

<https://doi.org/10.31649/1997-9266-2025-179-2-111-117>

УДК 004.8:004.75

О. О. Медяков¹
М. Ю. Хорканін¹

АВТОМАТИЗОВАНЕ ПЛАНУВАННЯ В ІНТЕЛЕКТУАЛЬНИХ РОЗПОДІЛЕНИХ СИСТЕМАХ МУЛЬТИАГЕНТНИМ ПІДХОДОМ НА БАЗІ LLM

¹Національний університет «Львівська політехніка»

Представлено новітній підхід до автоматизації управління інтелектуальними розподіленими інформаційними системами (PIS) на основі гібридної мультиагентної архітектури з використанням ШІ-агентів на базі великих мовних моделей (так звані LLM-агенти). Запропонована архітектура складається з дворівневої системи планування, яка включає агента-менеджера для централізованого глобальної декомпозиції завдань та агентів-виконавців для децентралізованого локального процесу задання та оптимізації обчислень на окремих кластерах інтелектуальної розподіленої системи. Для ефективності та оптимізації взаємодії агентів з відповідними середовищами запропоновано використовувати інноваційні методи доступу моделей до довгострокової та короткострокової пам'яті та уніфікацію процесів утилізації доступних агенту-моделі інструментів через генерацію, та відповідне виконання програмного коду, застосовуючи спеціалізований новітній метод промпт-інжинірингу CodeAct. Велика продуктивність CodeAct дозволяє створювати нових агентів, які взаємодіють з середовищами, виконують інтерпретований код та здатні співпрацювати з користувачами за допомогою природної мови. Запропонований підхід потенційно забезпечує агностичність архітектури щодо структури розподіленої системи, тобто дозволяє працювати з комплексними гетерогенними PIS, та щодо типів виконуваних завдань через спеціалізовані компоненти управління та доступу до пам'яті, а також програмним виконанням згенерованих агентами інструкцій. Робота включає опис необхідних вмінь та базових властивостей для використовуваних великих мовних моделей, структури двох типів необхідних агентів, та наборів необхідних інструментів для якісної роботи та взаємодії інтелектуальних агентів системи між собою та зовнішнім середовищем, зокрема керованою PIS. Дослідження пояснює перспективність використання великих мовних моделей та новітніх ШІ-агентів для інтелектуального планування в розподілених обчислювальних середовищах, розкриваючи потенціал штучного інтелекту в автоматизації складних процесів керування.

Ключові слова: LLM агенти, автоматизоване планування, мультиагентні системи, інтелектуальна розподілена інформаційна система, великі мовні моделі.

Вступ

З продовженням розвитку технологій розподілені інформаційні системи (PIS) стали важливими для підтримки важливих обчислювальних процесів у різних галузях [1], [2]. Такі системи характеризуються високим рівнем складності та динамічністю середовища, що створює виклики для їхнього ефективного управління. Традиційні підходи, з алгоритмічним підходом (статичне планування, стохастичні мережі Петрі тощо [3], [4]) або з ручним адмініструванням виявляються недостатньо ефективними в умовах швидких змін і великих масштабів обчислювальних операцій.

Великі мовні моделі (Large Language Models, LLMs) здатні генерувати високоякісний програмний код [5], аналізувати контекст і адаптуватися до нових завдань, тому останнім часом вони є найпотужнішою базою для нових ШІ-агентів [6]. У разі взаємодії кількох таких агентів, тобто створення мультиагентних систем [7], [8], стає можливо інтелектуалізувати процеси управління. Такі системи можуть не лише автоматизувати виконання складних задач, а й інтегрувати сучасні

інструменти, що дозволяють оптимізувати використання обчислювальних ресурсів.

У цій роботі запропоновано новітню архітектуру мультиагентного управління розподіленими інформаційними системами, де центральний ШІ-агент-менеджер забезпечує первинне планування та координацію, а агенти-виконавці спеціалізуються на організації окремих завдань на безпосередніх кластерах. У роботі приділяється спеціальна увага до інтеграції механізмів генерації коду для використання інструментів агентів, управління пам'яттю та оптимізації використання ресурсів. *Метою дослідження* є опис потенційно автономного інтелектуального підходу для управління складними обчислювальними середовищами, зокрема інтелектуальними розподіленими системами на базі взаємодії LMM агентів.

Аналітичний огляд літератури

Управління РІС усіх типів охоплює широкий спектр задач, а саме від моніторингу стану системи до прийняття рішень щодо алокації ресурсів та оптимізації навантаження. Сучасні наукові дослідження в цій галузі сконцентровані на розробці методів автоматизованого управління [9], здатних враховувати комплексні взаємозалежності між компонентами системи та адаптуватися до змін операційного середовища.

Застосування агентів на базі LLM уможливило реалізацію інтелектуальних підходів до управління, що забезпечують аналіз значних обсягів даних та прийняття рішень на основі складних патернів поведінки системи.

У сучасному контексті визначення та типологія ШІ-агентів на основі LLM характеризується значною різноманітністю, при цьому їхньою інтегральною характеристикою залишається здатність до інтерактивної взаємодії з навколишнім середовищем через застосування спеціалізованого інструментарію [6]. У широкому розумінні, ШІ-агентом вважається програмна система, логіка виконання якої визначається результатами генеративної діяльності базової LLM.

Актуальні наукові дослідження у галузі агентів штучного інтелекту на базі мовних моделей охоплюють низку фундаментальних напрямків, зокрема:

- вдосконалення механізмів логічного виведення в LLM [10];
- розвиток методів декомпозиції задач та планування їх розв'язання;
- проблеми та перспективи мультиагентних систем, їх взаємодії та комунікації [8];
- оптимізацію використання інструментів [11];
- імплементацію систем доповнення контексту на основі агентного Retrieval Augmented Generation (RAG) [12], [13];
- управління довготривалою та короткотривалою пам'яттю;
- дослідження практичних імплементацій подібних агентів для вирішення прикладних задач.

Описане дослідження належить до останнього напрямку та концентрується на застосуванні агентних систем у контексті планування інтелектуальних розподілених інформаційних систем.

Один з головних векторів розвитку агентів є оптимізація використання інструментів, зокрема у роботі [11] дослідники пропонують повністю замінити від JSON-орієнтованого використання інструментів до генерації програмного коду. Шляхом адаптації методології промпт-інжинірингу для агентів ReAct [14], авторами розроблено метод CodeAct [11]. Імплементація генерації коду замість традиційного виклику інструментів демонструє підвищення ефективності через редукцію кількості генерованих токенів та оптимальне використання вбудованих можливостей LLM щодо генерації програмного коду.

Розвиток типів та видів інструментів, придатних для роботи з комплексними середовищами, як РІС, теж отримав достатню увагу в останній час. У дослідженні [15] запропоновано концептуально новий клас інструментів — “middleware” (проміжного рівня), спеціально розроблених для забезпечення функціонування в складних гетерогенних середовищах.

Архітектура мультиагентного планувальника

Складність структури РІС має широкий спектр, і під час розв'язання задачі автоматичного планування за її використання необхідно враховувати цю різноманітність. Для цього запропоновано узагальнити структуру розподіленої інформаційної системи таким чином, щоб запропонований мультиагентний підхід був агностичним щодо цього аспекту.

Водночас варто зважати на те, що класичні РІС містять спеціалізовані чи додаткові підсистеми, існування яких зумовлене відсутністю інтелектуального планування, до прикладу, дистриб'ютор навантаження тощо. Проте запропонований спосіб планування не вимагає наявності таких compone-

нтів, саме тому РІСи повинні складатися лише з множини взаємопов'язаних, потенційно гетерогенних, кластерів.

Важливим аспектом архітектури будь-якої РІС є модель керування системою. Традиційно є два основних підходи: централізований та децентралізований. У централізованій моделі існує єдиний контролювальний вузол, який координує роботу всієї системи. Це спрощує процес керування та моніторингу, але створює потенційну точку відмови та може обмежувати масштабованість. Децентралізована модель розподіляє функції керування між множиною вузлів, що підвищує надійність та гнучкість системи, але ускладнює процеси координації та узгодження дій.

У контексті цієї роботи пропонується гібридний підхід, де загальне планування здійснюється централізовано агентом-менеджером, але виконання та локальна оптимізація задач відбувається децентралізовано на рівні окремих обчислювальних кластерів під керуванням інтелектуальних агентів-виконавців.

Схематичне представлення архітектури мультиагентного планувальника показано на рис. 1.

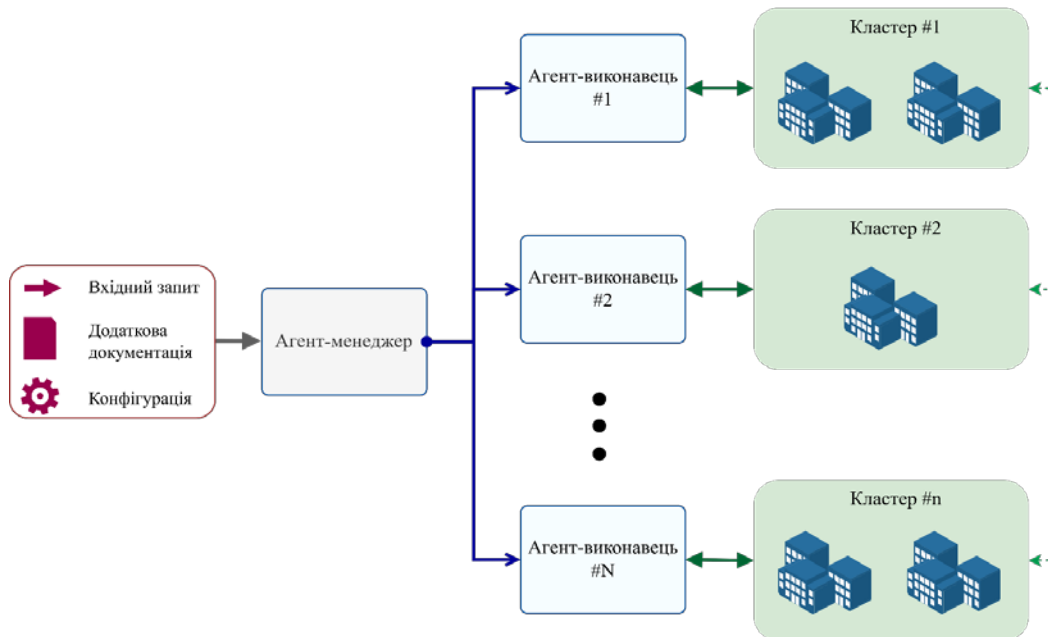


Рис. 1. Схема пропонуваного гібридного мультиагентного планувальника для РІС

Пропонований робочий процес мультиагентного планувальника для розподіленої інформаційної системи складається з таких етапів:

- Ініціалізація процесу через отримання вхідного запиту, що містить:
 - формалізований опис завдання;
 - набори даних для аналізу (з доступом до них);
 - супровідну документацію (за наявності).
- Опрацювання запиту мультиагентною системою:
 - декомпозиція завдання на підзадачі та формування плану їхнього розподілу між обчислювальними кластерами централізованим агентом-менеджером;
 - розподіл підзадач між кластерами;
 - локальне планування та оптимізація виконання завдань вузлами кластера під керуванням агента-виконавця.
- Паралельне виконання розподілених завдань на призначених обчислювальних кластерах.
- Агрегація результатів виконання через систему звітування кластерів та відповідних агентів про отримані та збережені результати.
- Оновлення контекстної пам'яті агента-менеджера на основі виконаного завдання.
- Формування та передача фінального результату користувачу.

Для оптимізації процесу розробки та експлуатації мультиагентної системи доцільно уніфікувати ключові характеристики агентів, методи їхньої взаємодії з середовищем та вимоги до базової LLM кожного агента. Як зазначено в попередньому розділі, найефективнішим способом взаємодії з інструментами є генерація та виконання програмного коду [8], що вимагає наявності відповідних програмних інтерфейсів для всіх інструментів у цільовій мові програмування. Застосування підхо-

ду CodeAct для роботи з інструментами не лише забезпечує ефективність агентів, але й легко інтегрується із запропонованим методом взаємодії агентів-виконавців з розподіленою інформаційною системою через програмні бібліотеки. Таким чином, вимога щодо якісної генерації коду базовою LLM одночасно задовольняє дві ключові характеристики агента.

Варто зазначити, що планування комплексних завдань у багатокластерній інтелектуальній розподіленій інформаційній системі передбачає численні виклики інструментів, що призводить до швидкого накопичення повідомлень у діалозі з LLM та перевищення обмежень контекстного вікна. Для вирішення цієї проблеми пропонується впровадження спеціального додаткового інструмента, який забезпечує доступ до різних типів пам'яті.

У роботі детально розглянуто архітектуру кожного типу агентів, починаючи з агента-менеджера, структурні компоненти якого показано на рис. 2.

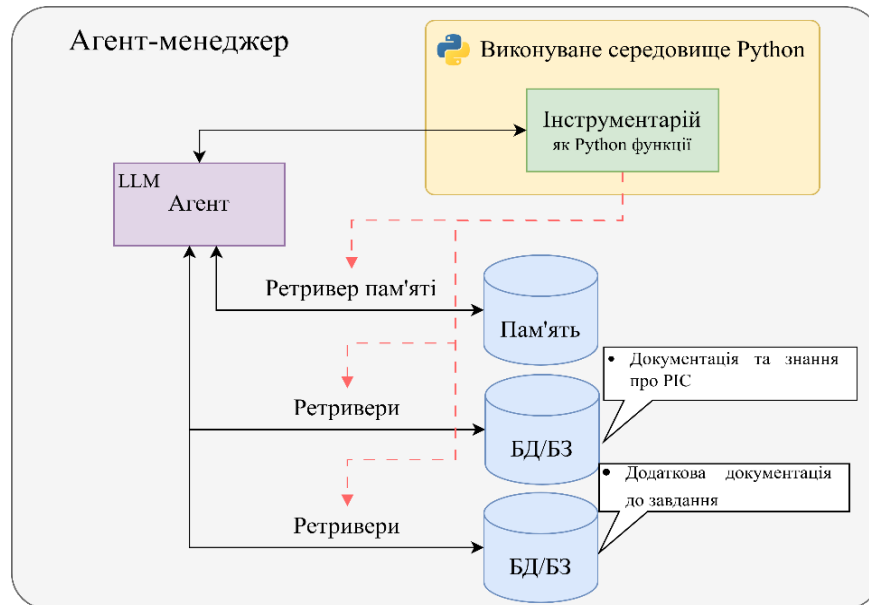


Рис. 2. Структура агента-менеджера

Основною функцією агента-менеджера є формування та розподіл підзадач для обробки даних між доступними кластерами РІС, враховуючи їхню гетерогенну структуру та характеристики. Для реалізації цієї функції агент використовує здібності логічного виведення базової LLM та спеціалізований набір інструментів.

Для забезпечення ефективного виконання цих комплексних завдань, агент-менеджер необхідно оснастити таким набором інструментів:

- інтерфейси доступу до бази знань РІС;
- компоненти взаємодії з предметно-орієнтованою базою знань;
- система пошуку в довготривалій історії моделі;
- програмний інтерфейс для конфігурування підзадач кластерів;
- комунікаційний інтерфейс для взаємодії з агентами-виконавцями.

Відповідно до визначених раніше вимог, всі взаємодії з інструментами реалізуються через генерацію та виконання Python коду [8], що вимагає відповідної реалізації програмних інтерфейсів усіх компонентів системи.

Наявність інструментів доступу до баз знань забезпечує можливість автономного доповнення контексту моделі за принципом класичного (наївного) RAG [12]. Для оптимізації реалізації пропонується використання гібридного пошуку, що поєднує семантичний пошук на основі ембедингів з пошуком за ключовими словами.

Зважаючи на широкий спектр можливих застосувань інтелектуальної РІС та варіативність результатів їхнього виконання, виникає необхідність селективного підходу до використання довготривалої пам'яті моделі. Для вирішення цього завдання пропонується додати спеціалізовані інструменти екстракції, що забезпечує вибірковий доступ до довготривалої пам'яті. Цей механізм передбачає включення до початкового пром프트 релевантних прикладів попередніх запитів, відібраних на основі ембединг-подібності до поточного завдання.

Після завершення етапу планування, агент-менеджер здійснює передачу конфігураційних да-

них агентам-виконавцям для подальшого використання обчислювальних ресурсів РІС. Взаємодія між агентами реалізується через стандартизований протокол комунікації, доступний агенту-менеджеру через спеціалізований Python API.

Після отримання підзадачі кластерним вузлом розподіленої інформаційної системи виникає необхідність забезпечити її виконання на відповідних обчислювальних ресурсах. З метою забезпечення універсальності системи планування щодо типів задач, що виконуються на РІС, необхідно реалізувати гнучку інфраструктуру програмного забезпечення для кластерних обчислень.

Враховуючи недоцільність розробки та підтримки великої кількості спеціалізованих програм для розподіленого виконання різноманітних інтелектуальних задач, пропонується делегувати відповідальність за створення виконуваного коду LLM-агенту, який визначається як агент-виконавець. Структура відповідного агента показана на рис. 3.

Основним завданням такого агента є підготовка програмного коду (через генерацію, модифікацію або адаптацію наявних рішень) для виконання інтелектуальних завдань у розподіленому середовищі. Зокрема, це передбачає створення Python-програм з використанням бібліотек, що підтримують розподілені обчислення. Після підготовки програмних файлів агент ініціює їхнє виконання на кластері через спеціалізований інструмент-тригер.

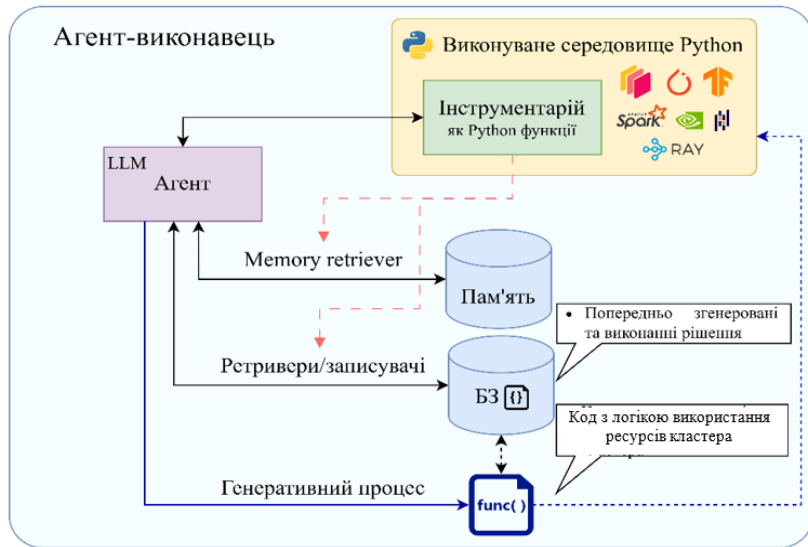


Рис. 3. Структура агента-виконавця

Оскільки всі агенти системи, згідно з раніше визначеними вимогами, повинні демонструвати високу якість генерації виконуваного коду, створення або модифікація програм для розподілених обчислень не потребує додаткових спеціалізованих можливостей базової LLM.

Для ефективної реалізації завдань, отриманих від агента-менеджера, агент-виконавець потребує доступу до такого набору інструментів:

- інтерфейс доступу до бази знань кластера;
- компонент взаємодії з предметно-орієнтованою базою знань;
- система реєстрації Python-файлів для виконання завдань;
- механізм тригерів для запуску обчислювальних завдань;
- компонент для доступу до бази попередніх пар «запит-реалізація» з метою повторного використання коду.

Запропонована архітектура мультиагентної системи для керування розподіленою інформаційною системою потенційно забезпечує гнучкий підхід до автоматизації планування та виконання складних обчислювальних задач. Використання спеціалізованих агентів на базі LLM, оснащених відповідними інструментами та механізмами доступу до баз знань, дозволяє ефективно проводити декомпозицію та розподіляти завдання між гетерогенними обчислювальними кластерами.

Ключовою перевагою запропонованого рішення є його універсальність щодо типів виконуваних завдань, що досягається через делегування відповідальності за генерацію виконуваного коду агентам-виконавцям, а також використання уніфікованих інтерфейсів взаємодії та спеціальних інструментних компонентів для ефективного управління контекстом та довготривалою пам'яттю системи. Така архітектура забезпечує масштабованість рішення та можливість його адаптації до різноманітних сценаріїв використання розподілених обчислювальних ресурсів.

Подальші дослідження можуть бути спрямовані на вдосконалення механізмів управління пам'яттю та оптимізацію інтеграції з різними типами обчислювальних середовищ, дослідити методи донавчання (fine-tuning) агентної LLM для поліпшення результатів та експериментальне впровадження запропонованої мультиагентної системи.

Висновки

Запропоновано новий підхід до автоматизації планування та керування розподіленими інформаційними системами на базі спеціалізованих LLM-агентів. Архітектура мультиагентного планувальника поєднує переваги централізованого та децентралізованого управління через впровадження гібридної системи планування, де агент-менеджер відповідає за глобальну декомпозицію та розподіл завдань, а агенти-виконавці забезпечують їхню локальну оптимізацію та реалізацію на рівні окремих обчислювальних кластерів.

Основною особливістю представленого рішення є його універсальність щодо типів виконуваних завдань та гетерогенності структури РІС, що досягається через делегування генерації виконувального коду агентам-виконавцям та використання уніфікованих інтерфейсів взаємодії. Застосування підходу CodeAct для роботи з інструментами та впровадження спеціалізованих інструментів для роботи з пам'яттю забезпечує ефективне управління контекстом та оптимальне використання обчислювальних ресурсів системи. Подальші дослідження в межах теми включатимуть експериментальне підтвердження результатів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi, "CoCoA: A General Framework for Communication-Efficient Distributed Optimization," *arXiv.org*, 2016. <https://arxiv.org/abs/1611.02189v2>. Accessed: Jan. 15, 2025.
- [2] S. Logie, D. Sabaz and W. A. Gruver, "Combinatorial sliding window scheduling for distributed systems," *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme – System Security and Assurance* (Cat. No.03CH37483), Washington, DC, USA, 2003, vol. 1, pp. 630-635. <https://doi.org/10.1109/ICSMC.2003.1243885>.
- [3] S. Teerapittayanon, B. McDanel, and K. H. T, "Distributed Deep Neural Networks over the Cloud, the Edge and End Devices," *arXiv.org*, 2017. <https://arxiv.org/abs/1709.01921v1>. Accessed: Jan. 15, 2025.
- [4] H. Karatza, "Scheduling in Distributed Systems," *Lecture Notes in Computer Science*, pp. 336-356, 2004, https://doi.org/https://doi.org/10.1007/978-3-540-24663-3_16.
- [5] J. Jiang, F. Wang, J. Shen, S. Kim, and S. Kim, "A Survey on Large Language Models for Code Generation," *arXiv.org*, Jun. 01, 2024. <https://arxiv.org/abs/2406.00515>.
- [6] C. Sypherd, and V. Belle, "Practical Considerations for Agentic LLM Systems," *arXiv.org*, 2024. <https://arxiv.org/abs/2412.04093>. Accessed: Jan. 16, 2025.
- [7] S. Han, Q. Zhang, Y. Yao, W. Jin, Z. Xu, and C. He, "LLM Multi-Agent Systems: Challenges and Open Problems," *arXiv.org*, Feb. 05, 2024. <https://arxiv.org/abs/2402.03578v1>.
- [8] T. Guo, et al., "Large Language Model based Multi-Agents: A Survey of Progress and Challenges," *arXiv.org*, 2024. <https://arxiv.org/abs/2402.01680v2>. Accessed: Jan. 17, 2025.
- [9] М. Хорканін, «Застосування машинного навчання для автоматизованого планування», *Вісник Хмельницького національного університету, серія: Технічні науки*, vol. 339, no. 4, pp. 11-14, 2024. <https://doi.org/10.31891/2307-5732-2024-339-4-1>.
- [10] Q. Wang, Z. Wang, Y. Su, H. Tong, and Y. Song, "Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key?" pp. 6106-6131, Jan. 2024. <https://doi.org/10.18653/v1/2024.acl-long.331>.
- [11] X. Wang, et al., "Executable Code Actions Elicit Better LLM Agents," *arXiv.org*, 2024. <https://arxiv.org/abs/2402.01030>.
- [12] P. Lewis, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv.org*, Apr. 12, 2021. <https://arxiv.org/abs/2005.11401>.
- [13] A. Singh, A. Ehtesham, S. Kumar, and Khoei Tala Talaei, "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG," *arXiv.org*, 2025. <https://arxiv.org/abs/2501.09136>. Accessed: Jan. 20, 2025.
- [14] S. Yao, et al., "ReAct: Synergizing Reasoning and Acting in Language Models," *arXiv.org*, Mar. 09, 2023. <https://arxiv.org/abs/2210.03629>.
- [15] Y. Gu, et al., "Middleware for LLMs: Tools Are Instrumental for Language Agents in Complex Environments," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7646-7663, Jan. 2024, <https://doi.org/10.18653/v1/2024.emnlp-main.436>.

Рекомендована кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ

Стаття надійшла до редакції 12.02.2025

Медяков Олександр Олександрович — студент Інституту комп'ютерних наук та інформаційних технологій, email: oleksandr.mediakov.mnsam.2023@lpnu.ua ;

Хорканін Максим Юрійович — аспірант кафедри інформаційних систем та мереж, email: maksym.y.khorkanin@lpnu.ua ;

Національний університет «Львівська політехніка», Львів

O. O. Mediakov¹
M. Yu. Khorkanin¹

Automated Planning in Intelligent Distributed Systems Using a Multi-Agent Approach Based on LLM

¹Lviv Polytechnic National University

The paper presents a novel approach to automating management of intelligent distributed information systems (DIS) using a hybrid multi-agent architecture with AI agents based on large language models (so-called LLM agents). The proposed architecture consists of a two-level planning system, which includes a manager agent for centralized global task decomposition and executor agents for decentralized local task assignment and computation optimization across individual clusters of the intelligent distributed system. To enhance agent interaction efficiency, the study proposes innovative methods for model access to long-term and short-term memory and tool utilization through code generation and execution, using the specialized prompt engineering method CodeAct. The CodeAct's high performance allows to create new agents, interacting with environments, execute interpreted code, and are able to collaborate with users using natural language. The proposed approach potentially provides architectural agnosticism regarding distributed system structure, allowing work with complex heterogeneous DIS and task types through specialized management components, memory access, and programmatic instruction execution generated by agents. The work includes descriptions of necessary skills and properties of the base large language models, the structure of two agent types, and the set of tools required for effective interaction of intelligent system's agents with each other and the external environment, including the managed DIS. The paper presents architectural diagrams of the manager agent and executor agents, describing the internal structure of all components and the mechanisms by which they interact. The research explains the promising potential of large language models and advanced AI agents for intelligent planning in distributed computing environments, revealing the potential of artificial intelligence in automating complex management processes

Keywords: LLM agents, automated planning, multi-agent systems, intelligent distributed information system, large language models.

Mediakov Oleksandr O. — Student of the Institute of Computer Science and Information Technology, email: oleksandr.mediakov.mnsam.2023@lpnu.ua ;

Khorkanin Maksym Y. — Post-Graduate Student of the Chair of Information Systems and Networks, email: maksym.y.khorkanin@lpnu.ua