

Касерес А.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Глоба Л.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ІНТЕГРАЦІЯ ШІ ТА МУЛЬТИАГЕНТНИХ СИСТЕМ ДЛЯ БАГАТОКРИТЕРІАЛЬНОГО АНАЛІЗУ У МУЛЬТИХМАРНОМУ СЕРЕДОВИЩІ

Сучасні вимоги до прийняття рішень у складних системах показують обмеження традиційних систем обробки з використанням багатокритеріального аналізу (MCDA) у динамічних, керованих даними середовищах. Статичні моделі MCDA не можуть масштабуватися та адаптуватися до реальних задач. Це ускладнює їх застосування до завдань інтеграції різноманітних і змінних критеріїв, наприклад, вартості, продуктивності і безпеки.

У цій роботі представлено ШІ-орієнтовану систему, яка включає великі мовні моделі (LLMs) і поєднує їх з мультиагентними системами для вирішення таких задач. Запропонована система автоматизує, масштабує та покращує процес прийняття рішень, динамічно реагуючи на зміни даних і забезпечуючи інтегровану оцінку за кількома критеріями. LLMs забезпечують аналіз вхідних даних, витягуючи складні залежності та вимоги, тоді як спеціалізовані агенти оптимізують ключові параметри. Агенти взаємодіють через централізоване ядро прийняття рішень, створюючи адаптивні рекомендації.

Ключовою інновацією є використання LLM для контекстуального аналізу вхідних даних та динамічних рекомендацій, що долає недоліки традиційних методів MCDA. Запропонована система протестована на прикладі вибору сервісів для оптимізації процесу обробки даних у мульти-хмарному середовищі, де показала покращення ефективності та зниження вартості використання ресурсів.

Співпраця з провідними консалтинговими компаніями у сфері хмарних технологій підтверджує актуальність запропонованого підходу, відкриваючи можливості для подальшого вдосконалення. Хоча система продемонстрована у сфері хмарних обчислень, її модульність дозволяє адаптацію до інших галузей, таких як охорона здоров'я, логістика та фінанси.

Таким чином, дослідження пропонує інноваційну, масштабовану систему, яка долає обмеження традиційних систем на базі MCDA, використовуючи LLM та мультиагентну архітектуру що забезпечує динамічну адаптацію обчислювального процесу до змін вхідних даних та інтегровану оцінку за заданим набором критеріїв. Подальша робота зосереджена на розширенні критеріїв, вдосконаленні взаємодії агентів і тестуванні у різних галузях.

Ключові слова: багатокритеріальний аналіз рішень, мультиагентні системи, великі мовні моделі, хмарні обчислення, автоматизація прийняття рішень.

Постановка проблеми. Мульти-хмарні середовища стають ключовим елементом сучасної технологічної інфраструктури, забезпечуючи масштабованість, гнучкість та підвищену продуктивність [1]. За останні роки використання мультихмарних рішень значно зросло, оскільки організації прагнуть уникнути залежності від одного провайдера і оптимізувати свої витрати [2]. Однак це створює нові виклики, пов'язані з інтеграцією сервісів, управлінням ресурсами та забезпеченням безпеки.

Для вирішення складних завдань вибору традиційно використовується такі підходи як бага-

токритеріальний аналіз (MCDA) [3]. Втім, його застосування у мультихмарних середовищах ускладнюється через динамічний характер цих систем, де дані та критерії змінюються в реальному часі. Зокрема, статичність моделей MCDA не дозволяє оперативно адаптуватися до змін умов, що робить їх малоефективними для вирішення реальних задач оптимізації у даній галузі.

Традиційні MCDA-моделі часто застосовуються для статичних середовищ, де параметри критеріїв не змінюються або змінюються дуже повільно [4]. Однак у мультихмарних системах рішення мають враховувати такі критерії [5]:

– Продуктивність – швидкість і затримки під час виконання завдань.

– Вартість – оптимізація витрат на використання хмарних ресурсів.

– Безпека – забезпечення конфіденційності та цілісності даних.

– Складність впровадження: необхідні витрати часу та зусиль на інтеграцію сервісів.

Значний рівень взаємозалежності між цими критеріями та динамічність умов створюють серйозні обмеження для використання MCDA.

З одного боку, мультихмарні середовища вимагають адаптивних, швидких і масштабованих рішень. З іншого боку, існуючі MCDA-моделі, попри свою точність, не враховують динамічність даних і залежності між критеріями. Отже, виникає необхідність у розробці інноваційного підходу, який би поєднував автоматизацію, адаптивність і можливість роботи з великими обсягами даних.

Аналіз останніх досліджень і публікацій. Багатокритеріальний аналіз рішень (MCDA) є фундаментальним інструментом для підтримки прийняття рішень у складних системах, де необхідно враховувати множинні, часто конфліктуючі критерії. Традиційні методи MCDA, такі як Аналітичний ієрархічний процес (АІП) та Метод аналізу переваг (TOPSIS), широко застосовуються для оцінки та ранжування альтернатив на основі визначених критеріїв [6]. Однак ці методи мають обмеження у динамічних та багатовимірних середовищах, оскільки вони зазвичай передбачають статичні моделі та не враховують швидкі зміни вхідних даних.

Мультиагентні системи (MAS) пропонують підхід до моделювання складних систем через взаємодію автономних агентів, кожен з яких має свої цілі та поведінку [7]. Мультиагентна система – набір агентів, кожен із яких відповідає за окремий критерій чи процес. Інтеграція MAS з MCDA дозволяє розподіляти процес прийняття рішень, що підвищує гнучкість та адаптивність системи. Дослідження показують, що використання MAS у поєднанні з MCDA сприяє ефективнішому вирішенню задач, де необхідно враховувати множинні критерії та динамічні зміни середовища [8].

Великі мовні моделі (LLM) – технології на базі штучного інтелекту, що аналізують дані та створюють текстові рекомендації [9]. LLM, такі як GPT-3, демонструють значний потенціал у розумінні та генерації природної мови, що відкриває нові можливості для автоматизації процесів прийняття рішень. LLM можуть аналізувати великі

обсяги текстових даних, витягувати релевантну інформацію та надавати рекомендації на основі контексту. Це особливо корисно у середовищах, де дані постійно оновлюються, і необхідно швидко адаптуватися до нових умов. Використання LLM для аналізу текстових даних, таких як новини та звіти, дозволяє агентам у MAS швидко адаптуватися до змін у постачаннях та приймати оптимальні рішення.

Ці приклади демонструють потенціал інтеграції LLM та MAS для розширення можливостей MCDA у різних галузях.

Постановка завдання. У цій роботі пропонується інтеграція мультиагентної AI-системи та великих мовних моделей аналізу контексту (LLM), що дозволяє вирішити зазначені вище проблеми. Основна мета полягає у:

- автоматизації процесу прийняття рішень;
- забезпеченні адаптивності до змінних даних;
- масштабуванні методів до динамічних середовищ.

Виклад основного матеріалу. Для ефективного впровадження багатокритеріального аналізу рішень у мультихмарних середовищах необхідно врахувати специфіку сучасних викликів. Зокрема, традиційні методи MCDA мають обмеження у динамічних, змінних середовищах, де швидко змінюються дані та критерії оптимізації. Серед основних проблем, які потрібно вирішити в рамках цього дослідження:

1. Як автоматизувати процес вибору оптимальних хмарних сервісів, враховуючи складність сучасних мультихмарних середовищ.

2. Як забезпечити динамічність і масштабованість системи у середовищах із постійно змінними параметрами.

3. Як інтегрувати великі мовні моделі для розширення можливостей традиційного MCDA.

4. Як адаптувати мультиагентний підхід для розв'язання задач, пов'язаних із оптимізацією хмарних сервісів.

Очікувані умови передбачають, що система повинна підтримувати адаптацію до змін у вимогах і критеріях, а також забезпечувати інтеграцію інструментів для автоматизованого прийняття рішень.

Для вирішення поставленої задачі пропонується мультиагентна система з інтеграцією великих мовних моделей для вдосконалення багатокритеріального аналізу рішень. Це дозволить забезпечити адаптивність, масштабованість та ефективність системи у динамічних середовищах з великою кількістю змінних критеріїв.

Завдання дослідження:

1. Розробити концептуальну архітектуру мультиагентної системи, що використовує великі мовні моделі для аналізу даних.

2. Розширити класичний підхід MCDA за рахунок інтеграції сучасних алгоритмів штучного інтелекту.

3. Забезпечити автоматизацію оцінки ключових критеріїв, таких як вартість, продуктивність та безпека, шляхом розробки відповідних агентів.

4. Провести тестування системи на реальних сценаріях мультиагентної оптимізації для оцінки її ефективності.

5. Здійснити аналіз зворотного зв'язку між критеріями для динамічної адаптації рекомендацій.

Очікувані результати: запропонований підхід має продемонструвати підвищення точності, швидкості та гнучкості у прийнятті рішень у складних середовищах, таких як мультиагентні системи.

Запропонований підхід базується на мультиагентній архітектурі, де кожен агент відповідає за аналіз конкретного критерію (наприклад, вартість, продуктивність, безпека) [10]. Велика мовна модель використовується як центральний компонент для обробки текстових запитів, аналізу залежностей та формулювання рекомендацій. Система побудована на принципах модульності, паралельного аналізу даних і інтеграції зворотного зв'язку для уточнення результатів (рисунок 1). Особливу увагу приділено доданню циклу зворотного зв'язку, що дозволяє системі динамічно адаптуватися до змін у вхідних параметрах та взаємозв'язках критеріїв.

Кожен агент у системі відповідає за оцінку одного з ключових критеріїв:

- Агент оптимізації вартості – аналізує ціни на сервіси, отримуючи актуальні дані з баз даних.
- Агент продуктивності – оцінює показники швидкодії та затримки для різних сервісів.
- Агент безпеки – визначає рівень ризиків, пов'язаних із доступом до даних та їх захистом.
- Агент впровадження – оцінює складність інтеграції конкретного сервісу в існуючу інфраструктуру.

Між агентами реалізована синхронна взаємодія через ядро прийняття рішень, яке забезпечує узгодженість результатів та формулювання оптимальної рекомендації. Нововведенням є механізм зворотного зв'язку, який дозволяє агентам переглядати оцінки в разі зміни контексту чи впливу одного критерію на інший.

Великі мовні моделі, зокрема LLAMA 3 [11], використовуються для:

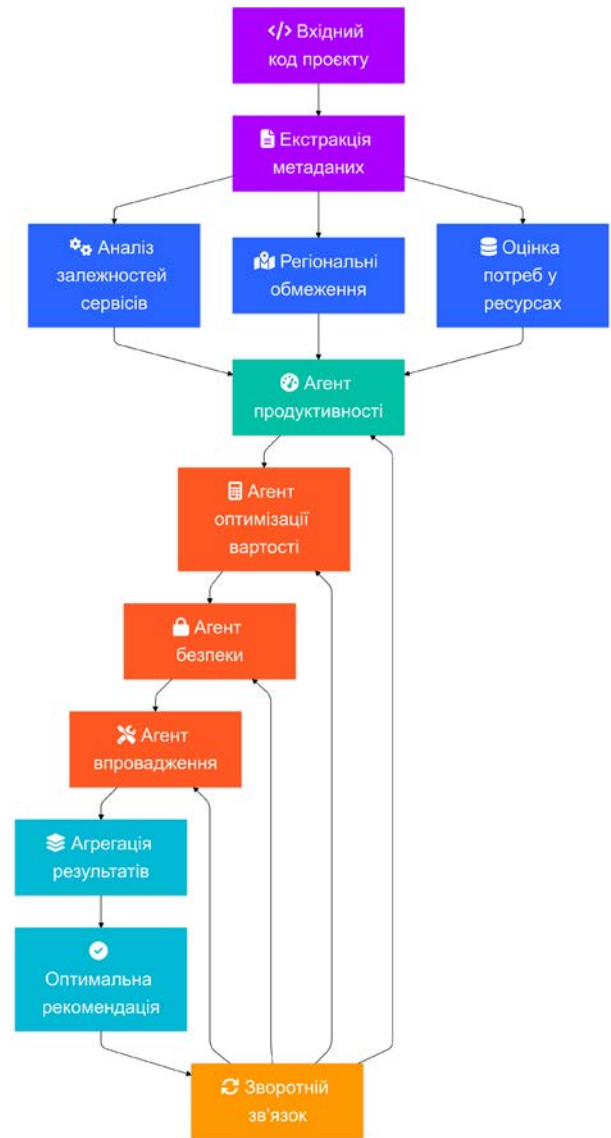


Рис. 1. Архітектура мультиагентної системи

- Розпізнавання залежностей між сервісами у вхідних даних.
- Формування контекстуальних запитів до агентів.
- Генерації текстових пояснень до запропонованих рекомендацій.

Цей підхід дозволяє автоматизувати аналіз вхідних даних, покращити якість оцінки критеріїв і забезпечити взаємодію між компонентами системи. LLM також використовується для обробки даних циклу зворотного зв'язку.

Робота системи включає такі основні етапи:

1. Система отримує запит користувача, що містить опис завдання та початкові обмеження.
2. LLM аналізує текстовий опис і виділяє ключові параметри для подальшого аналізу.

3. Кожен агент виконує аналіз свого критерію, використовуючи відповідні модулі обробки даних.

4. Ядро прийняття рішень об'єднує результати роботи агентів і формує оптимальну рекомендацію.

5. LLM генерує текстовий опис рекомендації, що надається користувачу.

6. Якщо результати виявляють невідповідність або вплив одного критерію на інші, система повторно ініціює аналіз із оновленими параметрами.

Для перевірки функціональності методології було створено прототип системи. У прототипі реалізовано базову взаємодію між агентами, використання LLM для екстракції метаданих та оцінки критеріїв. Також забезпечено автоматичне оновлення даних необхідних для роботи агентів (таких як ціни сервісів, список підтримуваних бібліотек та мов програмування, тощо).

Тестування прототипу на сценаріях оптимізації мультимарної інфраструктури показало:

- Підвищення точності рекомендацій.
- Зменшення часу на аналіз складних вхідних даних.
- Гнучкість у роботі зі змінними параметрами.

Цей підхід демонструє ефективність та перспективність у контексті динамічних середовищ.

Для реалізації системи було обрано такі технології:

– LangChain – для оркестрації взаємодії між агентами.

– Hugging Face Transformers – для інтеграції та налаштування великих мовних моделей.

– VectorDB – для зберігання контекстуальних даних.

– PostgreSQL – для управління структурованими даними, такими як метадані та параметри системи.

Причини вибору:

1. LangChain [12] забезпечує модульне та масштабоване середовище для побудови мульти-агентних систем.

2. Hugging Face [13] надає доступ до потужних попередньо навчених моделей, а також інструменти для їх адаптації до специфічних завдань.

3. VectorDB [14] підтримує швидкий доступ до контекстуально-структурованих даних, що є критично важливим для LLM.

4. PostgreSQL [15] – одна з найбільш популярних реляційних баз даних із широкими можливостями для роботи зі структурованими запитамі.

Ці технології відповідають вимогам щодо масштабованості, обробки даних у реальному часі та модульності.

Інтеграція VectorDB у системи із LLM вже продемонструвала успішність у проектах, пов'язаних із пошуком контекстуальної інформації [16]. Використання реляційних баз даних, таких як

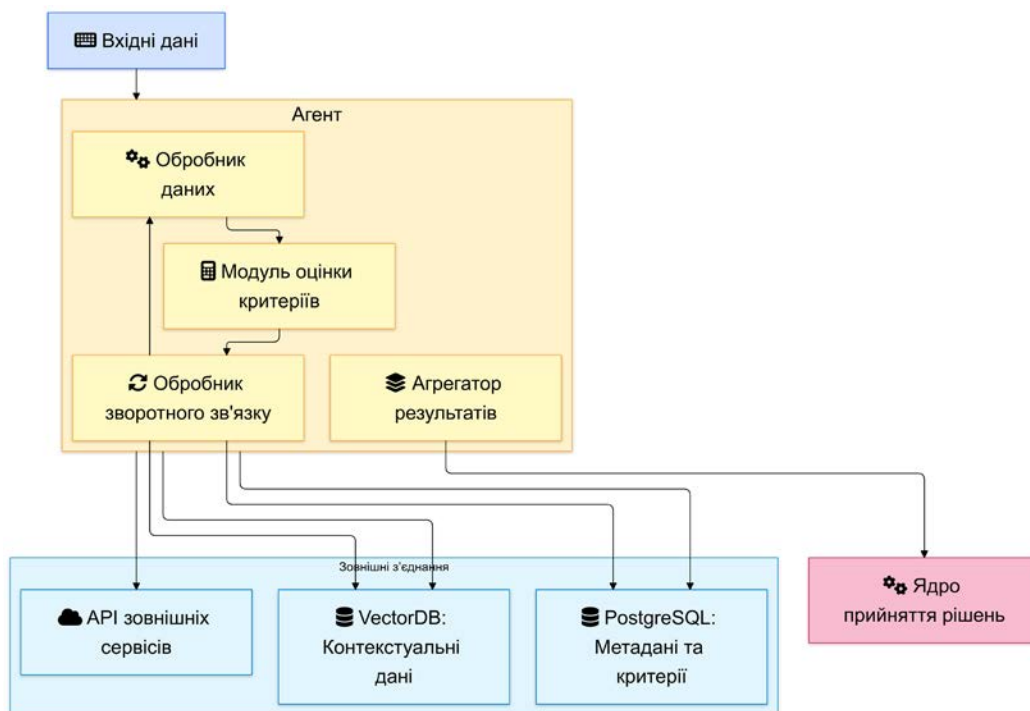


Рис. 2. Уніфікований шаблон для всіх агентів

PostgreSQL, довело свою ефективність у керуванні великими обсягами метаданих [17].

У розробці системи створено уніфікований шаблон для всіх агентів, що забезпечує модульність, масштабованість та підтримку зворотного зв'язку (рисунок 2). Цей шаблон спрощує інтеграцію нових функціональних модулів, оптимізуючи процес їх розробки та тестування.

Особливості шаблону:

1. Шаблон побудовано для підтримки різноманітних критеріїв аналізу. Наприклад, агент оптимізації вартості слугує базовою реалізацією, на основі якої створено інші агенти. Це автоматизує розробку довільних агентів по заданому шаблону.

2. Функціональність, яка виявляється в стандартизованому інтерфейсі для взаємодії з ядром прийняття рішень; інтеграції із PostgreSQL для доступу до структурованих метаданих і параметрів системи; отриманні контекстуальних даних із VectorDB для забезпечення більш точного аналізу критеріїв; взаємодії з API зовнішніх сервісів для отримання додаткових даних у реальному часі.

3. Підтримка зворотного зв'язку здійснюється у вигляді рекомендацій, сформованих агентами, які передаються до ядра прийняття рішень. Ядро аналізує результати роботи і формулює остаточний рейтинг альтернатив.

Серед переваги шаблону можна відзначити:

1. Легку інтеграцію нових агентів або оновлення існуючих.
2. Підтримку одночасної роботи кількох агентів із мінімальним впливом на продуктивність системи.

3. Завдяки зворотному зв'язку агенти здатні швидко адаптуватися до змін у критеріях чи даних.

4. Єдина архітектура полегшує перевірку коректності роботи нових агентів.

Розглянемо формування агенту оцінки на приклад агенту продуктивності, який аналізує такі показники, як затримка виконання та пропускна здатність передачі даних. Агент використовує VectorDB для розпізнавання залежностей між вхідними параметрами та PostgreSQL для збереження ключових фактів аналізу.

Інфраструктура обміну даними підтримує синхронну та асинхронну взаємодію між усіма компонентами системи, включаючи агентів, ядро прийняття рішень, та сховища даних (рисунок 3).

VectorDB використовується всіма агентами та LLM для отримання контекстуальних даних, необхідних для аналізу критеріїв. Наприклад, для аналізу залежностей між сервісами або формування запитів, які враховують раніше оброблену інформацію.

PostgreSQL слугує основним джерелом структурованих даних, таких як метадані (наприклад, специфікації сервісів і обмеження), поточні параметри критеріїв (наприклад, вартість, продуктивність). Кожен агент використовує PostgreSQL для доступу до результатів інших агентів або спільних даних.

Переваги цієї архітектури:

- Швидкий доступ до контекстуальної інформації через VectorDB.

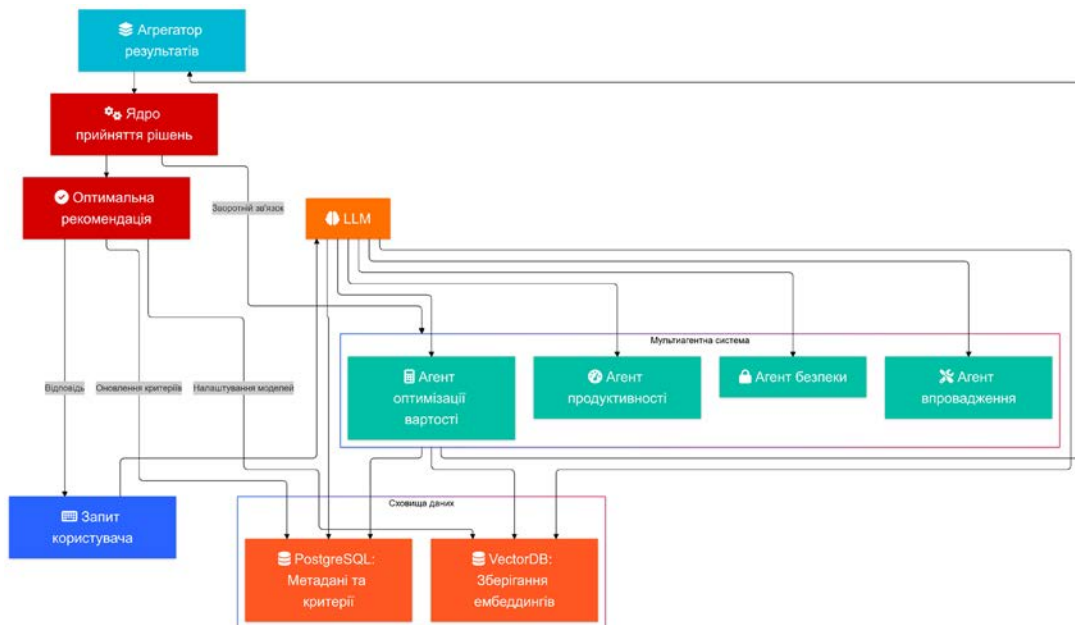


Рис. 3. Інфраструктура обміну даними

– Єдина база метаданих на основі PostgreSQL забезпечує узгодженість даних між агентами.

– Масштабованість: нова логіка агентів може бути додана без суттєвих змін в існуючих процесах.

Ця реалізація дозволяє забезпечити як ефективність, так і масштабованість системи, адаптуючи її до реальних сценаріїв використання.

Система була протестована в сценарії оптимізації мультихмарної інфраструктури. Сценарій передбачав вибір оптимальних хмарних сервісів за критеріями вартості (зменшення загальної вартості послуг), продуктивності (забезпечення низької затримки та високої пропускну здатності), безпеки (відповідність стандартам захисту даних) та складності впровадження (оцінка ресурсів, необхідних для інтеграції).

Процес тестування відбувався у такі етапи:

1. Було змодельовано запит користувача щодо підбору хмарних сервісів для виконання з описом завдань і обмежень.

2. Система отримувала вхідні дані, аналізувала їх за допомогою LLM і формувала запити до агентів.

3. Агентна система оцінювала критерії та повертала оптимальну рекомендацію через ядро прийняття рішень.

Результати демонструють, що запропонована система досягла значних переваг у порівнянні з традиційними підходами із застосуванням методу MCDA. Оцінка проводилася за показниками, які наведені у таблиці 1:

Результати свідчать про значний потенціал запропонованої системи.

Попри численні переваги, запропонована система має ряд обмежень:

1. Використання LLM вимагає значних обчислювальних потужностей, що може впливати на вартість впровадження в масштабних сценаріях.

2. Результати аналізу залежать від точності та повноти вхідних даних, що може бути проблемою в середовищах з недостатньо структурованими даними.

3. Система залежить від якості даних, які отримує з VectorDB і PostgreSQL, що створює окрему зовнішню залежність для коректної роботи.

Висновки. У статті представлено методологію, яка об'єднує потужність великих мовних моделей із модульністю мультиагентної архітектури. Це дозволило автоматизувати процес багатокритеріального аналізу рішень і покращити його адаптивність.

Запропонована система була протестована в контексті мультихмарної оптимізації обчислювальних ресурсів. Результати експериментів показали, що система може знижувати витрати, підвищувати продуктивність і забезпечувати вищу якість прийняття рішень у порівнянні з традиційними підходами на основі MCDA.

Використання модульної архітектури дозволяє легко додавати нові критерії, агенти або сценарії, що відкриває широкі можливості для адаптації системи до різноманітних завдань.

Запропоновану систему позитивно оцінили представники провідних хмарних консалтингових компаній, що підкреслює її актуальність для реального використання.

Основні напрямки подальших досліджень включають:

1. Планується додати нові критерії для аналізу, такі як відмовостійкість і відповідність вимогам фізичного розтушування даних.

2. Оптимізація механізмів зворотного зв'язку між агентами що дозволить забезпечити більш точну оцінку взаємозалежних критеріїв.

3. У співпраці з провідними консалтинговими компаніями буде проведено валідацію системи в реальних бізнес-сценаріях мультихмарної оптимізації.

4. Зниження вимог до обчислювальних ресурсів при використанні LLM стане ключовою задачею для зниження вартості системи.

5. Передбачається адаптація системи до роботи з новими мовними моделями.

Подальша робота дозволить розширити функціональність, покращити ефективність і підготувати систему до більш широкого впровадження.

Таблиця 1

Результат тестування системи

Показник	Результат
Зниження витрат	Система запропонувала оптимальні варіанти розгортання, які зменшили вартість на 15-20% у порівнянні з ручними методами аналізу
Покращення продуктивності	Використання агентів продуктивності дозволило скоротити затримку хмарних сервісів до 12-18 мс, що на 25% швидше за традиційні рішення.
Адаптивність	Впровадження зворотного зв'язку дозволило системі адаптувати критерії до змінних умов у реальному часі.
Автоматизація	Усі кроки процесу були виконані автоматично, зменшуючи втручання користувача та людські помилки.

Список літератури:

1. Saxena D., et al. A Survey and Comparative Study on Multi-Cloud Architectures: Emerging Issues And Challenges For Cloud Federation. *ArXiv*, 2021, abs/2108.12831. URL: <https://arxiv.org/abs/2108.12831> (дата звернення: 25.11.2024).
2. Flexera. Flexera 2024 State of the Cloud Report. URL: <https://info.flexera.com/CM-REPORT-State-of-the-Cloud> (дата звернення: 25.11.2024).
3. Linkov I., Moberg E. Multi-Criteria Decision Analysis: Environmental Applications and Case Studies. 1st ed. Boca Raton: CRC Press, 2012. 204 p.
4. Buriachenko A., Kuts N. Multi-Criterion Analysis Methods. 2021. URL: https://www.researchgate.net/publication/348774582_MULTI-CRITERION_ANALYSIS_METHODS (дата звернення: 26.11.2024).
5. Wang P., et al. Optimizing Data Placement for Cost-Effective and High Available Multi-Cloud Storage. *Computing and Informatics*, 2020, 39, pp. 51–82. DOI: 10.31577/cai20201-251.
6. Hanine M., Boutkhoum O., Tikniouine A., et al. Application of an Integrated Multi-Criteria Decision Making AHP-TOPSIS Methodology for ETL Software Selection. *SpringerPlus*, 2016, 5, p. 263. DOI: <https://doi.org/10.1186/s40064-016-1888-z> (дата звернення: 26.11.2024).
7. Dorri A., Kanhere, S. S., Jurdak R. Multi-Agent Systems: A Survey. *IEEE Access*, 2018, 6, pp. 28573–28593. DOI: 10.1109/ACCESS.2018.2831228.
8. Boutkhoum O., Hanine M., Agouti, T., Tikniouine, A. Multi-Agent Based Modeling Using Multi-Criteria Decision Analysis and OLAP System for Decision Support Problems. *World Academy of Science, Engineering and Technology*, 2015, 9, pp. 2243–2250.
9. Cloudflare. What is a Large Language Model (LLM)? URL: <https://www.cloudflare.com/learning/ai/what-is-large-language-model/> (дата звернення: 26.11.2024).
10. Li, Y., Tan, C. A Survey of the Consensus for Multi-Agent Systems. *Systems Science & Control Engineering*, 2019, 7(1), pp. 468–482. DOI: <https://doi.org/10.1080/21642583.2019.1695689> (дата звернення: 24.11.2024).
11. The Llama 3 Herd of Models. Llama Team, AI @ Meta. DOI: <https://doi.org/10.48550/arXiv.2407.21783> (дата звернення: 24.11.2024).
12. IBM. What is LangChain? URL: <https://www.ibm.com/topics/langchain>.
13. Lutkevich, B. What is Hugging Face? URL: <https://www.techtarget.com/whatis/definition/Hugging-Face#:~:text=Hugging%20Face%20lets%20users%20create,advance%20the%20field%20of%20NLP> (дата звернення: 26.11.2024).
14. VectorDB Documentation. URL: <https://vectordb.com/> (дата звернення: 26.11.2024).
15. PostgreSQL. PostgreSQL: The World's Most Advanced Open Source Relational Database. URL: <https://www.postgresql.org/> (дата звернення: 25.11.2024).
16. Smith, J., Brown, K. Contextual Information Retrieval Using Vector Databases. *Journal of AI Research*, 2023.
17. Johnson, R., Patel, V. Efficient Data Storage Solutions in Multi-Agent Systems. *Database Systems Review*, 2022.

Caceres A., Globa L.S. INTEGRATION OF AI AND MULTI-AGENT SYSTEMS FOR MULTI-CRITERIA DECISION ANALYSIS IN MULTI-CLOUD ENVIRONMENTS

Modern decision-making requirements in complex systems show the limitations of traditional multicriteria decision analysis (MCDA) in dynamic, data-driven environments. Static MCDA models cannot scale and adapt to real-world problems. Moreover, applying them to problems of integrating diverse and variable criteria, such as cost, performance, and safety, makes it difficult.

This paper presents an AI-based system that incorporates large language models (LLMs) and combines them with multi-agent systems to solve such problems. The proposed system automates, scales, and improves decision-making by dynamically responding to data changes and providing an integrated assessment across multiple criteria. LLMs provide input data analysis, extracting complex dependencies and requirements, while specialized agents optimize vital parameters. Agents interact through a centralized decision-making core to create adaptive recommendations.

The key innovation is using LLM for contextual analysis of input data and dynamic recommendations, eliminating gaps in traditional MCDA models. The proposed framework has been tested in cloud computing optimization, which has improved efficiency, accuracy, and usability. For example, the cost optimization agent calculates resource consumption, while the performance agent estimates latency and throughput. This collaborative approach has proven to be effective in real-world conditions.

Cooperation with leading cloud consulting companies confirms the relevance of this framework, offering opportunities for further improvement. The modularity and adaptability of the framework demonstrated in cloud systems make it applicable in other areas, such as healthcare, logistics, and finance.

In conclusion, this research presents a novel, scalable framework that overcomes the limitations of traditional MCDA by leveraging LLM and multi-agent systems. Ongoing work is focused on extending the criteria, improving agent interactions, and testing in various industries to ensure practical impact and applicability.

Key words: multi-criteria decision analysis, multi-agent systems, large language models, cloud computing, decision-making automation.