

М. Ю. ХОРКАНІНаспірант кафедри інформаційних систем та мереж
Національний університет «Львівська політехніка»
ORCID: 0009-0001-6263-6878**О. О. МЕДЯКОВ**студент кафедри інформаційних систем та мереж
Національний університет «Львівська політехніка»
ORCID: 0000-0002-2580-3155

КОНТРОЛЬОВАНЕ ЗАСТОСУВАННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ ДЛЯ ПЛАНУВАННЯ ЗАВДАНЬ У ІНТЕЛЕКТУАЛЬНИХ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Зростаючі потреби в автоматизації планування ресурсів у розподілених інформаційних системах створюють нові виклики щодо балансування між ефективністю автономних рішень та необхідністю забезпечення контролю спеціалістів. Традиційні підходи до автоматичного планування, включаючи статичні алгоритми та класичні методи машинного навчання, не повною мірою використовують потенціал сучасних технологій штучного інтелекту, тоді як повністю автономні системи на базі LLM-агентів несуть високі ризики при роботі з реальними обчислювальними ресурсами.

У даній роботі запропоновано гібридну мультиагентну архітектуру для автоматичного планування завдань у розподілених інформаційних системах з інтегрованою компонентою людина-в-циклі. Архітектура поєднує здібності логічного виведення великих мовних моделей з можливістю людського контролю, передбачаючи централізоване планування через агента-менеджера та децентралізоване виконання через агентів-виконавців. Ключовою особливістю запропонованого підходу є багаторівнева система валідації генерованого коду, що включає статичний аналіз, можливість попереднього виконання та обов'язкове підтвердження спеціалістом на критичних етапах прийняття рішень. Методологія дослідження базується на принципах гібридного планування, що дозволяє ефективно використовувати переваги LLM-агентів при мінімізації ризиків їх застосування в реальних обчислювальних середовищах. Визначено чіткі обмеження застосування системи до некритичних інтелектуальних завдань обробки даних, включаючи машинне навчання, аналіз даних, обробку природної мови та комп'ютерний зір. Запропонована архітектура вирішує проблему балансу між потребою в автоматизації планування ресурсів та необхідністю забезпечення безпеки при роботі з реальними розподіленими інфраструктурами, створюючи теоретичні засади для контрольованого впровадження агентних технологій у практику управління розподіленими системами.

Ключові слова: автоматичне планування, ШІ-агенти, людина-в-циклі, інтелектуальна розподілена інформаційна система.

M. YU. KHORKANINPostgraduate Student at the Department of Information Systems and Networks
Lviv Polytechnic National University
ORCID: 0009-0001-6263-6878**O. O. MEDIKOV**Student at the Department of Information Systems and Networks
Lviv Polytechnic National University
ORCID: 0000-0002-2580-3155

CONTROLLED APPLICATION OF MULTI-AGENT SYSTEMS FOR TASK PLANNING IN INTELLIGENT DISTRIBUTED INFORMATION SYSTEMS

The growing need for resource planning automation in distributed information systems creates new challenges regarding the balance between the efficiency of autonomous solutions and the need to ensure expert control. Traditional approaches to automatic planning, including static algorithms and classic machine learning methods, do not fully utilize the potential of modern artificial intelligence technologies, while fully autonomous systems based on LLM agents pose high risks when working with real computing resources.

This paper proposes a hybrid multi-agent architecture for automatic task planning in distributed information systems with an integrated human-in-the-loop component. Architecture combines the logical reasoning capabilities of large language models with the possibility of human control, providing centralized planning through a manager agent and decentralized execution through executor agents. A key feature of the proposed approach is a multi-level validation system for the generated code, which includes static analysis, the possibility of pre-execution, and mandatory confirmation by a specialist at critical decision-making stages. The research methodology is based on the principles of hybrid planning, which allows for the effective use of the advantages of LLM agents while minimizing the risks of their application in real computing environments. The system's application is clearly limited to non-critical intelligent data processing tasks, including machine learning, data analysis, natural language processing, and computer vision. The proposed architecture solves the problem of balancing the need for resource planning automation with the need to ensure safety when working with real distributed infrastructures, creating a theoretical basis for the controlled implementation of agent technologies in the practice of managing distributed systems.

Key words: automatic planning, AI-agents, human-in-the-loop, intelligent distributed information systems.

Постановка проблеми

Застосування більш автономних технологій, таких як агенти на базі великих мовних моделей, що дозволяють не лише приймати рішення щодо розподілу ресурсів по РІС, а й безпосередньо виконувати цей розподіл та запускати процес обчислення, дозволятимуть значно ефективніше виконувати задачі. Проте без правильної архітектури та контролю спеціалістів існують високі ризики щодо використання повністю автономних рішень. Саме тому актуальною є розробка гібридних підходів, що поєднують переваги автоматизованого планування з можливістю людського контролю.

Аналіз останніх досліджень і публікацій

Різновидів алгоритмів розподілу ресурсів та завдань по вузлам чи кластерам будь-якої РІС є широкий спектр, починаючи з класичних статичних алгоритмів [1], й до методів автоматичного планування з використання інтелектуальних підходів [2].

Незважаючи на популяризацію автоматичного планування на базі моделей машинного навчання [3] й досі пропонуються нові статичні алгоритми, здатні до оптимального розподілу завдань, наприклад алгоритми оптимізації колоною мурах використовуються у хмарних обчисленнях [4]. Проте головним вектором розвитку автоматичного планування залишається використання ШІ моделей, зокрема й великих мовних моделей (LLM) та агентів на їх базі [2].

Історичними першими було застосовано класичні моделі машинного навчання [3], зокрема навчання з підкріпленням для адаптивного автоматичного планування [5], й звичайно використання LLM-агентів для планування у хмарних середовищах, Spark-кластерах та інших розподілених середовищах [6].

У широкому розумінні, ШІ агентом вважається програмна система, логіка виконання якої визначається результатами генеративної діяльності базової LLM [7]. У контексті планування у розподілених інтелектуальних системах використання агентів дозволяє керувати розподілом задач між кластерами, порядок виконання цих завдань [2] та навіть генерацію чи оптимізацію коду, що безпосередньо містить логіку виконання завдання на вузлі чи кластері. Ця робота також є логічним продовженням запропонованої гібридної архітектури для автоматичного планування у розподілених системах на базі мультиагентної взаємодії мовних моделей у роботі [2].

Варто зауважити, що проблеми та перспективи мультиагентних систем, їх взаємодії та комунікації [8], незалежно від безпосереднього використання мультиагентного підходу – є дотичним до цієї роботи.

Як у теоретичному так і практичному полі застосування агентних систем для планування ресурсів, завдань у РІС, а також інших проблем, що вимагають прийняття рішень, які матимуть вплив на реальні обчислювальні ресурси чи зовнішній світ загалом розглядаються й з точки зору ризиків [6] та інших параметрів ефективності, якості та узагальнення [9]. Частина цих ризиків пов'язані із проблемами у базових мовних моделях, що керують агентною поведінкою.

Один з головних та останній методів, що дозволяє мінімізувати проблему генералізації та уникнути частину ризиків є додавання людини-в-циклі (HITL) до процесу прийняття рішень агентом. Оскільки така компонента є однією з ключових у межах архітектури, яка пропонується у даній роботі, необхідно розглянути останні роботи у напрямку використання HITL у мультиагентних системах.

У багатьох роботах було досліджено та експериментально показано, що додавання HITL дозволяє покращити узагальнювальні можливості агентів [9], додавати уточнення та додаткові пояснення для процесу прийняття рішень, чи застосувати експертизу спеціалістів у процесі виконання задачі планування [10].

Формулювання мети дослідження

Метою даного дослідження є проектування архітектури мультиагентного автоматичного планувальника у розподілених інформаційних системах з інтегрованою компонентою людина-в-циклі. Досягнення мети включає виконання таких основних задач: визначення обмежень застосування підходу, верхнього рівня огляду запропонованої архітектури, уточнення відповідальностей агентів різного рівня, визначення ролі людини-в-циклі у загальному потоці прийняття рішень та передачі інформації між агентами.

Викладення основного матеріалу дослідження

У контексті даної роботи пропонується гібридний підхід, де поєднується мультиагентна система для автоматичного планування та обчислень на PIC та компонента людини-в-циклі (HITL) для контролю процесу. Загальне планування здійснюється централізовано агентом-менеджером, але виконання та локальна оптимізація задач відбувається децентралізовано на рівні окремих обчислювальних кластерів під керуванням інтелектуальних агентів-виконавців. HITL компонента дозволить користувачам чи спеціалістам приймати чи змінювати рішення у різних частинах мультиагентної системи чи її взаємодії з реальним виконуваним середовищем.

Запропонована мультиагентна система планування та виконання завдань на розподіленій інформаційній системі має чітко визначену область застосування та обмеження, які є критично важливими для забезпечення безпеки та ефективності її функціонування.

Система призначена виключно для розв'язання інтелектуальних завдань обробки даних, що не передбачають критично важливих операцій реального часу або керування фізичними процесами. До таких завдань належать: аналіз даних з метою виявлення закономірностей, трендів та аномалій; машинне навчання та тренування моделей штучного інтелекту; обробка природної мови, включаючи класифікацію текстів, аналіз тональності, резюмування; комп'ютерний зір для класифікації зображень, виявлення об'єктів, сегментації; процеси витягування, трансформації та завантаження даних (ETL); статистичний аналіз та прогнозування.

Як визначено у основних задачах цього дослідження, необхідно розглянути та описати основні функції кожного рівня агентів (менеджера та виконавця), їхньої взаємодії, а також місце та функції застосування HITL. Узагальнення архітектури запропонованого рішення виведено на рис. 1.

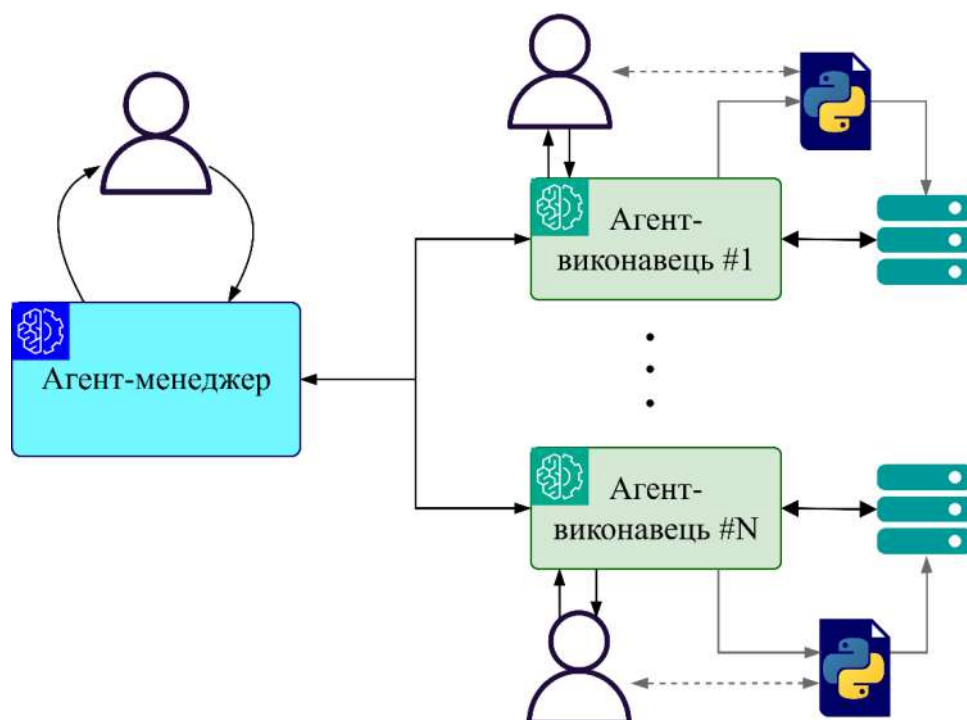


Рис. 1. Візуальне відображення запропонованої архітектури мультиагентного планувальника з HITL компонентою

Основною функцією агента-менеджера є формування та розподіл підзадач для обробки даних між доступними кластерами розподіленої інтелектуальної інформаційної системи, враховуючи їх гетерогенну структуру та характеристики. Для реалізації цієї функції агент використовує здібності логічного виведення базової LLM, спеціалізований набір інструментів, наявну інформацію про завдання та PIC, а також компоненти HITL.

Класичний потік обробки запиту включає планування підзадач (якщо такі є), перевірку наявних ресурсів (кластерів), алокацію підзадач відповідним агентам-виконавцям, запуск виконання підзадачі на кластері, формування кінцевого результату. Кожен агент-виконавець повинен повернути підтвердження про прийняття алокації підзадачі на контрольований ним кластер. Якщо ж агент-виконавець не підтверджує виконання, то менеджер повинен виконати реаллокацію.

Компоненти HITL на рівні агента-менеджера включають: ручне підтвердження людиною декомпозиції задачі на підзадачі з можливістю редагування списку, підтвердження розподілу підзадач на кластери, ручну алокацію та/або реаллокацію.

В свою чергу, основним завданням агента-виконавця є підготовка код-бази програмного забезпечення для виконання інтелектуального завдання у розподіленому середовищі (кластеру вузлів, за які відповідає конкретний агент). Зокрема, це передбачає створення Python-програм з використанням бібліотек, що підтримують розподілені обчислення. Після підготовки (генерації та пост-процесингу) програмних файлів агент ініціює їх виконання на кластері через спеціалізований інструмент-тригер.

Враховуючи визначені раніше методи алокації задачі на агента-виконавця, агент повинен проаналізувати запит від агента-менеджера та видати рішення щодо прийняття чи відхилення завдання. Якщо алокація відбувається за ручним запитом від користувача – вона автоматично приймається.

Крім того, безпосередня мовна модель, що є основним компонентом агента, не обов'язково повинна бути одною та ж для підтвердження алокації та написання коду, чи для виконання різних підзадач. Додавання компоненти HITL дозволяє навіть динамічно обирати іншу базову модель залежно від задачі.

Генерація коду, який потім буде виконуватися на реальній розподіленій інфраструктурі користувача, є процесом з високим ризиком без достатнього рівня перевірки. У цій роботі ми пропонуємо об'єднати два рівні перевірки – перевірку спеціалістом (HITL) та статичну перевірку коду у пост-процесингу.

До пост-процесингу належать класичні методи статичної перевірки Python коду. Зокрема, такі інструменти як Pylint [pylint], дозволяють перевірити генерований код на синтаксичні помилки. У разі, якщо у коду знайдено помилку, яку не можна виправити автоматично, код відправляється на доопрацювання та виправлення агентом.

Додатково до статичної перевірки може бути імплементовано так зване “sandbox” виконання коду перед тим, як допустити його до реальної інфраструктури. Це дозволить виявити потенційні помилки виконання та оцінити поведінку коду в контрольованому середовищі.

Обов'язковою частиною перевірки генерованого коду є підтвердження від HITL компоненти. Користувач матиме можливість не лише перевірити код, а й запросити виправлення певних його частин, або ж додати виправлення вручну. Така інтеграція людського контролю забезпечує додатковий рівень безпеки та якості генерованих рішень.

Висновки

У даній роботі було запропоновано архітектуру мультиагентної системи автоматичного планування для інтелектуальних розподілених інформаційних систем з інтегрованою компонентою людина-в-циклі. Ключовою особливістю запропонованого підходу є поєднання переваг автономного планування на базі LLM-агентів з можливістю людського контролю на критичних етапах прийняття рішень. Архітектура передбачає централізоване планування через агента-менеджера та децентралізоване виконання через агентів-виконавців, що забезпечує гнучкість та масштабованість рішення.

Запропонована архітектура включає перевірку генерованого коду, зокрема статичний аналіз, можливість “sandbox” виконання та обов'язкове підтвердження спеціалістом, дозволяє мінімізувати ризики застосування автономних агентів у реальних обчислювальних середовищах. Визначені обмеження застосування системи до некритичних завдань обробки даних забезпечують безпечне впровадження технології. Подальші дослідження можуть бути спрямовані на експериментальну валідацію запропонованої архітектури та оптимізацію взаємодії між компонентами людини-в-циклі та агентами різних рівнів.

Список використаної літератури

1. Khushboo Singh S. K. S. A survey of static scheduling algorithm for distributed computing system. *International journal of computer applications*. 2015. T. 129, № 2. С. 25–30. URL: <https://doi.org/10.5120/ijca2015906828>
2. Медяков О. О., Хорканін М. Ю. Автоматизоване планування в інтелектуальних розподілених системах мультиагентним підходом на базі llm. *Вісник Вінницького політехнічного інституту*. 2025. № 2. С. 111–117. URL: <https://doi.org/10.31649/1997-9266-2025-179-2-111-117>
3. Utilizing machine learning algorithms for task allocation in distributed agile software development / D. Al-Fraihat та ін. *Heliyon*. 2024. T. 10, № 21. С. e39926. URL: <https://doi.org/10.1016/j.heliyon.2024.e39926>
4. Wei X. Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing. *Journal of ambient intelligence and humanized computing*. 2020. URL: <https://doi.org/10.1007/s12652-020-02614-7>
5. Learning scheduling algorithms for data processing clusters / H. Mao та ін. URL: <https://doi.org/https://arxiv.org/abs/1810.01963>
6. Kankaniyage Don R. T. P., Ravi A., Toxtli C. Assessing the Task Management Capabilities of 6ed Agents. 2024. URL: <https://doi.org/10.13140/RG.2.2.11776.85768>
7. Sypherd C., Belle V. Practical considerations for agentic LLM systems. URL: <https://doi.org/https://arxiv.org/abs/2412.04093>
8. Large language model based multi-agents: a survey of progress and challenges / T. Guo та ін. URL: <https://doi.org/https://arxiv.org/abs/2402.01680>
9. HMCf: a human-in-the-loop multi-robot collaboration framework based on large language models / Z. Li та ін. URL: <https://doi.org/https://arxiv.org/abs/2505.00820>

10. A call for collaborative intelligence: why human-agent systems should precede AI autonomy / H. P. Zou та ін. URL: <https://doi.org/https://arxiv.org/abs/2506.09420>.

References

1. Khushboo Singh, S. K. S. (2015). A survey of static scheduling algorithm for distributed computing system. *International Journal of Computer Applications*, 129(2), 25–30. <https://doi.org/10.5120/ijca2015906828>
2. Mediakov, O., & Khorkanin, M. (2025). Automated Planning in Intelligent Distributed Systems Using a Multi-Agent Approach Based on LLM. “*Visnyk of Vinnytsia Politechnical Institute*”, (2), 111–117. <https://doi.org/10.31649/1997-9266-2025-179-2-111-117>
3. Al-Fraihat, D., Sharrab, Y., Al-Ghuwairi, A.-R., Alzabut, H., Beshara, M., & Algarni, A. (2024). Utilizing machine learning algorithms for task allocation in distributed agile software development. *Heliyon*, 10(21), e39926. <https://doi.org/10.1016/j.heliyon.2024.e39926>
4. Wei, X. (2020). Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-020-02614-7>
5. Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2019). *Learning scheduling algorithms for data processing clusters*. <https://doi.org/https://arxiv.org/abs/1810.01963>
6. Kankaniyage Don, R. T. P., Ravi, A., & Toxtli, C. (2024). *Assessing the task management capabilities of llm-powered agents*. <https://doi.org/10.13140/RG.2.2.11776.85768>
7. Sypherd, C., & Belle, V. (2024). *Practical considerations for agentic LLM systems*. <https://doi.org/https://arxiv.org/abs/2412.04093>
8. Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). *Large language model based multi-agents: A survey of progress and challenges*. <https://doi.org/https://arxiv.org/abs/2402.01680>
9. Li, Z., Wu, W., Wang, Y., Xu, Y., Hunt, W., & Stein, S. (2025). *HMCF: A human-in-the-loop multi-robot collaboration framework based on large language models*. <https://doi.org/https://arxiv.org/abs/2505.00820>
10. Zou, H. P., Huang, W.-C., Wu, Y., Miao, C., Li, D., Liu, A., Zhou, Y., Chen, Y., Zhang, W., Li, Y., Fang, L., Jiang, R., & Yu, P. S. (2025). *A call for collaborative intelligence: Why human-agent systems should precede AI autonomy*. <https://doi.org/https://arxiv.org/abs/2506.09420>

Дата першого надходження рукопису до видання: 30.09.2025

Дата прийнятого до друку рукопису після рецензування: 27.10.2025

Дата публікації: 28.11.2025