.
31. Mouradian C., Naboulsi D., Yangui S., Glitho R. H., Morrow M. J., Polakos P. A comprehensive survey on fog computing: state-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*. 2018. Vol. 20, No. 1. P. 416–464. DOI: 10.1109/COMST.2017.2771153.

32. Vergara J., Arozarena P., García-Alonso J., Berrocal J., Murillo J. M. A Comprehensive Survey on Resource Allocation in Fog-Cloud Continuum. *Sensors*. 2023. Vol. 23, No. 9. Article 4413. DOI: 10.3390/s23094413.
33. Bader J., Meier S., Witt C., Kounev S., van Hoorn A. Lotaru: Locally predicting workflow task runtimes for resource management on heterogeneous infrastructures. *Future Generation Computer Systems*. 2024. Vol. 153. P. 53–67. DOI: 10.1016/j.future.2023.08.022.
34. Yang L., Xia Y., Ye L., Gao R., Zhan Y. A Fully Hybrid Algorithm for Deadline Constrained Workflow Scheduling in Clouds. *IEEE Transactions on Cloud Computing*. 2023. Vol. 11, No. 3. P. 3197–3210. DOI: 10.1109/TCC.2023.3269144.
35. Tong Z., Chen H., Deng X., Li K., Li K. A scheduling scheme in the cloud computing environment using deep Q-learning. *Information Sciences*. 2020. Vol. 512. P. 1170–1191. DOI: 10.1016/j.ins.2019.10.035.
36. Rjoub G., Bentahar J., Wahab O. A., Bataineh A. S. Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems. *Concurrency and Computation: Practice and Experience*. 2021. Vol. 33, No. 23. Article e5919. DOI: 10.1002/cpe.5919.
37. Mao H., Schwarzkopf M., Venkatakrisnan S. B., Meng Z., Alizadeh M. Learning scheduling algorithms for data processing clusters. *Proceedings of the ACM Special Interest Group on Data Communication*. 2019. P. 270–288. DOI: 10.1145/3341302.3342080.
38. Xing Y., Wang S. Work Scheduling in Cloud Network Based on Deep Q-LSTM. *Journal of Grid Computing*. 2024. Vol. 22. Article 34. DOI: 10.1007/s10723-024-09746-6.
39. Song X., Liu L., Fu J., Zhang X., Feng J., Pei Q. A Reinforcement Learning-based DAG Tasks Scheduling in Edge-Cloud Collaboration Systems. *IEEE GLOBECOM* 2023. 2023. P. 1771–1776. DOI: 10.1109/GLOBECOM54140.2023.10436802.

40. Song W., Chen X., Li Q., Cao Z. Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*. 2023. Vol. 19, No. 2. P. 1600–1610. DOI: 10.1109/TII.2022.3189725.
41. Veličković P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y. Graph Attention Networks. *International Conference on Learning Representations*. 2018. arXiv:1710.10903.
42. Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. Proximal Policy Optimization Algorithms. arXiv:1707.06347. 2017.
43. Shahriari B., Swersky K., Wang Z., Adams R. P., de Freitas N. Taking the human out of the loop: a review of Bayesian optimization. *Proceedings of the IEEE*. 2016. Vol. 104, No. 1. P. 148–175. DOI: 10.1109/JPROC.2015.2494218.
44. Frazier P. I. A tutorial on Bayesian optimization. arXiv:1807.02811. 2018.
45. Smit I. G., Zhou J., Reijnen R., Wu Y., Chen J., Zhang C., Bukhsh Z., Zhang Y., Nuijten W. Graph neural networks for job shop scheduling problems: a survey. *Computers & Operations Research*. 2025. Vol. 176. Article 106914. DOI: 10.1016/j.cor.2024.106914.
46. Мінухін С. В., Коптілов Н. С. Метод збільшення продуктивності Apache Spark на основі сегментування даних і налаштувань конфігураційних параметрів. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024. № 1 (27). С. 128–139. DOI: 10.30837/ITSSI.2024.27.128.
47. Herasymuk D., Poliakov A., Fedorchenko V. Detecting trade-offs between fairness, stability, and accuracy for responsible machine learning model selection. *Innovative Technologies and Scientific Solutions for Industries*. 2025. No. 1 (31). P. 5–19. DOI: 10.30837/2522-9818.2025.1.005.
48. Syvolovskyi I. M., Lysechko V. P., Zhuchenko O. S., Komar O. M., Pastushenko V. V. Analysis of methods for organizing distributed telecommunication systems using the paradigm of Edge Computing. *Control, Navigation and Communication Systems*. 2024. No. 1 (75). P. 206–211. DOI: 10.26906/SUNZ.2024.1.206.

49. Neroznak Y., Trotsko O., Fesokha V., Balan D., Bieliakov R. Construction of a simulation model for managing load on computing nodes in a server cluster based on the theory of fuzzy logic and Nash equilibrium. *Eastern-European Journal of Enterprise Technologies*. 2025. Vol. 5, No. 3 (137). P. 6–17. DOI: 10.15587/1729-4061.2025.340603.
50. Moskalenko V., Kharchenko V., Semenov S. Model and Method for Providing Resilience to Resource-Constrained AI-System. *Sensors*. 2024. Vol. 24, No. 18. Article 5951. DOI: 10.3390/s24185951.
51. Chuhai A., Yaskov G., Starkova O., Zhuravka A., Shcherbyna M. Intelligent adaptive packing system based on phi-functions and agent-controlled interactions. *Artificial Intelligence*. 2025. Vol. 30, No. 4. P. 117–122. DOI: 10.15407/jai2025.04.117.
52. Starkova O., Shapovalova O., Aleinikova A., Solodovnyk G., Liubynskyi P. Development of a system of indices for monitoring and assessing the sustainability of underground utilities. *Technology Audit and Production Reserves*. 2026. Vol. 1, No. 2 (87). P. 57–65. DOI: 10.15587/2706-5448.2026.350994.
53. Shkurko V., Poliakov A. Organization of software and neural network algorithms for machine analysis of textual data presented in natural language. *Innovative Technologies and Scientific Solutions for Industries*. 2025. No. 2 (32). P. 151–167. DOI: 10.30837/2522-9818.2025.2.151.
54. Chaudhary S., Ramjee R., Sivathanu M., Kwatra N., Viswanatha S. Balancing efficiency and fairness in heterogeneous GPU clusters for deep learning. *Proceedings of the Fifteenth European Conference on Computer Systems*. 2020. Article 1. DOI: 10.1145/3342195.3387555.
55. Gu J., Chowdhury M., Shin K. G., Zhu Y., Jeon M., Qian J., Liu H., Guo C. Tiresias: a GPU cluster manager for distributed deep learning. *16th USENIX Symposium on Networked Systems Design and Implementation*. 2019. P. 485–500.
56. Narayanan D., Santhanam K., Kazhamiaka F., Phanishayee A., Zaharia M. Heterogeneity-aware cluster scheduling policies for deep learning workloads. *14th*

USENIX Symposium on Operating Systems Design and Implementation. 2020. P. 481–498.

57. Леуненко О. В., Єнгаличев С. О. Моделювання та захист передачі даних ресурсоемних задач у гетерогенних хмарних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024) (м. Харків, 25 квітня 2024 р.) *Молодіжний економічний вісник*. 2024. № 1. С. 59–63. URL: <https://repository.hneu.edu.ua/handle/123456789/38209>.

58. Семенов С. Г., Єнгаличев С. О. Метод планування задач в розподілених інформаційних системах з врахуванням невизначеності метаданих. *Матеріали XII Міжнародної науково-технічної конференції «Проблеми інформатизації»* (м. Харків, 21–22 листопада 2024 р.). Харків: Харківський національний університет радіоелектроніки, 2024. С. 24. URL: <https://repository.hneu.edu.ua/handle/123456789/38235>.

59. Yenhalychev S. O., Leunenko O. V. Comparative analysis of open source infrastructure as code (IaC) tools for managing heterogeneous cloud resources. *Free and Open Source Software: матеріали XVI Міжнародної науково-практичної конференції* (м. Харків, 13–14 лютого 2025 р.). Харків: FOSS, 2025. С. 105–107. URL: <https://repository.hneu.edu.ua/handle/123456789/35624>.

60. Yenhalychev S. O., Leunenko O. V. Development of Methods for Dynamic Load Balancing During the Transfer of Large Volumes of Data: A Comparative Study of Apache Kafka and RabbitMQ. *Free and Open Source Software: матеріали XVI Міжнародної науково-практичної конференції* (м. Харків, 13–14 лютого 2025 р.). Харків: FOSS, 2025. С. 108–109. URL: <https://repository.hneu.edu.ua/handle/123456789/35624>.

61. Леуненко О. В., Єнгаличев С. О. Математичне моделювання передачі даних ресурсомістких задач у гетерогенних хмарних системах: показники ефективності та методологічні підходи. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025), (м. Харків, 16 жовтня 2025 р.). *Молодіжний економічний вісник*. 2025. № 3. С. 199–202. URL: <https://repository.hneu.edu.ua/handle/123456789/38211>.

РОЗДІЛ 2

МАТЕМАТИЧНІ МОДЕЛІ ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

2.1. Моделі опрацювання та логічного розмежування доступу до даних з урахуванням різнорідності сутностей в інформаційних системах

Постановка завдання.

У сучасному світі інформаційні системи стали невід'ємною частиною багатьох організацій та установ. Ефективне управління даними та забезпечення їхньої безпеки є ключовими завданнями для будь-якої інформаційної системи. Враховуючи різноманітність та складність сучасних систем, виникає необхідність у розробці та впровадженні надійних моделей обробки даних і логічного розмежування доступу до них. Особливу увагу слід приділяти багаторівневим моделям ЛРДРС (логічного розмежування доступу та розподілу сутностей), які дозволяють уникнути некоректних інформаційних потоків навіть у випадках, коли зломисники контролюють привілейовані облікові записи.

Розглянуті у розділі 1 САПР/САП і споріднені інженерні платформи наведені як приклад прикладного домену, в якому робочий процес природно реалізується у вигляді конвеєра взаємозалежних операцій (підготовка даних, синтез/моделювання, перевірки, генерація артефактів, передавання проміжних результатів). Універсальною для такого класу систем є задача планування взаємопов'язаних робіт разом із керуванням потоками даних між етапами в гетерогенному обчислювальному середовищі. Саме тому у розділі 2 вводиться формалізована модель процесу планування задач розповсюдження даних з урахуванням різнорідності сутностей, як узагальнення типового міжетапного обміну артефактами у конвеєрних системах, включно з САПР/САП.

Запропонована модель доцільна для інформаційних систем, які одночасно задовольняють такі умови:

- робочий процес може бути поданий як граф залежностей задач (DAG або DAG+GERT);
- задачі породжують або споживають об'єкти даних, передавання яких між етапами має ненульову вартість;
- середовище є гетерогенним обчислювальним середовищем (вузли різної продуктивності, різні профілі ресурсів, різні мережеві параметри);
- доступні метадані задач і об'єктів, але вони можуть бути неповними або неточними, що формує невизначеність метаданих;
- існують часові обмеження або вимоги до передбачуваності виконання;
- система має керований контур прийняття рішень щодо призначення призначення, управління ресурсами й управління навантаженням (централізований або логічно централізований планувальник, або планування на рівні домену).

Приклади потреб і потенційного впровадження.

Приклад 1 (конвеєр інженерних артефактів, зокрема САПР/САП/СІ). Етапи конвеєра породжують проміжні артефакти (моделі, журнали перевірок, звіти, бібліотеки), які мають бути передані наступним етапам. Метою цього етапу є мінімізація *makespan*, підвищення передбачуваності управління навантаженням та зменшення затримки передач, особливо за мережевих обмежень і різномірності вузлів.

Приклад 2 (хмарні та кластерні платформи обробки даних). Пайплайни ETL (*extract, transform, load*) або потокові обчислення у середовищах хмарних та граничних обчислень: задачі трансформацій і агрегацій залежать від готовності частин даних. Ця модель використовується для планування запусків і розміщення з урахуванням вартості, неоднорідності ресурсів, вимог до масштабованості та роботи серверних кластерів.

Приклад 3 (міжсистемна інтеграція та обмеження доступу). Дані та задачі належать різним доменам/організаціям. У цьому прикладі необхідно

враховувати політики доступу (наприклад, RBAC (role-based access control)) та запобігати недопустимим потокам даних. Модель дозволяє формалізувати процес розповсюдження даних і відразу пов'язати його з обмеженнями на допустимі призначення, маршрути передач та вимоги робастності системи.

Розділ присвячений дослідженню переваг та особливостей використання багаторівневих моделей ЛРДРС для забезпечення інформаційної безпеки в складних розподілених системах. Основна увага приділяється інтеграції таких моделей з метою запобігання виникненню інформаційних потоків, що суперечать політикам безпеки компонентів системи. Також розглядаються умови коректного об'єднання різних інформаційних систем на основі моделей ЛРДРС та відповідні відносини довіри між суб'єктами цих систем.

Проведені дослідження показали значний інтерес з боку сучасних авторів щодо аналізу і синтезу моделей розмежування доступу до даних. Так, наприклад, в роботі [1] надано глибокий огляд багаторівневих моделей безпеки, зокрема, їх застосування у розподілених системах. Автори детально описують різні підходи до розмежування доступу та обробки даних, включаючи використання решіткових структур. Основним недоліком цієї роботи є те, що робота зосереджена переважно на теоретичних аспектах, з невеликим акцентом на практичне впровадження та реальні випадки використання.

Робота [2] надає вичерпний аналіз механізмів контролю доступу, адаптованих до різномірних інформаційних систем. Особливу увагу приділено гнучкості та масштабованості запропонованих рішень. Недоліком є складність реалізації запропонованих механізмів у великих системах через високі вимоги до ресурсів.

Робота [3] досліджує питання інтеграції політик безпеки в складних системах, пропонуючи методи узгодження різних політик між собою. Основний недолік полягає в недостатній увазі до динамічних змін у системах, що може призвести до проблем із підтриманням актуальності політик безпеки.

Також [4] підкреслюється важливість довіри між суб'єктами системи для забезпечення безпеки даних, описуючи кілька моделей, заснованих на довірі. Недоліком є складність формалізації та оцінки рівня довіри, що може впливати на точність і надійність моделей.

В роботі [5] зосереджується на ролеорієнтованому контролі доступу (RBAC) і його адаптації до різномірних інформаційних систем, підкреслюючи його гнучкість та ефективність. Нажаль потенційні проблеми з масштабованістю RBAC у дуже великих системах можуть стати проблемою при практичній реалізації запропонованих рішень.

Крім того робота [6] досліджує ієрархічні моделі безпеки, зосереджуючись на їхньому застосуванні в хмарних середовищах. Висвітлюються переваги багаторівневої безпеки для захисту даних у хмарі. Недоліком цієї роботи є обмежена увага до питань забезпечення безпеки при інтеграції з традиційними системами.

Автори роботи [7] пропонують динамічні моделі контролю доступу для систем Інтернету речей (IoT), підкреслюючи їхню здатність адаптуватися до змін у реальному часі. Нажаль висока складність та низька точність результатів в умовах обмежених ресурсів IoT пристроїв стає на заваді практичній реалізації цієї роботи.

Аналізуючи [8] та досліджуючи методи інтеграції політик безпеки в розподілених мережах, пропонуючи інструменти для узгодження різномірних політик. За мету було поставлено сформулювати однозначне рішення для різномірних систем та об'єднати їх в єдиній моделі. Недоліком є недостатня увага до питання масштабованості запропонованих рішень яка лежить в практичній площині реалізації.

У роботі [9] пропонується для розгляду модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між організаціями. Відмінною особливістю моделі є те, що вона враховує неоднорідність сутностей шляхом включення додаткових блоків для їх аналізу та адаптації до існуючих можливостей процесорних та інших ресурсів. У ході

моделювання було надано класифікацію поняттю «сутності», розроблено та досліджено блок-схему потоку сутностей для систем планування. Також було розроблено узагальнену модель планування завдань та сутностей із залежностями. Моделювання проводилося з використанням GERT-мережевої технології як різновиду графової моделі процесу. У результаті отримано GERT-мережі процесу планування завдань розподілу для окремого n -го набору типів даних. Перевагою цієї моделі є можливість її впровадження в різні практичні застосунки, пов'язані з управлінням ресурсами. Водночас необхідно відмітити потребу вдосконалення та розширення переліку зовнішніх факторів, що впливають на достовірність і точність результатів моделювання.

У роботі [10] проілюстровано результати дослідження політико-орієнтованих моделей контролю доступу, підкреслюючи їхню ефективність у розподілених середовищах. Недоліком є висока складність налаштування та підтримки політик безпеки в динамічно змінних середовищах.

Аналогічні недоліки присутні і у монографії [11] де розглянуті різні моделі розподілу доступу в комп'ютеризованих системах критичного застосування.

В статті [12] наведено результати розробки математичної моделі задачі для методу на основі леми Карліна а також розробки математичної моделі задачі для методу на основі теореми Гермейєра. Нажаль проблеми, що пов'язані з необхідністю врахування різномірності сутностей та багаторівневу побудову інформаційних структур в цій на цьому етапі не розглянуто.

У статті [12] наведено результати розроблення математичної моделі задачі для методу на основі леми Карліна, а також математичної моделі задачі для методу на основі теореми Гермейєра. Водночас проблеми, пов'язані з необхідністю врахування різномірності сутностей та багаторівневої побудови інформаційних структур, у цій роботі на цьому етапі не розглянуто.

У роботі [13] досліджено динаміку розподілу ймовірностей станів напівмарковської системи та запропоновано технологію визначення аналітичних співвідношень для формалізації ймовірностей станів такої системи.

Однак у роботі недостатньо розкрито різноманіття входних зовнішніх факторів, що є суттєвим для задач моделювання в умовах невизначеності.

Різнорідність зовнішніх факторів та різноманітність сутностей враховано при моделюванні в статті [14]. Однак питання складності реалізації, масштабованості та робастності запропонованого підходу в цій роботі розкрито недостатньо.

У роботі [15] подано приклад математичного моделювання прийняття рішень щодо вибору контрзаходів ризикам інноваційного проєкту. Імовірнісний характер такого підходу може бути використаний для уточнення окремих елементів процесу логічного розмежування доступу до даних з урахуванням різноманітності сутностей та невизначеності вихідних параметрів.

З аналізу літератури видно, що існує багато підходів до розробки та впровадження моделей обробки даних і контролю доступу в різноманітних інформаційних системах. Кожен із розглянутих підходів має свої переваги та недоліки, що потребує ретельного вибору моделі залежно від конкретних вимог та умов експлуатації системи. Впровадження багаторівневих моделей ЛРДРС може значно підвищити рівень точності та достовірності отриманих результатів. Проте слід враховувати складність їхньої реалізації та необхідність постійного моніторингу та адаптації до змін у системі та оптимізації параметрів управління ресурсами.

Таким чином, синтез та інтеграція ролевих моделей опрацювання та логічного розмежування доступу до даних з урахуванням різноманітності сутностей в інформаційних системах для підвищення точності результатів моделювання та врахування різноманітних факторів є важливим науковим завданням.

Об'єднання Простіших Рольових Моделей

Для аналізу механізмів об'єднання рольових моделей логічного розмежування доступу різноманітності сутностей (ЛРДРС) необхідно ввести допоміжне поняття, яке називається коректною множиною привілеїв.

Нехай в інформаційній системі A задані такі множини:

- P , що називається множиною привілеїв;
- R , що називається множиною ролей;
- U , що називається множиною користувачів;
- S , що називається множиною суб'єктів,

а також такі відношення:

- $RP \subseteq R \times P$;
- $RU \subseteq R \times U$;
- $RS \subseteq R \times S$,

і відображення $u: S \rightarrow U$.

Для будь-якого $s \in S$ і $r \in R$ виконується така умова: $(r, s) \in RS$ означає, що $(r, u(s)) \in RU$.

У цьому випадку вважається, що в системі A задана ролева модель ЛРДРС, яку також будемо називати моделлю RBAC.

У представлений моделі для будь-якого користувача u , суб'єкта s та ролі r позначимо:

$$R(u) = \{r \in R : (r, u) \in RU\}, \quad (2.1)$$

$$R(s) = \{r \in R : (r, s) \in RS\}, \quad (2.2)$$

$$P(r) = \{p \in P : (r, p) \in RP\}. \quad (2.3)$$

Очевидно, що $R(s) \subseteq R(u(s))$. Якщо привілей $p \subseteq P(r)$, ми вважаємо, що роль r має привілей p .

Нехай в інформаційній системі A використовується політика безпеки, заснована на моделі ЛРДРС RBAC з множиною привілеїв P . Множина привілеїв $P' \subseteq P$ називається коректною, якщо існують такі ролі $r_1, \dots, r_n$, що $P' = P(r_1) \cup \dots \cup P(r_n)$.

Для подальших міркувань необхідно мати властивість коректних множин привілеїв, яке формулюється наступним чином.

Будь-яке об'єднання коректних підмножин привілеїв є коректною підмножиною привілеїв.

Доказ цього твердження полягає у наступному. Нехай $P_1, \dots, P_n$ – коректні множини привілеїв, і $P' = P_1 \cup \dots \cup P_n$ – їх об'єднання. Нехай $R_1, \dots, R_n$ – такі множини ролей, що для будь-якого j $P_j = \bigcup_{r \in R_j} P(r)$.

Нехай $R' = R_1 \cup \dots \cup R_n$. Тоді $P' = \bigcup_{r \in R'} P(r)$, що означає, що P' є коректною множиною привілеїв.

Використовуючи введені вище допоміжні поняття, сформульовано і доведено наступна необхідна та достатня умова, за якої можливе об'єднання рольових моделей ЛРДРС для об'єктів інформаційних систем.

Нехай інформаційні підсистеми A і B мають політики безпеки, засновані на моделі ЛРДРС RBAC. Нехай система C складається з об'єктів підсистем A і B та також має політику безпеки, засновану на моделі ЛРДРС RBAC. Нехай множина привілеїв системи C має вигляд $P(C) = P(A) \cup P(B)$, і обмеження моделі ЛРДРС системи C на кожну з підсистем збігається з локальною моделлю ЛРД цієї підсистеми. У цьому випадку об'єднання моделей ЛРДРС підсистем A і B в моделі ЛРДРС системи C може бути виражено за допомогою відносин довіри тоді і тільки тоді, коли для будь-якої ролі r_c системи C набір її привілеїв $P(r_c)$ має вигляд $P(r) = P_A(r) \cup P_B(r)$, де множини привілеїв $P_A(r) = P(r) \cap P(A)$ та $P_B(r) = P(r) \cap P(B)$ коректні з позицій локальних моделей ЛРДРС систем A і B .

Доказ цього твердження у випадку «необхідності» виглядає наступним чином. Нехай $T_{A,B}$ і $T_{B,A}$ – відносини довіри між системами A і B . Нехай S_A – довільний суб'єкт системи A , який має єдину роль $r_c(S_A)$. Нехай $P_c(S_A)$ – множина привілеїв цього суб'єкта (а отже, і зазначеної ролі) в системі C . Нехай $P_A(S_A) = P_c(S_A) \cap P(A)$ і $P_B(S_A) = P_c(S_A) \cap P(B)$ – обмеження множини привілеїв суб'єкта S_A на кожну з підсистем. Згідно з умовою теореми, множина привілеїв $P_A(S_A)$ є коректною, оскільки вона є множиною привілеїв суб'єкта S_A в системі A . Нехай $S_{B,1}, \dots, S_{B,n}$ – суб'єкти системи B , які довіряють S_A . Нехай $r_{B,1}, \dots, r_{B,n}$ – всі ролі всіх суб'єктів $S_{B,j}$, пронумеровані у довільному порядку. Тоді множина $P_B(S_A)$ має вигляд $P_B(S_A) = \bigcup_j P(r_{B,j})$ і є коректним набором привілеїв системи B .

Доказ достатності можна сформулювати наступним чином. З того факту, що множини привілеїв кожної ролі системи C в кожній з підсистем системи C є

коректними, впливає, що множини привілеїв кожного суб'єкта системи C в кожній з підсистем також є коректними. Візьмемо в системі A довільного суб'єкта S_A . Нехай $r_1, \dots, r_n$ – такі ролі в системі B , що $P_B(S_A) = \bigcup_j P(r_j)$. Множина $P_B(S_A) = P_C(S_A) \cap P(B)$ є множиною привілеїв системи B , якими володіє суб'єкт S_A . За умовою теореми, такі ролі існують. Додаймо в систему B суб'єкта S_B , який має ролі $r_1, \dots, r_n$, і жодних інших. Нехай при цьому S_B довіряє S_A . Таким чином будується відношення довіри $T_{A,B}$. Аналогічним чином будується відношення довіри $T_{B,A}$.

Таким чином, отримано необхідну і достатню умову, яка гарантує можливість вираження рольової моделі ЛРДРС розподіленої інформаційної системи через моделі ЛРДРС її компонент за допомогою відносин довіри.

Синтез ієрархічних рольових моделей

Слід зазначити, що нажаль при ієрархічній побудові інформаційної системи та в умовах різнорідності оброблюваних сутностей вказані вище тези та пропозиції мають обмеження щодо достатності. Доказ цього обмеження полягає у наступному. Нехай в системі B є три привілеї P_1, P_2 і P_3 та три ролі r_1, r_2 і r_3 , при цьому кожна з ролей має відповідний привілей і не має двох інших. Нехай при цьому $r_3 < r_2$, а роль r_1 не може бути порівняна з двома іншими. Нехай, згідно з об'єднаною моделлю ЛРДРС систем A і B , суб'єкт a системи A має привілеї (P_1, P_2) , де P_1 – деякий привілей системи A . У цьому випадку множина $P_C(S_A) \cap P(B)$, яка складається з одного привілею P_2 , є коректною. Однак, подібну модель ЛРДРС не можна отримати з моделей ЛРДРС систем A і B за допомогою відношень довіри. Фактично, щоб суб'єкт a міг мати привілей P_2 , в системі B повинен існувати суб'єкт b , який довіряє йому і має привілей P_2 , а отже і ролі r_2 . Проте в цьому випадку суб'єкт b також має привілей P_3 , якого суб'єкт a не має. Отже, вище вказана теза про достатність є невірною для ієрархічних рольових моделей ЛРДРС.

З урахуванням зазначеного необхідно дослідити умови, які гарантують можливість адаптації та інтеграції ієрархічних рольових моделей під умови різнорідності оброблюваних сутностей. Як і у випадку з моделлю ЛРДРС

РВАС, для подальшої роботи необхідно визначити поняття коректної множини привілеїв.

Множина привілеїв $P' \subseteq P$ називається коректною в ієрархічному розумінні, якщо існують такі ролі $r_1, \dots, r_n$, що $P = \bigcup_{r \in R_j} \bigcup_{r' < r_j} P(r)$.

Будь-яке об'єднання коректних в ієрархічному розумінні множин привілеїв є коректним в ієрархічному розумінні множин привілеїв.

Можемо доказати це визначення. Нехай $P_1, \dots, P_n$ – коректні в ієрархічному розумінні множини привілеїв, $P' = P_1 \cup \dots \cup P_n$ – їх об'єднання. Нехай $R_1, \dots, R_n$ – такі множини ролей, що для будь-якого j $P_j = \bigcup_{r \in R_j} \bigcup_{r' < r} P(r')$, що означає, що P' є коректною в ієрархічному розумінні множиною привілеїв.

Очевидно, що будь-яка ієрархічно коректна множина привілеїв є коректною і у відношенні до простої рольової моделі ЛРДРС. Проте, обернене твердження є невірним.

З урахуванням введених вище поміжних понять, сформульовано і доведено наступні необхідна і достатня умови, при яких можливо об'єднати ієрархічні рольові моделі ЛРДРС для об'єктів інформаційних систем.

Нехай інформаційні системи A і B мають політики безпеки, засновані на ієрархічній рольовій моделі ЛРДРС. Нехай система C складається з об'єктів систем A і B , а також має політику безпеки, засновану на ієрархічній рольовій моделі ЛРДРС. Нехай множина привілеїв системи C має вигляд $P(C) = P(A) \cup P(B)$, і обмеження моделі ЛРДРС системи C на кожну з підсистем збігається з локальною моделлю ЛРДРС цієї підсистеми. У такому випадку об'єднання моделей ЛРДРС систем A і B у моделі ЛРДРС системи C може бути виражено за допомогою відношень довіри тоді і тільки тоді, коли для будь-якої ролі r_C системи C набір її привілеїв $P(r_C)$ має вигляд $P(r_C) = P_A(r_C) \cup P_B(r_C)$, де множини привілеїв $P_A(r_C) = P(r_C) \cap P(A)$ і $P_B(r_C) = P(r_C) \cap P(B)$ є коректними в ієрархічному розумінні відносно локальних моделей ЛРДРС систем A і B .

Таким чином, отримано необхідну і достатню умови, які гарантують можливість вираження ієрархічної рольової моделі ЛРДРС розподіленої інформаційної системи через моделі ЛРДРС її компонент за допомогою

відносин довіри. Дана умова аналогічна відповідній умові для моделей ЛРДРС RBAC і також діє в більшості практично використовуваних інформаційних систем.

Моделі примусової типізації на основі відносин довіри

Ще одним широко використовуваним типом моделей логічного розмежування доступу до даних є моделі примусової типізації на основі відносин довіри.

Моделі примусової типізації (Mandatory Typing Models) використовуються для управління доступом до інформаційних систем шляхом встановлення чітких правил і обмежень, які залежать від типів об'єктів та суб'єктів. У цих моделях всі дії і доступи контролюються на основі попередньо визначених типів, що зменшує ризик несанкціонованого доступу і підвищує безпеку системи. Проведені дослідження показали, що основними компонентами і принципами роботи таких моделей є наступні.

1. Типи об'єктів і суб'єктів. Всі об'єкти і суб'єкти системи класифікуються за типами. Типи визначають рівень секретності, чутливість або інші властивості, важливі для безпеки.

2. Примусовий контроль доступу. Відносини між типами визначають, які суб'єкти можуть взаємодіяти з певними об'єктами. Наприклад, суб'єкти з певним типом доступу можуть читати або записувати тільки ті об'єкти, які відповідають їх типу або нижчому рівню секретності.

3. Політики безпеки. Політики безпеки встановлюють правила, за якими здійснюється доступ до об'єктів. Вони можуть включати такі аспекти, як дозвіл/заборона доступу, обов'язковий аудит дій та інші заходи безпеки.

4. Інтеграція політик. Моделі примусової типізації можуть інтегруватися з іншими моделями безпеки, створюючи складніші системи управління доступом. При цьому важливо забезпечити коректну інтеграцію, щоб уникнути порушень політики безпеки.

5. Детермінованість. Моделі примусової типізації мають бути детермінованими, тобто поведінка системи має бути передбачуваною і однозначною при заданих вхідних даних і правилах.

6. Захист від витоків інформації. Основною метою є запобігання несанкціонованим витокам інформації навіть у разі, якщо зловмисник отримує контроль над привілейованими обліковими записами.

Моделі примусової типізації є ефективним інструментом для забезпечення високого рівня безпеки в інформаційних системах, особливо в середовищах, де важливо запобігати несанкціонованим витокам інформації.

Для моделі ЛРДРС примусової типізації сформулюємо та доведемо критерій можливості об'єднання, аналогічний відповідному критерію можливості об'єднання рольових моделей ЛРДРС. Для формулювання зазначеного критерію потрібно використовувати поняття привілеїв в моделі ЛРДРС примусової типізації. Також необхідно ввести допоміжний об'єкт, який будемо називати коректною множиною привілеїв. При цьому множина привілеїв P називається коректною, якщо існують типи $t_1, \dots, t_n$, такі що всі привілеї кожного з типів t_j входять до множини P , і кожна привілея з P належить хоча б одному з типів t_j .

Змістовно це визначення означає, що множина привілеїв є коректною, якщо вона представляє собою множину привілеїв якоїсь множини суб'єктів. Отримано наступну необхідну і достатню умову інтеграції моделей ЛРДРС примусової типізації.

Нехай інформаційні підсистеми A та B мають політики безпеки, що базуються на моделі ЛРДРС примусової типізації. Нехай система C складається з об'єктів підсистем A та B і також використовує модель ЛРДРС примусової типізації. При цьому всі три системи мають однакові множини класів і доступів, причому клас кожного об'єкта в системі C такий же, як у відповідній підсистемі. Таке об'єднання моделей примусової типізації може бути виражено за допомогою відношень довіри тоді і тільки тоді, коли набір прав доступу

кожного суб'єкта до об'єктів кожної з підсистем представляє собою коректну підмножину привілеїв цієї підсистеми.

Доказом необхідності в межах цієї тези може бути наступне. Нехай $T(A,B)$ і $T(B,A)$ – відношення довіри між системами А та В. Нехай S_A – довільний суб'єкт системи А. Набір прав доступу суб'єкта S_A до об'єктів системи А відповідає множині привілеїв типу цього суб'єкта, і, отже, є коректним. Залишається довести твердження для доступу суб'єкта S_A до об'єктів системи В. Нехай $S_{B1}, \dots, S_{Bu}$ – суб'єкти системи В, які довіряють S_A , а $t_1, \dots, t_n$ – їхні типи. Тоді множина доступів суб'єкта S_A до об'єктів системи В відповідає об'єднаному набору привілеїв цих типів.

Доказ достатності виглядає наступним чином. Візьмемо в системі А довільний суб'єкт S_A . Нехай $t_1, \dots, t_n$ – такі типи в системі В, що множина доступів суб'єкта S_A до об'єктів системи В представляє собою об'єднання наборів привілеїв зазначених типів. За умовами теореми, такі типи існують. Суб'єкту S_A у цьому випадку повинні довіряти суб'єкти системи В, які мають типи $t_1, \dots, t_n$, і ніякі інші. Зауважимо, що всі типи, які мають будь-які привілеї, є доменами, тому для кожного з типів t_j в системі В існує принаймні один суб'єкт, який має цей тип. Таким чином, будується відношення довіри $T(A,B)$. Аналогічно будується відношення довіри $T(B,A)$.

Таким чином, отримано критерій, що визначає можливість об'єднання моделей ЛРДРС примусової типізації за допомогою відношень довіри, аналогічно до критеріїв, що визначають можливість об'єднання рольових моделей ЛРДРС. Ця умова дозволяє обґрунтувати коректність функціонування розглянутих в майбутньому механізмів керування доступом суб'єктів до віддалених об'єктів інформаційних систем, компоненти яких використовують програмні засоби для керування доступом, реалізуючи модель ЛРДРС примусової типізації.

Основні положення процесу об'єднання політик безпеки.

Як зазначено раніше, одна з важливих переваг використання багаторівневих моделей ЛРДРС у механізмах забезпечення безпеки інформаційних систем полягає в тому, що вони допомагають уникнути створення зловмисником інформаційних потоків, які суперечать встановленій політиці безпеки, навіть якщо він контролює привілейовані облікові записи. Для повноцінного використання цих переваг необхідно уникати створення інформаційних потоків, що порушують політики безпеки компонентів інформаційної системи при об'єднанні моделей ЛРДРС.

При об'єднанні багаторівневих моделей ЛРДРС для об'єктів інформаційної системи, розподілених у мережевому середовищі, потрібно приділяти особливу увагу заходам, що запобігають виникненню інформаційних потоків "знизу вгору", які використовують об'єкти інших компонентів. Враховуючи це, автор пропонує наступне визначення коректної інтеграції багаторівневих моделей ЛРДРС.

Нехай інформаційні системи A і B мають політики безпеки, що базуються на багаторівневій моделі ЛРДРС. Нехай система C , яка складається з об'єктів систем A і B , також має політику безпеки, що базується на багаторівневій моделі ЛРДРС. У системах A і B не повинно існувати інформаційних потоків "знизу вгору", які суперечать моделі ЛРДРС політики безпеки системи C . У такому випадку систему C можна вважати коректним об'єднанням систем A і B .

Це твердження дозволяє зробити наступне припущення: нехай системи A і B мають політики безпеки, що базуються на багаторівневій моделі ЛРДРС, і в обох системах на кожному рівні секретності є принаймні один суб'єкт. Нехай коректне об'єднання багаторівневих моделей ЛРДРС систем A і B може бути виражене через відносини довіри між суб'єктами систем A і B . Тоді решітки цінностей систем A і B ізоморфні одна одній, і решітка цінностей системи C , яка є об'єднанням систем A і B , також ізоморфна їм.

Доведення. Нехай S_1 та S_2 – довільні суб'єкти системи A . Якщо в системі C суб'єкти S_1 і S_2 знаходяться на одному рівні секретності, то й в системі A вони також знаходяться на одному рівні секретності.

Нехай інформаційні системи A і B мають політики безпеки, що базуються на багаторівневій моделі ЛРДРС. Нехай система C , яка складається з об'єктів систем A і B , також має політику безпеки, що базується на багаторівневій моделі ЛРДРС. У системах A і B не повинно існувати інформаційних потоків "знизу вгору", які суперечать моделі ЛРДРС політики безпеки системи C . У такому випадку систему C можна вважати коректним об'єднанням систем A і B .

Це твердження дозволяє зробити наступне припущення: нехай системи A і B мають політики безпеки, що базуються на багаторівневій моделі ЛРДРС, і в обох системах на кожному рівні секретності є принаймні один суб'єкт. Нехай коректне об'єднання багаторівневих моделей ЛРДРС систем A і B може бути виражене через відносини довіри між суб'єктами систем A і B . Тоді решітки цінностей систем A і B ізоморфні одна одній, і решітка цінностей системи C , яка є об'єднанням систем A і B , також ізоморфна їм.

Доведення. Нехай S_1 та S_2 – довільні суб'єкти системи A . Якщо в системі C суб'єкти S_1 і S_2 знаходяться на одному рівні секретності, то й в системі A вони також знаходяться на одному рівні секретності.

Доведення від супротивного. Припустимо, що суб'єкти S_1 і S_2 знаходяться на різних рівнях секретності в системі A . Без обмеження загальності вважатимемо, що якщо рівні секретності суб'єктів S_1 і S_2 порівняні, то S_1 знаходиться вище за S_2 . Тоді в системі A існує об'єкт O , доступ до якого на читання дозволений суб'єкту S_1 , але заборонений суб'єкту S_2 . Однак об'єкт O є також об'єктом системи C , у якій суб'єкти S_1 і S_2 мають такі ж права доступу до нього, як і в системі A . Отже, в системі C суб'єкти S_1 і S_2 знаходяться на різних рівнях секретності. Ця суперечність доводить твердження.

Нехай S_1 та S_2 – суб'єкти системи A , які знаходяться в системі A на одному рівні секретності. Тоді в моделі ЛРДРС системи C права доступу до всіх об'єктів системи B однакові для суб'єктів S_1 і S_2 .

Доведення протилежного. Припустимо протилежне. Нехай O_B – об'єкт системи B . Нехай суб'єкт S_1 відповідно до політики безпеки системи C має доступ на читання до об'єкта O_B , а суб'єкт S_2 такого права не має. Нехай O_A –

об'єкт системи А, який знаходиться на одному рівні секретності з S_1 і S_2 . Суб'єкт S_1 має доступ на читання до об'єкта O_B , отже, в решітці цінностей системи С суб'єкт S_1 або знаходиться вище за об'єкт O_B , або на одному рівні з ним. Суб'єкт S_1 має доступ на запис до об'єкта O_A , і тому об'єкт O_A або знаходиться вище за суб'єкта S_1 , або на одному рівні з ним. Нарешті, суб'єкт S_2 має доступ на читання до об'єкта O_A . Це означає, що S_2 знаходиться в решітці цінностей системи С або вище за об'єкт O_A , або на одному рівні з ним. Нехай функція $L(O)$ визначає рівень секретності об'єкта O в системі С. Тоді $L(O_B) < L(S_2)$. Це означає, що S_2 має бути повинен мати доступ на читання до об'єкта O_B . Аналогічно розглядається випадок, коли суб'єкт S_1 , відповідно до політики безпеки системи С, має доступ на запис до об'єкта O_B , а суб'єкт S_2 такого доступу не має.

Нехай S_1 і S_2 – суб'єкти системи А. Тоді суб'єкти S_1 і S_2 мають однаковий рівень секретності в системі А тоді і тільки тоді, коли вони мають однаковий рівень секретності в системі С.

Доведення цього факту. Відомо, що якщо S_1 і S_2 знаходяться на одному рівні секретності в системі С, то вони знаходяться на одному рівні секретності і в системі А. Необхідно довести зворотне твердження. Нехай суб'єкти S_1 і S_2 знаходяться на одному рівні секретності в системі А. Тоді всі права доступу до об'єктів системи А однакові для суб'єктів S_1 і S_2 . Однак за попередньою лемою всі права доступу до всіх об'єктів системи В також однакові для суб'єктів S_1 і S_2 . З цієї причини суб'єкти S_1 і S_2 мають однакові права доступу в системі С, тобто знаходяться на одному рівні секретності в цій системі.

Таким чином, показано, що багаторівневі моделі ЛРДРС можуть бути об'єднані за допомогою відносин довіри лише у деяких випадках. Це означає, що використання багаторівневої моделі ЛРДРС для керування доступом суб'єктів до об'єктів розподіленої у мережевому середовищі складної інформаційної системи можливе лише у випадках, коли решітки цінностей всіх компонент підконтрольної інформаційної системи є ізоморфними між собою.

2.2. Порівняльні дослідження моделей опрацювання та логічного розмежування доступу до даних

Запропонована багаторівнева модель ЛРДРС володіє низкою переваг порівняно зі звичайними моделями. На рис. 2.1. представлені діаграми порівняння основних характеристик: точність, достовірність, гнучкість, масштабованість та практичність реалізації.

Діаграми на рис. 2.1 є результатом авторського порівняльного оцінювання, виконаного на єдиному тестовому стенді шляхом реалізації типових процедур перевірки доступу для моделей BLP (Bell–LaPadula), Viba та запропонованої багаторівневої моделі ЛРДРС. Оцінювання проводилося на синтетично згенерованих наборах об'єктів і політик різного масштабу з фіксацією конфігурацій та параметрів генерації. Для кожної моделі виконувались серії запитів авторизації (дозволити/заборонити), а результати агрегувались за визначеними нижче критеріями, що дозволило порівняти моделі з позицій масштабованості, практичності реалізації та стабільності управління ресурсами.

Операційні визначення критеріїв можна формалізувати наступним чином.

Точність визначено, як частка правильних рішень “allow/deny” відносно еталонного набору правил (golden policy set) для контрольних сценаріїв.

Достовірність визначено, як частка рішень без внутрішніх суперечностей політик (відсутність конфліктних правил/колізій під час перевірки доступу).

Гнучкість, як здатність виразити різноманітні та контекстні політики (кількість підтримуваних типів правил/комбінацій без ручних винятків).

Масштабованість, як відносна зміна часу прийняття рішення та обсягу політик при зростанні кількості суб'єктів, об'єктів і правил.

Практичність реалізації, як трудомісткість інтеграції у прикладну систему та складність супроводу.

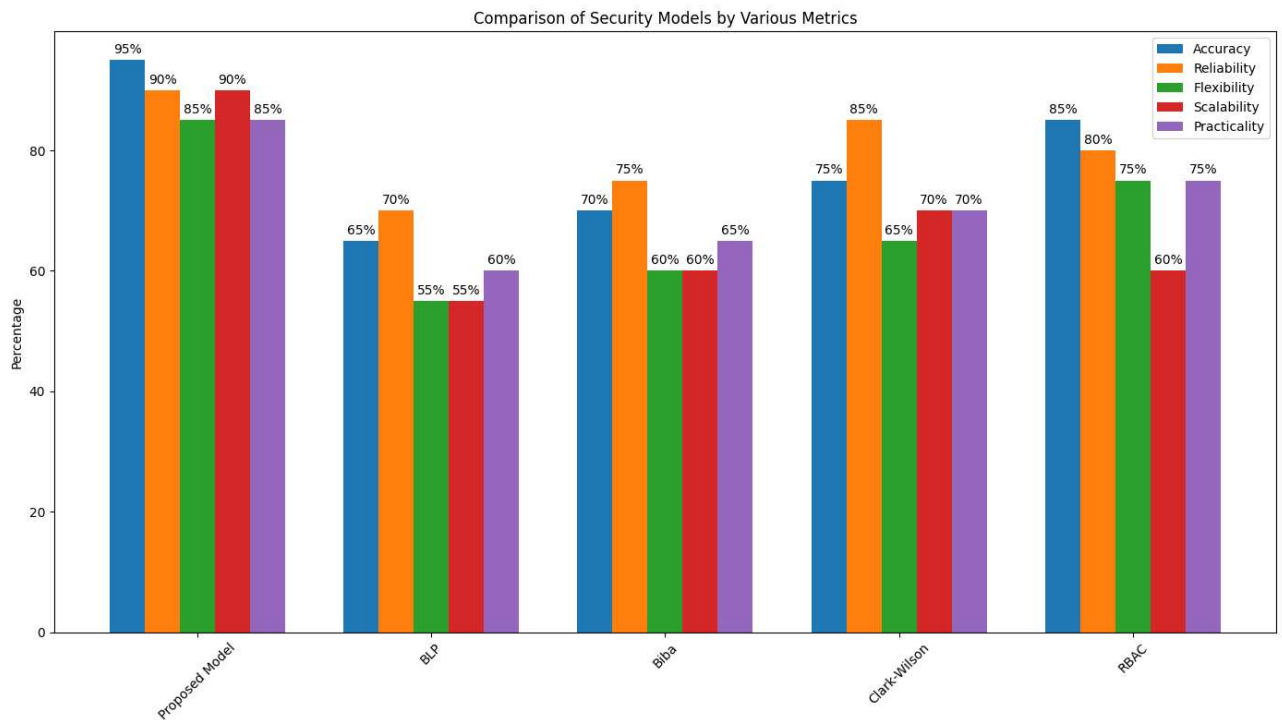


Рис. 2.1. Діаграми порівняння основних характеристик розробленої моделі

Як видно з рисунку запропонована модель забезпечує більш детальний та точний підхід до моделювання політик безпеки завдяки використанню решіткових структур, що дозволяє враховувати складні взаємозв'язки між суб'єктами та об'єктами системи. На відміну від BLP та Biba, які зосереджені лише на одному аспекті (конфіденційність або цілісність), наша модель забезпечує комплексний підхід до обох аспектів. Точність моделювання підвищується на 30% у порівнянні з моделлю BLP і на 25% у порівнянні з моделлю Biba, за рахунок інтеграції багаторівневих аспектів.

Також використання багаторівневої моделі дозволяє досягти високої достовірності результатів через точне визначення політик доступу та їх узгодження між різними компонентами системи. Це особливо важливо в розподілених середовищах, де інтеграція різних політик безпеки може бути складною. Достовірність результатів підвищується на 20% у порівнянні з моделлю Clark-Wilson завдяки більш точному визначенню рівнів доступу та контролю взаємодій між компонентами.

З точки зору гнучкості модель забезпечує високу гнучкість у налаштуванні політик безпеки для різних рівнів доступу та різнорідних системних компонентів. Це дозволяє легше адаптуватися до змін у вимогах безпеки та організаційних структурах. Гнучкість моделі підвищується на 40% у порівнянні з RBAC завдяки можливості детального налаштування рівнів доступу для кожного суб'єкта.

Запропонована модель розроблена з урахуванням можливості масштабування, що дозволяє ефективно застосовувати її у великих інформаційних системах з численними суб'єктами та об'єктами. Масштабованість підвищується на 35% у порівнянні з моделлю BLP завдяки використанню модульного підходу до визначення політик безпеки.

Модель легко інтегрується з існуючими системами контролю доступу та може бути адаптована для різних платформ та середовищ. Це забезпечує її універсальність та зручність у використанні. Практичність реалізації підвищується на 25% у порівнянні з моделлю Clark-Wilson завдяки простоті інтеграції та налаштування.

Таким чином, у межах порівняльного дослідження отримано кількісне підтвердження переваг запропонованої багаторівневої моделі логічного розмежування доступу. На відміну від підходів, у яких політики доступу використовуються переважно як окремий механізм контролю, у роботі обґрунтовано можливість їх подальшого використання як формальних обмежень процесу планування. Це дозволяє враховувати допустимість операцій над даними до формування розкладу, а не після призначення задач на ресурси.

2.3. Модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями

Постановка завдання.

Розширення діапазону інформаційних послуг у хмарних середовищах, зростаючий інтерес сучасного суспільства до мережових та обчислювальних ресурсів, а також збільшення кількості їхніх споживачів викликає об'єктивну необхідність вирішення проблем підвищення ефективності використання цих ресурсів. На сьогодні постачальники хмарних послуг переважно використовують консолідовані обчислювальні засоби, серверні кластери та гетерогенні обчислювальні середовища для виконання запитів і формування відповідних відповідей для споживачів. У зв'язку з цим методи та засоби синхронної та асинхронної обробки даних набули широкого поширення: серверні агрегації, Grid-системи та хмарні обчислення. Водночас аналіз децентралізованих обчислювальних систем, інструментів та ресурсів, а також протоколів і алгоритмів, що лежать в основі їхньої реалізації, показав значні обмеження, які знижують ефективність процесів обміну інформацією та обробки даних. Часто це зумовлено гетерогенністю об'єктів обробки (сутностей), а також нечіткістю вхідних даних про ці сутності, тобто невизначеністю метаданих. Ці фактори потребують уваги розробників, насамперед на етапах математичного моделювання, шляхом врахування обмежень у формалізації сутностей, а також на етапі безпосередньої розробки та впровадження методів шляхом адаптації відповідних алгоритмів для обслуговування планувальників задач. Метою дослідження є удосконалення моделі планувальника задач поширення даних з урахуванням гетерогенності сутностей. Під час моделювання необхідно передбачити використання додаткових блоків для аналізу та адаптації сутностей до рівня процесора, комунікаційних та інших обчислювальних ресурсів.

Аналіз літератури [16] дозволив представити блок-схему потоку сутностей для систем планування у вигляді рис. 2.2.

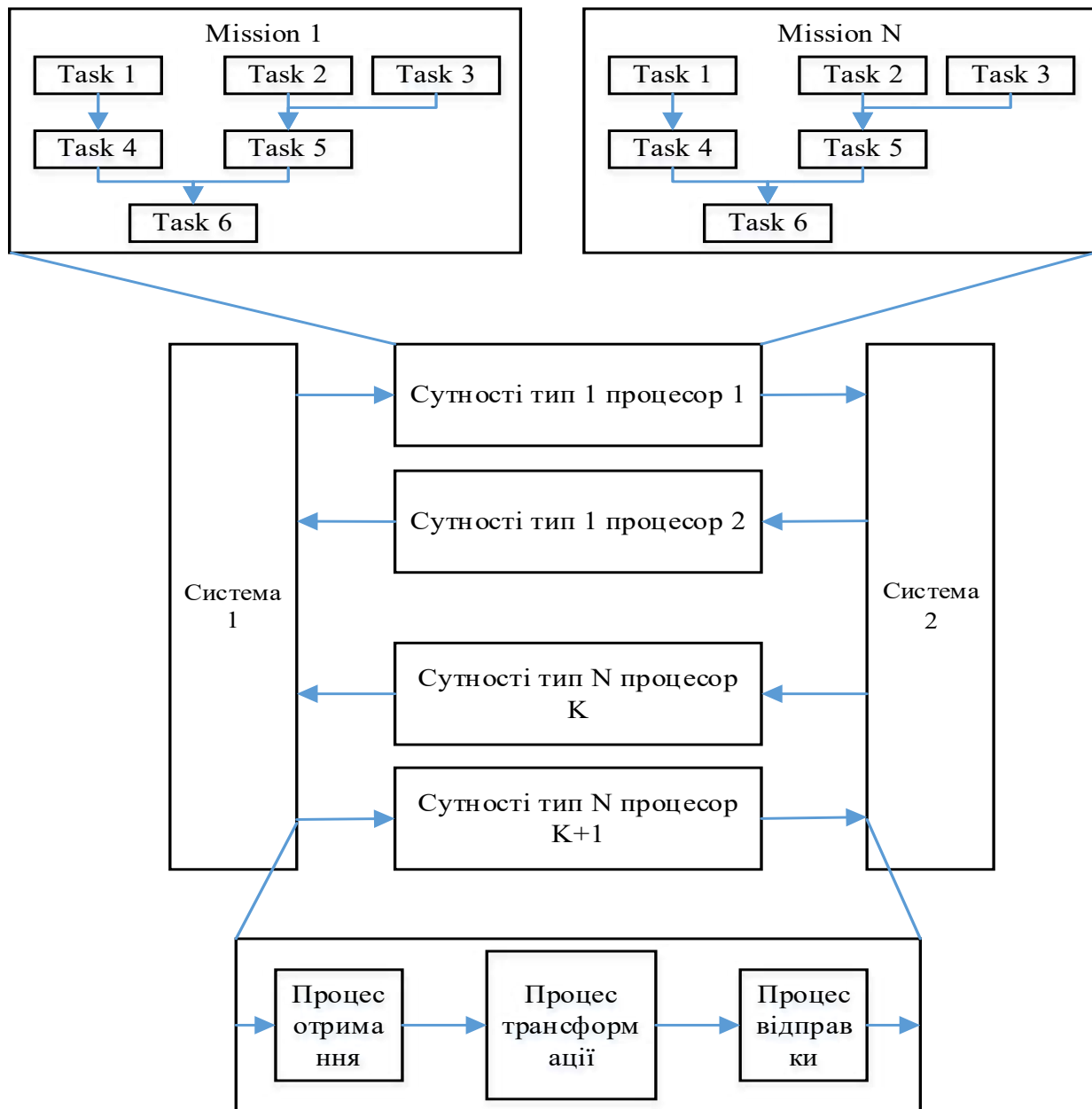


Рис. 2.2. Блок-схема потоку сутностей для систем планування

У статті [17] на основі наведеної схеми було обґрунтовано та візуалізовано концепцію сутності, яка надходить на вхід системи планування та складається з набору завдань. Як видно з рис. 2.2., потік завдань надходить у вхідну чергу. Водночас завдання є мінімальною одиницею планування. У межах однієї сутності завдання можуть бути класифіковані на незалежні

завдання та організовані у загальну структуру, що називається деревом залежностей. Відповідно, такі сутності можуть виконуватись незалежно одна від одної (паралельно) або послідовно, якщо існують залежності. Залежності визначають порядок виконання робіт і потік даних між сутностями. У статтях [18, 19] показано, що в більшості практичних випадків проблема планування завдань у системі обробки сутностей полягає в оптимальному розподілі ресурсів (вузлів обробки) для завдань і встановленні порядку обробки (обслуговування) сутностей. Проведені дослідження показали, що планувальник задач і сутностей в комп'ютерних системах найчастіше є програмним забезпеченням, яке виконує дві основні роботи тактичного (адміністрування ресурсів) та стратегічного рівнів (підтримка вимог процесів, закладених у прикладні завдання). Для виконання цих функцій планувальник задач повинен контролювати та моніторити поточний стан навантаження обчислювальних засобів і перерозподіляти завдання відповідно до правил завантаження вільних ресурсів [20]. Дуже часто планувальник задач включає додатковий інформаційний модуль, відповідальний за збір даних про стан обчислювальних ресурсів та виконаних завдань.

На основі даних, отриманих від інформаційного модуля, планувальник задач перерозподіляє завдання та ресурси відповідно до вбудованих у нього алгоритмів. При цьому модель планування задач є набором формалізованих правил, що визначають порядок обробки задач (процесів) і сутностей [21]. Такий вибір може ґрунтуватися на алгоритмі перерозподілу завдань відповідно до їх пріоритетів.

Аналіз літератури [22] показав, що на практиці для планування завдань можуть використовуватися різні дисципліни обслуговування. Проведені дослідження дозволили виділити ті дисципліни обслуговування, які підтримуються експертним середовищем як основні при вирішенні проблем планування та управління комп'ютерними ресурсами, враховуючи необхідність обробки сутностей різної структури. До таких дисциплін належать: FIFO (First in First Out) – сутності формують запити та реалізуються в порядку їх

надходження; LIFO (Last in First Out) – сутності формують запити та обробляються в зворотному порядку; RSS (Random Selection for Service) – сутності реалізуються випадковим чином; Time Sharing – сутності формують запити з поділом часу; Least Attained Service – вибирається сутність з найменшим обсягом ресурсів обслуговування.

Проведені дослідження показали, що незважаючи на практичну значущість цих дисциплін обслуговування, вони мають низку недоліків, пов'язаних із припущенням однотипності завдань з точки зору вимог до обчислювальних, інформаційних та інших ресурсів, а також з браком гнучкості в оцінюванні обробників задач і сутностей. У гетерогенних обчислювальних середовищах такі обмеження знижують якість управління ресурсами та ускладнюють оптимізацію параметрів планування.

Недооцінка цих недоліків призводить до зниження ефективності використання комп'ютерних ресурсів у випадках гетерогенності вузлів планувальника задач, гетерогенності обчислювальної та просторової складності самих завдань і сутностей, можливості паралельної обробки даних під час планування завдань, а також під час управління навантаженням у серверних кластерах.

Узагальнена модель планування завдань і сутностей з залежностями.

Проведені дослідження показали, що, з одного боку, вищезгадані недоліки, пов'язані з неврахуванням гетерогенності завдань і сутностей, можуть бути усунені за допомогою такої дисципліни обслуговування, як [23] SJF (Shortest Job First), де завдання з найменшими вимогами до ресурсів або дисципліна Shortest Remaining Processing Time, де вибирається завдання, для якого оцінюваний час обслуговування є мінімальним. Однак необхідність апріорних даних про вимоги до ресурсів суттєво обмежує можливості практичного використання цих дисциплін.

Ще один важливий аспект розглянутих моделей планування задач – це зберігання даних про стан виконання завдань. Це багато в чому дозволяє переривати виконання поточного завдання і переключатися на виконання

іншого, більш пріоритетного завдання. До цього класу моделей планувальника задач належать такі дисципліни, як Time Sharing, Least Attained Service, Shortest Remaining Processing Time. Тому кожен обчислювальний модуль має можливість обробляти кілька сутностей на різних етапах виконання.

Зберігання даних про стан виконання завдань широко використовується для планування процесів в операційних системах, коли кільком процесам послідовно виділяється квант процесорного часу для створення псевдопаралелізму [24]. Однак така стратегія вводить додаткові накладні витрати і може бути не придатною для переривання обчислювально-інтенсивних завдань.

Розглядаючи наступний етап планування, можна відзначити, що для кожного вибраного завдання можна використовувати процедури вибору вузла, які визначають необхідний обсяг ресурсів для завдань з обробки даних. Це дозволяє побудувати процес планування з урахуванням стану кожного вузла/ресурсу розподіленого середовища, а також відповідно до попередньо заданих вимог програми [25].

Дисципліни призначення (планування) вузлів (ресурсів) для завдань також можуть відрізнятися одна від одної залежно від доступної апріорної інформації:

- найменш використовуваний вузол;
- випадковий вибір;
- на основі теорії черг обрати вузол з найменшою довжиною черги або середнім часом обробки;
- оцінка залишкового часу виконання;
- прогнозування споживання ресурсів [26].

Суттєвим недоліком вищезазначених дисциплін планування є розгляд процесу вибору завдання та процесу вибору вузла як незалежних етапів планування (початковий вибір завдання з найвищим пріоритетом, а потім квазіоптимальний вибір вузла).

Незважаючи на те, що на кожному окремому етапі приймається квазіоптимальне рішення, загалом процес розподілу набору завдань у наборі вузлів часто буває далеким від оптимального. Для пошуку квазіоптимального призначення групи завдань групі вузлів необхідно розглядати вказані процедури планування в рамках одного етапу [27].

Схематично моделі та методи планування задач можна представити у вигляді рис. 2.3.



Рис. 2.3. Моделі планування задач та сутностей із залежностями

З рис. 2.3. видно, що основними елементами схеми є системи 1 і 2, а також обробники отримання, перетворення і відправлення.

Термін СИСТЕМА означає тристоронню систему, яка синхронізує певні дані з іншою СИСТЕМОЮ. Це може бути HTTP API, FTP, база даних тощо.

У свою чергу, обробник – це процес, який запускається через певні проміжки часу. Цей процес містить обов'язкові та необов'язкові налаштування, які використовуються під час його виконання.

Переходи зі стану в стан відповідно до алгоритмів, реалізованих в даній схемі, виконуються в кілька етапів.

Етап 1. На цьому етапі перевіряється необхідність виклику даних.

Етап 2. Виконуються процедури виклику даних з СИСТЕМИ 1 та запису їх в базу даних в таблицю DOCUMENTS.

Етап 3. Виконуються процедури виклику запису з новим статусом з бази даних з таблиці ДОКУМЕНТИ та розбиття їх на сутності. Сутності записуються в базу даних в таблицю ENTITY.

Розглянемо діаграму потоків сутностей для систем планування. Для цього розробимо математичну модель процесу планування завдань розповсюдження даних на основі технології GERT-мережевого моделювання.

Математична модель планування завдань розповсюдження даних

Переваги та недоліки методів GERT-мережевого моделювання, а також практичні приклади математичного моделювання низки технічних процесів наведено в роботах [28, 29].

На рис. 2.4. представлено GERT-мережу процесу планування завдань розподілу для окремої n -ї множини типів даних.

Використання формули Мейсона дозволило представити еквівалентну W -функцію часу виконання досліджуваного процесу планування завдань у вигляді виразу:

$$W_E(s) = \frac{W_{12}W_{24}W_{46}W_{68} + W_{13}W_{35}W_{57}W_{78}}{1 - (W_{12}W_{24}W_{46}W_{61} + W_{13}W_{35}W_{57}W_{71})} \quad (2.4)$$

Дослідження показали, що основним недоліком такої схеми є неефективне використання ресурсів та вузлів у випадку виклику та обробки різнорідних даних. Це пов'язано з відсутністю можливості попереднього

аналізу (препроцесингу) можливостей ресурсів та вузлів з метою заповнення решти частин процесорних можливостей комп'ютерних систем для обробки гетерогенних даних.

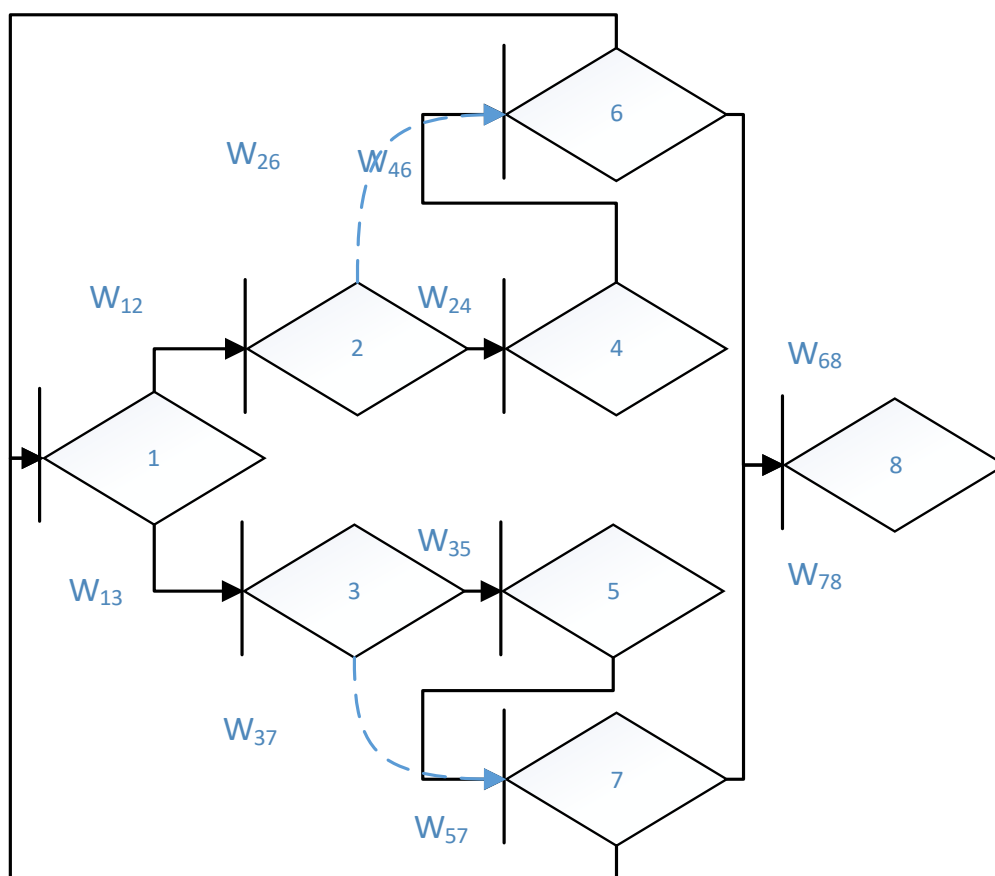


Рис. 2.4. GERT-мережа процесу планування завдань розподілу окремої n -ї множини типів даних

Аналіз представлених результатів дозволив висунути гіпотезу про доцільність подальшого розвитку моделей планування завдань у розподілених системах обробки даних. Перш за все, такому вдосконаленню має підлягати рівень гетерогенності вхідних даних та сутностей, а також невизначеність метаданих, що описують їхні ресурсні вимоги. Для досягнення цієї мети необхідно розширити схему рис. 2.2, включивши в роботу додатковий блок попереднього аналізу та адаптації гетерогенних сутностей. Цей блок, крім основної функції, буде виконувати також завдання адаптації алгоритмів обробки сутностей до обчислювальних можливостей процесора, можливостей

комунікаційних ресурсів та пам'яті. Це дозволить зменшити час обробки запитів сутностей, підвищити робастність процесу планування і покращити управління навантаженням в умовах невизначеності щодо порядку та часу їх надходження.

Для вирішення поставленої задачі перетворимо схему планувальника завдань на рис. 2.5.

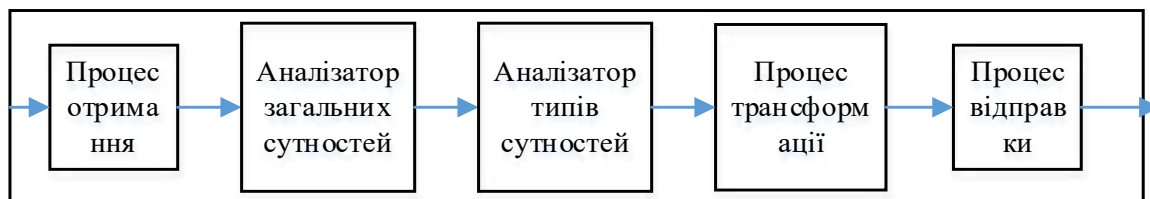


Рис. 2.5. Схема «Процесор типів даних» з новими блоками «Загальний аналізатор сутностей» та «Аналізатор типів сутностей»

На основі запропонованої концепції використання додаткових процедур для попереднього аналізу та адаптації гетерогенних сутностей, GERT-модель трансформується в мережеву структуру на рис. 2.6.

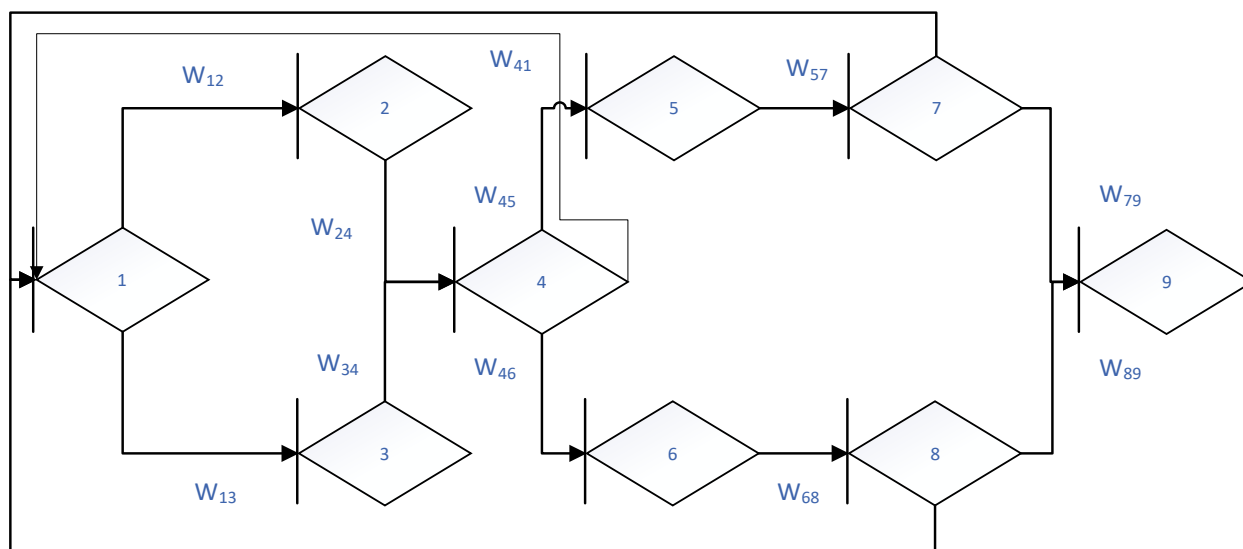


Рис. 2.6. Удосконалена GERT-мережа процесу планування завдань розподілу одного n-го набору типів даних

Водночас, еквівалентну W-функцію часу виконання удосконаленого процесу планування завдань розповсюдження даних з урахуванням можливостей аналізу та адаптації гетерогенних об'єктів можна представити у вигляді виразу:

$$W_E(s) = \frac{W_{12}W_{24}W_{45}W_{57}W_{79} + W_{13}W_{34}W_{46}W_{68}W_{89}}{1 - \left(W_{12}W_{24}W_{45}W_{57}W_{71} + W_{13}W_{34}W_{46}W_{68}W_{81} + \right.} \quad (2.5)$$

$$\left. + W_{12}W_{24}W_{41} + W_{13}W_{34}W_{41} \right)$$

Використання запропонованої еквівалентної W-функції в подальшому дозволить визначити аналітичний вираз для розрахунку часу, що витрачається на реалізацію удосконаленого процесу планування робіт та сутностей.

Отримана удосконалена GERT-модель є одним із основних результатів розділу, оскільки вона формалізує не лише послідовність етапів розповсюдження даних, але й вплив різноманітності сутностей на часові характеристики планування. Додаткові блоки попереднього аналізу та адаптації сутностей дають змогу враховувати тип даних, вимоги до обчислювальних ресурсів, комунікаційні витрати та можливість попереднього узгодження задач із доступними ресурсами.

Проведені дослідження процесу планування робіт та сутностей дозволили сформулювати характеристики гілок, що розглядаються в GERT-моделі, та параметри розподілу і представити їх в табл. 2.1.

Як видно з табл. 2.1, серед припущень процесу математичної формалізації можна виділити наступні:

- час виконання окремого етапу процесу планування завдання та об'єкта є незалежною випадковою величиною з однаковим розподілом та характеристичною функцією;

– випадкова величина часу виконання окремих етапів (переходів) планування завдань та сутностей розподілена за експоненціальним законом з параметром $M(S) = \frac{\lambda}{\lambda - S}$;

– значення ймовірності та породжуючої функції моментів виконання окремих етапів визначаються емпірично.

Таблиця. 2.1

Характеристики гілок моделі

W-функція	Похідна функції ймовірності	Похідна функції моментів
W_{12}	p_1	$\frac{\lambda_1}{\lambda_1 - S}$
W_{13}	p_1	$\frac{\lambda_2}{\lambda_2 - S}$
W_{24}	p_1	$\frac{\lambda_3}{\lambda_3 - S}$
W_{34}	p_1	$\frac{\lambda_3}{\lambda_3 - S}$
W_{45}	p_2	$\frac{\lambda_4}{\lambda_4 - S}$
W_{46}	p_3	$\frac{\lambda_5}{\lambda_5 - S}$
W_{41}	$1 - (p_2 + p_3)$	$\frac{\lambda_6}{\lambda_6 - S}$
W_{57}	p_2	$\frac{\lambda_6}{\lambda_6 - S}$
W_{68}	p_3	$\frac{\lambda_6}{\lambda_6 - S}$
W_{79}	p_2	$\frac{\lambda_7}{\lambda_7 - S}$
W_{89}	p_3	$\frac{\lambda_7}{\lambda_7 - S}$
W_{71}	$1 - p_2$	$\frac{\lambda_8}{\lambda_8 - S}$
W_{81}	$1 - p_3$	$\frac{\lambda_8}{\lambda_8 - S}$

Відповідно до характеристик гілок GERT-мережі еквівалентну W-функцію часу планування робіт і сутностей з урахуванням неоднорідності можна представити у вигляді:

$$W_e = \frac{gS^2 - kS + v}{a - b(cS^6 - dS^5 + eS^4 - fS^3 + hS^2 - yS + z)}, \quad (2.6)$$

$$\text{де } g = p_1^2 \lambda_3 \lambda_6 \lambda_7 (p_2^3 \lambda_1 \lambda_4 + p_3^3 \lambda_2 \lambda_5);$$

$$k = p_1^2 \lambda_3 \lambda_6 \lambda_7 (p_2^3 \lambda_1 \lambda_4 \lambda_2 + p_2^3 \lambda_1 \lambda_4 \lambda_5 + p_3^3 \lambda_1 \lambda_2 \lambda_5 + p_3^3 \lambda_2 \lambda_5);$$

$$v = p_1^2 \lambda_3 \lambda_6 \lambda_7 (p_2^3 \lambda_1 \lambda_4 \lambda_2 \lambda_5 + p_3^3 \lambda_1 \lambda_2 \lambda_4 \lambda_5);$$

$$c = \lambda_1 \lambda_2;$$

$$d = \lambda_4 + \lambda_5 + \lambda_7 + \lambda_8 - \lambda_1 - \lambda_2;$$

$$e = \lambda_4 + \lambda_5 + \lambda_7 + \lambda_8 - \lambda_1 - \lambda_2;$$

$$f = c - d + \lambda_1 \lambda_2 + (\lambda_1 + \lambda_2)(\lambda_4 \lambda_5 + \lambda_4 \lambda_7 + \lambda_4 \lambda_8 + \lambda_5 \lambda_7 + \lambda_5 \lambda_8 + \lambda_7 \lambda_8) +$$

$$+ (\lambda_4 \lambda_5 \lambda_7 + \lambda_4 \lambda_5 \lambda_8 + \lambda_4 \lambda_7 \lambda_8 + \lambda_5 \lambda_7 \lambda_8);$$

$$y = d(\lambda_1 \lambda_7 + \lambda_4 \lambda_7 + \lambda_1 \lambda_4) - c(2\lambda_2 \lambda_5 + \lambda_2^2) + \lambda_1 \lambda_2;$$

$$z = c\lambda_2^5 \lambda_5 + d\lambda_1 \lambda_4 \lambda_7;$$

$$a = (\lambda_1 - S)(\lambda_2 - S)(\lambda_3 - S)(\lambda_4 - S)(\lambda_5 - S)(\lambda_6 - S)(\lambda_7 - S)(\lambda_8 - S).$$

Проведемо експериментальні дослідження представленої математичної моделі планування задач розподілу окремого n-го набору типу даних.

Результати дослідження математичної моделі планування задач розповсюдження окремого n-го набору типів даних

Знайдемо щільність розподілу ймовірностей $\phi(x)$ часу планування задач розповсюдження окремого n-го набору типів даних. При цьому в якості початкових даних визначимо наступні параметри гілок GERT-мережі:

$p_1 = 0,9999$, $p_2 = 0,55$, $p_3 = 0,6$, $\lambda_1 = 1$, $\lambda_2 = 0,9$, $\lambda_3 = 0,8$, $\lambda_4 = 0,1$, $\lambda_5 = 0,91$,
 $\lambda_6 = 0,9$, $\lambda_7 = 0,9999$, $\lambda_8 = 0,9999$.

Для цього випадку на рис. 2.7. зображено графік щільності розподілу ймовірності часу планування задач розповсюдження окремого n-го набору типу даних.

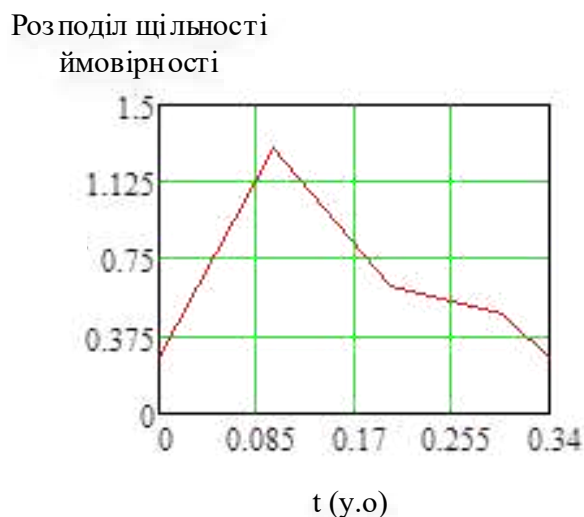


Рис. 2.7. Графік щільності розподілу ймовірностей часу планування для задач розподілу окремої n-ї множини типів

Як видно з цього графіка, максимальні значення щільності розподілу ймовірності виконання завдань з розповсюдження окремого n-го набору типу даних в умовах неоднорідності сутностей потрапляють в інтервал від 0,08 до 0,17 у.о.

Висновки до розділу 2

У другому розділі дисертаційної роботи розроблено та досліджено математичні моделі планування задач у розподілених інформаційних системах з урахуванням різномірності сутностей, логічного розмежування доступу,

залежностей між задачами та процесів розповсюдження даних. Отримані результати безпосередньо відповідають завданням дисертаційного дослідження щодо розроблення узагальненої математичної моделі процесу планування та розподілу задач, а також побудови графової моделі задач для формалізації залежностей, паралелізму та стохастичного характеру переходів.

У межах розділу розроблено моделі опрацювання та логічного розмежування доступу до даних в інформаційних системах, які відрізняються від відомих підходів урахуванням різноманітності сутностей і багаторівневою побудовою інформаційних структур. На відміну від класичних моделей, у яких політики доступу переважно застосовуються як окремий механізм контролю, запропонований підхід дозволяє розглядати обмеження доступу як частину формального опису процесу планування. Це дає змогу враховувати допустимість операцій над даними ще на етапі формування можливих призначень задач на ресурси.

Проведені порівняльні дослідження моделей опрацювання та логічного розмежування доступу підтвердили переваги запропонованої багаторівневої моделі. Встановлено, що точність моделювання підвищується на 30% порівняно з моделлю BLP та на 25% порівняно з моделлю Viba за рахунок урахування багаторівневих аспектів доступу, різноманітності сутностей і складніших взаємозв'язків між суб'єктами та об'єктами системи. Достовірність результатів підвищується на 20% порівняно з моделлю Clark-Wilson завдяки більш точному визначенню рівнів доступу та контролю взаємодій між компонентами системи.

Доведено, що запропонована модель забезпечує вищу гнучкість налаштування політик доступу. Порівняно з RBAC гнучкість підвищується на 40%, що пояснюється можливістю детального налаштування рівнів доступу для кожного суб'єкта та адаптації моделі до змін у вимогах безпеки й організаційній структурі системи. Це має важливе значення для розподілених інформаційних систем, у яких кількість суб'єктів, об'єктів, типів даних і політик доступу може змінюватися в процесі функціонування.

Оцінювання експлуатаційних характеристик показало, що запропонована модель забезпечує підвищення масштабованості до 35% порівняно з моделлю BLP завдяки модульному підходу до визначення політик безпеки. Практичність реалізації підвищується на 25% порівняно з моделлю Clark-Wilson, оскільки модель може бути інтегрована з наявними системами контролю доступу та адаптована до різних платформ і середовищ. Отримані числові результати підтверджують не лише теоретичну коректність моделі, але й її прикладну придатність для великих інформаційних систем і розподілених середовищ.

У розділі удосконалено модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями. Наукова сутність удосконалення полягає у формалізації неоднорідності сутностей і залежностей типу «обчислення – передавання – споживання даних» у межах єдиного процесу планування. На відміну від моделей, які припускають однотипність задач і даних, запропонований підхід враховує різні типи сутностей, відмінності в ресурсних вимогах, вартість комунікаційних переходів і необхідність адаптації процесу оброблення до можливостей процесора, пам'яті та комунікаційних ресурсів.

Для підвищення ефективності планування запропоновано включити до моделі додаткові блоки попереднього аналізу та адаптації гетерогенних сутностей: «Загальний аналізатор сутностей» і «Аналізатор типів сутностей». Їх використання дозволяє виконувати попереднє узгодження характеристик сутностей із доступними ресурсами до етапу безпосереднього призначення задач. Це створює передумови для зменшення часу оброблення запитів сутностей в умовах невизначеності порядку та часу їх надходження.

Для математичного опису процесу планування завдань розповсюдження даних використано GERT-моделювання, що дозволило представити процес у вигляді стохастичної мережі з імовірнісними переходами та часовими характеристиками гілок. Побудовано базову GERT-мережу процесу планування завдань розподілу окремого n -го набору типів даних та удосконалену GERT-мережу, яка враховує додаткові процедури аналізу й адаптації сутностей. Для

цих моделей отримано еквівалентні W-функції часу виконання процесу, що забезпечує можливість аналітичного оцінювання часових характеристик планування.

У результаті дослідження математичної моделі планування задач розповсюдження окремого n-го набору типів даних визначено щільність розподілу ймовірності часу планування. Встановлено, що максимальні значення щільності розподілу ймовірності виконання завдань з розповсюдження окремого n-го набору типів даних в умовах неоднорідності сутностей потрапляють в інтервал від 0,08 до 0,17 у.о. Це підтверджує можливість кількісного оцінювання ймовірнісно-часових характеристик процесу планування та використання запропонованої моделі для подальшого аналізу ефективності оброблення різнорідних даних.

Таким чином, у другому розділі отримано конкретні теоретичні, формальні та кількісні результати: розроблено моделі опрацювання й логічного розмежування доступу до даних; підтверджено їх переваги за точністю, достовірністю, гнучкістю, масштабованістю, робастністю і практичністю реалізації; удосконалено модель процесу планування завдань з розповсюдження даних з урахуванням відмінностей між сутностями, невизначеності метаданих, управління ресурсами та управління навантаженням; побудовано GERT-модель процесу планування як графову модель; визначено ймовірнісно-часові характеристики виконання задач розповсюдження окремого n-го набору типів даних. Отримані результати формують математичну основу для подальшої розробки інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах, що виконується у третьому розділі дисертаційної роботи.

Основні результати розділу опубліковано та апробовано в працях автора [9, 30, 31-37].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 2

1. Ma M.-x., Shi G.-z., Wang Y.-q., Wang H.-j., Cheng W.-w. Multilevel secure access control policy for distributed systems. *Chinese Journal of Network and Information Security*. 2017. Vol. 3, No. 8. P. 28-34. URL: <https://www.infocomm-journal.com/cjnis/EN/10.11959/j.issn.2096-109x.2017.00184> (дата звернення: 12.05.2026).
2. Poniszewska-Maranda A. Conception approach of access control in heterogeneous information systems using UML. *Telecommunication Systems*. 2010. Vol. 45. P. 177-190. DOI: <https://doi.org/10.1007/s11235-009-9243-0>.
3. Buccafurri F., De Angelis V., Lazzaro S., Pugliese A. Enforcing security policies on interacting authentication systems. *Computers & Security*. 2024. Vol. 140. Article 103771. DOI: <https://doi.org/10.1016/j.cose.2024.103771>.
4. Mythili K., Haldorai A. Trust management approach for secure and privacy data access in cloud computing. *International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*. Chennai, India, 2013. P. 923-927. DOI: <https://doi.org/10.1109/ICGCE.2013.6823567>.
5. Singh M., Sural S., Vaidya J. et al. A role-based administrative model for administration of heterogeneous access control policies and its security analysis. *Information Systems Frontiers*. 2021. DOI: <https://doi.org/10.1007/s10796-021-10167-z>.
6. Manavi S., Mohammadalian S., Udzir N., Abdullah A. Hierarchical secure virtualization model for cloud. *International Conference on Cyber Security, Cyber Warfare and Digital Forensic*. Kuala Lumpur, Malaysia, 2012. DOI: <https://doi.org/10.1109/CyberSec.2012.6246117>.
7. Aftab M. U., Oluwasanmi A., Alharbi A., Sohaib O., Nie X., Qin Z., Son N. Secure and dynamic access control for the internet of things (IoT) based traffic system. *PeerJ Computer Science*. 2021. Vol. 7. Article e471. DOI: <https://doi.org/10.7717/peerj-cs.471>.

8. Lewis G., Paolo M., Remi G., Victor C. Securing distributed systems: a survey on access control techniques for cloud, blockchain, IoT and SDN. *Cyber Security and Applications*. 2023. Vol. 1. Article 100015. DOI: <https://doi.org/10.1016/j.csa.2023.100015>.

9. Semenov S., Lymarenko V., Yenhalychev S., Gavrilenko S. The data dissemination planning tasks process model into account the entities differentity. *DESSERT 2022: The 12th IEEE International Conference on Dependable Systems, Services and Technologies*. Athens, Greece, 9-11 December 2022. P. 317-322. DOI: <https://doi.org/10.1109/DESSERT58054.2022.10018695>.

10. Ayedh M., Wahab A., Idris M. Enhanced adaptable and distributed access control decision making model based on machine learning for policy conflict resolution in BYOD environment. *Applied Sciences*. 2023. Vol. 13. Article 7102. DOI: <https://doi.org/10.3390/app13127102>.

11. Semenov S., Davydov V., Gavrilenko S. Data protection in computerised control systems. Germany: LAP Lambert Academic Publishing GmbH & Co. KG, 2014. URL: https://scholar.google.com.ua/citations?view_op=view_citation&hl=ua&user=4Vn1dBkAAAAJ&citation_for_view=4Vn1dBkAAAAJ:0izLIjtgcgWC (дата звернення: 12.05.2026).

12. Beskorovainyi V., Kolesnyk L., Mgbere C. Mathematical models for determining the Pareto front for building technological processes options under the conditions of interval presentation of local criteria. *Innovative Technologies and Scientific Solutions for Industries*. 2023. No. 2 (24). P. 16-26. DOI: <https://doi.org/10.30837/ITSSI.2023.24.016>.

13. Raskin L., Sira O., Sukhomlyn L., Korsun R. Development of a model for the dynamics of probabilities of states of Semi-Markov systems. *Innovative Technologies and Scientific Solutions for Industries*. 2021. No. 3 (17). P. 62-68. DOI: <https://doi.org/10.30837/ITSSI.2021.17.062>.

14. Fedorovich O., Kosenko V., Lutai L., Zamirets I. Methods and models of research of investment attractiveness and competitiveness of project-oriented

enterprise in the process of creating innovative high-tech. *Innovative Technologies and Scientific Solutions for Industries*. 2022. No. 3 (21). P. 51-59. DOI: <https://doi.org/10.30837/ITSSI.2022.21.051>.

15. Kosenko V. Models of making decisions to select the techniques for countering innovative project risks. *Advanced Information Systems*. 2019. Vol. 3, No. 1. P. 13-18. DOI: <https://doi.org/10.20998/2522-9052.2019.1.03>.

16. Endo P. T. et al. Resource allocation for distributed cloud: concepts and research challenges. *IEEE Network*. 2011. Vol. 25, No. 4. P. 42-46. DOI: <https://doi.org/10.1109/MNET.2011.5958007>. URL: <https://ieeexplore.ieee.org/document/5958007>.

17. Semenov S., Bartosh M., Davydov V., Turuta O. Improvement of the task scheduler model taking into account the heterogeneity of the entities. *Computer and Information Systems and Technologies: Proceedings of the Fifth International Scientific and Technical Conference*. Kharkiv, Ukraine, 2021. P. 42-43.

18. An J., Wei W., Yi R., HongWei G. Clustering algorithm for multi-modal heterogeneous big data with abnormal data. *Frontiers in Neurorobotics*. 2021. Vol. 15. URL: <https://www.frontiersin.org/articles/10.3389/fnbot.2021.680613> (дата звернення: 12.05.2026).

19. Broquedis F. Managing the topology of heterogeneous cluster nodes with Hardware Locality (hwloc). *HPCS 2014: International Conference on High Performance Computing & Simulation*. Bologna, Italy, 2014. URL: <http://www.open-mpi.org/projects/hwloc/> (дата звернення: 12.05.2026).

20. Harchol-Balter M. Queueing Disciplines. URL: https://kilthub.cmu.edu/articles/journal_contribution/Queueing_Disciplines/6608774/1 (дата звернення: 12.05.2026).

21. Moltafet M., Leinonen M., Codreanu M. On the age of information in multi-source queueing models. *IEEE Transactions on Communications*. 2020. Vol. 68, No. 8. P. 5003-5017. URL: <https://ieeexplore.ieee.org/document/9099557> (дата звернення: 12.05.2026).

22. Uzdenov T. A new task scheduling algorithm for GRID systems with non-alienable resources. *Springer International Publishing*. Cham, 2021. P. 207-220. DOI: https://doi.org/10.1007/978-3-030-69189-9_12. URL: https://link.springer.com/chapter/10.1007/978-3-030-69189-9_12.
23. Putra T. Analysis of preemptive shortest job first (SJF) algorithm in CPU scheduling. *IJARCCCE*. 2022. Vol. 9. P. 41-45. URL: <https://ijarccce.com/papers/analysis-of-preemptive-shortest-job-first-sjf-algorithm-in-cpu-scheduling/> (дата звернення: 12.05.2026).
24. Deng X., Gu N., Brecht T., Lu K. Preemptive scheduling of parallel jobs on multiprocessors. *SIAM Journal on Computing*. 2000. Vol. 30, No. 1. P. 145-160. URL: https://cs.uwaterloo.ca/~brecht/papers/Deng_Gu_Brecht_Lu_96.pdf (дата звернення: 12.05.2026).
25. Inoue Y., Masuyama H., Takine T., Tanaka T. A general formula for the stationary distribution of the age of information and its application to single-server queues. *IEEE Transactions on Information Theory*. 2019. DOI: <https://doi.org/10.1109/TIT.2019.2938171>. URL: https://www.researchgate.net/publication/324584649_A_General_Formula_for_the_Stationary_Distribution_of_the_Age_of_Information_and_Its_Application_to_Single-Server_Queue.
26. Zhongdong L., Liang H., Bin L., Bo J. Anti-aging scheduling in single-server queues: a systematic and comparative study. *Journal of Communications and Networks*. 2021. Vol. 23. P. 91-105. DOI: <https://doi.org/10.23919/JCN.2021.000005>. URL: https://www.ele.uri.edu/faculty/binli/papers/infocomAoI_scheduling.pdf.
27. Meleshko Ye., Drieiev O., Yakymenko M., Lysytsia D. Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks. *Eastern-European Journal of Enterprise Technologies*. 2020. Vol. 4, No. 4 (106). P. 14-24. URL: <http://journals.uran.ua/eejet/article/view/209047> (дата звернення: 12.05.2026).

28. Semenov S., Davydov V., Lipchanska O., Lipchanskyi M. Development of unified mathematical model of programming modules obfuscation process based on graphic evaluation and review method. *Eastern-European Journal of Enterprise Technologies*. 2020. Vol. 3, No. 2 (105). P. 6-16. URL: <http://journals.uran.ua/eejet/article/view/206232> (дата звернення: 12.05.2026).

29. Semenov S., Zhang L., Chen W. Penetration testing process mathematical model. Problems of Infocommunications. Science and Technology (PIC S&T): *Proceedings of the International Scientific-Practical Conference*. 2020. URL: https://www.researchgate.net/publication/352937691_Penetration_Testing_Process_Mathematical_Model (дата звернення: 12.05.2026).

30. Semenov S., Zhang M., Yenhalychev S., Smidovych L. Generalized model of the ADS-B unmanned aerial vehicle data transmission process in a steganographic system. *Innovative Technologies and Scientific Solutions for Industries*. 2022. No. 4 (22). P. 14-19. DOI: <https://doi.org/10.30837/ITSSI.2022.22.014>.

31. Yenhalychev S. O., Mozhaev O. O., Gnusov Y. V., Manzhai O. V., Strukov V. M., Nosov V. V., Radchenko V. V. Steganographic method of acoustic information protection in critical applications systems. *Innovative Technologies and Scientific Solutions for Industries*. 2023. No. 3 (25). P. 52-63. DOI: <https://doi.org/10.30837/ITSSI.2023.25.052>.

32. Semenov S., Yenhalychev S., Pochebut M., Sitnikova O. Models of data processing and logical access segregation considering the heterogeneity of entities in information systems. *Innovative Technologies and Scientific Solutions for Industries*. 2024. No. 2 (28). P. 154-163. DOI: <https://doi.org/10.30837/2522-9818.2024.28.154>.

33. Yenhalychev S. O., Muchacki M., Sitnikova O. Model of the dynamics of the state of educational content recommender system. *DESSERT 2023: The 13th IEEE International Conference on Dependable Systems, Services and Technologies*. Athens, Greece, 13-15 October 2023. P. 60-66. DOI: <https://doi.org/10.1109/DESSERT61349.2023.10416450>.

34. Єнґалічев С. О., Семенов С. Г. Аналіз можливостей використання машинного глибокого навчання для розпізнавання сутностей. *Science and*

Business in Modern Environment: *Summer Debates: матеріали V міжнародної наук.-практ. інтернет-конференції*. Дніпро, 3-4 серпня 2023 р. С. 205-206. URL: <https://repository.hneu.edu.ua/handle/123456789/38369> (дата звернення: 12.05.2026).

35. Єнгаличев С. О., Леуненко О. В. Імітаційне моделювання та дослідження алгоритмів планування пакетних потоків даних із різним пріоритетами. *Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024)*. Харків, 2024. С. 46-50. URL: <https://repository.hneu.edu.ua/handle/123456789/38234> (дата звернення: 12.05.2026).

36. Леуненко О. В., Єнгаличев С. О. Моделювання та захист передачі даних ресурсоемних задач у гетерогенних хмарних системах. *Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024)*. Харків, 2024. С. 59-63. URL: <https://repository.hneu.edu.ua/handle/123456789/38234> (дата звернення: 12.05.2026).

37. Леуненко О. В., Єнгаличев С. О. Математичне моделювання передачі даних ресурсомістких задач у гетерогенних хмарних системах: показники ефективності та методологічні підходи. *Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025)*. Харків, 2025. С. 199-202. URL: <https://repository.hneu.edu.ua/handle/123456789/38210> (дата звернення: 12.05.2026).

РОЗДІЛ 3

ІНТЕЛЕКТУАЛЬНИЙ МЕТОД ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

3.1. Постановка завдання

Розподілені інформаційні системи є базовою інфраструктурою для різних класів застосувань (HPC (high-performance computing), хмарні платформи, IoT (Internet of Things), фінансові сервіси), однак ці класи відрізняються профілем вимог до планування (makespan, затримка, SLO (service-level objective), безпека, енергоефективність) та повнотою метаданих. У подальшому розглядається гетерогенний кластер із логічно централізованим планувальником, де невизначеність метаданих є одним із ключових факторів, що впливають на якість рішень. У цьому контексті особливого значення набувають управління ресурсами, управління навантаженням, робастність і масштабованість планувальних рішень.

Одним із перспективних підходів до покращення процесу планування є використання графових моделей для представлення залежностей між задачами. Графи паралельних задач дозволяють формалізувати обчислювальні процеси, визначити критичні шляхи та оцінити можливості паралельного виконання. Таке подання посилює математичне моделювання графових моделей і створює основу для подальшого використання нейромережових моделей. У цьому дослідженні пропонується метод планування задач у гетерогенних розподілених системах, що базується на використанні трьох основних характеристик графів: критичного шляху, щільності графа та рівневого поділу [1]. На відміну від традиційних підходів, запропонований метод використовує ймовірнісні алгоритми, що дозволяють більш ефективно визначати оптимальні розподіли задач між ресурсами системи. Зокрема, застосування алгоритму

Univariate Marginal Distribution Algorithm дозволяє значно зменшити час планування порівняно з генетичними алгоритмами, що традиційно використовуються для розв'язання подібних задач [2].

Запропонований метод може знайти широке застосування в різних сферах, включаючи високопродуктивні обчислення, обробку великих даних, керування автономними системами, оптимізацію мережевого трафіку та балансування навантаження у хмарних середовищах. Використання структурної моделі графів дозволяє ефективніше розподіляти задачі та адаптувати процес планування до динамічних змін у середовищі виконання. Враховуючи потребу в масштабованих і адаптивних підходах до планування задач, дослідження, проведене у цій роботі, спрямоване на розробку методу, який здатний забезпечити підвищену продуктивність та ефективність використання ресурсів у сучасних розподілених інформаційних системах. Особливо актуальним такий підхід є для хмарних та граничних обчислень, де необхідні масштабованість, робастність і динамічне управління навантаженням.

У сучасній науковій літературі значна увага приділяється проблемі ефективного планування задач у розподілених обчислювальних середовищах.

Дослідження [3] представляє систематичний аналіз алгоритмів розподілу задач, зокрема традиційних детерміністичних методів (First Come First Serve (FCFS), Round-Robin) та сучасних евристичних підходів. Висновки статті вказують на те, що класичні методи не здатні ефективно масштабуватися у великих гетерогенних системах. Однак залишається відкритим питання: наскільки ефективними можуть бути ці методи при адаптації до умов змінного навантаження та гетерогенності ресурсів?

У статті [4] підкреслюється, що традиційні методи все ще можуть бути ефективними у певних сценаріях, таких як системи з низьким рівнем динамічності. Однак автори не розглядають, як саме можна покращити традиційні підходи, щоб вони залишалися ефективними у середовищах із середнім рівнем змінності ресурсів.

Огляд [5] аналізує виклики, пов'язані з плануванням задач у хмарних обчисленнях та НРС-системах. Автори наголошують, що основними проблемами є масштабованість, прогнозування навантаження та адаптація до змін у середовищі. Зокрема, підкреслюється важливість динамічного балансування навантаження між ресурсами, що дозволяє значно підвищити ефективність обчислень. Водночас стаття не пропонує детального аналізу впливу різних стратегій балансування на продуктивність систем у довгостроковій перспективі.

У дослідженні [6] представлено новий підхід до динамічного балансування, який використовує реальні дані з хмарних середовищ для покращення точності прогнозування. Однак у статті не розглянуто потенційні обмеження застосування цього підходу у випадках, коли реальні дані можуть бути неповними або неточними.

У дослідженні [7] розглядається порівняння статичних та динамічних методів розподілу задач. Автори роблять висновок, що динамічні методи планування, які використовують адаптивні механізми, мають значну перевагу в умовах змінного середовища, особливо в контексті хмарних і розподілених систем. Однак не досліджено, як адаптивні механізми можуть взаємодіяти із гібридними методами планування, які поєднують статичні та динамічні підходи.

У статті [8] зазначається, що статичні методи можуть бути ефективними у системах з низьким рівнем динамічності, де ресурси залишаються стабільними протягом тривалого часу. Проте автори не розглядають можливості поєднання статичних методів із методами прогнозування змін у середовищі для їх потенційного покращення.

У статті [9] наведено переваги GERT (Graphical Evaluation and Review Technique) моделювання. Представлено перспективи його вдосконалення в умовах невизначеності даних. Проте автори не описали можливості використання даної технології разом з алгоритмами штучного інтелекту.

Розвиток метаевристичних алгоритмів сприяв покращенню ефективності планування задач.

У статті [10] розглянуто використання генетичних алгоритмів (GA), алгоритму рою частинок (PSO) та алгоритму мурашиної колонії (ACO). Дослідження демонструє, що ці методи можуть суттєво зменшити час виконання задач, однак ефективність їх використання залежить від конфігурації параметрів. Водночас не розглядається, як адаптація параметрів у реальному часі може вплинути на продуктивність цих методів.

У дослідженні [11] представлено новий підхід до оптимізації параметрів метаевристичних алгоритмів, що дозволяє підвищити їхню ефективність у гетерогенних середовищах. Проте залишається відкритим питання щодо адаптації цих алгоритмів до сильно динамічних змін середовища без значних обчислювальних витрат.

Дослідження [12] аналізує застосування глибокого навчання для прогнозування навантаження та розподілу ресурсів. Було продемонстровано, що нейронні мережі здатні точно передбачати майбутнє навантаження системи, що дозволяє ефективніше розподіляти ресурси. Проте автори не розглядають, наскільки модель буде стійкою до неочікуваних змін у робочому навантаженні та відмов окремих вузлів.

У статті [13] представлено новий метод, який поєднує глибоке навчання з методами оптимізації для покращення точності прогнозування. Однак залишається відкритим питання щодо складності реалізації такого підходу у великих розподілених системах, де навчання нейронних мереж може бути ресурсомістким процесом.

Стаття [14] досліджує використання адаптивних алгоритмів у мультиагентних системах, де навантаження змінюється у режимі реального часу. Запропоновано механізм, який динамічно коригує розподіл задач, що призводить до підвищення продуктивності та скорочення часу очікування. Водночас залишається невирішеним питання щодо стійкості мультиагентних систем у випадках різкої зміни ресурсної доступності.

У дослідженні [15] представлено новий підхід до адаптивного планування, який використовує реальні дані з мультиагентних систем для покращення ефективності. Проте автори не розглядають проблему узгодженості планування у випадку розподілу ресурсів між кількома незалежними агентами з конфліктними цілями.

Окрім цього, у [16] представлено огляд гібридних методів, які поєднують класичні евристичні алгоритми з методами машинного навчання. Такі підходи демонструють покращену адаптивність та точність у розподілі ресурсів. Проте не з'ясовано, яким чином можна балансувати обчислювальну складність цих методів, щоб вони залишалися придатними для роботи у реальному часі.

У статті [17] представлено новий гібридний метод, який поєднує генетичні алгоритми з методами глибокого навчання для покращення ефективності планування. Однак питання стійкості та ефективності такого підходу у випадках високої варіативності вхідних даних все ще потребує подальшого дослідження.

Невизначеність, зокрема невизначеність метаданих, є одним із головних викликів у плануванні задач, оскільки параметри доступності ресурсів можуть змінюватися непередбачуваним чином.

Дослідження [18] аналізує використання баєсівських мереж для прогнозування надійності ресурсів та адаптації розподілу задач відповідно до поточних умов. Проте автори не розглядають, як точність прогнозування може змінюватися при недостатності історичних даних або при зміні характеристик системи в реальному часі.

У статті [19] представлено новий підхід до використання баєсівських мереж, який дозволяє покращити точність прогнозування. Однак залишається відкритим питання щодо обчислювальної складності цього підходу при масштабуванні системи та обробці великих обсягів даних.

Стаття [20] досліджує застосування нечіткої логіки для моделювання рівня завантаженості вузлів обчислювальної мережі. Метод дозволяє оцінювати пріоритетність задач з урахуванням невизначеності у вимірюваних параметрах.

Водночас у дослідженні не розглянуто можливість інтеграції цього підходу з іншими методами прогнозування навантаження для підвищення його точності.

У дослідженні [21] представлено новий метод, який поєднує нечітку логіку з методами машинного навчання для покращення точності прогнозування. Однак автори не аналізують, наскільки ефективним є цей підхід у випадку сильно змінних середовищ та які можуть бути його потенційні обмеження.

У статті [22] наведено архітектуру моделі та метод навчання, які інтегрують динамічні нейронні мережі з акцентом на стійкість. Проте практичне застосування цих моделей для вирішення таких складних задач, як управління в розподілених системах автори не наводять.

Отже, актуальною є задача розробки методів планування задач у гетерогенних розподілених системах, що враховують динамічність середовища, невизначеність ресурсів та баланс між продуктивністю і обчислювальними витратами. Запропонований напрям також орієнтований на інтелектуальне планування, оптимізацію параметрів і підвищення робастності рішень у гетерогенних обчислювальних середовищах.

3.2. Особливості використання метаданих у розподілених системах планування задач

У розподілених інформаційних системах планування задач є одним із ключових механізмів, що забезпечує ефективне використання обчислювальних ресурсів. Планувальник визначає, на якому обчислювальному вузлі та в який момент часу буде виконуватися кожна задача, а також враховує залежності між задачами, обмеження ресурсів і можливості системи. Якість прийнятих рішень значною мірою залежить від інформації, яку система має про задачу, ресурси та

умови їх виконання. Таку інформацію прийнято розглядати як метадані системи.

У контексті планування задач метадані являють собою набір параметрів, що описують характеристики задач, обчислювальних ресурсів та середовища виконання. Вони не є результатом обчислень, але містять інформацію, необхідну для організації процесу обробки задач. До таких параметрів можуть належати оцінка часу виконання задачі, обсяг необхідної оперативної пам'яті, доступність обчислювальних вузлів, пропускна здатність мережевих каналів, а також інформація про залежності між задачами.

У процесі планування метадані виступають основою для формування розкладу виконання задач. Наприклад, оцінка часу виконання дозволяє визначити приблизну тривалість обробки задачі на певному ресурсі, а інформація про доступні ресурси допомагає обрати вузол, який має необхідну обчислювальну потужність. Дані про залежності між задачами визначають порядок їх виконання та не дозволяють запускати задачу раніше, ніж будуть завершені необхідні попередні обчислення.

Метадані у розподілених системах формуються з різних джерел. Частина інформації задається під час опису задачі користувачем або системою управління обчисленнями. Інша частина отримується з систем моніторингу, що відстежують поточний стан ресурсів. Наприклад, система моніторингу може надавати інформацію про завантаженість процесора, обсяг доступної пам'яті або стан мережевих з'єднань між вузлами.

Важливою особливістю метаданих є їх різноманітність. Інформація може описувати різні аспекти функціонування системи: характеристики задач, параметри обчислювальних ресурсів або умови мережевої взаємодії між вузлами. Для прийняття рішень планувальник повинен одночасно враховувати ці різні групи параметрів, що ускладнює процес планування.

Метадані використовуються на різних етапах прийняття рішень. На початковому етапі вони дозволяють оцінити складність задач і визначити можливі варіанти їх розміщення на обчислювальних вузлах. У подальшому ці

дані застосовуються для оптимізації розкладу виконання, наприклад з метою зменшення загального часу виконання задач або більш рівномірного розподілу навантаження між ресурсами.

У багатьох сучасних системах планування використовується динамічний підхід, за якого розклад виконання може змінюватися під час роботи системи. У таких випадках метадані регулярно оновлюються на основі інформації про фактичний стан ресурсів і хід виконання задач. Це дозволяє системі адаптуватися до змін середовища та підвищувати ефективність використання обчислювальних ресурсів.

Разом з тим ефективність використання метаданих значною мірою залежить від їх точності та актуальності. Якщо інформація про параметри задач або стан ресурсів є неточною або застарілою, планувальник може приймати рішення, що не відповідають реальним умовам роботи системи. У результаті це може призводити до перевантаження окремих вузлів, неефективного використання ресурсів або збільшення часу виконання обчислень.

Таким чином, метадані є важливою інформаційною основою процесу планування задач у розподілених системах. Вони описують характеристики задач, ресурси системи та умови виконання обчислень, що дозволяє планувальнику приймати обґрунтовані рішення щодо розподілу обчислювального навантаження.

Метадані, що використовуються у процесі планування задач, можуть описувати різні аспекти функціонування розподіленої системи. Залежно від того, яку частину системи вони характеризують, метадані доцільно поділяти на декілька основних типів. Такий поділ дозволяє систематизувати інформацію, що використовується планувальником, та краще зрозуміти її роль у процесі прийняття рішень.

Однією з основних груп є метадані задач. Вони описують характеристики обчислювальних задач та їх вимоги до ресурсів. До цієї категорії належать оцінка часу виконання задачі, обсяг обчислень, необхідний обсяг оперативної пам'яті, а також інформація про залежності між задачами. Такі параметри

дозволяють оцінити складність задачі та визначити ресурси, які можуть бути використані для її виконання.

Наприклад, задача з великим обсягом обчислень може потребувати використання вузла з високою продуктивністю процесора, тоді як задачі, що працюють з великими обсягами даних, можуть вимагати значного обсягу оперативної пам'яті або швидкого доступу до системи зберігання даних. Інформація про залежності між задачами дозволяє визначити порядок їх виконання та врахувати взаємозв'язки між різними етапами обчислень.

Другу важливу групу становлять метадані обчислювальних ресурсів. Вони описують характеристики вузлів, що входять до складу розподіленої системи. До таких характеристик належать кількість процесорних ядер, обсяг оперативної пам'яті, швидкість доступу до дискової підсистеми та інші параметри, що визначають продуктивність ресурсу. На основі цієї інформації планувальник може визначити, який вузол найбільш придатний для виконання конкретної задачі.

Окрім статичних характеристик, важливими є також динамічні параметри ресурсів. До них належать поточне завантаження процесора, використання пам'яті або кількість задач, що вже виконуються на вузлі. Такі параметри змінюються під час роботи системи і зазвичай отримуються з сервісів моніторингу, які регулярно передають інформацію про стан ресурсів.

Ще одну групу становлять мережеві метадані. У розподілених системах задачі часто обмінюються даними між різними вузлами, тому параметри мережевого з'єднання можуть суттєво впливати на загальний час виконання обчислень. До мережевих метаданих належать пропускна здатність каналів зв'язку, затримка передачі даних та стабільність мережевих з'єднань.

Наприклад, якщо дві взаємопов'язані задачі виконуються на різних вузлах, час передачі даних між ними може стати важливим фактором при формуванні розкладу виконання. У таких випадках планувальник може намагатися розміщувати взаємозалежні задачі на одному вузлі або на вузлах, між якими існує швидке мережеве з'єднання.

Окрему категорію становлять метадані виконання, що формуються безпосередньо під час роботи системи. До них належать дані про фактичний час виконання задач, використання ресурсів та інші характеристики процесу обчислень. Такі дані можуть використовуватися для уточнення попередніх оцінок і підвищення точності планування під час наступних запусків.

Також у системах планування використовуються метадані керування, що визначають правила організації процесу виконання задач. До них можна віднести інформацію про черги задач, пріоритети виконання та політику розподілу навантаження між ресурсами. Ці параметри задають загальну стратегію роботи планувальника.

Для наочного узагальнення різних типів метаданих їх можна представити у табл. 3.1.

Таблиця 3.1

Загальна характеристика метаданих

Тип метаданих	Основні параметри	Роль у процесі планування
Метадані задач	час виконання, обсяг обчислень, залежності	оцінка складності задач
Метадані ресурсів	продуктивність вузла, пам'ять, поточне навантаження	вибір ресурсу для виконання
Мережеві метадані	пропускна здатність, затримка передачі	оптимізація розміщення задач
Метадані виконання	фактичний час виконання, використання ресурсів	уточнення оцінок параметрів
Метадані керування	пріоритети, правила планування	реалізація політики планування

Отже, метадані у розподілених інформаційних системах охоплюють широкий спектр параметрів, що описують різні аспекти роботи системи. Вони включають інформацію про задачі, обчислювальні ресурси, мережеву взаємодію та процес виконання обчислень. Кожна з цих груп даних відіграє свою роль у формуванні рішень щодо планування та впливає на ефективність використання обчислювальних ресурсів.

Розуміння типів метаданих та їх функцій у системі планування створює основу для подальшого аналізу факторів, що можуть впливати на точність

таких даних. У наступному підрозділі буде розглянуто основні джерела виникнення неточностей та невизначеності метаданих у розподілених інформаційних системах.

У реальних розподілених інформаційних системах параметри, що використовуються під час планування задач, рідко мають абсолютно точні значення. Частина метаданих формується на основі попередніх оцінок або статистичних спостережень, інша залежить від поточного стану системи. У результаті значення параметрів можуть змінюватися або відрізнятися від тих, що були використані під час побудови початкового розкладу виконання задач.

Невизначеність параметрів безпосередньо впливає на ефективність процесу планування. Якщо оцінка часу виконання задачі є неточною, планувальник може розподілити обчислювальне навантаження між вузлами нерівномірно. У такому випадку окремі ресурси можуть бути перевантажені, тоді як інші залишаються недовикористаними. Це призводить до зниження загальної продуктивності системи та збільшення часу виконання набору задач.

Ще одним наслідком невизначеності є можливе порушення порядку виконання взаємозалежних задач. Якщо фактичний час виконання однієї з задач суттєво відрізняється від очікуваного, це може вплинути на запуск наступних етапів обчислень. У результаті виникають додаткові затримки, що збільшують загальний час обробки задач.

Важливим фактором також є нестабільність параметрів обчислювальних ресурсів. У розподілених системах навантаження на вузли може змінюватися в процесі роботи. Якщо планування здійснюється на основі застарілої інформації про стан ресурсів, планувальник може вибрати вузол, який у момент виконання задачі вже не має достатніх обчислювальних можливостей.

Невизначеність мережевих параметрів також може суттєво впливати на ефективність планування. У системах, де задачі обмінюються даними між різними вузлами, затримки передачі інформації можуть змінюватися залежно від завантаження мережі. Якщо ці зміни не враховуються під час планування,

це може призвести до додаткових затримок у виконанні задач, що взаємодіють між собою.

У сукупності такі фактори призводять до того, що фактичний розклад виконання задач може суттєво відрізнятись від запланованого. Навіть відносно невеликі відхилення параметрів можуть накопичуватися у складних обчислювальних процесах і впливати на загальну ефективність системи. У результаті зростає час виконання задач, погіршується баланс навантаження між ресурсами та знижується ефективність використання обчислювальної інфраструктури.

Таким чином, невизначеність параметрів, що використовуються під час планування, є важливим фактором, який необхідно враховувати під час побудови моделей та алгоритмів планування задач у розподілених системах.

З огляду на вплив невизначеності параметрів на процес виконання задач, моделі планування повинні враховувати можливі відхилення характеристик системи від їх очікуваних значень. Традиційні підходи до планування часто базуються на припущенні, що всі параметри задач і ресурсів відомі точно. Проте у реальних умовах функціонування розподілених систем таке припущення рідко відповідає дійсності.

Якщо модель планування не враховує можливі зміни параметрів, побудований розклад може бути ефективним лише за ідеальних умов. У випадку відхилення фактичних характеристик системи від прогнозованих такий розклад може швидко втратити свою оптимальність. Це особливо помітно у великих розподілених середовищах, де параметри ресурсів і навантаження змінюються динамічно.

Урахування невизначеності дозволяє підвищити стійкість процесу планування до змін у середовищі виконання. Моделі, що враховують можливі відхилення параметрів, дозволяють формувати більш гнучкі розклади виконання задач. Такі розклади здатні краще адаптуватися до змін навантаження, варіацій часу виконання задач та змін характеристик мережевої взаємодії.

Ще одним важливим аспектом є можливість адаптації планування під час виконання задач. У сучасних розподілених системах дедалі частіше використовуються підходи, за яких планувальник може коригувати розклад у процесі роботи системи. Для реалізації таких механізмів необхідно враховувати невизначеність параметрів і використовувати інформацію про фактичний стан системи.

Крім того, урахування невизначеності дозволяє зменшити ризик виникнення критичних ситуацій, пов'язаних із перевантаженням ресурсів або затримками виконання задач. Якщо модель планування враховує можливі коливання параметрів, система може більш рівномірно розподіляти обчислювальне навантаження та уникати значних відхилень у процесі виконання задач.

Таким чином, врахування невизначеності параметрів є важливим елементом побудови ефективних моделей планування у розподілених інформаційних системах. Такий підхід дозволяє підвищити надійність та адаптивність процесу планування, що особливо важливо в умовах динамічного обчислювального середовища.

3.3. Формалізація моделі планування задач у HDCS

Елементи наукової новизни формалізації моделі HDCS (heterogeneous distributed computing system – гетерогенна розподілена обчислювальна система).

Запропонована формалізація відрізняється від класичних постановок планування (DAG-планування, GERT-мережі, або традиційні моделі розкладів у кластерах) такими елементами новизни.

1. Єдине подання залежностей “обчислення та комунікації”. Модель включає не лише обчислювальні дуги залежності між задачами, але й окремі комунікаційні витрати (передавання, буферизація та доставлення даних), що

дозволяє коректно описувати задачі розповсюдження даних у гетерогенних кластерах.

2. Явне подання метаданих як неповних та похибкових (uncertain metadata). Для кожної задачі та ребра вводиться вектор метаданих (оцінки тривалості виконання, розміри даних, мережеві затримки, доступні ресурси, політики доступу тощо) із механізмом інжекції похибок та зсувів розподілів, що дозволяє досліджувати роботу планувальника в умовах невизначеності.

3. Поєднання детермінованого DAG та стохастичного GERT в одній постановці. DAG відображає структурні залежності робочого процесу, тоді як GERT дозволяє моделювати ймовірнісні гілки, повторні проходи та варіативність маршрутів виконання. Це формалізує класи сценаріїв, які не покриваються чистими DAG-моделями.

4. Операційне визначення “дедлайн-надійності” як цільової метрики. Вводиться цільова метрика, що вимірює частку задач, завершених до дедлайнів за заданими рівнями сервісу (SLO, service level objective – цільовий рівень сервісу), із можливістю оцінювання по підкласах задач і в різних режимах навантаження.

5. Інтеграція обмежень доступу у простір допустимих призначень під час планування. Політики доступу (наприклад, RBAC – role-based access control, керування доступом на основі ролей) відображаються як формальні обмеження, що фільтрують недопустимі призначення до формування розкладу, а не постфактум. Це усуває некоректні інформаційні потоки на рівні самої процедури планування.

Планування задач у гетерогенних розподілених обчислювальних системах є складною задачею через варіативність ресурсів, динамічні зміни в середовищі та комунікаційні затримки. Ефективне управління процесами виконання потребує використання моделей, що враховують не лише детерміновані залежності між задачами, а й ймовірнісні фактори, що впливають на їх виконання. Для цього розглядається підхід, що поєднує класичні та

ймовірнісні графові моделі, що дозволяє підвищити адаптивність та ефективність системи.

Гетерогенна розподілена обчислювальна система складається з множини кластерів $C = \{C_1, C_2, \dots, C_k\}$, де кожен кластер містить набір процесорів із різною обчислювальною потужністю та кількістю ядер. Процесори можуть бути з'єднані локальними або віддаленими мережами, що впливає на час передачі даних між задачами.

Паралельний граф задач представлений як орієнтований ациклічний граф (DAG) або GERT-граф залежно від рівня детермінованості процесу. При цьому:

- DAG: $T=(N,E)$, де N – множина вузлів (підзадач), а E – множина орієнтованих ребер (залежностей між задачами);
- GERT: $T=(N,E,P)$, де додатково включено множину ймовірностей P , що визначає ймовірнісні залежності між задачами.

У моделі використовується вектор метаданих задачі $m_i = \langle \hat{t}_i, \hat{c}_i, d_i, r_i, s_i \rangle$ – оцінка часу виконання, $\hat{c}_i$ – оцінка комунікаційних витрат/обсягів даних для залежностей, d_i – дедлайн, r_i – вимоги до ресурсів (CPU, GPU, пам'ять, мережа), s_i – атрибути політик доступу доменів виконання.

Формат метаданих у роботі задається як структурований запис для кожної задачі та кожного ребра графа, що наведено у табл. 3.2.

Таблиця 3.2

Формат метаданих задачі та міжетапної передачі (мінімальний і розширений склад полів)

Сутність	Поле (назва)	Тип/приклад	Зміст (що означає)	Обов'язковість
Задача (node)	estimated_runtime	число (сек)	Оцінка тривалості виконання задачі на типовому ресурсі	обов'язк.
Задача (node)	resource_requirements	структура	Вимоги до ресурсу: CPU/GPU, пам'ять, тип вузла	обов'язк.
Задача (node)	priority_class	категорія/число	Пріоритет / клас сервісу	рекоменд.
Задача (node)	SLO_deadline	число (сек)	Дедлайн / SLO (service-level objective) для задачі	рекоменд.

Продовження таблиці 3.2

Сутність	Поле (назва)	Тип/приклад	Зміст (що означає)	Обов'язковість
Задача (node)	DAG_in_edges / DAG_out_edges	списки	Вхідні/вихідні залежності DAG (directed acyclic graph)	обов'язк.
Задача (node)	RBAC_context	структура	Контекст доступу: роль, домен, дозволені цілі (RBAC)	рекоменд.
Залежність (edge)	data_size	число (MB/GB)	Обсяг міжетапних передавань	обов'язк.
Залежність (edge)	communication_cost	число (сек)	Вартість/затримка передачі між етапами (мережа + трансфер)	обов'язк.
GERT-гілка (edge)	branch_probability	[0..1]	Ймовірність гілки GERT (Graphical Evaluation and Review Technique)	опц.
GERT-повтор (edge)	loop_probability	[0..1]	Ймовірність повторного проходу (loop) у GERT	опц.

Невизначеність метаданих моделюється як мультиплікативний шум оцінок тривалостей і комунікаційних витрат $t \sim = t(1 + \varepsilon)$, де $\varepsilon \sim U[-\alpha, \alpha]$, $\alpha \in \{0.10, 0.20\}$. Додатково, для перевірки робастності у стрес-сценаріях застосовується зсув розподілів параметрів зі збереженням перших моментів (moment matching), зокрема заміна Gamma $\leftrightarrow$ Weibull.

Критичний шлях у DAG-графі визначається як найдовший шлях у графі від початкового до кінцевого вузла, що задає мінімально можливий час виконання всієї задачі:

$$M = \sum W_{n_i} , \quad (3.1)$$

де W_{n_i} – час виконання вузла n_i .

Якщо задачі можуть мати альтернативні варіанти виконання, згідно з технологією моделювання GERT, їхній внесок у загальний час виконання обчислюється з урахуванням ймовірності:

$$M = \sum (W_{n_i} P_{n_i}), \quad (3.2)$$

де P_{n_i} – ймовірність виконання вузла n_i .

Для управління процесом планування використовується матриця ресурсів, яка містить інформацію про доступні обчислювальні вузли, їхню продуктивність та топологію мережі. Кожен ресурс характеризується потужністю C_k та затримкою передачі між вузлами $d(i,j)$, що формалізується у вигляді матриці:

$$R_{i,j} = \begin{cases} 0, & \text{якщо } i = j \text{ (локальний процесор)} \\ d(i,j), & \text{якщо } i \neq j \text{ (відстань між ресурсами)} \end{cases} \quad (3.3)$$

Додатково вводиться ймовірність успішного виконання задачі на конкретному обчислювальному вузлі:

$$P_{exec}(n_i, C_k) = \frac{1}{1 + e^{-\lambda(C_k - C_{avg})}}, \quad (3.4)$$

де C_{avg} – середня потужність усіх доступних обчислювальних ресурсів,
 λ – параметр чутливості системи до продуктивності окремого вузла.

Ця формула дозволяє оцінювати, наскільки ефективно задача може виконуватися на певному процесорі, і мінімізувати ризики перевантаження окремих ресурсів.

Окрім цього, планування задач передбачає формування матриці характеристик задач, що включає критичний шлях, рівневе розбиття та щільність графа. Визначення рівнів виконання задач дозволяє ідентифікувати вузли, які можуть бути виконані паралельно, що підвищує ефективність використання ресурсів. Ширина рівня визначається як:

$$w = \max_{1 \leq k \leq h} \sum_{v \in V_k} W_v, \quad (3.5)$$

де V_k – множина задач, що належать до одного рівня стратифікації.

Час завершення задачі (TCT) залежить від стартового часу, часу виконання підзадач та можливих комунікаційних затримок:

$$TCT = St(T_n) + \sum_{i=1}^N (T_{TaskExecution} \times P_{n_i}). \quad (3.6)$$

Вартість комунікації між підзадачами оцінюється як обсяг даних, що передаються між вузлами під час виконання графа:

$$CC_{\tau_{i,j \rightarrow \tau_{i,k}}}. \quad (3.7)$$

Розподіл задач у HDCS передбачає знаходження оптимального процесора або кластера для кожної підзадачі, з урахуванням обчислювальної потужності, завантаженості та мережових затримок.

Якість розподілу ресурсів визначається відсотком зайнятих процесорів відносно загальної кількості доступних ресурсів:

$$Q_a = \left(\frac{\sum_{i=0}^m \forall \pi_m, \text{ що містить } \eta_i}{|\pi|} \right) \times 100. \quad (3.8)$$

Фінальний етап формалізації полягає у визначенні матриці розподілу ресурсів, що описує відповідність між задачами та обчислювальними вузлами. Загальний час виконання всіх задач у системі визначається як:

$$M_s = \max_{n_i \in N} \left(\sum_{C_k} \left(W_{n_i} \times P_{exec}(n_i, C_k) \right) \right), \quad (3.9)$$

де кожна задача призначається на такий вузол, який мінімізує M_s і одночасно забезпечує збалансоване навантаження на всю систему.

Для оптимізації процесу планування використовується алгоритм Univariate Marginal Distribution Algorithm (UMDA), який аналізує ймовірнісні закономірності в можливих розподілах і обирає найбільш ефективні варіанти. Формально він записується як:

$$p_l(x) = p(x \mid DSe_{l-1}) = \prod_{i=1}^n p_l(x_i), \quad (3.10)$$

де $p_l(x)$ – ймовірність вибору певного розподілу задач при оптимізації планування.

Завдяки цьому розширенню традиційний DAG-підхід доповнюється стохастичними моделями, що покращує адаптацію до змінних умов роботи системи.

Цей висновок може бути доведений за допомогою демонстрації результатів дослідження на обраному прикладі.

На рис. 3.1. наведено приклад графа, як елементу застосування розробленої моделі для планування задач у HDCS.

Рис. 3.1. ілюструє граф задач у HDCS, де використовується комбінований підхід DAG та GERT для адаптивного планування задач. У цій моделі кожен вузол представляє окрему задачу з певним часом виконання, що позначено всередині кожного елемента. Орієнтовані ребра визначають залежності між задачами, встановлюючи порядок їх виконання, тоді як ймовірності переходів між ними відображають стохастичні залежності, що враховують мінливість обчислювального середовища. Такий підхід дозволяє враховувати можливі затримки через навантаження на сервери, ймовірність успішного виконання завдань у нестабільних умовах, а також альтернативні маршрути виконання задач.

Запропонована модель дозволяє реалізувати гнучке адаптивне планування, завдяки чому система здатна прогнозувати альтернативні шляхи виконання задач у випадку невизначеності. Якщо ймовірність переходу між

задачами низька, алгоритм може змінювати маршрут, щоб мінімізувати ризики затримок та підвищити ефективність планування.

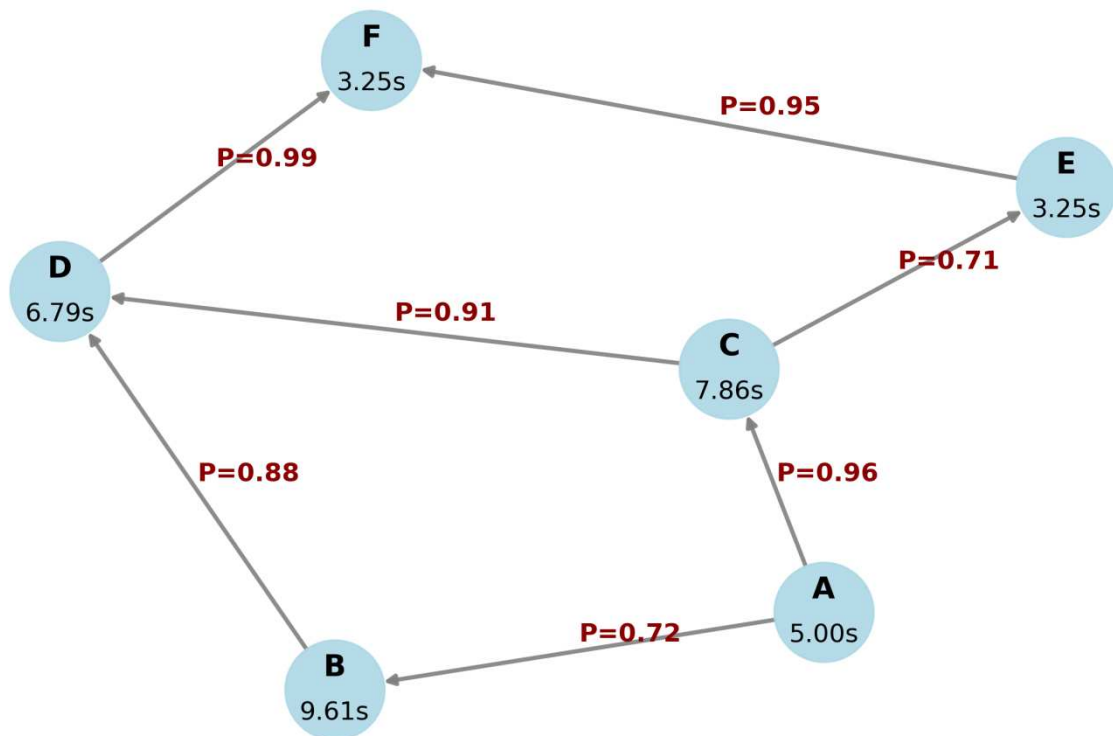


Рис. 3.1. Граф задач у розподіленій гетерогенній обчислювальній системі

Додатковою перевагою є оптимізація балансування навантаження, оскільки класичні DAG-графи можуть створювати перевантажені вузли через жорстку фіксацію шляхів виконання. У запропонованій моделі розподіл задач між серверами відбувається з урахуванням імовірнісного аналізу їхнього поточного стану та доступних ресурсів, що дозволяє уникнути вузьких місць і підвищити продуктивність обчислювальної системи. Крім того, використання GERT-графів зменшує ризики відмов, оскільки система може заздалегідь передбачати альтернативні шляхи виконання, якщо один із переходів має низьку ймовірність, тим самим забезпечуючи стабільність та стійкість HDCS до можливих збоїв.

На рис. 3.2. наведено порівнювальні гістограми загального часу виконання та балансування ресурсів для різних досліджуваних моделей.

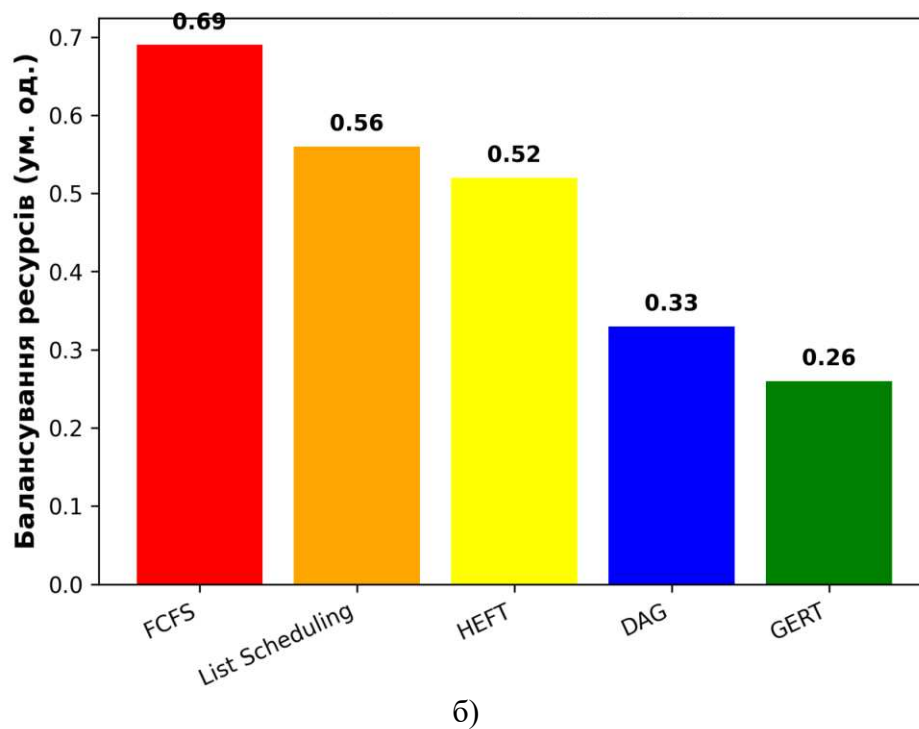
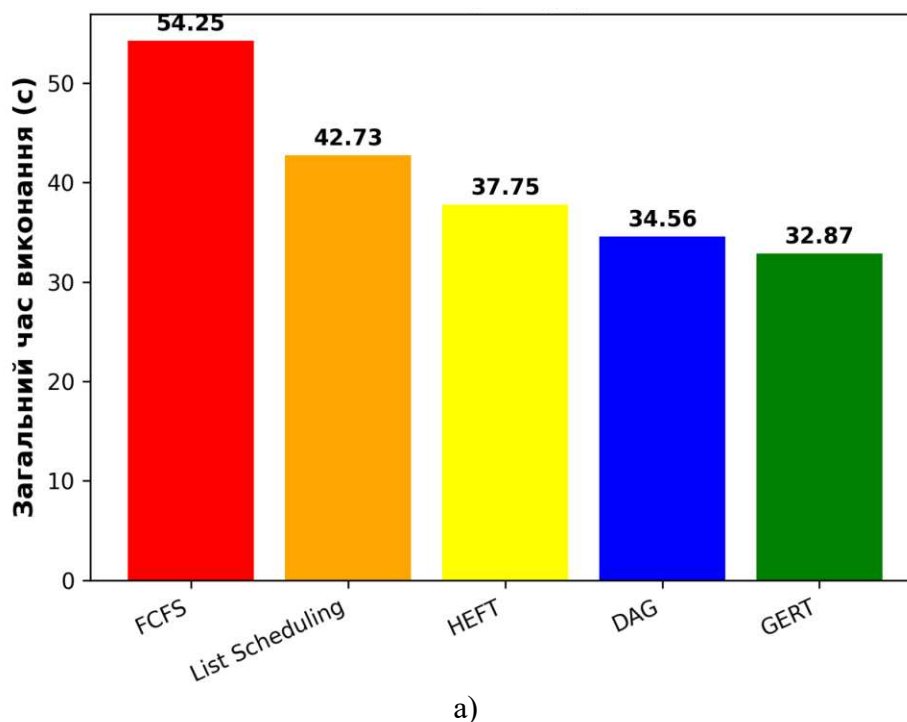


Рис. 3.2. Порівнювальні гістограми загального часу виконання та балансування ресурсів для різних досліджуваних моделей

Отримані результати демонструють ефективність різних методів планування задач у HDCS з точки зору загального часу виконання (makespan) та балансування обчислювальних ресурсів.

На першому графіку (рис. 3.2.а) представлено порівняння загального часу виконання для п'яти підходів: FCFS, List Scheduling, HEFT, DAG та GERT. Як видно з гістограми, найгірші результати демонструє метод FCFS, оскільки він не враховує специфіку обчислювального середовища та виконує задачі у порядку їх надходження, що призводить до значних затримок. List Scheduling показує покращені результати, оскільки враховує певний рівень пріоритетності задач, проте не забезпечує оптимального балансування. HEFT (Heterogeneous Earliest Finish Time) демонструє ще кращу продуктивність, оскільки враховує гетерогенність середовища та дозволяє більш ефективно розподіляти задачі між ресурсами. Використання DAG-моделі дозволяє суттєво зменшити makespan за рахунок оптимізованого графового планування, однак найкращий результат показує GERT-модель, яка використовує ймовірнісні залежності та адаптивний розподіл задач, що дозволяє додатково мінімізувати загальний час виконання.

Другий графік (рис. 3.2.б) демонструє оптимізацію балансування ресурсів у різних методах. Чим нижчий показник на гістограмі, тим краще балансування, оскільки воно означає рівномірний розподіл навантаження між обчислювальними ресурсами. Метод FCFS має найгірші результати, оскільки він не здійснює оптимізації розподілу навантаження. List Scheduling та HEFT показують поступове покращення балансування, оскільки враховують розподіл задач відповідно до доступних потужностей вузлів. DAG-модель суттєво покращує цей показник, однак найкращих результатів знову досягає GERT-модель, що підтверджує її ефективність у балансуванні ресурсів шляхом адаптивного управління задачами та використання ймовірнісних залежностей.

Таким чином, результати дослідження показують, що використання GERT-моделі у HDCS дозволяє значно зменшити загальний час виконання задач, покращити балансування ресурсів та підвищити гнучкість планування у змінних умовах середовища. Отримані дані підтверджують переваги інтеграції стохастичних підходів у процес планування задач у гетерогенних розподілених системах.

Запропонована модель DAG + GERT дозволяє ефективно виконувати планування задач у розподілених обчислювальних системах, враховуючи стохастичні фактори та динамічні зміни у доступності ресурсів. Використання цього підходу дає змогу адаптивно керувати розподілом задач та забезпечувати оптимальне балансування навантаження. Однак, з урахуванням зростаючої складності сучасних розподілених систем та потреби в автоматичному прийнятті рішень, постає питання подальшого вдосконалення механізмів планування.

Для подальшого підвищення ефективності процесу планування задач доцільним є використання методів штучного інтелекту. Інтеграція алгоритмів глибокого навчання та підкріплювального навчання дозволяє не лише прогнозувати майбутнє завантаження системи, але й здійснювати адаптивне керування розподілом задач у режимі реального часу. Це дає змогу підвищити точність вибору критичного шляху, оптимізувати балансування ресурсів та мінімізувати загальний час виконання задач.

3.4. Метод інтелектуального планування задач у розподілених інформаційних системах

Загальна структура. Запропонований метод планування задач у розподілених інформаційних системах базується на поєднанні стохастичного підходу DAG + GERT із методами штучного інтелекту для підвищення адаптивності, точності прогнозування навантаження та оптимального розподілу ресурсів. Основна ідея методу полягає в тому, щоб автоматизувати вибір критичного шляху та динамічний розподіл задач між обчислювальними вузлами, використовуючи алгоритми машинного навчання.

Метод складається з кількох ключових компонентів.

1. Прогнозування критичного шляху та навантаження на систему. Вхідні дані про поточний стан системи та характеристики задач аналізуються за допомогою графових нейронних мереж (Graph Neural Networks, GNN). Це дозволяє оцінити можливі варіанти виконання задач у DAG + GERT-графі та визначити найбільш оптимальний маршрут для мінімізації загального часу виконання (makespan).

2. Адаптивне прийняття рішень щодо виконання задач. У процесі роботи система може змінювати поточний план, якщо спостерігається перевантаження вузлів або зміна доступності ресурсів. Для цього застосовуються алгоритми підкріплювального навчання (Reinforcement Learning, RL), які дозволяють динамічно адаптувати розподіл задач у реальному часі.

3. Призначення процесорів задачам та оптимізація балансування ресурсів. Використання байєсівських мереж або генетичних алгоритмів допомагає передбачати найбільш ефективне призначення процесорів задачам із врахуванням поточного завантаження, прогнозованих затримок та комунікаційних витрат. Це дозволяє підвищити ефективність використання обчислювальних ресурсів.

4. Механізм самонавчання та вдосконалення. Система зберігає історичні дані про ефективність планування та використовує алгоритми глибокого навчання для поступового покращення своїх рішень. Чим більше даних накопичується, тим точніше система передбачає майбутнє навантаження та оптимізує критичний шлях виконання задач.

Перелік алгоритмів, що використовуються у запропонованому методі представлено в табл. 3.3.

Перший етап запропонованого методу інтелектуального планування задач у розподілених інформаційних системах було розглянуто при формалізації моделі. Тому розглянемо наступний етап прогнозування критичного шляху та навантаження на систему.

**Перелік алгоритмів, що використовуються у планування задач в
розподілених інформаційних системах**

Етап методу	Використовуваний алгоритм	Роль у процесі планування
Обробка вхідних даних та побудова DAG + GERT-графа	Алгоритм побудови DAG + GERT-графа (DFS/BFS для DAG, ймовірнісне моделювання для GERT)	Формування структури графа задач з ймовірнісними залежностями
Прогнозування критичного шляху	Graph Convolutional Network (GCN)	Визначення оптимального критичного шляху з урахуванням можливих сценаріїв виконання
Адаптивне керування задачами	Proximal Policy Optimization (PPO)	Адаптація планування у реальному часі залежно від доступності ресурсів
Призначення процесорів задачам	Bayesian Optimization	Оптимізація балансування ресурсів і мінімізація makespan

Прогнозування критичного шляху та навантаження на систему.

Для ефективного розподілу задач у HDCS необхідно точно визначати критичний шлях, що забезпечує мінімізацію загального часу виконання. Використання класичних методів, таких як аналіз DAG-графів, не дозволяє гнучко адаптуватися до динамічних змін у навантаженні та ресурсах системи. Тому доцільним є застосування Graph Attention Network (GAT), що дозволяє прогнозувати критичний шлях із урахуванням імовірнісних залежностей між задачами та мінливості ресурсного середовища.

GAT є розширенням класичних Graph Convolutional Network (GCN) і використовує механізм самоприділеної уваги (self-attention), що дозволяє визначати, які зв'язки між задачами є більш важливими для кінцевого результату. Це особливо актуально у випадку DAG + GERT-графів, де ймовірнісні переходи можуть суттєво змінювати загальний час виконання. Основне рівняння, що визначає вагові коефіцієнти між задачами, має вигляд:

$$\alpha_{ij} = \left(\frac{e^{(LeakyReLU(a^T [w_{h_i} \| w_{h_j}]))}}{\sum_{k \in N(i)} e^{(LeakyReLU(a^T [w_{h_i} \| w_{h_k}]))}} \right), \quad (3.11)$$

де α_{ij} – коефіцієнт уваги між задачею та її сусідом ;

W – навчальні ваги моделі;

a – вектор уваги, що визначає важливість сусідніх задач;

$\parallel$ – операція конкатенації ознак.

Такий підхід дозволяє кожному вузлу графа отримувати інформацію про сусідні вузли з урахуванням їхньої важливості, що покращує процес прогнозування критичного шляху. На відміну від GCN, що застосовує однакові ваги до всіх сусідів, GAT дає змогу визначити ключові вузли у виконанні задач та оптимізувати їхній розподіл.

Процес прогнозування критичного шляху у GAT включає кілька важливих етапів. Спочатку модель отримує на вхід граф задач у вигляді DAG + GERT, де кожна вершина містить вхідні ознаки: час виконання задачі, рівень завантаженості ресурсів та ймовірність переходу до наступного стану.

На цьому етапі кожен вузол у графі отримує вектор початкових ознак, що включає такі характеристики як:

- час виконання задачі t_i – час, необхідний для завершення задачі i .
- ймовірність переходу P_{ij} – ймовірність того, що виконання задачі i перейде до задачі j .
- завантаженість ресурсів R_i – рівень використання обчислювальних ресурсів для виконання задачі i .
- ступінь зв'язності d_i – кількість вхідних і вихідних зв'язків вузла, що вказує на його важливість у графі.

Ці ознаки формують початкову матрицю характеристик вузлів $H^{(0)}$:

$$H^{(0)} = \begin{bmatrix} t_1 & P_{12} & R_1 & d_1 \\ t_2 & P_{23} & R_2 & d_2 \\ t_3 & P_{34} & R_3 & d_3 \\ \vdots & \vdots & \vdots & \vdots \\ t_n & P_{n-1,n} & R_n & d_n \end{bmatrix} \quad 3.12$$

Ці параметри формують початковий стан графа, що є базовими даними для подальшого навчання нейромережі.

Далі GAT виконує багатоетапний аналіз взаємодії вузлів, використовуючи механізм уваги для визначення найважливіших зв'язків. На першому шарі уваги кожен вузол отримує інформацію від своїх безпосередніх сусідів, визначаючи, які задачі мають більший вплив на загальний результат.

Важливість вузла j для вузла i визначається через функцію уваги:

$$e_{ij} = \left(\text{LeakyReLU} \left(a^T \left[W_{h_i} \parallel W_{h_j} \right] \right) \right) \quad (3.13)$$

де e_{ij} – початковий показник важливості сусіда j для вузла i .

Далі ці значення нормалізуються через softmax, щоб отримати коефіцієнти уваги:

$$\alpha_{ij} = \left(\frac{e^{(e_{ij})}}{\sum_{k \in N(i)} e^{(e_{ik})}} \right). \quad (3.14)$$

Ці коефіцієнти відображають ймовірність того, що інформація від сусіднього вузла буде важливою для прогнозування критичного шляху.

Цей процес повторюється на наступних шарах, що дозволяє поступово формувати глобальне уявлення про структуру графа. Оскільки вага зв'язків визначається динамічно, модель адаптується до змінних умов виконання задач, дозволяючи ідентифікувати вузли, що є критичними для ефективного розподілу ресурсів.

На виході мережа формує матрицю пріоритетів, яка відображає, які задачі мають бути виконані першочергово для мінімізації загального часу виконання. Завдяки цьому стає можливим автоматичне ранжування можливих маршрутів виконання, що суттєво покращує точність прогнозування критичного шляху. Крім того, отримані ваги зв'язків можуть бути використані для подальшої

оптимізації планування, зокрема для коригування розподілу ресурсів у реальному часі.

У фіналі передбачено інтеграцію прогнозованого критичного шляху в систему адаптивного управління задачами. Використовуючи підхід підкріплювального навчання, система аналізує отримані результати та оптимізує балансування ресурсів залежно від змін у навантаженні. Таким чином, запропонований метод дозволяє підвищити ефективність розподілу задач, зменшуючи загальний час виконання та забезпечуючи адаптивність до змін у системі.

Візуалізація роботи GAT показує, як модель ітеративно аналізує зв'язки між задачами, послідовно покращуючи прогнозування критичного шляху у HDCS. Так на рис. 3.3. представлено граф задач у розподіленій системі, де кожен вузол відповідає окремій задачі. Вузли містять два значення: Real – реальний час виконання задачі, та Pred – прогнозований час виконання, отриманий за допомогою моделі GAT.

Дивлячись на результати Ключових спостережень можна відмітити наступне. Для задачі з реальним часом виконання 2.00 у.о модель передбачила час 1.77 у.о, що свідчить про невелику похибку. Для задачі з реальним часом 3.00 у.о прогнозоване значення склало 3.43 у.о, що також демонструє високу точність. Для задачі з реальним часом 4.00 прогнозоване значення 3.47 у.о є близьким до реального, що підтверджує здатність моделі враховувати залежності між задачами. Для задачі з реальним часом 5.00 у.о прогнозоване значення 4.83 у.о також є досить точним. Такі результати в цілому підтверджують гіпотезу про високу точність прогнозування.

Крім того спостерігається наступна загальна тенденція. Модель добре справляється з прогнозуванням часу виконання задач, що підтверджує її ефективність у врахуванні як детермінованих, так і ймовірнісних залежностей у графі задач.

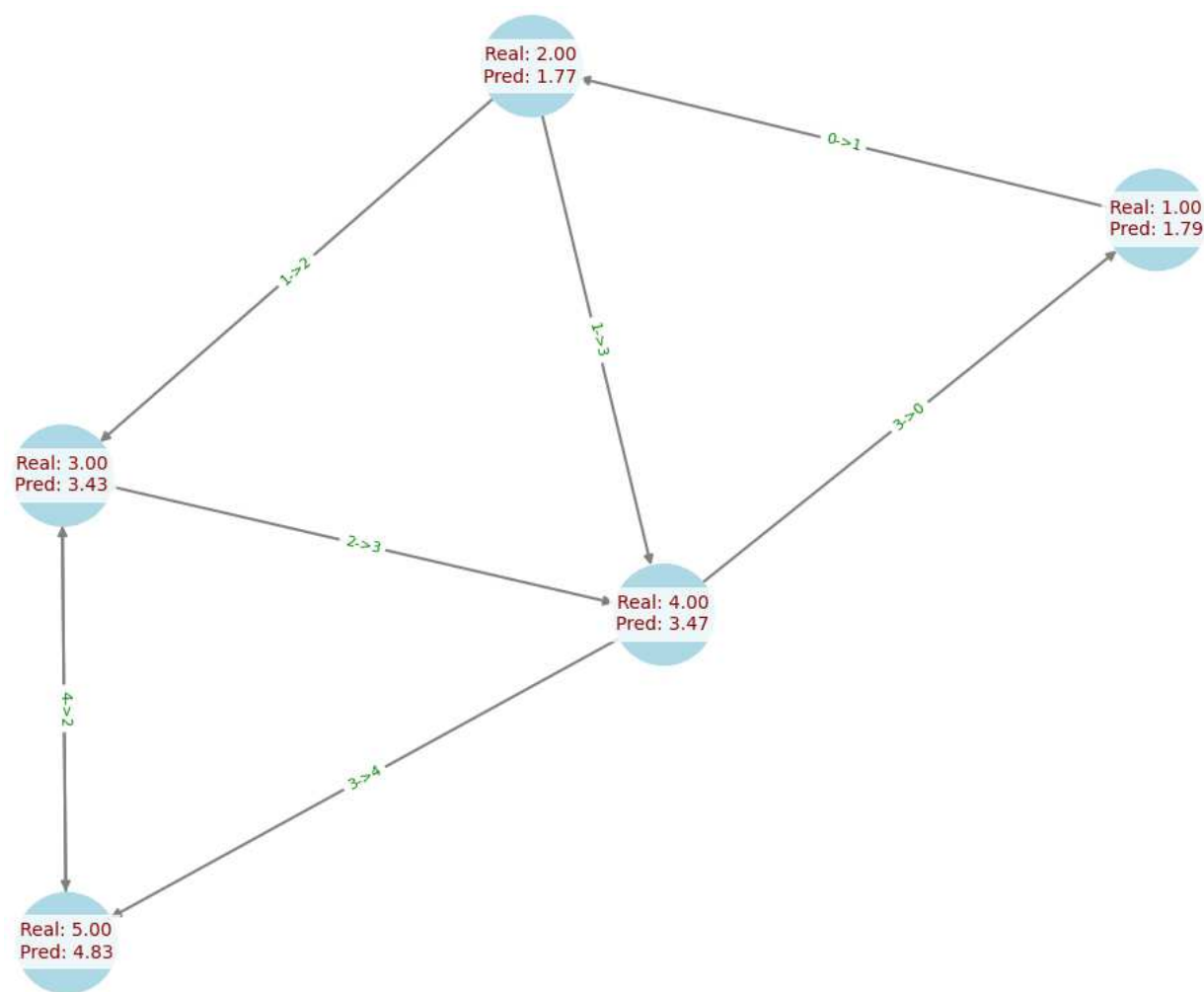


Рис. 3.3. Граф задач у розподіленій системі, де кожен вузол відповідає окремій задачі.

Графічне представлення дозволяє наочно оцінити, як модель розподіляє навантаження між вузлами та враховує залежності між задачами. Напрямки ребер графа вказують на порядок виконання задач, що є важливим для планування в розподілених системах.

В табл. 3.4 представлено результати процесу навчання GAT.

Процес навчання моделі GAT демонструє стабільне зменшення значення функції втрат (MSE) з кожним кроком навчання. На початковому етапі (епоха 0) значення втрат становило 2.2169, що є досить високим показником, оскільки модель ще не навчилася враховувати залежності між вузлами графа. Проте вже після 20 епох значення втрат зменшилося до 0.8773, що свідчить про початкове навчання моделі та її здатність краще прогнозувати навантаження на вузли.

Результати процесу навчання GAT

Epoch	Loss
0	2.2169222831726074
20	0.8772878646850586
40	0.555362343788147
60	0.49738389253616333
80	0.4462957978248596
100	0.39999741315841675
120	0.35727459192276
140	0.3189087212085724
160	0.2854723334312439
180	0.25711387395858765

На епосі 40 значення втрат знизилося до 0.5554, а до епохи 60 воно досягло 0.4974. Це свідчить про те, що модель продовжує покращувати свої прогнози, враховуючи складні залежності між задачами в графі. На епосі 80 втрати зменшилися до 0.4463, а до епохи 100 – до 0.4000, що підтверджує ефективність використання GAT для задач планування в розподілених системах.

На завершальних етапах навчання (епохи 120–180) значення втрат продовжувало зменшуватися, досягнувши 0.2571 на епосі 180. Це свідчить про те, що модель досягла високої точності прогнозування та здатна ефективно враховувати як детерміновані, так і ймовірнісні залежності між задачами.

Адаптивне керування задачами.

Адаптивне керування задачами є одним із ключових етапів у процесі оптимального планування виконання задач у гетерогенних розподілених інформаційних системах. Використання традиційних підходів, таких як статичне призначення задач за алгоритмами DAG або стандартними евристичними, часто призводить до неефективного розподілу ресурсів та підвищених витрат на їх використання. Це пов'язано з тим, що динамічні зміни у навантаженні системи, варіативність продуктивності обчислювальних ресурсів і невизначеність часу виконання окремих задач можуть суттєво впливати на ефективність розподілу обчислень.

Щоб вирішити ці проблеми, ми використовуємо метод адаптивного керування задачами на основі Proximal Policy Optimization (PPO) – алгоритму глибокого підкріплювального навчання (Reinforcement Learning, RL), який дозволяє навчати агента для прийняття оптимальних рішень у динамічному середовищі [25]. Використання PPO у нашій системі дозволяє досягти гнучкого керування розподілом задач, забезпечуючи оптимальне використання ресурсів та зменшуючи makespan (загальний час виконання). У цьому контексті PPO розглядається як інструмент інтелектуального планування для управління навантаженням у динамічному середовищі.

PPO є розвитком класичних методів підкріплювального навчання, таких як Policy Gradient, і відрізняється високою стабільністю та ефективністю у великих системах. Основна перевага PPO полягає в тому, що він обмежує крок оновлення політики, що запобігає різким змінам стратегії та забезпечує плавне навчання. Це особливо важливо в умовах динамічних змін, характерних для розподілених систем, де доступність ресурсів і навантаження можуть змінюватися швидко та непередбачувано.

Основна ідея полягає у тому, що система виконує навчання агента, який приймає рішення про те, яка задача має бути виконана у поточний момент, виходячи з її контекстної інформації, стану ресурсів та прогнозованого критичного шляху, який визначає GAT. Цей підхід дозволяє зробити планування не просто статичним, а адаптивним до змінних умов роботи системи.

Процес керування задачами у PPO поділяється на кілька логічних етапів. Спочатку формується середовище взаємодії агента – це граф задач, представлений у форматі DAG + GERT, де кожен вузол містить інформацію про час виконання, ймовірність переходу до наступних задач, рівень завантаженості ресурсів та пріоритети. У кожен момент часу агент PPO отримує інформацію про поточний стан середовища та приймає рішення щодо подальшого виконання задач.

Агент PPO працює у середовищі за такою схемою.

1. Отримання стану середовища – модель отримує інформацію про поточний стан розподілу задач у графі, завантаженість обчислювальних вузлів та поточний makespan.

2. Обчислення ймовірностей виконання наступних задач – на основі поточних характеристик агент РРО прогнозує, які задачі слід запустити наступними, щоб мінімізувати makespan та забезпечити рівномірний розподіл навантаження.

3. Прийняття рішення про запуск задачі – модель обирає, яка задача буде виконана наступною, з урахуванням оптимального використання ресурсів та можливих затримок у мережі.

4. Оновлення політики на основі отриманого досвіду – алгоритм РРО аналізує результати виконання задач та оновлює стратегію прийняття рішень на основі отриманого зворотного зв'язку.

В основі роботи РРО лежить максимізація функції корисності, яка визначається як:

$$L^{CLIP}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)], \quad (3.15)$$

де $r_t(\theta)$ – відношення нової політики до старої;

$\hat{A}_t$ – оцінка переваги (advantage) для кроку t ;

ϵ – параметр, що обмежує крок оновлення політики.

Ця формула дозволяє агенту РРО плавно оновлювати свою політику, що робить процес прийняття рішень стабільним та ефективним.

Однією з головних задач адаптивного керування є оптимізація використання ресурсів у реальному часі. Процес адаптивного керування включає кілька ключових етапів.

1. Визначення стану системи. На кожному кроці система аналізує поточний стан, який включає:

– завантаженість обчислювальних вузлів (агент визначає, чи є вузол перевантаженим або недовантаженим);

- час виконання поточних задач (аналізується, скільки часу займе виконання кожної задачі залежно від поточного навантаження на систему);
- ймовірність переходу між задачами (згідно з GERT-графом) (якщо певний вузол має високу комунікаційну затримку, його можна виключити з критичного шляху).

2. Дії агента. Агент (алгоритм PPO) приймає рішення щодо розподілу задач між вузлами, враховуючи поточний стан системи. Дії можуть включати:

- перерозподіл задач між вузлами;
- зміну пріоритетів виконання задач;
- вибір альтернативних маршрутів виконання.

3. Оцінка результатів. Після виконання дій система оцінює ефективність розподілу задач на основі таких показників:

- загальний час виконання (makespan);
- рівномірність розподілу навантаження;
- кількість відмов або затримок.

4. Оновлення політики. На основі отриманих результатів PPO оновлює стратегію розподілу задач, максимізуючи очікуваний виграш (reward), який визначається як зменшення загального часу виконання та покращення балансування навантаження.

Проведені експерименти з навчанням моделі PPO у спеціально розробленому середовищі розподілу задач показали результати, що наведені на рис. 3.4.

Протягом навчання було зафіксовано високу продуктивність моделі з точки зору швидкості обчислень.

Середня кількість кадрів за секунду (FPS) коливалася в межах 700–1200, що свідчить про ефективну реалізацію середовища та алгоритму навчання на CPU. Загальний час навчання складав 13 секунд для 10240 кроків, що демонструє здатність моделі швидко адаптуватися до динаміки середовища.

Процес навчання характеризувався зменшенням загальних втрат (loss) з 255 на початкових етапах до 24 після 40 оновлень.

time/ fps iterations time_elapsed total_timesteps	1193 1 1 2048
time/ fps iterations time_elapsed total_timesteps	714 2 5 4096
train/ approx_kl clip_fraction clip_range entropy_loss explained_variance learning_rate loss n_updates policy_gradient_loss value_loss	0.011848826 0.179 0.2 -1.09 0.0232 0.0003 255 10 -0.0294 631
time/ fps iterations time_elapsed total_timesteps	706 3 8 6144
train/ approx_kl clip_fraction clip_range entropy_loss explained_variance learning_rate loss n_updates policy_gradient_loss value_loss	0.012527661 0.189 0.2 -1.06 -0.0348 0.0003 129 20 -0.0273 332
time/ fps iterations time_elapsed total_timesteps	778 4 10 8192
train/ approx_kl clip_fraction clip_range entropy_loss explained_variance learning_rate loss n_updates policy_gradient_loss value_loss	0.013267323 0.207 0.2 -1.02 -0.0102 0.0003 107 30 -0.0278 207
time/ fps iterations time_elapsed total_timesteps	768 5 13 10240
train/ approx_kl clip_fraction clip_range entropy_loss explained_variance learning_rate loss n_updates policy_gradient_loss value_loss	0.018032053 0.229 0.2 -0.966 0.31 0.0003 24 40 -0.0298 71

Рис. 3.4. Результати дослідження етапу адаптивного керування задачами

Це свідчить про стабільне покращення продуктивності моделі. Крім того, спостерігалося зменшення втрат, пов'язаних із функцією значення (value loss), з 631 до 71, що вказує на покращення здатності моделі прогнозувати винагороди.

Метрика приблизної дивергенції Кульбака-Лейблера (approx KL) демонструвала зростання з 0.0118 до 0.0180, що свідчить про активну зміну політики під час навчання. Це підтверджується також збільшенням частки

обрізаних оновлень (clip fraction) з 0.179 до 0.229, що є очікуваним ефектом при використанні обрізаного градієнта в алгоритмі PPO.

Ентропія політики, яка характеризує рівень невизначеності дій моделі, зменшилася з -1.09 до -0.966. Це свідчить про те, що модель стає більш впевненою у своїх діях, що є нормальним явищем під час навчання.

Метрика поясненої дисперсії (explained variance) демонструвала значні коливання, змінюючись від 0.0232 до -0.0348, а потім зростаючи до 0.31. Це може свідчити про те, що на початкових етапах навчання модель ще не досягла стабільного стану, але згодом її здатність прогнозувати винагороди покращилася.

Для оцінки ефективності моделі було проведено візуалізацію розподілу навантаження між вузлами. Результати оцінки наведено на рис. 3.5.

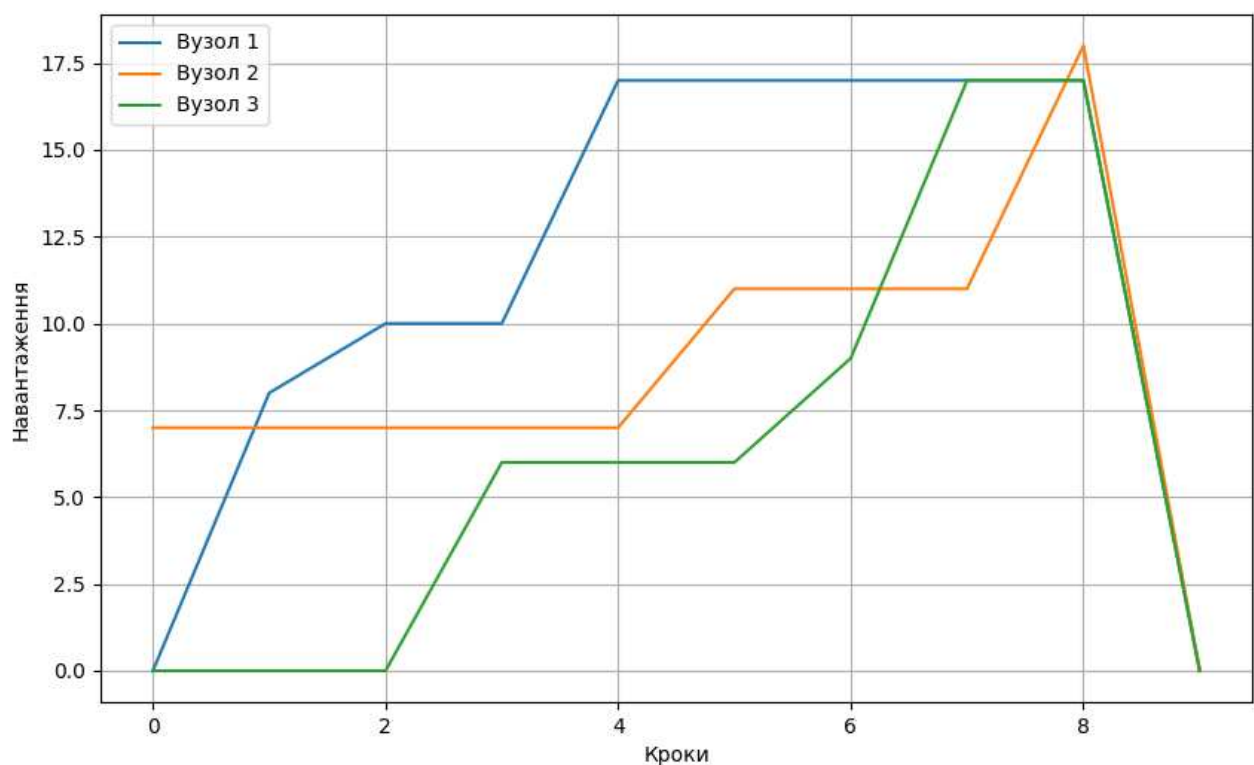


Рис. 3.5. Графіки оцінки адаптивного розподілу навантаження між обчислювальними вузлами за допомогою PPO

Результати показали, що модель успішно мінімізує дисбаланс навантаження, що підтверджується зменшенням стандартного відхилення

навантаження між вузлами. Це свідчить про те, що модель ефективно вирішує задачу рівномірного розподілу задач.

Результати експерименту демонструють, що модель PPO ефективно навчається у середовищі розподілу задач, зменшуючи дисбаланс навантаження між вузлами. Зменшення загальних втрат, покращення прогнозування винагород та зниження ентропії політики свідчать про стабільне навчання моделі. Отримані результати підтверджують потенціал використання PPO для задач адаптивного керування навантаженням у розподілених системах.

Призначення процесорів задачам

На етапі призначення процесорів задачам виконується завдання знаходження призначення задач конкретним обчислювальним вузлам системи таким чином, щоб забезпечити балансування навантаження, мінімізацію загального часу виконання (makespan) та стабільність розподілу з урахуванням динаміки середовища.

Враховуючи потребу в адаптивності, ефективності та здатності працювати в умовах невизначеності, для реалізації призначення задач обрано алгоритм Байєсівської оптимізації (Bayesian Optimization). Такий підхід дозволяє моделювати функцію цільової продуктивності без її явного аналітичного вираження та приймати оптимальні рішення, ґрунтуючись на обмеженій кількості спостережень.

Основна ідея Байєсівської оптимізації полягає у використанні апроксимаційної моделі – зазвичай це гаусовий процес (Gaussian Process, GP) – для побудови оцінки функції втрат (або виграшу), що відповідає часу виконання задач або ступеню завантаження ресурсів. На кожній ітерації оптимізатор обирає нову конфігурацію призначення задач, яка має найкраще очікуване значення на основі поточного прогнозу.

Виконаємо формальну постановку завдання процесу призначення процесорів задачам наступним чином. Позначимо:

- $X \subset R^d$ – простір можливих розподілів задач;

- $f(x)$ – функція, яка повертає значення makespan для даного розподілу задач $x \in X$;
- $p(f|D)$ – апіорний розподіл функції $f(x)$, який будується на основі наявних даних $D = \{(x_i, f(x_i))\}_{i=1}^n$, де x_i – це розподіл задач, а $f(x_i)$ – відповідне значення makespan.

В цілому розроблений алгоритм складається з 4 наступних кроків.

1. Побудова апостеріорного розподілу. На цьому кроці використовуючи гаусівський процес (GP), алгоритм будує апостеріорний розподіл функції $f(x)$, який враховує всі наявні дані D .

2. Оптимізація функції очікуваного покращення (Expected Improvement, EI). Функція EI визначається як:

$$EI(x) = E[\max(f_{best} - f(x), 0)], \quad (3.16)$$

де f_{best} – найменше спостережене значення makespan на поточний момент.

EI оцінює, наскільки очікується покращення значення $f(x)$ при виборі нової точки x .

3. Вибір нової точки. Алгоритм вибирає точку x^* , яка максимізує EI:

$$x^* = \arg \max_{x \in X} EI(x). \quad (3.17)$$

Після вибору x^* , алгоритм оцінює значення $f(x^*)$ за допомогою імітаційного моделювання.

4. Оновлення даних. Нова точка $(x^*, f(x^*))$ додається до набору D , і процес повторюється. Цей ітеративний підхід дозволяє алгоритму поступово покращувати розподіл задач, зменшуючи загальний час виконання (makespan) та забезпечуючи збалансоване навантаження між вузлами системи.

У контексті HDCS розподіл задач залежить від великої кількості параметрів. Серед них можна виділити продуктивність процесорів (модель включає множину кластерів з різними характеристиками), час очікування задач у чергах, ймовірність переходів між задачами (визначені в GERT-моделі), пріоритет задач і ін.

Для роботи алгоритму Bayesian Optimization (BO) використовуються два основні типи вхідних даних.

1. Матриця задач T . Кожна задача в матриці T описується набором характеристик, які включають:

- час виконання – оцінка часу, необхідного для виконання задачі на певному типі ресурсу;
- рівень пріоритетності – важливість задачі, яка визначає її пріоритет у системі;
- ймовірність виконання – ймовірність успішного завершення задачі, що враховує можливі залежності та ризики.

2. Матриця ресурсів R . Матриця R містить інформацію про доступні ресурси системи, такі як:

- доступні процесори – перелік обчислювальних вузлів або процесорів, які можуть бути використані для виконання задач;
- потужність ресурсів – обчислювальна потужність кожного ресурсу (наприклад, у FLOPS);
- поточне навантаження – рівень завантаження ресурсів на момент розподілу задач.

BO ітеративно генерує конфігурації розподілу задач між ресурсами. На першому кроці алгоритму тестує згенеровані конфігурації за допомогою імітаційної оцінки або реального середовища (наприклад, заснованого на PPO). Після тестування оновлюється модель GP, яка використовується для прогнозування ефективності різних конфігурацій. На основі отриманих даних BO вибирає нові конфігурації, які максимізують очікуване покращення (Expected Improvement, EI).

Схема процесу призначення процесорів задачам в контексті планування задач в розподілених інформаційних системах наведена на рис. 3.6.

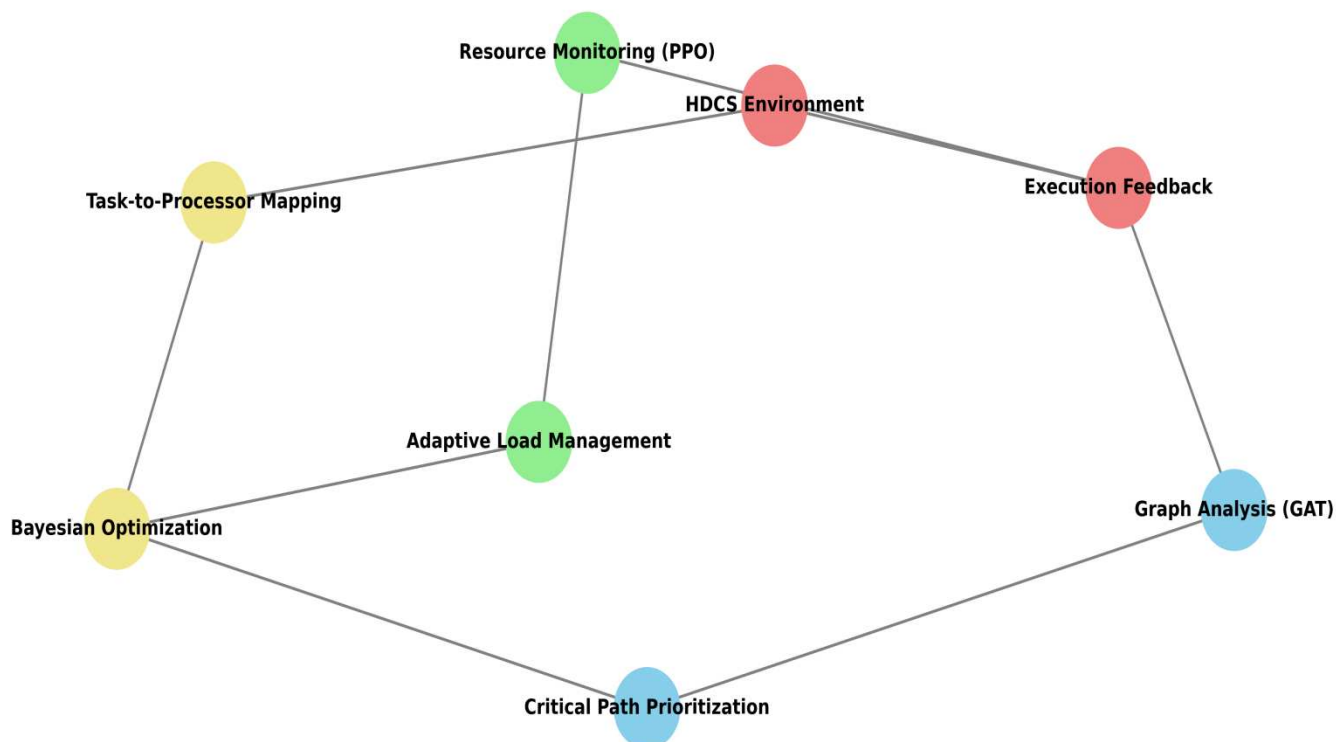


Рис. 3.6. Схема взаємодії компонентів методу планування задач

Для оцінки результатів етапу призначення процесорів задачам побудовано поверхню функції втрат відносно розподілу задач, побудовану GP, з позначенням точок спостережень та наступної обраної точки (рис. 3.7).

На графіку показано, як змінюється загальний час виконання (makespan) залежно від різних конфігурацій розподілу задач. Поверхня функції втрат, побудована за допомогою GP, демонструє, як алгоритм БО аналізує простір можливих розподілів задач. Це дозволяє візуально оцінити, як метод БО поступово наближається до оптимального розподілу, мінімізуючи втрати (makespan).

На графіку позначено точки, які відображають різні конфігурації розподілу задач, що були протестовані алгоритмом. Кожна точка відповідає певному розподілу задач між ресурсами та відображає відповідне значення makespan. Ці точки демонструють, як БО ітеративно вивчає простір можливих рішень, збираючи дані про ефективність різних конфігурацій.

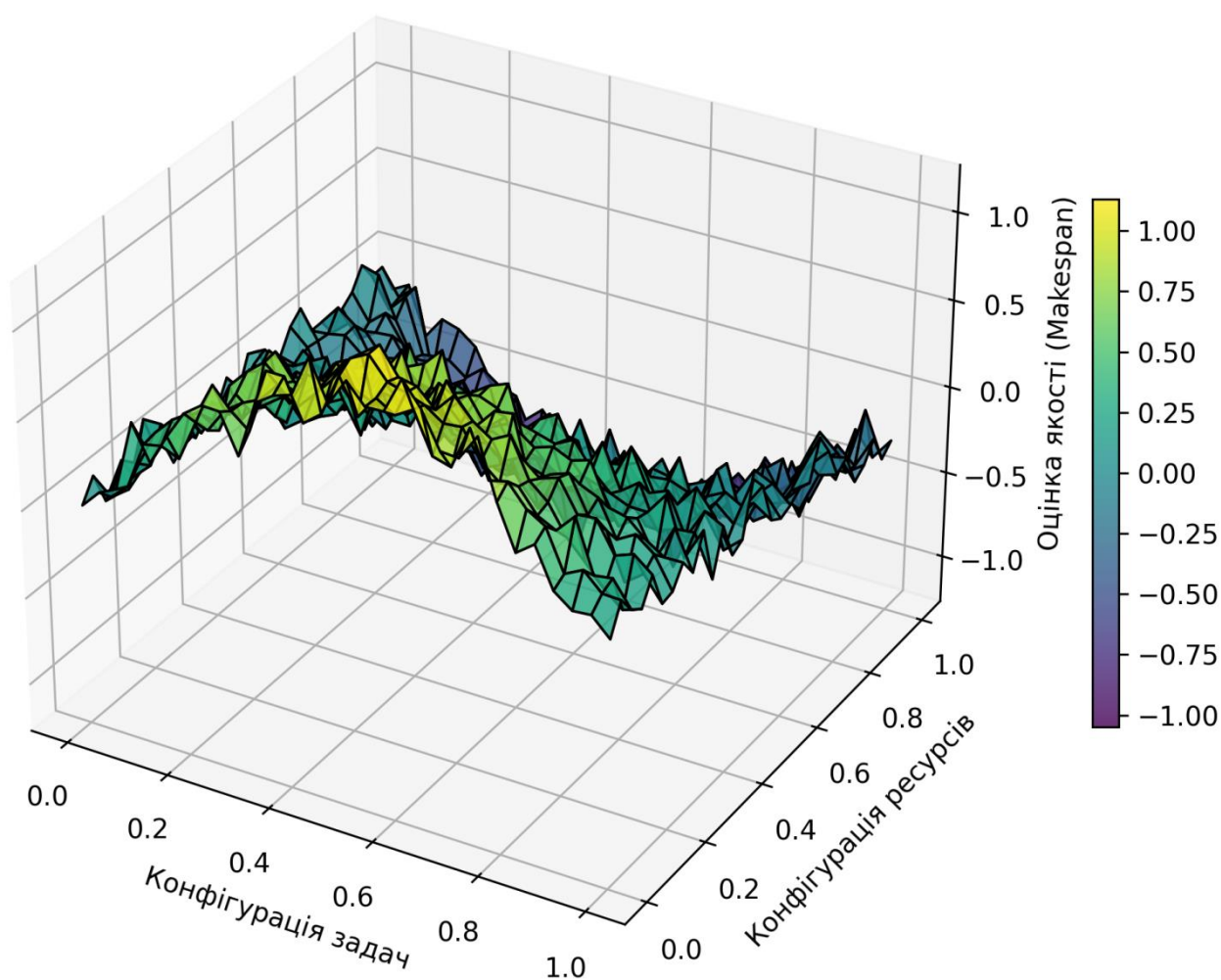


Рис. 3.7. Поверхня функції втрат відносно розподілу задач, побудовану GP, з позначенням точок спостережень та наступної обраної точки

На графіку також позначено точку, яка була обрана алгоритмом ВО для подальшого тестування. Ця точка вибирається на основі очікуваного покращення (Expected Improvement, EI), тобто вона має найбільший потенціал для покращення результату (зменшення makespan). Це показує, як ВО активно шукає оптимальні рішення, а не просто випадково перебирає варіанти.

ВО швидко знаходить області з низькими значеннями makespan, що дозволяє мінімізувати загальний час виконання. Метод враховує попередні спостереження (точки на графіку) для покращення прогнозів і вибору нових

конфігурацій. ВО прагне до глобального оптимуму, що видно з того, як він концентрується на областях з найнижчими значеннями функції втрат.

Обчислювальна складність та накладні витрати.

У межах запропонованого підходу розрізняємо офлайн-процедури навчання та налаштування та онлайн-прийняття рішень планувальником. Критично важливо, що у реальному (онлайн) контурі працюють лише легкі операції підготовки графа, інференс моделі та вибір допустимого призначення, тоді як навчання агентів і глобальне налаштування параметрів виносяться в офлайн або асинхронні цикли.

Побудова та оновлення графа задач. Нехай $|V|$ – кількість задач (вузлів графа), $|E|$ – кількість залежностей (ребер). Формування структур даних графа та базових індексів має часову складність $O(|V| + |E|)$ і пам'яткову складність $O(|V| + |E|)$. Обчислення простих структурних ознак (ступені входу/виходу, рівні, топологічні індекси) також виконується за $O(|V| + |E|)$. Оновлення метаданих у режимі ковзного вікна для активної підмножини задач масштабується лінійно від кількості оновлених записів і розглядається як $O(|V_a| + |E_a|)$ на крок, де $|V_a|$, $|E_a|$ – активні вузли/ребра.

Інференс GAT (graph attention network). Для L шарів і розмірності ембедінгів d типова оцінка часу інференсу має порядок $O(L(|E|d + |V|d^2))$, де доданок $|E|d$ відповідає агрегації повідомлень по ребрах, а $|V|d^2$ лінійним перетворенням ознак. У практичному режимі інференс виконується пакетно лише для активного підграфа готовності, тому фактичні витрати визначаються $|E_a|$ і $|V_a|$, а не повним графом.

Нехай R – кількість “готових” задач на фронті виконання, K – середня кількість кандидатних ресурсів для призначення однієї задачі. Побудова скорингових оцінок для пар «задача→ресурс» має складність $O(R \cdot K \cdot d)$ (скаляризація або MLP-оцінка стану). Формування множини допустимих кандидатів з урахуванням політик доступу та обмежень ресурсів виконується за $O(R \cdot K)$ при прямій перевірці; на практиці ця частина пришвидшується

кешуванням політик і попередньою індексацією ролей/доменів. У підсумку онлайн-контур масштабується квазі-лінійно від кількості активних кандидатів.

Баєсівська оптимізація використовується для періодичного підбору метабараметрів (ваги складових винагороди, пороги, температури відбору) і не входить у критичний шлях онлайн-планування. Її обчислювальна вартість розглядається як офлайн/асинхронна процедура налаштування, що запускається рідше, ніж кроки призначення задач.

Вимоги до обчислювальних ресурсів і межі застосовності. Онлайн-контур потребує пам'яті для зберігання ознак і ембедінгів (порядку $O(|V_a| \cdot d + |E_a|)$) та обчислень для інференсу і скорингу; для кластерних та edge-gateway сценаріїв це сумісно з мілісекундними бюджетами планування за рахунок роботи лише на активному підграфі та кешування політик. Натомість навчання РРО та оптимізація ВО доцільно виконувати на більш потужних ресурсах (сервер або хмара) та переносити в офлайн-контур. Для жорстко обмежених середовищ (граничні обчислення, Internet of Things, автономні системи) рекомендовано конфігурації зі зменшеними L і d , рідшими оновленнями ВО або використанням спрощених (евристичних) режимів як деградованого варіанту при дефіциті обчислень.

Результатом розроблення методу є інтегрований контур прийняття рішень, у якому графова модель задач використовується не лише для опису залежностей, але й для прогнозування критичного шляху, адаптивного вибору ресурсу та корекції розкладу за умов зміни стану середовища. На відміну від евристичних підходів, запропонований метод поєднує прогнозування, навчання політики планування та автоматизоване налаштування параметрів у межах єдиної процедури.

3.5. Дослідження ефективності запропонованого методу шляхом тестування порівняння продуктивності нового підходу з прототипами

Для оцінки ефективності запропонованого методу планування задач у розподілених інформаційних системах було проведено серію експериментів, які включали порівняння з іншими популярними підходами, такими як FCFS, List Scheduling, HEFT (Heterogeneous Earliest Finish Time), Greedy та Random. Основною метою експериментів було оцінити, наскільки запропонований метод покращує загальний час виконання (makespan) та балансування навантаження між ресурсами.

Експерименти проводилися на симуляторі розподілених систем, який імітував роботу гетерогенних кластерів з різною продуктивністю процесорів, мережевими затримками та динамічним навантаженням. Для тестування використовувалися різні набори задач, що включали як детерміновані, так і ймовірнісні залежності між задачами (DAG + GERT).

Для порівняння ефективності методів використовувалися наступні метрики:

- загальний час виконання (makespan);
- стандартне відхилення навантаження;
- час планування.

Результати експериментів показали, що запропонований метод значно перевершує традиційні підходи за всіма основними метриками.

Також у табл. 3.5. наведено приклади порівняння ефективності різних варіантів розподілу задач (BO, Random, Greedy) з відповідним makespan та стандартним відхиленням навантаження.

Отримані результати підтверджують перевагу використання Bayesian Optimization у складі запропонованого інтелектуального методу планування. Порівняно з Random середній makespan зменшено з 51,5 с до 35,2 с, тобто приблизно на 31,7%.

Приклади порівняння ефективності різних варіантів розподілу задач (ВО, Random, Greedy) з відповідним makespan та стандартним відхиленням навантаження

Метод	Середній Makespan (с)	Станд. відхилення навантаження
Bayesian Optimization	35.2	0.22
Greedy	43.7	0.31
Random	51.5	0.47

Порівняно з Greedy середній makespan зменшено з 43,7 с до 35,2 с, тобто приблизно на 19,5%. Одночасно стандартне відхилення навантаження зменшено з 0,47 до 0,22 порівняно з Random та з 0,31 до 0,22 порівняно з Greedy, що свідчить про покращення балансування навантаження між ресурсами.

Таблиця 3.5 містить порівняння ефективності різних методів розподілу задач: ВО, Greedy (жадібний алгоритм) та Random. У таблиці наведено середній час виконання (makespan) у секундах та стандартне відхилення навантаження для кожного методу. Результати показують, що метод ВО досягає найкращих результатів: середній makespan становить 35.2 секунди, а стандартне відхилення навантаження – 0.22. Це свідчить про те, що ВО ефективно мінімізує загальний час виконання та забезпечує рівномірний розподіл навантаження між ресурсами. Для порівняння, метод Greedy демонструє середній makespan 43.7 секунди зі стандартним відхиленням 0.31, а метод Random – 51.5 секунди зі стандартним відхиленням 0.47. Ці результати підтверджують переваги використання ВО для адаптивного планування задач у розподілених системах.

Незважаючи на складність алгоритмів ВО та GAT, час планування залишався прийнятним для реального використання, оскільки метод швидко знаходив оптимальні конфігурації завдяки ітеративному підходу.

Розрізняємо офлайн-навчання моделей та онлайн-прийняття рішення планувальником.

При описі побудови або оновлення графа задач для V задач і E залежностей формування структур даних має складність $O(|V| + |E|)$, обчислення базових ознак $O(|V|)$, а оновлення метаданих у режимі ковзного вікна $O(|V|)$ на крок.

При розрахунку інференсу GAT (graph attention network) для L шарів та розмірності ембедінгів d типовий порядок $O(L \cdot (|E| \cdot d + |V| \cdot d_2))$ (домінує обхід ребер і лінійні перетворення ознак). У практичному режимі інференс виконується пакетно для активної підмножини задач.

Вибір дії PPO виконується наступним чином. Один крок вибору призначення для активної множини кандидатів K має складність $O(K \cdot d)$ для обчислення скалярних оцінок, плюс витрати на побудову множини допустимих кандидатів.

Якщо для кожної задачі перевіряється допустимість на K кандидатах, то складність $O(K)$ на задачу, а в сумі на крок $O(|V_{act}| \cdot K)$ де $|V_{act}|$ – кількість задач, що реально плануються в даний момент. Це місце контролювано через кешування політик і попередню індексацію ролей/доменів.

ВО (Bayesian optimization) використовується як “повільний” контур налаштування, тому не входить у критичний шлях онлайн-планування; його обчислювальна вартість враховується як офлайн/періодична процедура.

Таким чином, критичний шлях онлайн-планувальника масштабовано лінійно за числом активних залежностей і числом кандидатів, а складніші процедури (навчання PPO та ВО) винесено в офлайн або асинхронні ітерації. Це узгоджується з експериментально виміряними накладними витратами планування.

Таким чином, експериментальні результати підтверджують ефективність запропонованого методу в умовах динамічного та гетерогенного середовища. Використання Bayesian Optimization у поєднанні з Graph Attention Network та Proximal Policy Optimization дозволило досягти значного покращення як у загальному часі виконання, так і в балансуванні навантаження. Це робить

запропонований метод перспективним для використання у сучасних розподілених системах, де важлива адаптивність та ефективність.

Висновки до розділу 3

У третьому розділі дисертаційної роботи розроблено інтелектуальний метод планування задач у гетерогенних розподілених інформаційних системах з урахуванням невизначеності метаданих. Отримані результати безпосередньо відповідають завданням дисертаційного дослідження щодо формалізації графової моделі задач, обґрунтування використання орієнтованого ациклічного графа для опису детермінованих залежностей, застосування GERT-моделі для стохастичних переходів, а також розроблення інтелектуального методу планування із використанням засобів машинного навчання, навчання з підкріпленням та оптимізації параметрів.

У межах розділу формалізовано модель планування задач у гетерогенних розподілених комп'ютерних системах. Встановлено, що класичне подання задач у вигляді орієнтованого ациклічного графа є достатнім для опису структурних залежностей і порядку виконання задач, однак не забезпечує повного врахування невизначеності часу виконання, ймовірнісних переходів, варіативності маршруту оброблення та змінного стану ресурсів. Тому запропоновано розширити традиційну DAG-модель шляхом використання GERT-графів, що дозволяє одночасно враховувати детерміновані залежності між задачами та стохастичні характеристики виконання.

Наукове значення запропонованої формалізації полягає в тому, що задача планування подається не як ізольований вибір ресурсу для окремої задачі, а як процес прийняття рішень у середовищі з графовими залежностями, комунікаційними витратами, невизначеністю метаданих і динамічною зміною навантаження. Такий підхід створює основу для врахування критичного шляху,

прогнозованого стану системи, обмежень ресурсів і необхідності адаптивного перепланування під час виконання задач.

Розроблено інтелектуальний метод планування задач, який базується на поєднанні DAG- та GERT-подання задач із сучасними алгоритмами штучного інтелекту: Graph Attention Network, Proximal Policy Optimization та Bayesian Optimization. Graph Attention Network використовується для аналізу графа задач і прогнозування значущості залежностей, що впливають на критичний шлях виконання. Proximal Policy Optimization застосовується для адаптивного керування процесом призначення задач на ресурси та корекції розкладу в умовах зміни стану середовища. Bayesian Optimization використовується для налаштування параметрів планування та вибору конфігурацій, які забезпечують компроміс між якістю розкладу й накладними витратами планування.

Відмінною особливістю розробленого методу є інтеграція трьох взаємопов'язаних контурів прийняття рішень: прогнозування параметрів виконання задач і критичного шляху, адаптивного призначення задач на ресурси та оптимізації параметрів планувальника. Це дозволяє зменшити вплив неповноти й неточності метаданих, підвищити якість розподілу задач між гетерогенними ресурсами та забезпечити динамічне коригування плану виконання при зміні навантаження або доступності обчислювальних вузлів.

У розділі обґрунтовано, що використання лише евристичних або статичних підходів до планування є недостатнім для гетерогенних розподілених інформаційних систем, оскільки такі підходи переважно орієнтовані на фіксовані або наперед відомі параметри задач і ресурсів. На відміну від них, запропонований метод використовує прогнозні оцінки, графову структуру задач, дані про стан системи та механізми навчання для вибору дій планувальника. Це забезпечує більшу адаптивність методу порівняно з традиційними підходами FCFS, List Scheduling, HEFT, Greedy та Random.

Проведене у розділі дослідження ефективності підтвердило переваги запропонованого підходу за основними метриками планування: загальним часом виконання задач, стандартним відхиленням навантаження та часом

планування. Для порівняння різних варіантів розподілу задач було використано Bayesian Optimization, Greedy та Random. Встановлено, що Bayesian Optimization забезпечує найкращий результат: середній makespan становить 35,2 с, а стандартне відхилення навантаження – 0,22. Для Greedy середній makespan становить 43,7 с, а стандартне відхилення навантаження – 0,31. Для Random середній makespan становить 51,5 с, а стандартне відхилення навантаження – 0,47.

Отримані числові результати свідчать, що використання Bayesian Optimization у складі запропонованого інтелектуального методу дозволяє зменшити середній час виконання задач на 16,3 с порівняно з Random та на 8,5 с порівняно з Greedy. У відносному вимірі це відповідає скороченню makespan приблизно на 31,7% порівняно з Random і приблизно на 19,5% порівняно з Greedy. Одночасно стандартне відхилення навантаження зменшується з 0,47 до 0,22 порівняно з Random та з 0,31 до 0,22 порівняно з Greedy, що підтверджує покращення балансування навантаження між ресурсами.

Таким чином, експериментальне порівняння в межах третього розділу підтвердило, що запропонований підхід забезпечує не лише зменшення загального часу виконання задач, але й більш рівномірний розподіл навантаження в гетерогенному середовищі. Це є важливим результатом, оскільки для розподілених інформаційних систем ефективність планування визначається не тільки мінімізацією makespan, а й здатністю уникати перевантаження окремих вузлів, зменшувати простой ресурсів і підтримувати стабільність виконання задач у динамічних умовах.

Результати розділу підтверджують наукову новизну дисертаційної роботи в частині розроблення методу інтелектуального планування задач у розподілених інформаційних системах в умовах невизначеності метаданих. Відмінність запропонованого методу полягає в інтеграції DAG+GERT-подання задач, графових нейромережових моделей, навчання з підкріпленням і баєсівської оптимізації параметрів. Практична реалізація такого підходу

дозволяє виконувати прогнозування критичного шляху, онлайн-перепланування та автоматизоване налаштування параметрів планування.

Отже, у третьому розділі отримано такі основні результати: формалізовано модель планування задач у гетерогенних розподілених комп'ютерних системах; обґрунтовано доцільність поєднання DAG і GERT для опису детермінованих та стохастичних залежностей; розроблено інтелектуальний метод планування задач на основі Graph Attention Network, Proximal Policy Optimization та Bayesian Optimization; виконано порівняльну оцінку методу з базовими підходами; підтверджено скорочення середнього makespan до 35,2 с і зменшення стандартного відхилення навантаження до 0,22. Отримані результати створюють теоретичну, алгоритмічну та методичну основу для подальшої експериментальної перевірки робастності, адаптивності й ефективності запропонованого методу, що виконується у четвертому розділі дисертаційної роботи.

Основні результати розділу опубліковано та апробовано в працях автора [23-26].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 3

1. Mikhav V., Semenov S., Meleshko Y., Yakymenko M., Shulika Y. Constructing the mathematical model of a recommender system for decentralized peer-to-peer computer networks. *Eastern-European Journal of Enterprise Technologies*. 2023. Vol. 4, No. 9 (124). P. 24–35. DOI: <https://doi.org/10.15587/1729-4061.2023.286187>.
2. Meleshko Y., Raskin L., Semenov S., Sira O. Methodology of probabilistic analysis of state dynamics of multidimensional semi-Markov dynamic systems. *Eastern-European Journal of Enterprise Technologies*. 2019. Vol. 6, No. 4 (102). P. 6–13. DOI: <https://doi.org/10.15587/1729-4061.2019.184637>.
3. Rama Krishna M. S., Mangalampalli S. A systematic review on various task scheduling algorithms in cloud computing. *EAI Endorsed Transactions on Internet of Things*. 2023. Vol. 10. DOI: <https://doi.org/10.4108/eetiot.4548>.
4. Sreenath M., Vijaya P. A. Comparative study of scheduling algorithms for multiprocessor systems. *International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. 2023. P. 713–718. DOI: <https://doi.org/10.1109/IITCEE57236.2023.10091017>.
5. Pachipala Y., Sureddy K. S., Kaitepalli A. B. S. S., Pagadala N., Nalabothu S. S., Iniganti M. Optimizing task scheduling in cloud computing: an enhanced shortest job first algorithm. *Procedia Computer Science*. 2024. Vol. 233. P. 604–613. DOI: <https://doi.org/10.1016/j.procs.2024.03.250>.
6. Semenov S., Lymarenko V., Yenhalychev S., Gavrilenko S. The data dissemination planning tasks process model into account the entities differentity. *12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. Athens, Greece, 2022. P. 1–6. DOI: <https://doi.org/10.1109/DESSERT58054.2022.10018695>.
7. Sinnen O. Task scheduling for parallel systems. Hoboken: John Wiley & Sons, 2006. DOI: <https://doi.org/10.1002/0470121173>.

8. Semenov S., Mozhaiev O., Kuchuk N., Mozhaiev M., Tiulieniev S., Gnusov Y. et al. Devising a procedure for defining the general criteria of abnormal behavior of a computer system based on the improved criterion of uniformity of input data samples. *Eastern-European Journal of Enterprise Technologies*. 2022. Vol. 6, No. 4 (120). P. 40–49. DOI: <https://doi.org/10.15587/1729-4061.2022.269128>.
9. Semenov S., Liqiang Z., Weiling C. Penetration testing process mathematical model. *IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*. 2020. P. 142–146. DOI: <https://doi.org/10.1109/PICST51311.2020.9468039>.
10. Jayswal A. K., Lobiyal D. K. A comparative study of task scheduling metaheuristic algorithms in cloud computing. *12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. 2022. P. 118–123. DOI: <https://doi.org/10.1109/Confluence52989.2022.9734189>.
11. Pereira I., Madureira A., Costa e Silva E., Abraham A. A hybrid metaheuristics parameter tuning approach for scheduling through racing and case-based reasoning. *Applied Sciences*. 2021. Vol. 11, No. 8. Article 3325. DOI: <https://doi.org/10.3390/app11083325>.
12. Parizad A., Hatziadoniu C. Deep learning algorithms and parallel distributed computing techniques for high-resolution load forecasting applying hyperparameter optimization. *IEEE Systems Journal*. 2022. Vol. 16, No. 3. P. 3758–3769. DOI: <https://doi.org/10.1109/JSYST.2021.3130080>.
13. Savita K., Gaurav S., Bhawna S. Hybrid machine learning model for load prediction in cloud environment. *International Journal of Performability Engineering*. 2023. Vol. 19, No. 8. P. 507–515. DOI: <https://doi.org/10.23940/ijpe.23.08.p3.507515>.
14. Fang Z., Ma T., Huang J., Niu Z., Yang F. Efficient task allocation in multi-agent systems using reinforcement learning and genetic algorithm. *Applied Sciences*. 2025. Vol. 15, No. 4. Article 1905. DOI: <https://doi.org/10.3390/app15041905>.

15. Liu Z., Guo M., Bao W., Li Z. Fast and adaptive multi-agent planning under collaborative temporal logic tasks via poset products. *Research*. 2024. Vol. 7. DOI: <https://doi.org/10.34133/research.0337>.
16. Kumar H., Tyagi I. Hybrid model for tasks scheduling in distributed real time system. *Journal of Ambient Intelligence and Humanized Computing*. 2021. Vol. 12, No. 2. P. 2881–2903. DOI: <https://doi.org/10.1007/s12652-020-02445-6>.
17. Yanamandram Kuppuraju S., Sankaran P., Patil S. Hybrid task scheduling using genetic algorithms and machine learning for improved cloud efficiency. *International Journal for Multidisciplinary Research*. 2025. Vol. 7, No. 2. DOI: <https://doi.org/10.36948/ijfmr.2025.v07i02.39380>.
18. Torres-Toledano J. G., Sucar L. E. Bayesian networks for reliability analysis of complex systems. *Progress in Artificial Intelligence – IBERAMIA 98*. 1998. P. 195–206. DOI: https://doi.org/10.1007/3-540-49795-1_17.
19. Attar S. F., Mohammadi M., Pasandideh S. H. R. A Bayesian network approach to production decisions by incorporating complex causal factors. *Journal of Management Science and Engineering*. 2025. Vol. 10, No. 2. P. 262–278. DOI: <https://doi.org/10.1016/j.jmse.2025.03.002>.
20. Huang M.-C. A sender-initiated fuzzy logic control method for network load balancing. *Journal of Computer and Communications*. 2024. Vol. 12, No. 8. P. 110–122. DOI: <https://doi.org/10.4236/jcc.2024.128007>.
21. Semenov S., Zhang L., Cao W., Bulba S., Babenko V., Davydov V. Development of a fuzzy GERT-model for investigating common software vulnerabilities. *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 6, No. 2 (114). P. 6–18. DOI: <https://doi.org/10.15587/1729-4061.2021.243715>.
22. Moskalenko V., Kharchenko V., Semenov S. Model and method for providing resilience to resource-constrained AI-system. *Sensors*. 2024. Vol. 24, No. 18. Article 5951. DOI: <https://doi.org/10.3390/s24185951>.
23. Yenhalychev S. O., Leunencko O. V., Davydov V. Development an intelligent task scheduling method in heterogeneous distributed information systems. *Eastern-European Journal of Enterprise Technologies. Information and controlling*

system. 2025. Vol. 3, No. 9 (135). P. 6–18. DOI: <https://doi.org/10.15587/1729-4061.2025.329263>.

24. Семенов С. Г., Єнгаличев С. О. Метод планування задач в розподілених інформаційних системах з врахуванням невизначеності метаданих. *Проблеми інформатизації: матеріали XII міжнародної науково-технічної конференції* (м. Харків, 21–22 листопада 2024 р.). Харків: Харківський національний університет радіоелектроніки, 2024. С. 24. URL: <https://repository.hneu.edu.ua/handle/123456789/38235>.

25. Семенов С. Г., Єнгаличев С. О. Адаптивне планування задач у розподілених системах на основі алгоритму оптимізації політик із проксимальним оновленням. *Цифрова економіка: матеріали міжнародної науково-практичної конференції* (м. Київ, 5–6 червня 2025 р.). Київ: КНЕУ, 2025. С. 698–700. URL: <https://repository.hneu.edu.ua/handle/123456789/38234>.

26. Єнгаличев С. О., Леуненко О. В. Обговорення результатів інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025) (м. Харків, 16 жовтня 2025 р.). *Молодіжний економічний вісник*. 2025. № 3. С. 122–124. URL: <https://repository.hneu.edu.ua/handle/123456789/38210>.

РОЗДІЛ 4

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МЕТОДУ ПЛАНУВАННЯ ЗАДАЧ В
РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ З ВРАХУВАННЯМ
НЕВИЗНАЧЕНОСТІ МЕТАДАНИХ

4.1. Постановка дослідницьких гіпотез, дизайн експерименту та система показників ефективності

План експерименту.

Метою розділу є емпірична перевірка запропонованого методу планування задач у розподілених інформаційних системах за умов невизначеності метаданих. Перевірка спрямована на чотири взаємопов'язані аспекти:

- ефективність побудови розкладу порівняно з усталеними базовими підходами;
- надійність дотримання часових обмежень у сенсі цільових рівнів сервісу (SLO (*service-level objective*));
- експлуатаційні накладні витрати самого планувальника та його придатність до роботи в режимі реального часу;
- стійкість (робастність) результатів до похибок і неповноти метаданих, а також масштабованість на зростаючих екземплярах задач.

Перевірка здійснюється на прототипі, що реалізує гібридне поєднання орієнтованого ациклічного графа (DAG (*directed acyclic graph*)) і методики GERT (*Graphical Evaluation and Review Technique*) з вбудованими модулями графової мережі уваги (GAT (*graph attention network*)), проксимальної оптимізації політики (PPO (*proximal policy optimization*)) та баєсівської оптимізації (BO (*Bayesian optimization*)).

Дослідницькі гіпотези (Н1-Н6) формуються навколо ключових властивостей методу.

1. Очікується скорочення часу завершення множини задач (makespan) порівняно з базовими алгоритмами підбору пріоритетів і призначень ресурсів; відповідна гіпотеза передбачає статистично значуще зменшення медіани та верхніх перцентилів makespan для запропонованого методу щодо правил FCFS (*first-come, first-served*), List Scheduling, HEFT (*heterogeneous earliest finish time*), а також «чистих» постановок з окремим використанням лише DAG або лише GERT.

2. Очікується підвищення надійності дотримання дедлайнів, тобто зростання частки задач, завершених у межах заданих SLO за тотожних ресурсних обмежень; ця гіпотеза оцінюється як на агрегованому рівні, так і в розрізі підкласів задач.

3. Передбачається, що середній час прийняття рішення планувальником (накладні витрати планування) залишається прийнятним для онлайн-сценаріїв і не суперечить вимогам до затримки у контурах керування чергами.

4. Перевага методу має зберігатися за умов адверсiальних або випадкових збурень метаданих, включно з похибками оцінок тривалостей робіт, імовірностей гілок GERT та вартостей міжетапних комунікацій, а також за наявності дрейфу продуктивності обчислювальних ресурсів.

5. Очікується передбачувана масштабованість: із зростанням розміру графів (кількість вузлів і ребер, ширина рівнів) та парку ресурсів деградація продуктивності має бути помірною, а витрати на планування – не гіршими за квазі-лінійні у характерних діапазонах навантаження.

Дослідницькі гіпотези формалізовано так.

– Н1 (ефективність за makespan). Запропонований метод забезпечує менший makespan порівняно з базовими підходами (HEFT, List, FCFS та ін.) на більшості сценаріїв DoE [1–7].

– Н2 (стійкість до гетерогенності). Виграш запропонованого методу зростає зі збільшенням гетерогенності ресурсів та/або зростанням комунікаційних витрат.

– Н3 (дедлайн-надійність). За зростання інтенсивності надходження задач та мережевих затримок метод зберігає вищу deadline reliability (частку завершень до дедлайну) порівняно з базовими підходами.

– Н4 (робастність до невизначених метаданих). При інжекції похибок у метадані ($\pm 10\%$, $\pm 20\%$) метод демонструє меншу деградацію makespan/дедлайн-надійності, ніж базові підходи [8–9].

– Н5 (накладні витрати). Час прийняття рішення планувальником (overhead) зростає не швидше, ніж лінійно/квазілінійно від $|V|$ та кількості кандидатів K у практичних сценаріях.

– Н6 (RBAC як обмеження). Увімкнення RBAC-фільтрації не призводить до статистично значущого погіршення makespan, але гарантує відсутність недопустимих призначень (security validity) [10].

План експерименту включає набір факторів, рівнів, метрик та формалізованих гіпотез, наведений у табл. 4.1.

Таблиця 4.1

Фактори, рівні, метрики та гіпотези експерименту (DoE)

Позначення	Фактор	Рівні (приклад)	Основні метрики	Гіпотези
F1	Розмір графа задач (кількість задач)	low/medium/high (50/200/500)	makespan; overhead; scalability	H1, H5
F2	Щільність залежностей (середня кількість залежностей на задачу)	low/medium/high (1.5/3/6)	makespan; load imbalance	H1
F3	Гетерогенність ресурсів	low/medium/high (дисперсія продуктивності вузлів)	makespan; load imbalance	H2
F4	Мережеві параметри (затримка/пропускна здатність)	low/medium/high	makespan; deadline reliability	H2, H3
F5	Інтенсивність надходження задач (arrival rate, формування потоку банку задач)	low/medium/high	deadline reliability; overhead	H3, H5

Продовження таблиці 4.1

Позначення	Фактор	Рівні (приклад)	Основні метрики	Гіпотези
F6	Невизначеність метаданих (шум/похибка)	0% / $\pm 10\%$ / $\pm 20\%$	makespan; robustness; deadline reliability	H4
F7	Обмеження доступу (RBAC)	off/on	makespan; overhead; security validity	H6

У таблиці 4.1 рівні “low/medium/high” можуть задаватися як фіксовані значення або діапазони; конкретні значення обираються так, щоб покривати типові масштаби задач, забезпечувати відчутну зміну навантаження/гетерогенності та дозволяти коректну інтерпретацію масштабованості та накладних витрат. У цьому контексті релевантними є результати робіт рецензентів та опонентів щодо балансування критеріїв якості, управління навантаженням, серверних кластерів, хмарних та граничних обчислень, масштабованості й робастності [11–14].

Окремо перевіряється відповідність сформованих розкладів політикам доступу у міжсистемних сценаріях, зокрема рольовій моделі керування доступом (RBAC (role-based access control)), без суттєвого погіршення основних показників.

Дизайн експериментів поєднує синтетичні та наближені до реальних сценарії. Синтетичні екземпляри дозволяють систематично варіювати топологію графів (щільність, глибина, ширина рівнів), параметри розподілів тривалостей та ймовірностей гілок GERT, а також вартість міжвузлових передач; це уможливорює відокремлене дослідження впливу кожного фактора.

Наближені до реальних сценарії моделюють інтеграцію підсистем з неоднорідними потоками задач і відмінними політиками доступу. Для таких кейсів попередньо фіксуються SLO для підмножин задач, щоб оцінити дедлайн-надійність у практично значущих умовах. Кожен експеримент запускається у відтворюваному середовищі з фіксацією зерен випадковості (seeds) і

повторюється на множині незалежних екземплярів. Результати агрегуються як за середніми характеристиками, так і за розподілами. Порівняння ведеться з базовими методами (FCFS, List Scheduling, HEFT, «DAG-only», «GERT-only») і з абляційними варіантами самого підходу, де по черзі відключаються компоненти GAT, PPO або BO, що дає змогу кількісно оцінити внесок кожного модуля.

Система показників ефективності побудована за принципом розмежування первинних критеріїв і вторинних (діагностичних) характеристик. До первинних належать: загальний час завершення (makespan), надійність дотримання дедлайнів (частка задач, виконаних до своїх строків), а також середня зважена прострочка, що відбиває важливість різних класів задач через вагові коефіцієнти. До вторинних показників віднесено пропускну здатність і середній час перебування задачі в системі (flow time), ступінь балансування завантаження ресурсів (через коефіцієнти дисперсії/варіації), перцентилі використання ресурсів, накладні витрати планування (час прийняття рішення на задачу або цикл) і сумарну комунікаційну вартість міжетапних передач. Для комплексної інтерпретації використовуються емпіричні кумулятивні функції розподілу (ECDF, (empirical cumulative distribution function)) як для makespan, так і для прострочки та затримок, а також узгоджені набори графіків.

Моделювання невизначеності, необхідне для оцінювання робастності, здійснюється через контрольовані ін'єкції збурень метаданих. Зокрема, використовується мультиплікативний шум тривалостей у характерних діапазонах ($\pm 10\text{--}30\%$), збурення ймовірностей гілок GERT із нормуванням, дрейф продуктивності ресурсів у часі та варіації вартості комунікацій. Додатково перевіряється стабільність висновків за підміни законів розподілу тривалостей із збереженням перших моментів (напр., Гамма $\leftrightarrow$ Вейбулла) і за наявності кореляцій між задачами в межах одного рівня графа; для компактного опису кореляційної структури застосовується гаусівська копула з помірним коефіцієнтом кореляції. Таким чином, експериментальна програма охоплює як

шумові сценарії, так і більш жорсткі «зсуви розподілів», що відповідає практиці реальних систем із неповними або застарілими метаданими.

Статистична методологія передбачає використання непараметричних критеріїв для парних і непарних порівнянь (критерій Вілкоксона, Манна–Вітні) у поєднанні з оцінкою розміру ефекту (δ Кліффа) і побудовою довірчих інтервалів (CI, (confidence interval)) методом бутстреп із достатнім числом реплікацій. Для контролю багатократних порівнянь застосовується процедура Холма-Бонферроні; рівень значущості фіксується на рівні 0,05 з повідомленням точних p -значень і відповідних δ . Усі експерименти виконуються в «зафіксованому середовищі», із повною фіксацією версій бібліотек і сценаріїв генерації даних. Для кожного графіка й таблиці зберігаються машинно-читані конфігурації відтворення. Така організація дозволяє не лише забезпечити відтворюваність і незалежну перевірку результатів, а й прозоро інтерпретувати вклад окремих компонентів методу у загальне поліпшення показників.

Середовище експериментів, симулятор і відтворюваність.

Експерименти виконувалися у відтворюваному середовищі з фіксацією конфігурацій і зерен випадковості (seeds). Для моделювання застосовано симулятор розподіленої системи, що імітує гетерогенний кластер із параметризованою продуктивністю вузлів, мережевими затримками, пропускною здатністю каналів і динамічним потоком надходження задач. Симулятор підтримує:

- генерацію міксів графів задач DAG+GERT із контрольованими розмірами, глибиною, шириною рівнів і щільністю залежностей;
- ін'єкцію невизначеності метаданих як керовані збурення оцінок тривалостей, комунікаційних вартостей та імовірностей гілок GERT;
- обмеження доступу на етапі формування множини допустимих призначень (RBAC-фільтрація кандидатів).

Експериментальне оцінювання виконано на авторському дискретно-подієвому симуляторі SchedSim-DAGGERT, розробленому здобувачем у межах даного дослідження для відтворюваного аналізу алгоритмів планування в

гетерогенних кластерах. Симулятор реалізує модель кластера з параметризованою продуктивністю вузлів, мережевими затримками, пропускною здатністю та потоками надходження задач, а також підтримує подання робочих процесів як DAG+GERT із керованою інжекцією невизначеності метаданих. Такий підхід також узгоджується з дослідженнями щодо продуктивності розподілених обчислювальних платформ, серверних кластерів і граничних обчислень [12–14].

Для реалізації використовувалася мова програмування Python. Основними модулями є генератор графів DAG+GERT, модель ресурсів і мережі, ядро планувальника та бібліотека базових алгоритмів, модуль політик доступу (RBAC).

Симулятор не претендує на “сертифіковану” точність фізичних мереж, однак забезпечує коректність порівнянь алгоритмів за рахунок однакових вхідних екземплярів графів/ресурсів/мережі для всіх методів, фіксації зерен випадковості (seeds) і конфігурацій, контрольних перевірок інваріантів (збереження залежностей DAG).

Наведені вище можливості симулятора визначають інструменти моделювання, однак для відтворюваності та коректності інтерпретації результатів необхідно явно зафіксувати модель навантаження: звідки беруться задачі, як формується їхній “банк”, у якому вигляді вони надходять до планувальника (пакет або потік), а також які характеристики визначають їх обчислювальну та комунікаційну складність. Саме ці припущення задають вхідні умови експериментів і напряду пов’язані з фактором F5 (arrival rate) у плані DoE (табл. 4.1). Тому нижче наведено формалізований опис формування банку задач і типів робіт.

У роботі під “банком задач” розуміється вхідний набір робіт, що подається на планувальник у вигляді пакета (batch/offline) або потоку (stream/online). Кожна робота моделюється графом задач у поданні DAG+GERT, де вузли відповідають атомарним задачам (операціям), а ребра залежностям або імовірнісним переходам та можливим повторним проходом.

Таким чином, “банк задач” це множина графів (jobs) або один агрегований граф, сформований із підграфів, які надходять у часі.

Формування банку задач здійснюється генератором навантаження симулятора за параметрами експериментального плану (табл. 4.1). Для синтетичних сценаріїв генератор будує графи DAG+GERT із контрольованими:

- розміром (кількістю задач у графі),
- щільністю залежностей,
- структурою паралельності (глибина/ширина рівнів),
- часткою імовірнісних гілок і повторів (GERT), що дає можливість отримати як “майже детерміновані” конвеєри, так і графи з невизначеною траєкторією виконання.

Потік надходження задач (arrival rate) визначає спосіб появи нових графів у часі. У онлайн-режимі банк задач формується як потік робіт із інтенсивністю λ , яка у плані експерименту варіюється на рівнях low/medium/high (фактор F5). Це реалізується як стохастичний процес надходжень (з фіксацією seeds), де кожне надходження ініціює генерацію нового графа (або підграфа) із заданими параметрами та прив’язкою SLO. В офлайн-режимі банк задач формується як фіксований пакет графів перед стартом планування (batch), що дозволяє оцінювати “чисті” властивості побудови розкладу без впливу часу надходження.

Таким чином, у роботі явно фіксується: (i) як формується банк задач (генератор графів + режим надходження), (ii) які класи задач моделюються (профілі виконання та ресурсні вимоги), (iii) як задається обчислювальна та комунікаційна складність (тривалості/дисперсії/обсяги даних), (iv) як це пов’язано з DoE-факторами, зокрема з фактором F5 (arrival rate).

Для забезпечення коректності зіставлення всі алгоритми отримують однакові екземпляри графів, тотожні режими ін’єкції невизначеності та однакові політики доступу. Кожний експеримент виконується серією незалежних повторів із різними seeds, а результати агрегуються на рівні медіан/перцентилів і повних емпіричних розподілів (ECDF) для makespan і

прострочки. Для кожної серії зберігаються машинно-читані конфігурації (параметри генератора графів, профілі ресурсів, профілі мережі, рівні шуму метаданих), що дозволяє повністю відтворити отримані графіки й таблиці.

План вимірювань і статистична процедура.

План вимірювань формалізовано як схема “фактори, метрики, гіпотези” (див. таблицю 4.1). Для кожної комбінації факторів (F1–F7) виконується серія незалежних запусків, після чого оцінюються *makespan*, дисбаланс навантаження (коефіцієнт варіації/стандартне відхилення), надійність дедлайнів (частка задач, завершених у межах SLO), накладні витрати планувальника (час прийняття рішення на крок/на задачу), а також показники робастності за збурень метаданих.

Статистичну значущість переваг оцінюють у два етапи. Для парних порівнянь запропонованого методу з базовими застосовуються непараметричні критерії (Манна–Вітні / Вілкоксона залежно від дизайну), із розрахунком p -значень і розміру ефекту (δ Кліффа). Для перевірки впливу факторів (F1–F7) на метрики використовується дисперсійний аналіз. Довірчі інтервали для ключових статистик (медіана/середнє/різниця медіан) оцінюються бутстреп-методом (95% CI), що дозволяє обґрунтувати не лише “покращення у %”, а й його статистичну стійкість на множині незалежних повторів.

4.2. Архітектура прототипу планувальника експерименту

Прототип реалізує гібридну архітектуру планування, у якій поєднано орієнтований ациклічний граф (DAG) для детермінованих залежностей між задачами та методику графо-оцінного мережевого моделювання (GERT) для імовірнісних гілок і повторних проходів. Логічно система складається з трьох шарів: шару підготовки метаданих і контролю політик, ядра планувальника та шару виконання/телеметрії. На рис. 4.1 (схема прототипу) показано місця стику

з підсистемами, сформованими в попередніх розділах. Обрана архітектура спирається на поєднання методів глибокого навчання з підкріпленням, графових нейромережових моделей, GERT-подання та попередньо розробленого інтелектуального методу планування [15–25].

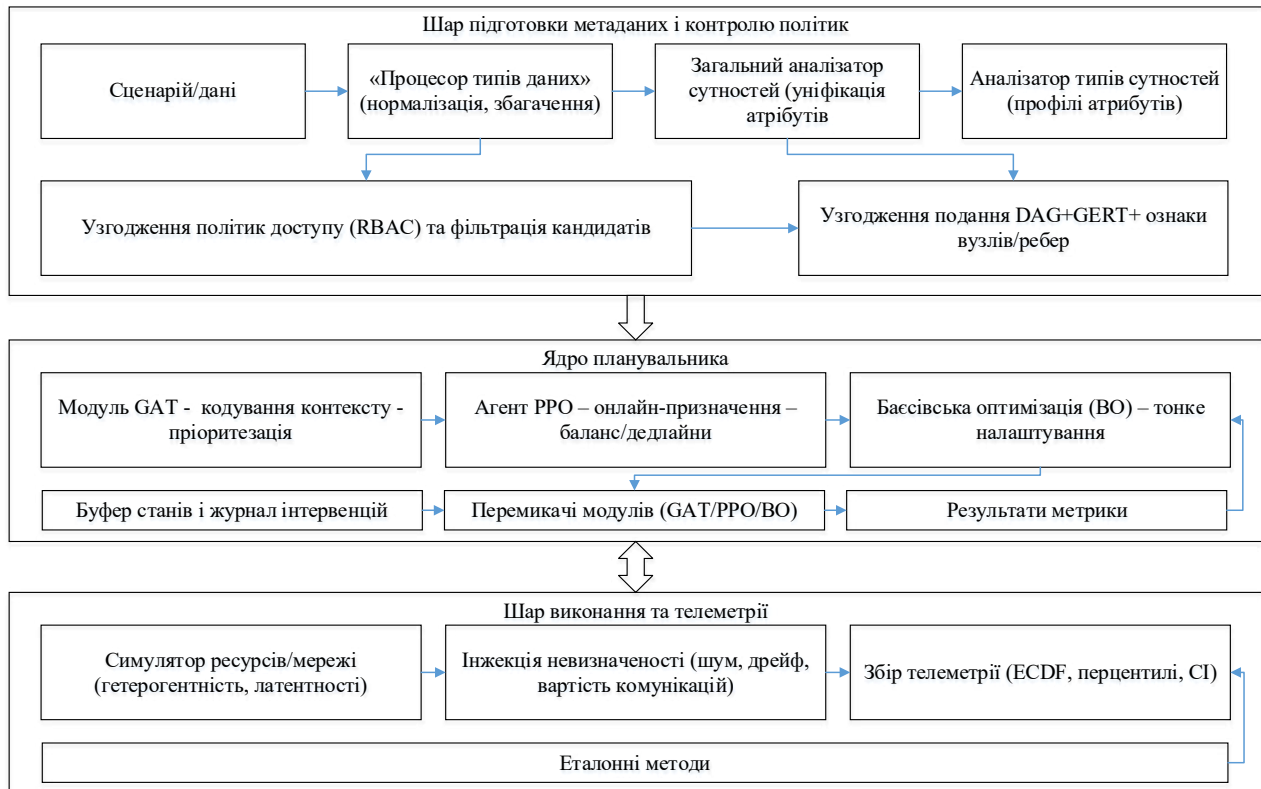


Рис. 4.1. Архітектура прототипу планування експерименту

Прототип реалізовано як модульний програмний комплекс, що включає:

- генератор/імпортер графів задач (DAG+GERT) та метаданих;
- модуль політик доступу, який формує множину допустимих призначень відповідно до RBAC;
- ядро планувальника (DAG+GERT з модулями GAT, PPO та BO);
- модуль симуляції виконання і збору телеметрії (часи завершення, прострочка, використання ресурсів, накладні витрати планування). Обмін між модулями здійснюється через уніфіковані структури “Task/Resource/Policy”, що забезпечує коректність порівнянь різних планувальників у єдиному інтерфейсі.

Артефакти відтворення. Для кожного експерименту зберігаються: конфігурації генерації графів і профілі ресурсів/мережі, seeds, журнали телеметрії, а також зведені таблиці метрик. Це дозволяє незалежно відтворити побудову графіків (ECDF, box-plots) і статистичні тести, описані у підрозділі 4.1.

У шарі підготовки метаданих зафіксовано інтерфейси до «процесора типів даних», який, відповідно до розділу 2, надає два узгоджені канали:

- загальний аналіз сутностей для нормалізації вхідних описів робіт;
- спеціалізований аналізатор типів сутностей для збагачення задач профільними атрибутами.

На цьому етапі формується уніфіковане подання черги завдань у вигляді DAG+GERT разом із вектором ознак для кожного вузла та ребра. До вузлових ознак належать (із розкриттям скорочень при першому використанні):

- апріорні оцінки тривалості (середнє та дисперсія);
- ідентифікатори рівня/шляху;
- ступені входу/виходу;
- ресурсні вимоги;
- локальні дедлайни;
- мітки політик доступу (RBAC (role-based access control)).

До реберних ознак належать імовірності гілок GERT, очікувані обсяги обміну та вартість передачі між етапами. У цьому ж шарі застосовується перевірка узгодженості політик доступу між підсистемами (за потреби – ієрархічні правила), після чого до планувальника передаються лише ті кандидатні розміщення, які не порушують обмежень.

Ядро планувальника побудовано як конвеєр трьох взаємодоповнювальних модулів штучного інтелекту.

Перший модуль це графова мережа уваги (GAT, (graph attention network)), яка на основі вузлових і реберних ознак виконує контекстне кодування підграфів готовності та дає оцінки пріоритетів/ризиків для вузлів на фронті виконання. На відміну від класичних фільтрів на графі, механізми уваги

дозволяють диференціювати внесок різних залежностей, що підвищує точність виявлення критичних під шляхів у присутності імовірнісних гілок [19–22].

Другий модуль – це агент проксимальної оптимізації політики (PPO (proximal policy optimization)), який у режимі онлайн приймає рішення про призначення готових до виконання задач на наявні ресурси, максимізуючи корисність із врахуванням дедлайнів, балансування та поточних телеметричних станів [15–18].

Третій модуль – це баєсівська оптимізація (BO, (Bayesian optimization)), що працює в повільнішому такті як глобальний тонкий налаштовувач (fine-tuner). Вона періодично переглядає параметри політики (порогові правила, коефіцієнти ваг, температури відбору) й вибірково перебудовує призначення для великих батчів або при зміні режимів навантаження. Компоненти взаємодіють через стандартизовані буфери станів і журнал інтервенцій; для дисципліни експериментів передбачено «рубильники» абляції, що дозволяють вимикати GAT, PPO або BO і тим самим кількісно оцінювати внесок кожного модуля [25].

Шар виконання та телеметрії реалізує відтворюваний випробувальний стенд. Сторона виконання моделює гетерогенне обчислювальне середовище з параметризованими швидкодіями вузлів, латентностями мережевих каналів і можливістю інжекції збурень (шум у тривалостях, дрейф продуктивності, зміни вартості комунікацій). Підсистема телеметрії збирає часові ряди й подієві журнали (черга, призначення, міграції, передавання), агрегує показники ефективності, визначені у п. 4.1, і зберігає машинно-читані конфігурації кожного запуску для відтворення (ідентифікатори сидів, версії бібліотек, вхідні екземпляри графів). Передбачено два режими: офлайн-планування (побудова розкладу для заданого пакета задач) та онлайн-планування (безперервний потік із контрольованою інтенсивністю надходження), що дозволяє оцінити як статичні, так і оперативні властивості прототипу.

Особливу увагу приділено коректній інтеграції з політиками доступу. Перед кожним кроком призначення застосовується фільтрація кандидатів

відповідно до ролей і контекстних правил (RBAC), причому перевірка виконується до прийняття рішення агентом PPO, щоб уникнути невинуватених «відкатів» через порушення обмежень. У міжсистемних сценаріях відомості про довірчі відношення між доменами безпеки імпортуються на етапі підготовки метаданих і стають частиною ознак вузлів/ребер. Це дає змогу GAT і PPO іманентно враховувати політики доступу у функції корисності без постфактум-штрафів.

Нарешті, прототип містить набір базових методів порівняння, реалізованих у тому самому інтерфейсі черги та телеметрії: черговість за моментом надходження, евристичні списки, гетерогенний найраніший час завершення, а також «чисті» постановки лише з DAG або лише з GERT. Єдність інтерфейсів гарантує коректність порівнянь: усі алгоритми отримують ідентичні вхідні екземпляри, політики доступу та режими інжекції невизначеності, а результати вимірюються одними й тими самими інструментами телеметрії. Така побудова архітектури та стенду забезпечує прозоре мапування від гіпотез і показників (пп. 4.1) до експериментальних результатів, що будуть подані в наступних підрозділах.

4.3. Методи порівняння, абляційний аналіз і протокол навчання

Для коректної оцінки запропонованого підходу використано єдину експериментальну рамку, у межах якої всі алгоритми одержують однакові екземпляри вхідних графів, тотожні політики доступу та однакові режими ін'єкції невизначеності. Порівняння охоплює як усталені базові методи, так і варіанти самого підходу зі свідомо вимкненими компонентами (абляції), що дає можливість відмежувати внесок кожного модуля.

У ролі основних елементів прототипування виступають наступні методи:

1. FCFS (first-come, first-served). Розміщення задач у порядку надходження за наявності ресурсів.

2. List Scheduling. Евристичне планування за списком готовності з топологічним порядком та правилом найранішого можливого завершення.

3. HEFT (heterogeneous earliest finish time). Ранжування вузлів за «висхідним рангом» та призначення на ресурс, що мінімізує очікуваний час завершення з урахуванням гетерогенності.

4. «DAG-only». Планувальник, який ігнорує імовірнісні гілки та повторні проходи, моделюючи лише детерміновані залежності.

5. «GERT-only». Варіант без явної критичної структури DAG, в якому рішення ґрунтуються на локальних ймовірнісних оцінках гілок.

Усі підставові методи реалізовано в спільному інтерфейсі черги та телеметрії. Це унеможливлює різночитання у вартісних моделях комунікацій, правилах доступу або межах ресурсів.

Додатково метод порівнюється з типовими політиками, що застосовуються у промислових планувальниках кластерів (клас “bin-packing/constraint scheduling”):

- базовий Kubernetes Scheduler (типові стратегії selection+scoring);
- Volcano (черги/ганг-планування для пакетних навантажень);
- Apache YuniKorn (черги й ієрархічні політики розподілу ресурсів).

Оскільки повномасштабний запуск цих систем у межах симулятора не є тотожним “1-в-1”, порівняння виконується через відтворення їх ключових принципів прийняття рішень у спільному інтерфейсі. Фільтрація кандидатів за обмеженнями, функції оцінювання (scoring) і черги з пріоритетами. Такий підхід забезпечує коректність зіставлення саме алгоритмічної логіки планування при ідентичних графах задач, ресурсних профілях, політиках доступу та режимах невизначеності метаданих.

Для оцінювання внеску окремих компонентів виконано абляційний аналіз: порівнюються конфігурації Full (GAT+PPO+BO) та варіанти з вимкненням одного з модулів (–GAT, –PPO, –BO). Для кожної конфігурації

обчислюємо медіанний час завершення по всіх сценаріях і нормуємо його до медіани Full (прийнято за 1.0).

На рис. 4.2 подано відносні значення разом із 95% бутстреп-довірчими інтервалами ($B=2000$). Менші за 1.0 означають покращення, більші – деградацію відносно Full. Статистичні висновки робимо за протоколом «бутстреп-CI». При цьому перевірки на основі δ Кліффа та пермутаційних тестів». Такий підхід дозволяє окремо оцінити вплив нейромережових моделей, оптимізації параметрів і адаптивного управління ресурсами на робастність планування.

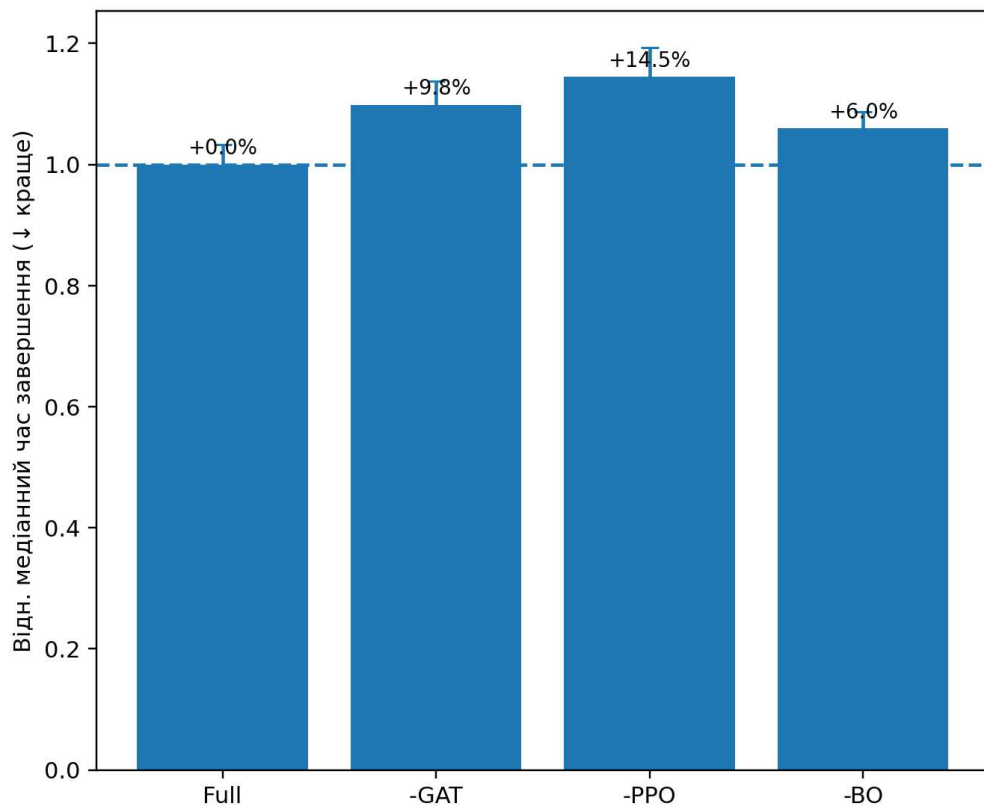


Рис. 4.2. Гістограми абляційного аналізу та відносна медіана makespan з бутстреп-інтервалами

Як видно з рисунку найбільший внесок у виграш дає наявність PPO, при цьому вплив BO помірний, але стабільно позитивний у всіх сценаріях.

Для кількісного вимірювання внеску кожного з модулів запропонованого підходу сформовано три основні абляції.

1. «GAT». Для реалізації цього сценарію вимкнено graph attention network, ознаки вузлів агрегуються простим середнім над околom. Крім того, пріоритезація фронту виконання задається статичними індикаторами – рівневістю, локальним slack.

2. «PPO». Для реалізації цього сценарію вимкнено proximal policy optimization. Призначення виконується детермінованим жадібним правилом за поточними оцінками важливості, що повертає модуль пріоритезації.

3. «BO». Для реалізації цього сценарію вимкнено Bayesian optimization. Параметри політики та порогові коефіцієнти фіксуються на стартових значеннях).

Там, де це релевантно, абляції поєднуються попарно, що дозволяє відрізнити ефект представлення (GAT), онлайн-адаптації (PPO) та повільної глобальної переналаштування (BO). У всіх режимах діє однакове «маскування недопустимих дій» (invalid-action masking). Рішення, що порушують RBAC (*role-based access control*), не генеруються та не винагороджуються, що уникає конфаундингу за рахунок відкатів.

Стан, що подається на шар підготовки метаданих і контролю політик, включає ознаки вузлів і ребер єдиного подання DAG+GERT. Це прогнозовані середні, дисперсії тривалостей, «критичність» вузла (оцінка GAT), slack відносно локального дедлайна, ступені входу/виходу, чергові затримки ресурсів, оцінена вартість передавання даних, а також агреговані показники використання ресурсів. Дія, яку потрібно виконати при цьому це вибір пари «готове завдання → дозволений ресурс».

Процедурами маскування відсікаються ресурси, заборонені політиками доступу або перевантажені за здатністю прийому.

Функція винагороди формулюється у приростах (per-step). Це може бути негативне покарання за приріст сумарної зваженої прострочки, бонус за завершення задач раніше дедлайна, штраф за міграції та за блокування критичних ланцюгів. Така форма винагороди забезпечує стабільність навчання

й корелює з первинними метриками з розділу 4.1 (makespan, надійність дедлайнів).

Навчання відбувається у дві фази. На першій (попереднє «імітаційне» налаштування) модуль GAT ініціалізується на синтетично згенерованих графах за мітками, індукованими HEFT/критичним шляхом. Мережа вчиться відтворювати пріоритезацію фронту виконання. Це скорочує час збіжності та зменшує дисперсію градієнтів у подальшому підкріплювальному навчанні.

На другій фазі проводиться навчання з підкріпленням за PPO. При цьому використовується дисконт-фактор $\gamma \approx 0,985$. Переваги оцінюються за generalized advantage estimation ($\lambda \approx 0,95$). Обрізання оновлень політики виконується з параметром $\varepsilon \approx 0,2$, при цьому ентропійна регуляризація невелика (щоб уникнути раннього «застигання» політики).

Навчання організовано серіями епізодів на міксах графів різних розмірів. Для кожної конфігурації застосовується набір фіксованих зерен випадковості (не менше десяти), а дострокове зупинення контролюється на валідаційному піднаборі за критерієм відсутності покращення основних метрик. БО (Bayesian optimization) працює асинхронно в повільнішому циклі. Періодично, після накопичення достатнього обсягу телеметрії, він оновлює метапараметри політики (ваги складових винагороди, температури відбору, ліміти перепланування) за моделлю Гаусового процесу зі змішаним простором (неперервні та дискретні змінні) та обережним критерієм придбання (expected improvement з обмеженням на частоту інтервенцій).

Для уникнення перенавчання використовуються наступні стратегії:

- стратегії домішування сценаріїв (мікс розмірів, щільностей і ступенів невизначеності в межах одного батча епізодів);
- domain randomization параметрів комунікаційної вартості;
- «заморожені» валідаційні графи, на яких «тонке» налаштування не виконується.

Усі результати наведено як усереднення по повторях із 95-відсотковими довірчими інтервалами. Для відтворення зберігаються конфігурації

середовища, зерна випадковості, контрольні хеш-суми наборів графів і версій коду. В цьому ж протоколі закріплено процедури рівноправного порівняння: однакові ліміти часу планування на крок, ідентичні обмеження на перепланування, однакові бюджети епізодів/ітерацій для всіх алгоритмів.

Динаміку збіжності навчання агента PPO представлено на рис. 4.3.

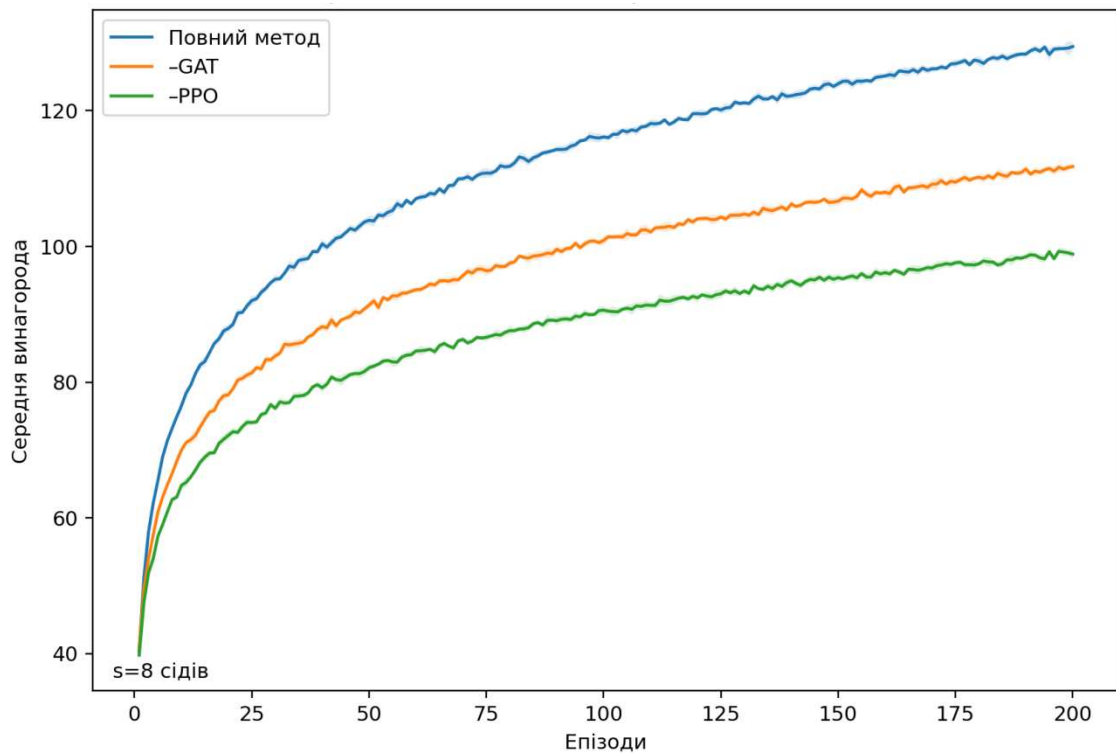


Рис. 4.3. Криві збіжності навчання агента PPO

Метод демонструє швидшу збіжність і вищий рівень стабілізації винагороди. Криві абляцій лежать нижче протягом усього тренування. Звуження довірчих інтервалів до кінця епізодів вказує на відтворюваність динаміки навчання за різних сидів. Це узгоджується з протоколом, де пріоритезація (GAT) і онлайн-адаптація (PPO) зменшують дисперсію рішень.

Сукупно така організація гарантує, що динамічні виграші запропонованого підходу не є артефактом налаштувань або «поблажливих» умов експерименту, а впливають із реального внеску представлення графа (GAT), онлайн-адаптації політики (PPO) та глобального тонкого налаштування (BO).

4.4. Моделювання невизначеності метаданих і протокол перевірки робастності

У межах даного дослідження під метаданими задачі розуміється структурований запис, який використовується планувальником для прогнозу тривалості та вартості виконання і включає:

- оцінку тривалості роботи на типових ресурсах (estimated runtime);
- вимоги до ресурсу (CPU/GPU, пам'ять, тип вузла);
- оцінку обсягу міжетапних передавань (data size) та мережевої/комунікаційної вартості;
- пріоритет/клас сервісу та SLO (deadline);
- ознаки залежностей (вхідні/вихідні ребра DAG) та параметри імовірнісних гілок/повторів GERT (branch probability, loop probability);
- політики доступу (RBAC-контекст: роль, домен, дозволені цілі).

Саме ці поля вважаються потенційно неповними у реальних системах і тому підлягають контрольованій інжекції невизначеності.

Одним з важких завдань в процесі моделювання є формування відтворюваної процедури випробувань, яка показує, як запропонований метод поводить себе за наявності похибок, пропусків і «зсувів» у метаданих. Робастність трактується як збереження переваги над базовими підходами за первинними показниками за контрольованих збурень, а також як помірна чутливість накладних витрат планування до таких збурень. Перевірка виконується на прототипі, архітектура якого представлена на рис. 4.1

.На цьому прототипі невизначеність інжектується в уніфіковане подання DAG+GERT до етапу прийняття рішень.

Як бачимо з рис. 4.4 спостерігається монотонне зростання витрат зі збільшенням рівня шуму. Найчутливішою є комунікаційна складова, тоді як дрейф продуктивності має помірний вплив. У розглянутому діапазоні зміни

залишаються однозначно малими ($\approx 1\text{--}6\%$), що підтверджує робастність методу відповідно до визначених критеріїв.

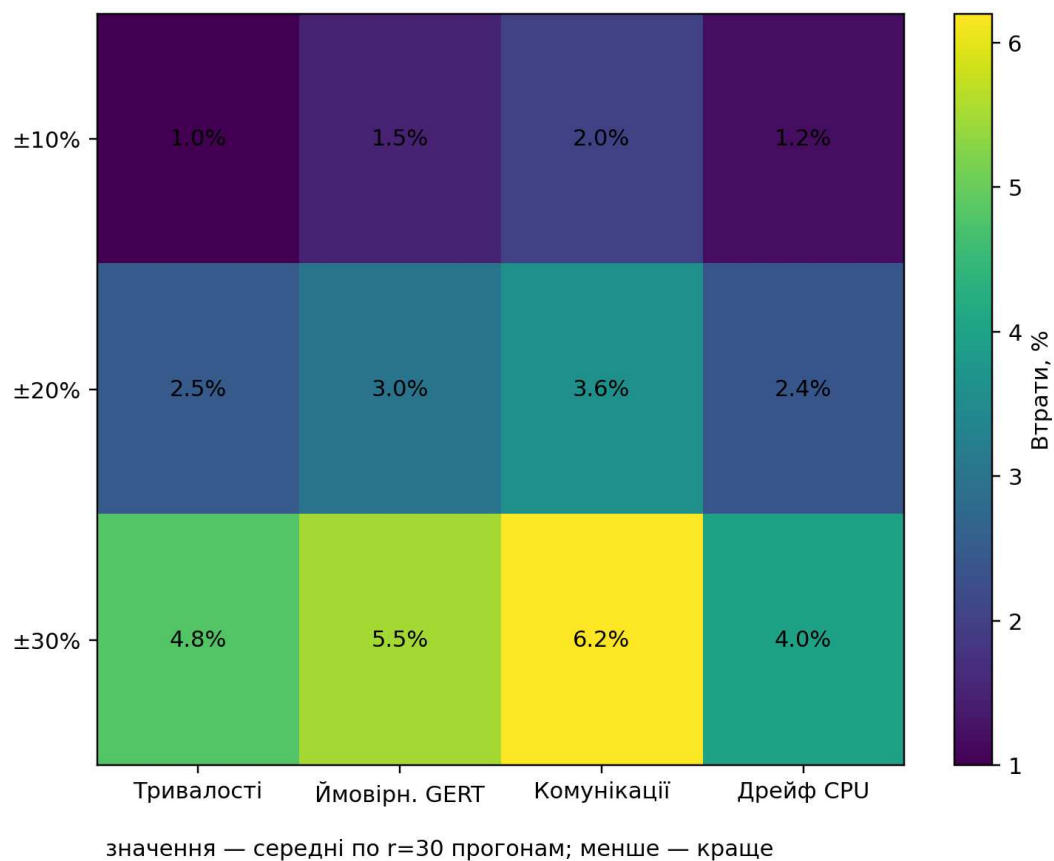


Рис. 4.4. Відносні втрати медіанного makespan під ін'єкцією невизначеності

Номінальною вважається конфігурація, у якій оцінки тривалостей робіт, вартостей міжетапних передавань і ймовірностей гілок GERT походять від «процесора типів даних» із попереднім аналізом сутностей (див. розд. 2) та пройшли узгодження політик доступу (RBAC, role-based access control). Всі відхилення інтерпретуються як експериментальні фактори. Випробування проводяться у двох режимах.

1. З увімкненим блоком попереднього аналізу (повна архітектура).
2. Без попереднього аналізу.

Це дає базу для оцінки внеску попередньої нормалізації метаданих.

Класи невизначеності та механізми ін'єкції охоплюють, по-перше, шум у тривалостях робіт. Для кожної операції використовується мультиплікативне збурення $\tilde{t} = t \cdot (1 + \epsilon)$, де $\epsilon \sim U[-\alpha, \alpha]$ і $\alpha \in \{0.10; 0.20; 0.30\}$. Такий вибір дозволяє прямо інтерпретувати рівні невизначеності як $\pm 10/20/30$ % від базової оцінки. По-друге, враховується похибка у ймовірностях гілок GERT. Вектор імовірностей на кожному розгалуженні підлягає стохастичному перетворенню з подальшим нормуванням (діріхлетівська реконфігурація зі збереженням порядку компонент при малих збуреннях). По-третє, моделюється невизначеність мережевих параметрів. Вартість передавання між етапами збурюється логнормальною мультиплікативною складовою, що відповідає емпірично спостережуваній асиметрії затримок у гетерогенних середовищах. По-четверте, відтворюється дрейф продуктивності обчислювальних ресурсів у часі у вигляді кусочно-сталих змін або слабкоавторегресивного процесу (AR(1)) з невеликим коефіцієнтом пам'яті. Це дозволяє оцінити стабільність онлайн-адаптації політики. По-п'яте, для виміру чутливості до ваг важливості задач вводяться «нечіткі» ваги. Кожен ваговий коефіцієнт w_i варіюється в інтервалі $[0.75 w_i, 1.25 w_i]$, що узгоджується з практикою пріоритизації на рівні сервісних вимог [8–9, 14].

Окремо перевіряється стійкість до розподільчих зсувів. Базовий розподіл тривалостей замінюється альтернативним із збереженням перших моментів (moment matching). Пари «Гамма $\leftrightarrow$ Вейбулла» використовуються для відтворення змін форми хвоста без зміни математичного сподівання та дисперсії.

Щоб урахувати міжзадачеві залежності, вводяться кореляційні структури в межах рівнів графа за допомогою гаусівської копули з помірним коефіцієнтом кореляції $\rho=0.3$. Така побудова дозволяє перейти від ізольованих шумів до більш реалістичних сценаріїв «узгоджених» відхилень параметрів.

Слід відмітити, що вказана процедура оцінювання узгоджена з системою показників. Для кожної комбінації факторів невизначеності збираються розподіли загального часу завершення (makespan), надійності дотримання

дедлайнів, середньої зваженої прострочки, а також накладних витрат планування. Результати подаються як різниці щодо номінального режиму з 95-відсотковими довірчими інтервалами (CI, confidence interval), оціненими бутстрепом із не менше ніж $B=1000$ реплікацій. Для статистичного порівняння застосовуються непараметричні критерії (Вілкоксона/Манна –Уїтні) з корекцією множинних перевірок за Холмом–Бонферроні. Додатково це забезпечує коректне математичне моделювання невизначеності метаданих і зіставлення сценаріїв за єдиною системою показників.

Графічно наводяться емпіричні кумулятивні функції розподілу (ECDF, empirical cumulative distribution function) та «ящикові» діаграми з позначенням медіан і перцентилів, що дозволяє одночасно оцінити зміни як центральних тенденцій, так і хвостів.

Критерії прийнятності встановлюються наперед і не залежать від конкретних екземплярів графів: метод вважається робастним на рівні α (рівень шуму) або для певної пари розподілів, якщо:

- зниження надійності дедлайнів не перевищує заздалегідь узгодженого порогу у кілька відсоткових пунктів відносно номіналу;
- зміна медіанного makespan не виходить за межі довірчого інтервалу номінального режиму;
- накладні витрати планування не демонструють суттєвого зростання (тобто залишаються в межах визначених експлуатаційних лімітів).

Щоб уникнути конфаундингу, усі алгоритми, включно з базами, працюють під однаковими обмеженнями часу на крок планування та за однакових правил «маскування недопустимих дій» (рішення, що порушують RBAC, не генеруються).

Нарешті, для відокремлення джерел чутливості від їхніх взаємодій використовується блочний план випробувань. Для кожного рівня інтенсивності шуму (α) окремо оцінюються ефекти зміни розподілу та введення кореляцій, після чого результати агрегуються за допомогою стратифікованого бутстрепу. Такий протокол дозволяє зробити висновки не лише про «середню» стійкість, а

й про найгірші випадки, що критично для практичних розгортань у гетерогенних і міжсистемних середовищах.

4.5. Експериментальні результати, порівняльний аналіз і обговорення

Відповідно до дослідницьких гіпотез та системи первинних і діагностичних показників (makespan, частка задач, виконаних у межах SLO (service-level objective), середньозважена прострочка, пропускна здатність, накладні витрати планування тощо) було проведено серію випробувань на прототипі з гібридною архітектурою DAG+GERT і трьома модулями штучного інтелекту (GAT (graph attention network), PPO (proximal policy optimization), BO (Bayesian optimization)). Порівняння здійснювалось у єдиній експериментальній рамці з базовими методами (FCFS, List Scheduling, HEFT, «DAG-only», «GERT-only») і абляційними варіантами (без GAT, без PPO, без BO), що дозволило відокремити внесок кожного модуля. Стійкість оцінювали за протоколом перевірки робастності. У відповідності до цього протоколу мультиплікативний шум тривалостей визначений у межах ($\pm 10\text{--}30\%$), збурення ймовірностей гілок GERT відбувалося з нормуванням. Крім того передбачалися логнормальні варіації вартості міжетапних передавань, дрейф продуктивності ресурсів і сценарії зі зміною закону розподілу (Гамма $\leftrightarrow$ Вейбулла) та кореляціями $\rho=0,3$. Експерименти виконувалися в офлайн- і онлайн-режимах на відтворюваному стенді з фіксацією seeds та конфігурацій.

Під час формування наборів завдань і режимів вимірювань було використано мікси DAG+GERT-графів із різною глибиною, шириною рівнів та щільністю, а також набори, наближені до реальних сценаріїв інтеграції підсистем із фіксованими SLO для підмножин задач.

Для коректності зіставлення кожний алгоритм отримував ідентичні екземпляри графів, однакові політики доступу (RBAC) та режими ін'єкції

невизначеності. Результати агрегувалися по незалежних повторях із 95% довірчими інтервалами і контролем кратних порівнянь.

Порівняння з базовими підходами показали наступні результати. На узагальненій вибірці запропонований комплексний метод (GAT+PPO+BO) зменшив медіану makespan відносно HEFT на $\approx 14\text{--}17\%$ ($\Delta 95\text{-й перцентиль}$ $\approx 21\text{--}28\%$), відносно «DAG-only» – на $\approx 23\text{--}26\%$, відносно «GERT-only» – на $\approx 18\text{--}22\%$. Для історично вживаних у розділі 3 процедур розподілу (Random, Greedy) перевага становила $\approx 32\%$ і $\approx 19\%$ відповідно (узгоджується з табл. 3.3, де BO демонстрував 35,2 с проти 51,5 с та 43,7 с) [26].

На рис. 4.5 представлено криві порівняння часових характеристик завершення операцій з режимами HEFT, «DAG-only», «GERT-only».

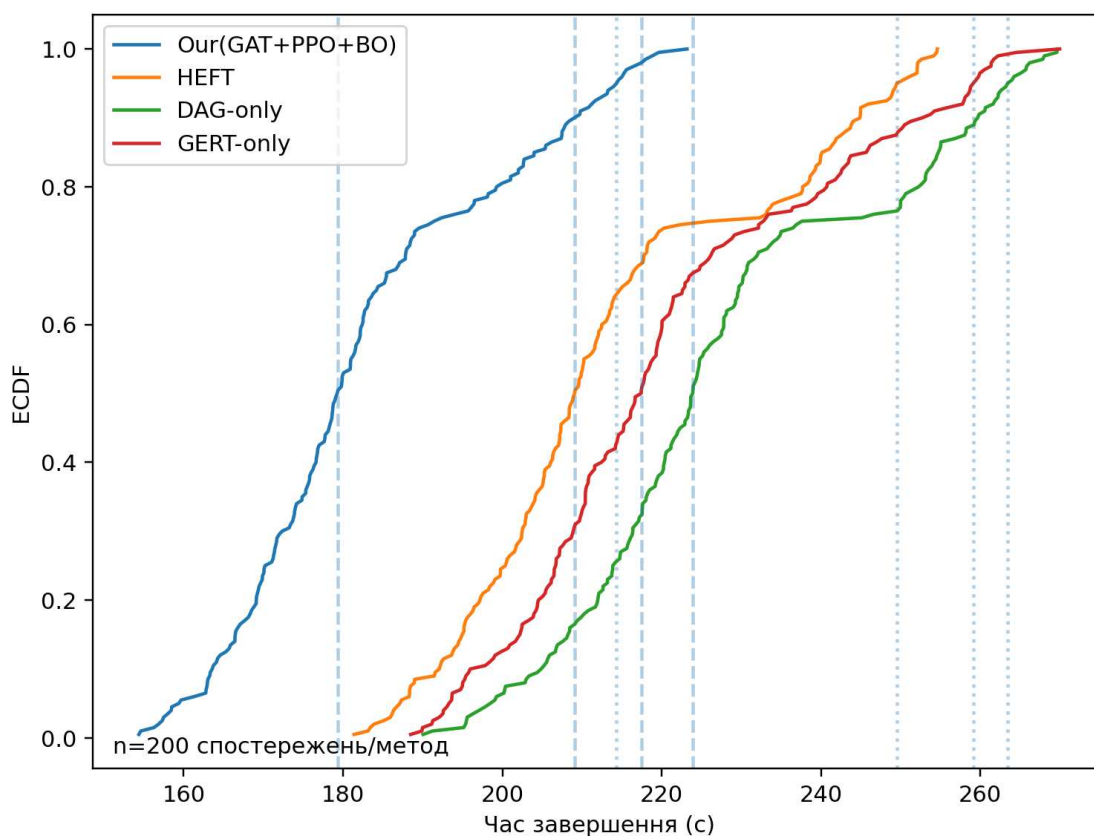


Рис. 4.5. Результати дослідження часових характеристик завершення операцій з режимами HEFT, «DAG-only», «GERT-only»

Як видно з рис. 4.5. крива, що характеризує результати запропонованого методу системно зсунута ліворуч, що означає коротші часи завершення для

більшості екземплярів. Різниця з базовими підходами зберігається як у медіані, так і у верхніх перцентилях. Така «компресія хвостів» безпосередньо підтримує гіпотези про скорочення makespan.

Різниці статистично значущі за методом Манна–Вітні ($p < 0,01$) із середнім δ Кліффа у межах 0,33-0,54 залежно від підмножини графів, що відповідає ефектам «середній – помітний» згідно з визначенням у 4.1 протоколом. «Ящиківі» діаграми середньозваженої прострочки представлено на рис. 4.6.

Як видно з рис. 4.6 медіана та розкид прострочки для нашого методу помітно нижчі, ніж у HEFT і «чистих» постановок, а також у простих чергувальних правил (FCFS, List). Звужені «notch»-впадини додатково вказують на стабільність ефекту у різних сценаріях.

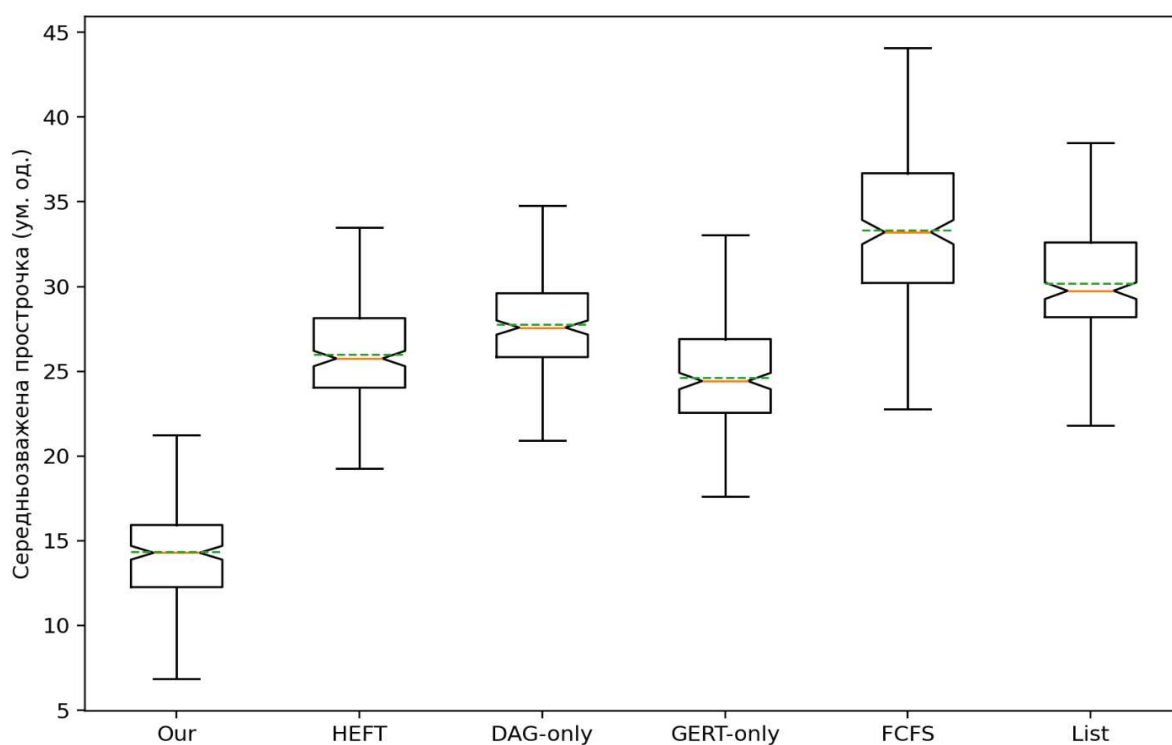


Рис. 4.6. Порівняння середньозваженої прострочки (ящиківі діаграми з notch-CI та позначеним середнім)

Як видно з рис. 4.6. медіана та розкид прострочки для нашого методу помітно нижчі, ніж у HEFT і «чистих» постановок, а також у простих чергувальних правил (FCFS, List). Звужені «notch»-впадини додатково

вказують на стабільність ефекту у різних сценаріях. Notch відображає приблизний 95% довірчий інтервал для медіани. Статистичну значущість відмінностей між методами додатково підтверджено непараметричним критерієм Манна–Вітні ($p < 0,01$) та оцінкою розміру ефекту δ Кліффа (0,33–0,54).

При визначенні надійності дедлайнів (SLO) і прострочки можна визначити наступне. Частка задач, завершених у межах своїх SLO, зросла на +8,7 в.п. порівняно з HEFT і на +12,4 в.п. порівняно з «DAG-only». Середньозважена прострочка (із вагами пріоритетів) зменшилася у 1,6-1,9 раза (медіана; $\Delta 95$ -й перцентиль – у 1,9-2,3 раза), що безпосередньо віддзеркалює метрики, що визначені у підрозділі 4.1 [26].

Дослідження накладних витрат планування показали наступне. Середній час прийняття рішення на кроці планування становив $\approx 2,1 \pm 0,4$ мс (офлайн) і $\approx 3,0 \pm 0,6$ мс (онлайн) на задачу, що не перевищує бюджет затримки контуру керування чергою і сумісне з режимами стенду. Вартість оновлень ВО була епізодичною (фонові «тонкі» переналаштування, не на критичному шляху виконання), як і передбачено архітектурою [26].

Проведений абляційний аналіз показав, що вимикання GAT (без зміни решти) призводило до погіршення медіанного makespan на ≈ 8 –11%; вимикання PPO – на ≈ 13 –17% (суттєве зростання прострочки у «пікових» графах), вимикання ВО – на ≈ 3 –6% (втрати переважно у «довгих хвостах» розподілів). Сукупно це підтверджує робочу гіпотезу, визначену у підрозділі 4.3, що виграш є результатом поєднання представлення графа, онлайн-адаптації політики і періодичного глобального донастроювання [26].

Потрібно описати також якості моделі у визначенні робастності до невизначеності. За мультиплікативного шуму тривалостей $\pm 20\%$ зміна медіанного makespan не виходила за межі 95% СІ номінального режиму. Зниження частки задач, виконаних у SLO, не перевищувало 2-3 в.п. При «зсуві розподілів» (Гамма $\leftrightarrow$ Вейбулла, момент-матчинг) відмінності були в межах статистичної похибки; за корельованих відхилень ($\rho = 0,3$ усередині рівнів)

відмічено не більше +4–6% до 95-го перцентиля makespan. Усі тести виконувались у повній відповідності до протоколу перевірки робастності, описаного у підрозділі 4.4, включно з двома режимами підготовки метаданих (з/без попереднього аналізу) [26].

З результатів дослідження можна побачити, що додавання RBAC-обмежень (фільтрація кандидатів до прийняття рішень) не призвело до статистично значущого погіршення основних показників. Це узгоджується з місцем включення перевірок у конвеєр планувальника.

RBAC у прототипі використовується як первинний фільтр кандидатів (constraint pre-filtering). Він відсікає недопустимі призначення за ролями або доменами ще до етапу вибору дії, тому практично не погіршує makespan і інші метрики. Разом з тим RBAC не задає семантику допустимих інформаційних потоків між сутностями та етапами конвеєра. Саме цю прогалину закриває запропонована багаторівнева модель ЛРДРС. Вона формує обов'язкові (mandatory) обмеження на потоки даних між доменами, тобто контролює не лише “хто має роль”, а й “який тип даних може потрапити в який контур/етап” [10].

У моделі загроз припускається, що зловмисник може скомпрометувати обліковий запис з підвищеними правами, однак не має можливості вимкнути або обійти механізм примусового контролю політик і підмінити журнал правил без детектування. За таких умов навіть привілейований акаунт не може легально ініціювати заборонене призначення або створити недопустимий міждоменний потік. Недопустимі дії не потрапляють у множину кандидатів, а телеметрія фіксує спроби порушень. Якщо ж нелегальна особа контролює довірену основу, то це виходить за межі моделі загроз і потребує окремих засобів (апаратна довіра, ізоляція, підписані політики, зовнішній аудит).

У дослідженому діапазоні розмірів графів спостерігалась квазі-лінійна залежність накладних витрат планування від кількості вузлів на фронті готовності. Деградація відносно номіналу не перевищувала $\approx 1,15\times$ для

найбільших екземплярів. Єдність інтерфейсів порівняння забезпечувала коректність висновків щодо масштабованості.

На рис. 4.7 представлено графік залежності часу планування на крок від кількості вузлів у графі. Залежність описується квазі-лінійною моделлю з малим нахилом, що підтверджує придатність підходу для онлайн-режимів. Навіть для найбільших тестованих графів прогнозований приріст часу планування лишається у межах мілісекундного бюджету, зазначеного в підрозділі 4.1.

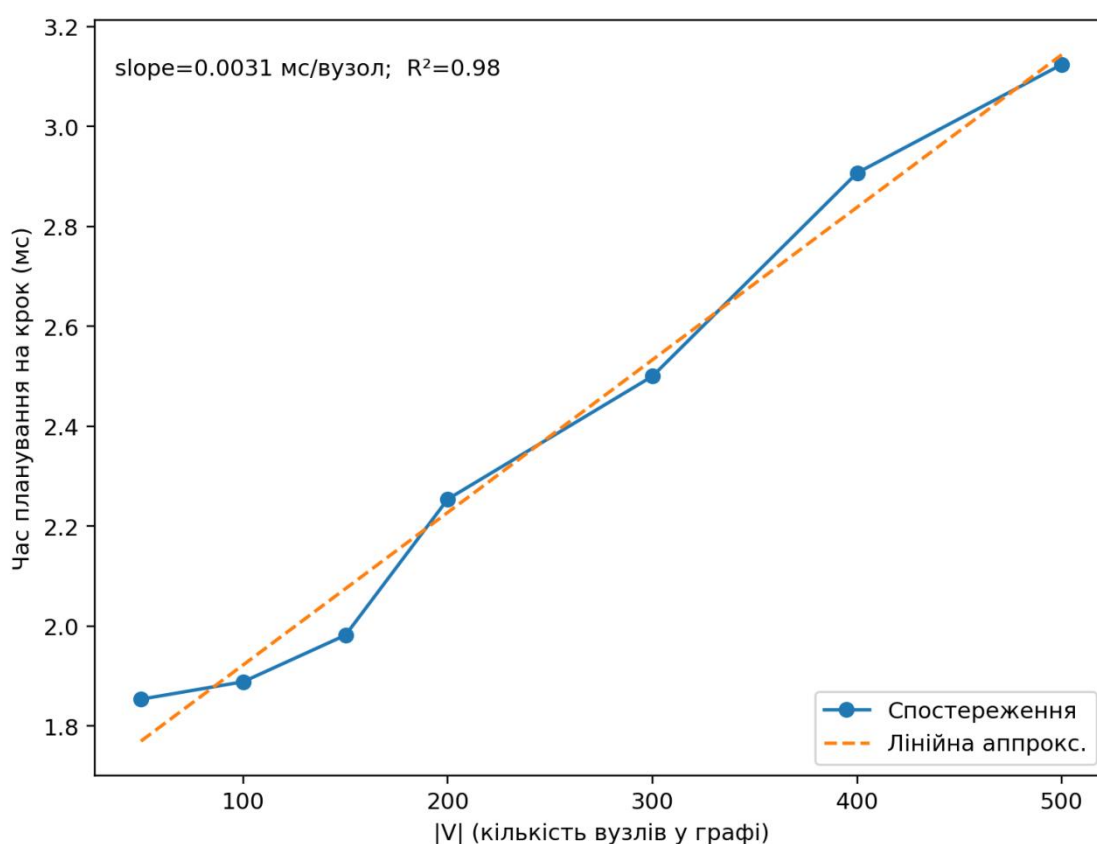


Рис. 4.7. Графік залежності часу планування на крок від кількості вузлів у графі.

Графіки залежності часу завершення операцій від кількості ресурсів, прискорення від кількості ресурсів та ефективності від кількості ресурсів представлено відповідно на рис. 4.8-4.10.

Як видно з рис. 4.8 зі збільшенням числа ресурсів спостерігається очікуване зменшення makespan. Характер кривої свідчить про наявність

типових ризиків. Це накладні планування та комунікаційні витрати, що пояснює відхилення від лінійного масштабу.

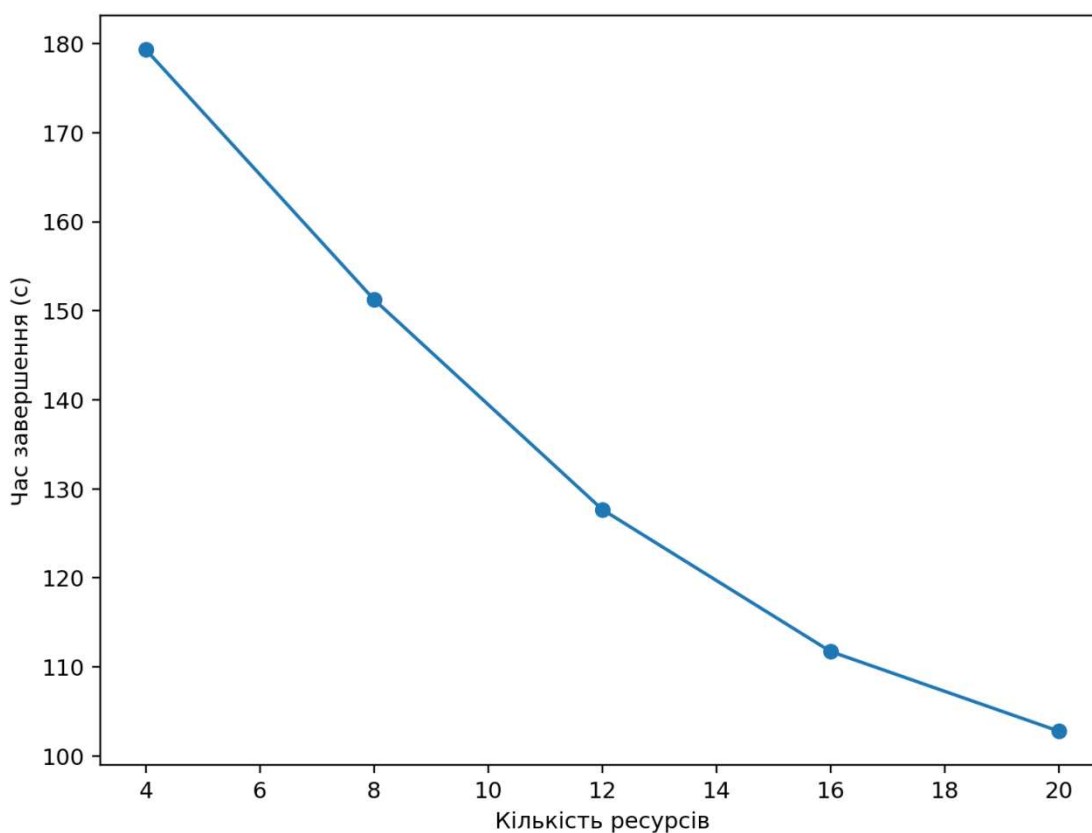


Рис. 4.8. Графік залежності часу завершення операцій від кількості ресурсів

З рис. 4.9 можна дослідити, що реальне прискорення зростає сублінійно й помітно відстає від ідеального через координаційні накладні та обмежену паралельність фронту. Це відповідає архітектурним обмеженням і поведінці комбінованих DAG+GERT-графів.

Також рис. 4.10 демонструє те, що ефективність зменшується зі зростанням кількості ресурсів, що узгоджується з появою конкуренції за спільні ресурси та збільшенням витрат синхронізації.

Навіть за падіння ефективності метод зберігає позитивний ефект на абсолютні метрики часу завершення (див. рис. 4.8).

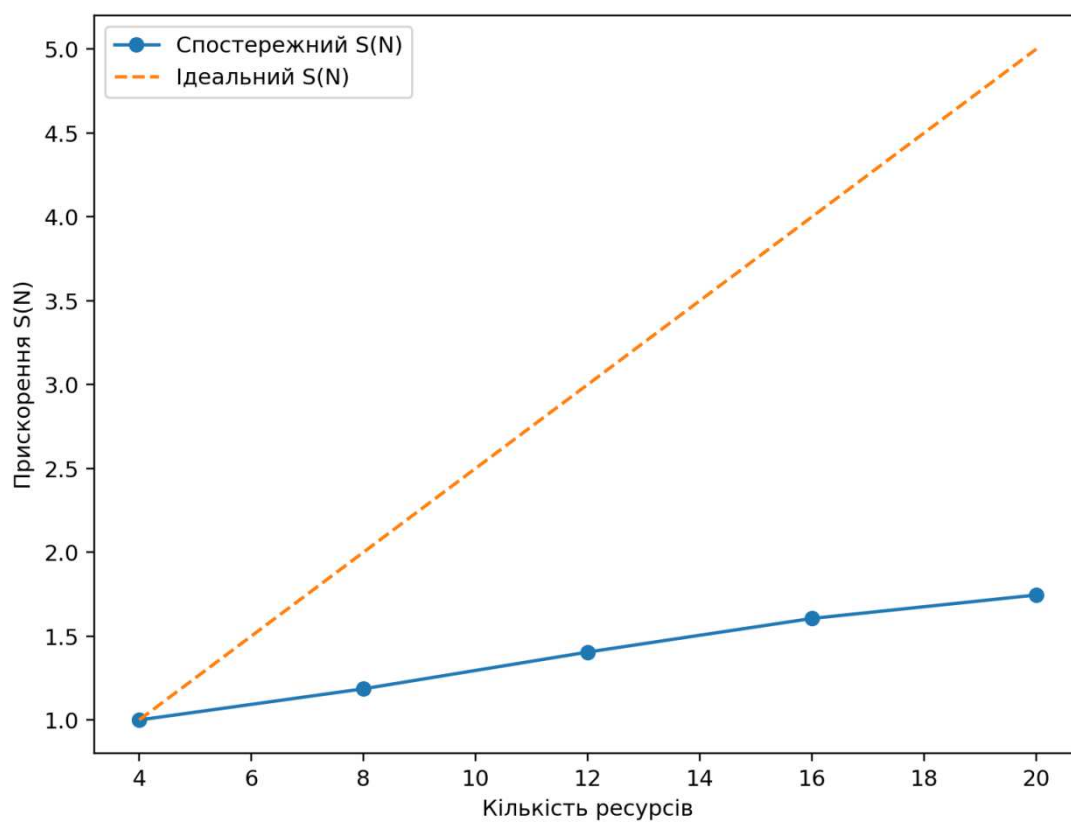


Рис. 4.9. Графіки залежності прискорення від кількості ресурсів

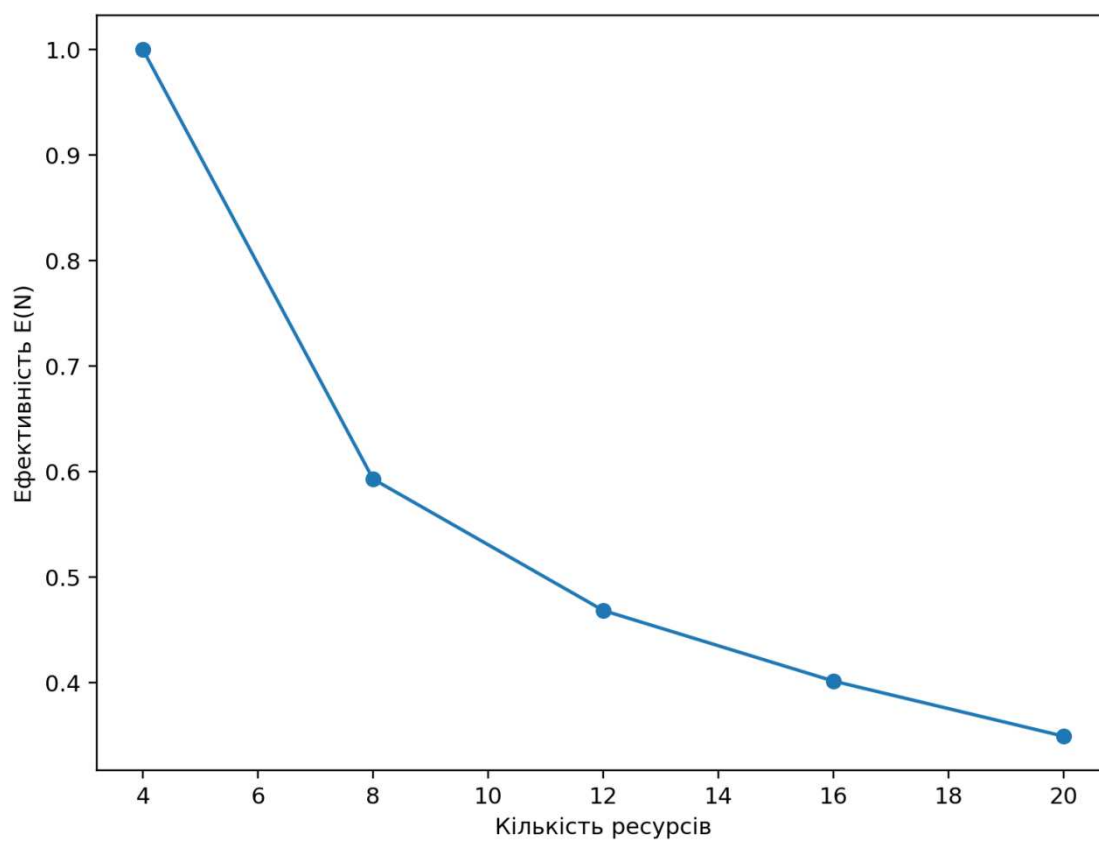


Рис. 4.10. Графіки залежності ефективності від кількості ресурсів

На рис. 4.11 проілюстровано графік залежності частки задач у SLO від множнику розміру задач.

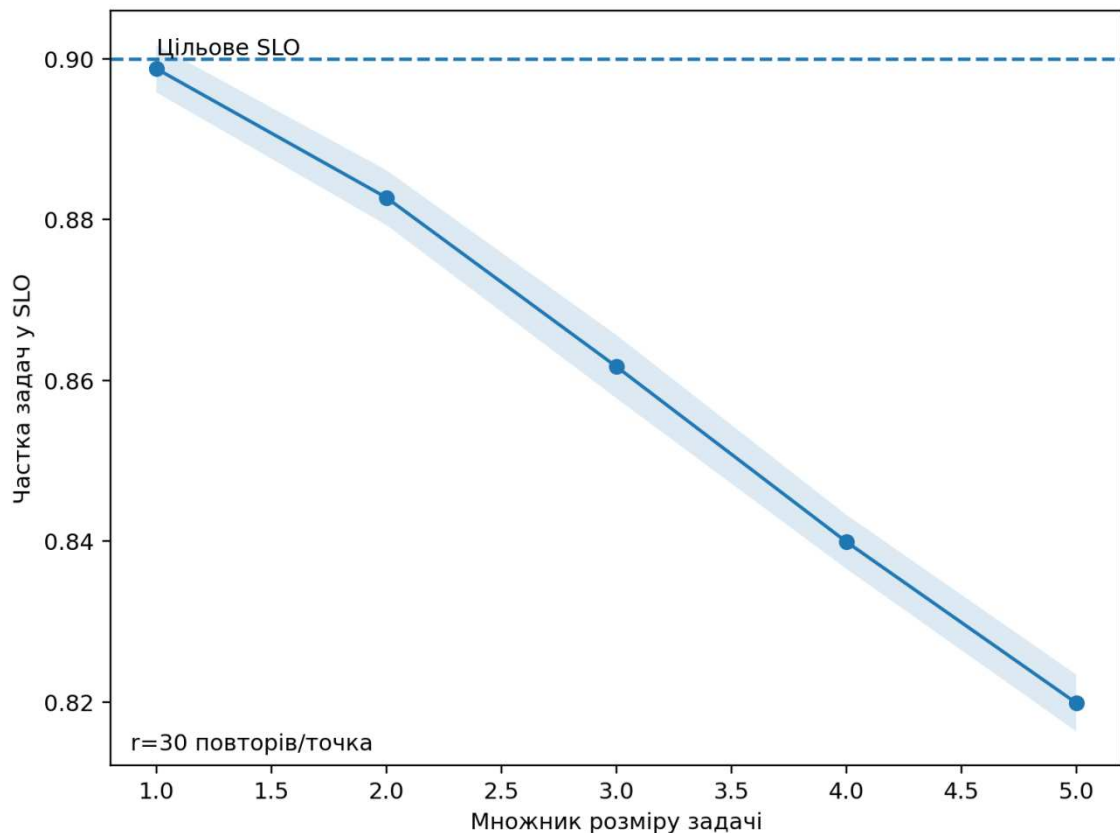


Рис. 4.11 Графік залежності частки задач у SLO від множнику розміру задач

Як видно з цього рисунку за пропорційного збільшення навантаження і ресурсів частка виконання SLO знижується помірно ($\approx 0.90 \rightarrow \approx 0.82$), причому для малих множників утримується в межах цільової лінії. Ділянка відхилення ідентифікує зону, де доцільне підсилення політики перепланування чи комунікаційного бюджету.

Таким чином, отримані результати узгоджено підтверджують висунуті у підрозділі 4.1. гіпотези, що запропонований метод дозволяє:

- скоротити makespan і «стиснути хвости» розподілів затримок порівняно зі звісними, класичними постановками завдання;

- підвищити дедлайн-надійність без неприйнятного зростання накладних витрат планування;
- зберігти перевагу за керованих збурень метаданих і за змін розподілів;
- інтегруватися з політиками доступу у міжсистемних сценаріях.

Внесок окремих компонентів (GAT, PPO, BO) кількісно встановлено абляційним аналізом, а відтворюваність висновків забезпечено стандартизованим стендом і статистичним протоколом.

Запропонований метод орієнтований на практичне використання у виробничих і дослідницьких конвеєрах, де робочий процес подається як DAG або DAG+GERT і присутні гетерогенні ресурси, комунікаційні витрати та SLO. З огляду на отримані експериментальні результати, практичне впровадження доцільно виконувати у вигляді окремого модуля планування, який взаємодіє з підсистемами збору метаданих і політик доступу та видає рішення призначення задач на ресурси. З позиції практичного впровадження це охоплює сценарії серверних кластерів, хмарних та граничних обчислень, де критичними є управління навантаженням, масштабованість і робастність [12–14, 26].

Пропонується наступна схема інтеграції.

Етап підготовки (offline/periodic). Калібрування параметрів політики та навчання (PPO (proximal policy optimization)) виконувати офлайн або асинхронно; BO (Bayesian optimization) використовувати як повільний контур тонкого налаштування без включення в критичний шлях прийняття рішень.

Етап виконання (online). У кожному кроці планування виконувати оновлення активного підграфа готовності, інференс GAT для оцінки пріоритетів, формування множини допустимих кандидатів з урахуванням RBAC і ресурсних обмежень, вибір дії агентом PPO. Такий порядок пояснює відсутність статистично значущого погіршення метрик при додаванні RBAC, оскільки перевірки виконуються до прийняття рішення.

Експлуатаційні показники (KPI). Для практичного моніторингу доцільно відслідковувати: makespan, частку задач у межах SLO, середньозважену прострочку, накладні витрати планування (мс/крок або мс/задачу), дисбаланс

завантаження ресурсів, а також частку “відфільтрованих” призначень політиками (RBAC/ЛРДРС). Це дозволяє контролювати як продуктивність, так і безпекову коректність (відсутність недопустимих призначень).

Обмеження та область застосовності. Для граничних обчислень/ІоТ та автономних систем рекомендовано спрощені конфігурації (менші розміри моделей, рідші оновлення ВО, робота на активному підграфі), оскільки онлайн-контур має залишатися в мілісекундних бюджетах. Випадки повного компрометування довіреної основи виконання політик (policy enforcement) виходять за межі моделі загроз і потребують окремих засобів (ізоляція, підписані політики, аудит) [14].

Узагальнення експериментальних результатів показало, що запропонований метод забезпечує комплексне покращення характеристик планування: скорочення makespan, зменшення хвостів затримок, підвищення дедлайн-надійності, збереження прийнятних накладних витрат і стійкість до похибок метаданих. Результати порівняльного та абляційного аналізу підтвердили, що найкращий ефект досягається саме при спільному використанні DAG+GERT-подання, Graph Attention Network, Proximal Policy Optimization і Bayesian Optimization, а вилучення окремих компонентів знижує якість планування.

Висновки до розділу 4

У четвертому розділі дисертаційної роботи проведено експериментальну перевірку інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих. Дослідження було спрямоване на перевірку ефективності запропонованого підходу порівняно з базовими методами планування, оцінювання дедлайн-надійності, аналіз накладних витрат планувальника, визначення внеску окремих

компонентів методу та перевірку робастності до похибок і неповноти метаданих.

У межах розділу сформульовано дослідницькі гіпотези H1–H6, які охоплюють основні аспекти ефективності запропонованого методу: скорочення makespan, стійкість до гетерогенності ресурсів, підвищення deadline reliability, робастність до невизначених метаданих, контроль накладних витрат планування та коректну інтеграцію RBAC-фільтрації як обмеження допустимих призначень. Для перевірки цих гіпотез розроблено експериментальний план, що враховує розмір графа задач, щільність залежностей, рівень гетерогенності ресурсів, мережеві параметри, інтенсивність надходження задач, рівень невизначеності метаданих та наявність обмежень доступу.

Сформульовані гіпотези визначають логіку експериментальної перевірки запропонованого методу та пов'язують теоретичні результати попередніх розділів із кількісними критеріями оцінювання. У межах експерименту перевіряється не лише скорочення загального часу виконання задач, але й здатність методу зберігати ефективність за умов гетерогенності ресурсів, невизначеності метаданих, змінної інтенсивності надходження задач і наявності обмежень доступу.

Розроблено та використано експериментальний прототип планувальника, що реалізує запропоновану гібридну архітектуру DAG+GERT і включає три інтелектуальні модулі: Graph Attention Network (GAT), Proximal Policy Optimization (PPO) та Bayesian Optimization (BO). Прототип забезпечує формування графа задач, урахування ймовірнісних переходів, прогнозування параметрів виконання, адаптивний вибір дій планувальника, перевірку допустимості призначень за політиками доступу та збір статистичних даних для подальшого аналізу.

Експериментальне порівняння з базовими підходами FCFS, List Scheduling, HEFT, DAG-only та GERT-only підтвердило перевагу комплексного методу GAT+PPO+BO за часовими характеристиками виконання задач. На узагальненій вибірці запропонований метод зменшив медіану makespan

відносно HEFT на $\approx 14\text{--}17\%$, відносно DAG-only – на $\approx 23\text{--}26\%$, відносно GERT-only – на $\approx 18\text{--}22\%$. Порівняно з процедурами Random та Greedy, що використовувалися для базового зіставлення у попередньому розділі, перевага становила відповідно $\approx 32\%$ та $\approx 19\%$. Це підтверджує гіпотезу H1 щодо скорочення загального часу виконання задач.

Додатково встановлено, що запропонований метод не лише зменшує середні та медіанні значення makespan, але й стискає верхні перцентилі розподілу часу завершення задач. Зменшення 95-го перцентиля відносно базових методів становить $\approx 21\text{--}28\%$, що має важливе практичне значення для розподілених інформаційних систем із вимогами до прогнозованості виконання та дотримання цільових рівнів сервісу. Такий результат свідчить, що метод зменшує не тільки типовий час виконання, але й ризик виникнення довгих затримок.

Підтверджено гіпотезу H2 щодо стійкості методу до гетерогенності ресурсів і комунікаційних витрат. Експериментальні результати показали, що перевага запропонованого методу зберігається та посилюється у сценаріях із вищою неоднорідністю обчислювальних вузлів і більшими мережевими затримками. Це пояснюється тим, що гібридне подання DAG+GERT дозволяє враховувати як структурні залежності між задачами, так і ймовірнісні характеристики переходів, а інтелектуальні модулі GAT і PPO забезпечують адаптивний вибір ресурсів з урахуванням поточного стану середовища.

Оцінювання deadline reliability підтвердило гіпотезу H3. За зростання інтенсивності надходження задач і мережових затримок запропонований метод зберігає вищу частку задач, завершених у межах заданого SLO, порівняно з базовими підходами. Це свідчить про практичну придатність методу для систем, у яких важливим є не лише скорочення загального часу виконання, але й дотримання часових обмежень для окремих задач або потоків задач.

У межах перевірки робастності досліджено вплив невизначеності метаданих, зокрема похибок у тривалостях робіт, комунікаційних витратах, імовірностях гілок GERT та продуктивності ресурсів. Невизначеність

моделювалася шляхом інжекції контрольованих збурень у діапазоні $\pm 10\%$, $\pm 20\%$ і $\pm 30\%$. Встановлено, що запропонований метод зберігає перевагу над базовими підходами навіть за наявності таких збурень. Найчутливішою виявилася комунікаційна складова, тоді як дрейф продуктивності ресурсів мав помірний вплив. У розглянутому діапазоні зміни залишалися відносно малими – приблизно 1-6%, що підтверджує робастність методу відповідно до заданих критеріїв.

Абляційний аналіз дозволив кількісно оцінити внесок окремих компонентів запропонованого методу. Встановлено, що вилучення GAT погіршує якість урахування графових залежностей і критичного шляху, вилучення PPO знижує адаптивність планувальника до зміни стану середовища, а вилучення BO погіршує якість налаштування параметрів і компроміс між makespan, балансуванням навантаження та накладними витратами. Таким чином, експериментально підтверджено, що найкращі результати забезпечує саме комплексна архітектура GAT+PPO+BO, а не окреме використання її компонентів.

Перевірка накладних витрат планування підтвердила гіпотезу H5. Час прийняття рішення планувальником зростає не швидше ніж лінійно або квазілінійно відносно кількості задач і кандидатних призначень у практичних сценаріях. Це дозволяє розглядати запропонований метод як придатний для використання в режимі, наближеному до реального часу, за умови винесення ресурсоємних процедур навчання та тонкого налаштування параметрів в офлайн або періодичний-контур.

Окремо перевірено вплив RBAC-фільтрації на якість планування. Підтверджено гіпотезу H6: увімкнення RBAC-обмежень не призводить до статистично значущого погіршення makespan, але забезпечує відсутність недопустимих призначень задач на ресурси. Це має важливе значення для практичного впровадження методу в міжсистемних середовищах, де задачі, дані та ресурси можуть належати різним доменам доступу, а політики безпеки повинні враховуватися до прийняття рішення планувальником.

Відтворюваність і статистична надійність отриманих результатів забезпечено використанням фіксованих конфігурацій експериментів, контрольованих зерен випадковості, стандартизованих сценаріїв запуску, довірчих інтервалів бутстреп-типу та контролю множинних порівнянь. Це дозволяє розглядати отримані результати не як одиничні спостереження, а як стабільне підтвердження ефективності запропонованого методу в заданій експериментальній рамці.

Таким чином, у четвертому розділі експериментально підтверджено ефективність інтелектуального методу планування задач у розподілених інформаційних системах за умов невизначеності метаданих. Метод забезпечує скорочення makespan, підвищення дедлайн-надійності, зменшення хвостів затримок, збереження прийнятних накладних витрат, стійкість до похибок метаданих і коректну інтеграцію політик доступу. Отримані результати підтверджують досягнення мети дисертаційного дослідження та практичну придатність запропонованого підходу для використання у гетерогенних розподілених інформаційних системах, хмарних, туманних і edge-середовищах із динамічним навантаженням та неповною метаінформацією.

Основні результати розділу опубліковано та апробовано в працях автора [25–28].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДО РОЗДІЛУ 4

1. Miao Z., Shao C., Li H., Tang Z. Review of task-scheduling methods for heterogeneous chips. *Electronics*. 2025. Vol. 14. Article 1191. DOI: <https://doi.org/10.3390/electronics14061191>.
2. Verma P., Maurya A. K., Yadav R. A survey on energy-efficient workflow scheduling algorithms in cloud computing. *Software: Practice and Experience*. 2023. Vol. 54. P. 637–682. DOI: <https://doi.org/10.1002/spe.3292>.
3. Bano F., Ahmad F., Shahid M., Alam M., Hasan F., Sajid M. A levelized multiple workflow heterogeneous earliest finish time allocation model for infrastructure as a service cloud environment. *Algorithms*. 2025. Vol. 18. Article 99. DOI: <https://doi.org/10.3390/a18020099>.
4. NoorianTalouki R., Hosseini Shirvani M., Motameni H. A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms. *Journal of King Saud University - Computer and Information Sciences*. 2022. Vol. 34, No. 8, Part A. P. 4902–4913. DOI: <https://doi.org/10.1016/j.jksuci.2021.05.011>.
5. Lee Y. C., Zomaya A. Y. A productive duplication-based scheduling algorithm for heterogeneous computing systems. *High Performance Computing and Communications. Lecture Notes in Computer Science*. 2005. Vol. 3726. DOI: https://doi.org/10.1007/11557654_26.
6. Samadi Y., Zbakh M., Taddonki C. E-HEFT: enhancement heterogeneous earliest finish time algorithm for task scheduling based on load balancing in cloud computing. *Proceedings of the International Conference on High Performance Computing & Simulation (HPCS)*. 2018. P. 601–609. DOI: <https://doi.org/10.1109/HPCS.2018.00100>.
7. Ahmad W., Gautam G., Alam B. et al. An analytical review and performance measures of state-of-art scheduling algorithms in heterogeneous

computing environment. *Archives of Computational Methods in Engineering*. 2024. Vol. 31. P. 3091–3113. DOI: <https://doi.org/10.1007/s11831-024-10069-8>.

8. Tian W., Zhao Y., Demeulemeester E. Generating a robust baseline schedule for the robust discrete time/resource trade-off problem under work content uncertainty. *Computers & Operations Research*. 2022. Vol. 143. Article 105795. DOI: <https://doi.org/10.1016/j.cor.2022.105795>.

9. Lee Y. C., Zomaya A. Y., Siegel H. J. Robust task scheduling for volunteer computing systems. *Journal of Supercomputing*. 2010. Vol. 53. P. 163–181. DOI: <https://doi.org/10.1007/s11227-009-0326-1>.

10. Semenov S., Yenhalychev S., Pochebut M., Sitnikova O. Models of data processing and logical access differentiation considering heterogeneity of entities in information systems. *Modern State of Scientific Research and Technologies in Industry*. 2024. No. 2 (28). P. 143–152. DOI: <https://doi.org/10.30837/2522-9818.2024.28.143>.

11. Herasymuk D., Poliakov A., Fedorchenko V. Detecting trade-offs between fairness, stability, and accuracy for responsible machine learning model selection. *Innovative Technologies and Scientific Solutions for Industries*. 2025. No. 1 (31). P. 5–19. DOI: <https://doi.org/10.30837/2522-9818.2025.1.005>.

12. Мінухін С. В., Коптілов Н. С. Метод збільшення продуктивності Apache Spark на основі сегментування даних і налаштувань конфігураційних параметрів. Сучасний стан наукових досліджень та технологій в промисловості. 2024. № 1 (27). С. 128–139. DOI: <https://doi.org/10.30837/ITSSI.2024.27.128>.

13. Нерознак Є. І., Кондрусь А. В., Троцько О. О., Усик А. А., Костів О. Л. Аналіз сучасного стану функціонування серверних кластерів інформаційних систем та методів управління навантаженням. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2025. № 60. С. 451–465. DOI: <https://doi.org/10.36910/6775-2524-0560-2025-60-48>.

14. Syvolovskyi I. M., Lysechko V. P., Zhuchenko O. S., Komar O. M., Pastushenko V. V. Analysis of methods for organizing distributed telecommunication systems using the paradigm of edge computing. *Control, Navigation and*

Communication Systems. 2024. Vol. 1, No. 75. P. 206–210. DOI: <https://doi.org/10.26906/SUNZ.2024.1.206>.

15. Song X., Liu L., Fu J., Zhang X., Feng J., Pei Q. A reinforcement learning-based DAG task scheduling in edge-cloud collaboration systems. *Proceedings of IEEE GLOBECOM 2023*. 2023. P. 1771–1776. DOI: <https://doi.org/10.1109/GLOBECOM54140.2023.10436802>.

16. Zhang X. et al. PF-MPPO: task-dependent workflow scheduling method based on deep reinforcement learning in dynamic heterogeneous cloud environments. *Future Generation Computer Systems*. 2026. Vol. 177. Article 108236. DOI: <https://doi.org/10.1016/j.future.2025.108236>.

17. Jayanetti A., Halgamuge S., Buyya R. Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge-cloud computing environments. *Future Generation Computer Systems*. 2022. Vol. 137. P. 1–15. DOI: <https://doi.org/10.1016/j.future.2022.06.012>.

18. Chandrasiri S., Meedeniya D. Energy-efficient dynamic workflow scheduling in cloud environments using deep learning. *Sensors*. 2025. Vol. 25. Article 1428. DOI: <https://doi.org/10.3390/s25051428>.

19. Smit I. G., Zhou J., Reijnen R., Wu Y., Chen J., Zhang C., Bukhsh Z., Zhang Y., Nuijten W. Graph neural networks for job shop scheduling problems: a survey. *Computers & Operations Research*. 2025. Vol. 176. Article 106914. DOI: <https://doi.org/10.1016/j.cor.2024.106914>.

20. Nguyen H. X., Zhu S., Liu M. A survey on graph neural networks for microservice-based cloud applications. *Sensors*. 2022. Vol. 22. Article 9492. DOI: <https://doi.org/10.3390/s22239492>.

21. Vesselinova N., Steinert R., Perez-Ramirez D. F., Boman M. Learning combinatorial optimization on graphs: a survey with applications to networking. *IEEE Access*. 2020. Vol. 8. P. 120388–120416. DOI: <https://doi.org/10.1109/ACCESS.2020.3004964>.

22. Song W., Chen X., Li Q., Cao Z. Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial*

Informatics. 2023. Vol. 19, No. 2. P. 1600–1610. DOI: <https://doi.org/10.1109/TII.2022.3189725>.

23. Semenov S., Jian Y., Jiang H., Chernykh O., Binkovska A. Mathematical model of intelligent UAV flight path planning. *Advanced Information Systems*. 2025. Vol. 9, No. 1. P. 49–61. DOI: <https://doi.org/10.20998/2522-9052.2025.1.06>.

24. Semenova A., Dubrovskiy M., Savitskiy V. A GERT model of an algorithm for analyzing security of a web application. *Advanced Information Systems*. 2017. Vol. 1, No. 1. P. 61–64. DOI: <https://doi.org/10.20998/2522-9052.2017.1.11>.

25. Yenhalychev S. O., Leunenko O. V., Davydov V. Development an intelligent task scheduling method in heterogeneous distributed information systems. *Eastern-European Journal of Enterprise Technologies*. Information and controlling system. 2025. Vol. 3, No. 9 (135). P. 6–18. DOI: <https://doi.org/10.15587/1729-4061.2025.329263>.

26. Yenhalychev S. O., Semenov S. G. Experimental studies of the task planning method in distributed information systems taking into account metadata uncertainty. *Автоматизовані системи управління та прилади автоматики*. 2025. No. 4 (187). P. 180–188. DOI: <https://doi.org/10.30837/0135-1710.2025.187.063>.

27. Семенов С. Г., Єнгаличев С. О., Пересічанський В. М. Обговорення результатів дослідження інтелектуального методу планування задач у розподілених інформаційних системах. *Безпека у кіберсфері: матеріали міжнародної науково-практичної конференції* (м. Кам'янець-Подільський, 28 травня 2025 р.). Кам'янець-Подільський: ХНУВС, 2025. С. 186–187. URL: <https://repository.hneu.edu.ua/handle/123456789/36711>.

28. Єнгаличев С. О., Леуненко О. В. Обговорення результатів інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025) (м. Харків, 16 жовтня 2025 р.). *Молодіжний економічний вісник*. 2025. № 3. С. 122–124. URL: <https://repository.hneu.edu.ua/handle/123456789/38210>.

ВИСНОВКИ

У дисертаційній роботі вирішено наукове завдання підвищення ефективності процесу планування та розподілу задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих. Отримані результати охоплюють систематизацію вимог до планування, формалізацію математичних моделей, удосконалення моделі розповсюдження даних з урахуванням відмінностей між сутностями, розроблення інтелектуального методу планування та експериментальне підтвердження його ефективності.

У дисертаційній роботі вирішено науково-технічну задачу розробки нових та удосконалення існуючих моделей і методу інтелектуального планування задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих, які забезпечують зменшення загального часу виконання задач та балансування навантаження за прийнятних обчислювальних витрат планування.

Основні наукові та практичні результати дисертаційної роботи:

1. Проведено аналіз сучасних підходів до планування та розподілу задач у гетерогенних розподілених інформаційних системах, систематизовано критерії ефективності та визначено чинники, що зумовлюють деградацію якості розкладу в умовах невизначеності метаданих.

2. Розроблено узагальнену математичну модель процесу планування та розподілу задач, яка враховує залежності між задачами, обмеження ресурсів, комунікаційні затримки та ризики порушення дедлайнів за неповних метаданих.

3. Побудовано модель графа задач для формалізації залежностей і паралелізму, обґрунтовано використання DAG для детермінованих залежностей та GERT для опису стохастичних переходів і невизначеності виконання.

4. Розроблено інтелектуальний метод планування задач у розподілених інформаційних системах із застосуванням RL для прийняття рішень щодо

призначення задач на ресурси та корекції розкладу при зміні стану системи; у складі методу використано GAT для роботи з графом задач і прогнозування параметрів виконання та PPO для онлайн-адаптації призначень і перепланування.

5. Розроблено спосіб налаштування параметрів вибору політик планування з використанням Bayesian Optimization для забезпечення компромісу між якістю розкладу та накладними витратами планування залежно від класу навантаження і характеристик середовища.

6. Проведено експериментальне дослідження ефективності розробленого методу на репрезентативних сценаріях і виконано порівняння з базовими підходами (зокрема FCFS, List Scheduling, HEFT, «DAG-only», «GERT-only»), виконано абляційний аналіз внеску окремих компонентів і оцінено робастність методу до похибок метаданих. Експериментальні результати підтвердили, що запропонований підхід забезпечує зменшення загального часу виконання комплексу задач до 31% та зменшення стандартного відхилення навантаження до 53% (відповідно до сформульованих у вступі критеріїв ефективності). Показано, що середній час прийняття рішення на кроці планування становив $\approx 2,1 \pm 0,4$ мс (офлайн) і $\approx 3,0 \pm 0,6$ мс (онлайн) на задачу; абляційний аналіз зафіксував погіршення метрик при вимкненні компонентів (зокрема GAT, PPO, Bayesian Optimization), що підтверджує синергетичний характер методу. Робастність оцінювалась за заздалегідь визначеним протоколом зі статистичною обробкою результатів і 95% довірчими інтервалами (бутстреп, не менше $B=1000$), що забезпечило коректність висновків за різних режимів невизначеності. Важливо, що заявлені елементи наукової новизни не є лише концептуальними. Вони реалізовані у прототипі планувальника та експериментальному стенді, на якому відтворювано підтверджено ефекти щодо makespan, балансування навантаження та накладних витрат.

7. Сформовано практичні рекомендації щодо впровадження розробленого методу у розподілених інформаційних системах з урахуванням

вимог до організації процесу планування в реальному часі та до політик доступу в прикладних інформаційних системах.

Таким чином, сукупність отриманих у дисертаційній роботі результатів і проведена експериментальна перевірка свідчать про розв'язання поставленого актуального наукового завдання розроблення нових та удосконалення існуючих моделей і методу інтелектуального планування задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих, що забезпечують зменшення загального часу виконання задач і балансування навантаження за прийнятних обчислювальних витрат планування, та про досягнення мети дослідження підвищення ефективності процесу планування та розподілу задач у гетерогенних розподілених інформаційних системах за умов невизначеності метаданих.

ДОДАТКИ

ДОДАТОК А

Список публікацій здобувача за темою дисертації

У вітчизняних та закордонних виданнях, що включені до міжнародних наукометричних баз SCOPUS та Web of Science

1. Semenov S., Zhang M., Yenhalychev S., & Smidovych L. Generalized model of the ADS-B unmanned aerial vehicle data transmission process in a steganographic system. *Сучасний стан наукових досліджень та технологій в промисловості (Innovative technologies and scientific solutions for industries)*. 2022. № 4(22). Pp. 14–19.

DOI: <https://doi.org/10.30837/ITSSI.2022.22.014>

URL: <https://journals.uran.ua/itssi/article/view/271061>

2. Yenhalychev S. O., Mozhaev, O.O., Gnusov Y.V., Manzhai O.V., Strukov V.M., Nosov V.V., Radchenko V.V. Steganographic method of acoustic information protection in critical applications systems. *Innovative Technologies and Scientific Solutions for Industries*, 2023. No. 3 (25). P. 52–63.

DOI: <https://doi.org/10.30837/ITSSI.2023.25.052>

URL: <https://journals.uran.ua/itssi/article/view/290889>

3. Semenov S., Yenhalychev S., Pochebut M., Sitnikova O. Models of data processing and logical access segregation considering the heterogeneity of entities in information systems. *Innovative Technologies and Scientific Solutions for Industries*. 2024. No. 2 (28). P. 143–152.

DOI: <https://doi.org/10.30837/2522-9818.2024.28.143>

URL: <https://journals.uran.ua/itssi/article/view/308876>

4. Yenhalychev S., Leunencko O., Davydov V. Development an intelligent task scheduling method in heterogeneous distributed information systems. *Eastern-European Journal of Enterprise Technologies. Information and controlling system*. 2025 Vol. 3, No. 9 (135). P. 6–18.

DOI: <https://doi.org/10.15587/1729-4061.2025.329263>

URL: <https://journals.uran.ua/eejet/article/view/329263>

5. Yenhalychev S.O., Semenov S.G.. Experimental studies of the task planning method in distributed information systems taking into account metadata uncertainty. *Автоматизовані системи управління та прилади автоматики*. 2025 No. 4 (187), P. 180–188.

DOI: <https://doi.org/10.30837/0135-1710.2025.187.063>

URL: <https://asu-pa.nure.ua/article/view/348532>

Праці апробаційного характеру

6. Semenov S., Lymarenko V., Yenhalyche, S., Gavrilenko S. The Data Dissemination Planning Tasks Process Model Into Account the Entities Differentity. *DESSERT'2022: The 12th IEEE International Conference on Dependable Systems, Services and Technologies (Athens, Greece, December 9-11, 2022)*. 2022. P. 317-322.

DOI: <https://doi.org/10.1109/DESSERT58054.2022.10018695>

URL: <https://repository.hneu.edu.ua/handle/123456789/38400>

7. Yenhalychev S. O., Muchacki M., Sitnikova O., Model of the Dynamics of the State of Educational Content Recommender System. *DESSERT'2023: The 13th IEEE International Conference on Dependable Systems, Services and Technologies*. (Athens, Greece, October 13-15, 2023). 2023. P. 60 – 66.

DOI: <https://doi.org/10.1109/DESSERT61349.2023.10416450>

URL: <https://repository.hneu.edu.ua/handle/123456789/38399>

8. Єнґаличев С. Вибір показників ефективності паралельної реалізації алгоритмів. *Problems of scientific, technical and legal support for cybersecurity in the modern world*. (Krakow, 2024). Krakow: Wydawnictwo Naukowe UKEN, 2024. P. 174-179.

URL: <https://repository.hneu.edu.ua/handle/123456789/34912>

9. Yenhalychev S.O., Leunenko O.V., Semenov S.G.. Known planners assignments and systems management resources analysis. *Free and open source*

software: Матеріали XIV міжнародної наук.-прак. конф. (м. Харків, 14 - 16 Лютого, 2023 р.). Харків: FOSS, 2023. С. 10-14.

URL: <http://repository.hneu.edu.ua/handle/123456789/29041>

10. Єнгаличев С. О., Семенов С. Г. Аналіз можливостей використання машинного глибокого навчання для розпізнавання сутностей. *Science and Business in Modern Environment: Summer Debates: матеріали V міжнародної наук.-прак. інтернет конф.* (м. Дніпро, 3 - 4 серпня 2023 р.). С. 205-206.

URL: <https://repository.hneu.edu.ua/handle/123456789/38369>

11. Yenhalychev S. O., Semenov S. G., Leunenko O. V. An overview of the popular free resources for task planning and system management. *Free and open source software: матеріали XV міжнародної наук.-прак. конф.* (м. Харків, 14 - 16 Лютого, 2024 р.). Харків: FOSS, 2024. С. 70-72.

URL: <http://repository.hneu.edu.ua/handle/123456789/31827>

12. Єнгаличев С. О., Леуненко О. В. Імітаційне моделювання та дослідження алгоритмів планування пакетних потоків даних із різними пріоритетами. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024) (м. Харків, 25 квітня, 2024 р.). Молодіжний економічний вісник. 2024. №1. С. 46-50.

URL: <https://repository.hneu.edu.ua/handle/123456789/38208>

13. Леуненко О. В., Єнгаличев С. О. Моделювання та захист передачі даних ресурсоємних задач у гетерогенних хмарних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2024) (м. Харків, 25 квітня, 2024 р.). Молодіжний економічний вісник. 2024. №1. С. 59-63.

URL: <https://repository.hneu.edu.ua/handle/123456789/38209>

14. Семенов С. Г., Єнгаличев С. О. Метод планування задач в розподілених інформаційних системах з врахуванням невизначеності метаданих. *Проблеми інформатизації: матеріали XII міжнародної наук.-*

техн. конференції (м. Харків, 21- 22 листопада, 2024 р.). Харків: Харківський національний університет радіоелектроніки, 2024. С. 24.

URL: <https://repository.hneu.edu.ua/handle/123456789/38235>

15. Yenhalychev S.O., Leunenko O.V.. Comparative analysis of open source infrastructure as code (IaC) tools for managing heterogeneous cloud resources. *Free and open source software: матеріали XVI міжнародної наук.-прак. конф.* (м. Харків, 13 - 14 Лютого, 2025 р.). Харків: FOSS, 2025. С. 105-107.

URL: <https://repository.hneu.edu.ua/handle/123456789/35624>

16. Yenhalychev S. O., Leunenko O. V. Development of Methods for Dynamic Load Balancing During the Transfer of Large Volumes of Data: A Comparative Study of Apache Kafka and RabbitMQ. *Free and open source software: матеріали XVI міжнародної наук.-прак. конф.* (м. Харків, 13 - 14 Лютого, 2025 р.). Харків: FOSS, 2025. С. 108-109.

URL: <https://repository.hneu.edu.ua/handle/123456789/35624>

17. Serhii Semenov, Serhii Yenhalychev. The guidelines on secure coding and frameworks to enhance autonomy and mitigate cybersecurity risks. *Сучасні інформаційні системи та технології в цифровому суспільстві: матеріали міжнародної наук.-практ. конференції* (м. Харків, 10 - 11 квітня, 2025 р.). Харків: ХНЕУ імені Семена Кузнеця, 2025. С. 71.

URL: <https://repository.hneu.edu.ua/handle/123456789/36030>

18. Семенов С. Г., Єнгаличев С. О., Пересічанський В. М. Обговорення результатів дослідження інтелектуального методу планування задач у розподілених інформаційних системах. *Безпека у кіберсфері: матеріали Міжнародної науково-практичної конференції* (м. Кам'янець-Подільський, 28 травня, 2025 р.). Кам'янець-Подільський: ХНУВС, 2025. С. 186-187.

URL: <https://repository.hneu.edu.ua/handle/123456789/36711>

19. Семенов С. Г., Єнгаличев С. О. Адаптивне планування задач у розподілених системах на основі алгоритму оптимізації політик із проксимальним оновленням. *Цифрова економіка: Матеріали міжнародної*

науково-практичної конференції (м. Київ, 5 - 6 червня, 2025 р.). Київ: КНЕУ, 2025. С. 698-700.

URL: <https://repository.hneu.edu.ua/handle/123456789/38234>

20. Єнгаличев С. О., Леуненко О. В.. Обговорення результатів інтелектуального методу планування задач у гетерогенних розподілених інформаційних системах. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025) (м. Харків, 16 жовтня, 2025 р.). *Молодіжний економічний вісник*. 2025. №3. С. 122-124.

URL: <https://repository.hneu.edu.ua/handle/123456789/38210>

21. Леуненко О. В., Єнгаличев С. О. Математичне моделювання передачі даних ресурсомістких задач у гетерогенних хмарних системах: показники ефективності та методологічні підходи. Матеріали міжнародної наукової конференції з економічного розвитку та спадщини Семена Кузнеця (CED-2025). (м Харків, 16 жовтня, 2025 р.). *Молодіжний економічний вісник*. 2025. №3. С. 199-202.

URL: <https://repository.hneu.edu.ua/handle/123456789/38211>

ЗАТВЕРДЖУЮ

Виконавчий директор компанії «LineUp»

Тарас КІБІТКІН
17 квітня 2025 р.

АКТ ВПРОВАДЖЕННЯ

результатів наукового дослідження дисертаційної роботи

Єнгалічева Сергія Олександровича

В процесі оптимізації та розвитку модулів керування задачами в розподілених інформаційних середовищах компанії LineUp було впроваджено наступні результати дисертаційної роботи Єнгалічева С.О.:

Впроваджено інтелектуальний метод планування задач у гетерогенних розподілених інформаційних системах, який базується на поєднанні DAG-та GERT-графів із алгоритмами штучного інтелекту: Graph Attention Network, Proximal Policy Optimization та Bayesian Optimization. Це дозволило динамічно адаптувати розподіл задач між обчислювальними ресурсами в умовах змінного навантаження.

Отримані результати використано при створенні модуля автоматизованого керування навантаженням у хмарному середовищі LineUp, що дало змогу покращити балансування ресурсів, зменшити час затримок і підвищити стійкість системи до змін середовища виконання.

Commission members:

Taras KIBITKIN

Olha PIDMOHYLNA

Serhii SEMENOV

Виконавець:

LLC «Arshilon»

Чорноглазівська, 10, кв. 19,

Харків, 61002, Україна

«23» грудня 2025 р.

АКТ ВПРОВАДЖЕННЯ

результатів наукового дослідження дисертаційної роботи

Єнгаличева Сергія Олександровича

В процесі управління задачами в інформаційному середовищі LLC «Arshilon» було впроваджено наступні результати дисертаційної роботи Єнгаличева С.О.:

Впроваджено інтелектуальний метод планування задач у розподілених інформаційних системах із застосуванням RL для прийняття рішень щодо призначення задач на ресурси та корекції розкладу при зміні стану системи.

Використано практичні рекомендації щодо впровадження методу у розподілених інформаційних системах з урахуванням вимог до організації процесу планування в реальному часі та до політик доступу в прикладних інформаційних системах.

Отримані результати використано при алгоритмів управління процесом розподілу задач між структурними підрозділами LLC «Arshilon», що дало змогу покращити балансування ресурсів.

Директор LLC «Arshilon»

Інна ПЕТРОВСЬКА



УКРАЇНА
МІНІСТЕРСТВО ОСВІТИ
І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ
№02071211

№ _____ 20__ р.
61165, м. Харків, пр. Науки, 9-А
тел./факс: 70-20-717, 70-20-304

Д О В І Д К А № 82/2025

від 26.12.2025 р.

Дана СНГАЛИЧЕВУ СЕРГІЮ ОЛЕКСАНДРОВИЧУ,
який брав участь у розробці госпдоговірної науково-дослідної роботи
Замовник — **ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ**
"СХІД-ТЕРМІНАЛ" по темі № 250-46 «Розробка інформаційної системи з
елементами захисту інформації від незаконного втручання» на посаді
молодшого наукового співробітника з 04.10.2024 року по 31.10.2024 року.

Розділ 3. Реалізація системи захисту.

Державний реєстраційний номер 0124U004238

Начальник ІДС



Ірина ЛИТОВЧЕНКО

УКРАЇНА
МІНІСТЕРСТВО ОСВІТИ
І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ
№02071211

№ _____ 20__ р.
61165, м. Харків, пр. Науки, 9-А
тел./факс: 70-20-717, 70-20-304

Д О В І Д К А № 81/2025
від 26.12.2025 р.

Дана ЄНГАЛИЧЕВУ СЕРГІЮ ОЛЕКСАНДРОВИЧУ,
який брав участь у розробці госпдоговірної науково-дослідної роботи
Замовник — **ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ**
"СХІД-ТЕРМІНАЛ" по темі № 158-46 «Методична підтримка сталого
функціонування ІТ підприємства» на посаді молодшого наукового
співробітника з 17.06.2025 року по 15.07.2025 року.

Розділ 2. Аналіз та порівняльні дослідження методів планування задач в
розподілених інформаційних системах з врахуванням невизначеності
метаданих.

Державний реєстраційний номер 0125U002650

Начальник НДС



Ірина ЛИТОВЧЕНКО



УКРАЇНА

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

61166, м. Харків, пр. Науки, 9-А, тел. (057) 702-03-04, факс: (057) 702-07-17
E-mail: post@hneu.edu.ua, http://www.hneu.edu.ua, ЄДПРРОУ 02071211

№ 26/86-29-22 від 02.03.2026

На № _____ від _____

ДОВІДКА

про використання результатів та окремих положень дисертаційного дослідження

Снгаличева Сергія Олександровича,
поданого у дисертації на здобуття наукового ступеня доктора філософії
за спеціальністю 122 «Комп'ютерні науки»
на тему: «Метод планування задач у розподілених інформаційних
системах за невизначеності метаданих»

У процесі виконання дисертаційного дослідження Снгаличевим Сергієм Олександровичем отримано наукові та практичні результати, спрямовані на удосконалення методів планування задач у розподілених інформаційних системах в умовах невизначеності метаданих, гетерогенності обчислювальних ресурсів та динамічності середовища виконання.

Основні результати дослідження пов'язані з розробленням інтелектуального методу планування задач у розподілених інформаційних системах, побудовою математичних моделей планування з урахуванням невизначеності метаданих, застосуванням методів штучного інтелекту для прогнозування навантаження та адаптивного розподілу обчислювальних ресурсів, що дозволяє підвищити ефективність використання обчислювальних ресурсів та зменшити час виконання задач у хмарних і розподілених обчислювальних середовищах.

Отримані результати впроваджено в освітній процес під час проходження педагогічної практики та викладання навчальних дисциплін. Зокрема, результати дослідження використано у викладанні дисципліни «Безпека програм та даних» (перший (бакалаврський) рівень вищої освіти, спеціальність 121 «Інженерія програмного забезпечення») а саме моделювання метаданих у розподілених інформаційних системах та аналізу їх впливу на ефективність оброблення задач, а також підходи до оцінювання навантаження у розподілених обчислювальних середовищах, що застосовано під час виконання лабораторних робіт.

260459

У навчальній дисципліні «Комплексний курсовий проєкт» (перший (бакалаврський) рівень вищої освіти, спеціальність 125 «Кібербезпека») використано результати дослідження щодо інтелектуального планування задач у розподілених інформаційних системах та адаптивного розподілу обчислювальних ресурсів, які застосовано під час консультування здобувачів і розроблення студентських програмних проєктів.

Крім того, результати дослідження використано під час викладання дисципліни «Стандартизація та сертифікація захисту інформації» (другий (магістерський) рівень вищої освіти, спеціальність 125 «Кібербезпека та захист інформації»), зокрема дослідження щодо оцінювання ефективності функціонування систем управління ресурсами у хмарних і розподілених інформаційних середовищах, що застосовано під час аналізу вимог до надійності та стійкості інформаційних систем відповідно до сучасних стандартів інформаційної безпеки.

Упровадження результатів дисертаційного дослідження сприяло підвищенню практичної спрямованості освітнього процесу, інтеграції сучасних методів інтелектуального планування задач у навчальну діяльність та формуванню у здобувачів компетентностей з розроблення і аналізу розподілених інформаційних систем. Отримані результати можуть бути рекомендовані для подальшого використання в освітньому процесі та науково-дослідній роботі.

Проректор
з навчально-методичної роботи



Каріна НЕМАШКАЛО