Інформаційні технології, комп'ютерні науки

УДК 004.415.2:004.8

Киселевич Володимир Володимирович
ORCID: 0009-0006-6449-710X
Асистент кафедри комп'ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка

ВАРТІСТЬ ВОЛОДІННЯ АГЕНТНОЮ АРХІТЕКТУРОЮ ЯК АРХІТЕКТУРНИЙ КРИТЕРІЙ

Розглянуто розрив між оптимізацією моделі та оптимізацією архітектури агентної системи. Побудовано модель оцінювання вартості агентної архітектури через кількість викликів моделі як функцію числа агентів і глибини кооперації із сімома явно виписаними припущеннями. Установлено, що число викликів лінійно залежить від числа агентів, натомість обсяг поданого контексту для схеми з обміном між усіма агентами пропорційний квадрату їх числа, тому саме обсяг контексту визначає межу застосовності схеми. Сформульовано умову доцільності мультиагентної схеми та обґрунтовано вартість як рівноправний архітектурний критерій поряд із надійністю. Ключові слова: агентні системи, великі мовні моделі, вартість володіння, архітектурний критерій, глибина кооперації, обсяг контексту.

Постановка проблеми. Проєктування програмних систем на основі агентів великих мовних моделей склалося так, що вартість обчислень під час висновування трактується як експлуатаційна характеристика, яку вимірюють після того, як схема кооперації агентів уже реалізована. Однак порядок величини цієї характеристики задається саме архітектурним рішенням. Систематичне вимірювання споживання токенів в агентних задачах кодування показало, що агентні завдання споживають приблизно у 1000 разів більше токенів, ніж режими міркування над кодом і діалогу про код, причому на тому самому завданні витрати різняться до 30 разів за загальним числом токенів, а вище споживання не переходить у вищу точність, бо точність часто досягає піку за проміжної вартості й далі насичується [1]. Системний аналіз агентів на рівні інфраструктури, виконаний з позицій архітектури обчислювальних систем, кількісно оцінив споживання ресурсів, затримки та енергоспоживання різних агентних архітектур і зафіксував швидко спадну віддачу від збільшення обчислень, зростання розкиду затримок та нестійкі інфраструктурні витрати [2]. Аналіз агентних бенчмарків показав, що вузьке зосередження на точності без урахування інших метрик призводить до появи систем, складність і вартість яких не є виправданими для досягнутого результату [3]. Проблема полягає в тому, що переважна більшість оптимізаційних робіт спрямована на модель, тобто на квантизацію, дистиляцію та скорочення числа параметрів, тоді як рішення про число агентів і схему їх взаємодії, яке визначає порядок величини витрат, ухвалюється без вартісного обґрунтування.

Аналіз останніх досліджень. Числові оцінки, наведені далі, походять переважно з препринтів (усі позиції списку джерел, окрім [2] та [4], мають статус препринта), тому їх слід трактувати як попередні, до завершення рецензування. Перший і значно численніший напрям робіт оптимізує модель та бюджет висновування. Показано, що ефективність стратегій масштабування обчислень під час висновування критично залежить від складності запиту, а розподіл бюджету за принципом оптимального розподілу обчислень підвищує ефективність масштабування більш ніж у чотири рази порівняно з базовим best-of-N, до того ж за однакового обсягу обчислень обчислення на етапі висновування дають перевагу над моделлю у 14 разів більшою на задачах, де менша модель має нетривіальний рівень успішності [5]. Емпіричний аналіз оптимального за обчисленнями висновування підтверджує цей висновок: модель Llemma-7B у поєднанні з алгоритмом деревного пошуку стабільно перевищує модель Llemma-34B на наборі MATH за всіх перевірених стратегій висновування [6]. Повторне генерування зразків підвищує частку розв'язаних задач на SWE-bench Lite для DeepSeek-Coder-V2-Instruct з 15,9 відсотка за одного зразка до 56 відсотків за 250 зразків, що перевищує найкращий результат за одного зразка, який становить 43 відсотки [7]. Бюджетування токенів методом примусового обмеження бюджету після донавчання на наборі s1K з 1000 питань дало моделі s1-32B перевагу над o1-preview до 27 відсотків на змагальних математичних задачах [4]. Вартісний бік цього напряму описує праця про гетерогенність тарифів різних програмних інтерфейсів великих мовних моделей, де плата може відрізнятися на два порядки, а каскадні стратегії знижують витрати до 98 відсотків за збереження рівня якості GPT-4 [8]. Другий напрям, представлений значно меншою кількістю праць, розглядає архітектуру композитної схеми. Встановлено немонотонність, а саме: продуктивність схем Vote і Filter-Vote спершу зростає, а потім спадає залежно від кількості викликів моделі, бо збільшення кількості викликів підвищує якість на легких запитах і знижує її на складних, для чого запропоновано аналітичну модель визначення оптимальної кількості викликів з малої кількості зразків [9]. Аналіз траєкторій виконання 30 завдань розробки програмного

забезпечення у середовищі ChatDev показав, що ітеративна стадія рецензування коду у середньому споживає 59,4 відсотка токенів, а вхідні токени стабільно становлять найбільшу частку споживання, у середньому 53,9 відсотка, тобто основна вартість лежить не в первинній генерації коду, а в автоматизованому доопрацюванні та верифікації [10]. Просте генерування зразків із голосуванням масштабує продуктивність із кількістю задіяних агентів, причому ступінь приросту корелює зі складністю завдання [11]. Автоматична генерація агентних робочих потоків у середньому дає приріст 5,7 відсотка порівняно з найкращими базовими методами й дає змогу меншим моделям перевищувати GPT-4o на окремих задачах за 4,55 відсотка його вартості висновування [12]. Водночас спроможність моделі не передбачає її кооперативності: за ідентичних інструкцій максимізувати спільний дохід модель o3 досягає лише 17 відсотків оптимальної колективної продуктивності, тоді як слабша o3-mini досягає 50 відсотків [13]. Спільним для перелічених праць є те, що вартість фіксується як наслідок обраної схеми, а не як змінна проєктування, і аналітичного зв'язку між схемою кооперації, числом агентів та очікуваним обсягом поданого контексту в них не побудовано.

Мета. Метою роботи є побудова моделі оцінювання вартості володіння агентною архітектурою через кількість викликів моделі як функцію кількості агентів і глибини кооперації з явно виписаними припущеннями, формулювання умови доцільності мультиагентної схеми та обґрунтування вартості як рівноправного архітектурного критерію поряд із надійністю.

Виклад основного матеріалу. Запропоновано модель оцінювання вартості агентної архітектури, що спирається на сім явно виписаних припущень. Перше: одиницею вартості є один виклик моделі, бо тарифікація постачальників прив'язана до обсягу поданих і згенерованих токенів [8]. Друге: число агентів-виконавців позначено через n, глибину кооперації (число раундів обміну) через d, а частку кроків з окремою перевіркою через v. Третє: одноагентна схема виконує d ітерацій циклу «міркування, дія, спостереження». Четверте: оркестрована схема витрачає на кожному раунді n викликів виконавців та один виклик оркестратора. П'яте: кооперативна схема з обміном між усіма агентами витрачає на кожному раунді по одному виклику на агента, але подає кожному з них виходи всіх інших. Шосте: обсяг контексту накопичується, тобто k-й виклик несе історію попередніх кроків. Сьоме: модель не враховує кешування префікса контексту та відмінності тарифів для поданих і згенерованих токенів. Останнє припущення означає, що отримані оцінки обсягу контексту є верхньою межею.

Число викликів для одноагентної схеми дорівнює глибині кооперації. Для оркестрованої схеми воно становить добуток глибини на число агентів, збільшене на одиницю. Для кооперативної схеми воно є добутком глибини на число агентів. Уведення окремого кроку перевірки збільшує підсумкове число викликів у стільки разів, скільки становить одиниця плюс частка кроків з перевіркою. Обсяг поданого контексту оцінюється інакше. Для оркестрованої схеми він пропорційний добуткові числа агентів, збільшеного на одиницю, на суму натурального ряду до глибини кооперації, тобто на половину добутку глибини на глибину, збільшену на одиницю. Для кооперативної схеми він пропорційний добуткові квадрата числа агентів на ту саму суму. Коефіцієнтом пропорційності є середній обсяг одного кроку в токенах. Числові значення моделі для трьох схем і трьох значень глибини кооперації подано в табл. 1, де середній обсяг одного кроку прийнято однаковим для всіх схем.

Таблиця 1

Число викликів моделі та обсяг поданого контексту за схемами кооперації

Схема	Агентів n	Глибина d	Викликів	Контекст, тис. токенів
одноагентна	1	3	3	4,2
одноагентна	1	5	5	10,5
одноагентна	1	8	8	25,2
оркестрована	3	3	12	16,8
оркестрована	3	5	20	42,0
оркестрована	3	8	32	100,8
оркестрована	5	5	30	63,0
оркестрована	5	8	48	151,2
кооперативна	3	5	15	94,5
кооперативна	3	8	24	226,8
кооперативна	5	5	25	262,5
кооперативна	5	8	40	630,0

Джерело: розробка автора.

Кратності, що випливають з моделі, розходяться за двома показниками. За числом викликів оркестрована схема дорожча за одноагентну в чотири рази за трьох агентів і в шість разів за п'яти, а кооперативна схема дорожча відповідно втричі та в п'ять разів, тобто залежність є лінійною щодо кількості агентів. За обсягом поданого контексту оркестрована схема зберігає ті самі кратності, натомість для кооперативної схеми обсяг

контексту пропорційний квадрату числа агентів і є дев'ятикратним для трьох агентів та двадцятип'ятикратним для п'яти. Головний висновок моделі полягає в тому, що межу застосовності схеми з обміном між усіма агентами визначає обсяг контексту, а не число викликів: за п'яти агентів і восьми раундів обміну кооперативна схема потребує менше викликів, ніж оркестрована (40 проти 48), але подає у понад чотири рази більший обсяг контексту (630,0 проти 151,2 тис. токенів). Цей результат узгоджується з емпіричним спостереженням, що вхідні токени становлять найбільшу частку споживання [10], і дає одне з можливих пояснень того, чому розкид витрат на тому самому завданні досягає 30 разів [1].

Виокремлено різницю в незворотності двох класів проєктних рішень. Оптимізація моделі є замінною: квантизовану чи дистилювану модель можна замінити іншою без зміни коду системи, а гетерогенність тарифів дає змогу перенести навантаження на дешевшого постачальника [8]. Архітектурне рішення є важкозмінним: перехід від кооперативної схеми до оркестрованої означає переписування протоколу обміну, форматів повідомлень і механізму агрегації результатів. Оскільки обсяг поданого контексту для кооперативної схеми пропорційний квадрату числа агентів, помилка на етапі вибору схеми не компенсується жодною подальшою оптимізацією моделі, бо остання змінює множник, а не порядок величини. Практичним наслідком є те, що вибір числа агентів і схеми обміну доцільно фіксувати разом з оцінкою обсягу поданого контексту для найгіршого сценарію глибини кооперації, тобто до написання коду, а не за результатами першого промислового навантаження. Звідси вартість слід уводити не як експлуатаційну характеристику після впровадження, а як рівноправний архітектурний критерій поряд із надійністю, оскільки рішення з більшою незворотністю мусить ухвалюватися з урахуванням вартості на етапі проєктування.

Сформульовано умову доцільності мультиагентної схеми: така схема виправдана лише тоді, коли приріст частки успішно розв'язаних задач перевищує кратність зростання вартості, помножену на граничну вартість одиниці якості для конкретного застосування. Практичне наповнення цієї умови ілюструють наявні вимірювання. Приріст з 15,9 відсотка до 56 відсотків за 250 зразків [7] відповідає кратності зростання вартості близько 250, тому умова виконується лише за високої граничної вартості одиниці якості, характерної для задач з дорогою ручною альтернативою. Натомість приріст 5,7 відсотка за 4,55 відсотка вартості моделі найвищого рівня [12] задовольняє умову за значно нижчого порогу. Немонотонність якості за числом викликів [9] означає, що ліва частина умови не є монотонною, тому перевірку слід виконувати не для максимального, а для кількох проміжних значень глибини кооперації; результати про насичення точності за вищої вартості [1] та про швидко спадну віддачу [2] задають верхню межу розумної глибини. Додаткове застереження дає результат про те, що спроможніша модель кооперує гірше [13], бо він унеможливлює оцінювання очікуваного приросту точності з характеристик окремої моделі, узятої поза схемою кооперації. Отже, ліву частину умови доводиться вимірювати для схеми загалом, а не виводити з опублікованих показників базової моделі. Проста схема з генеруванням зразків і голосуванням [11] є доречною точкою відліку для перевірки умови, оскільки її вартість збігається з кратністю числа агентів, а приріст якості вимірюється безпосередньо.

Обмеження. Робота не містить власного експерименту, тому запропонована модель перевірена лише на узгодженість з опублікованими вимірюваннями, а не з прямим вимірюванням витрат у робочій системі. Усі числові кратності отримані аналітично з припущень моделі, а не з реальних траєкторій виконання, і зберігають чинність лише тоді, коли ці припущення виконуються в конкретній реалізації. Припущення про накопичення повної історії кроків дає завищену оцінку, бо кешування префікса контексту та стратегії скорочення історії знижують фактичний обсяг поданих токенів, і модель цього не враховує. Трактування одного виклику як однорідної одиниці вартості є умовним, оскільки тарифи для поданих і згенерованих токенів різняться, а плата між постачальниками відрізняється на два порядки [8]. Статистичної перевірки не виконано: не оцінено ні дисперсії кратностей, ні значущості розбіжностей між схемами, а середній обсяг одного кроку в токенах прийнято однаковим для всіх трьох схем, хоча кооперативна схема потребує довших службових повідомлень. Вибірка опрацьованих джерел обмежена тринадцятьма працями, переважна частина яких має статус препринта, а частина агентних оцінювань має проблеми з відтворюваністю [3]. Сформульована умова доцільності містить величину граничної вартості одиниці якості, яка задається зовні предметною галуззю і в роботі не визначається.

Висновки. Показано, що переважна більшість робіт з економіки агентних систем оптимізує модель та бюджет висновування, тоді як архітектурні рішення, які визначають порядок величини витрат, залишаються поза вартісним аналізом. Побудовано модель оцінювання вартості агентної архітектури через кількість викликів моделі як функцію кількості агентів і глибини кооперації із сімома явно виписаними припущеннями та наведено її числові значення для трьох схем кооперації. Головний результат моделі полягає в тому, що кількість викликів моделі лінійно залежить від кількості агентів, натомість обсяг поданого контексту для схеми з обміном між усіма агентами пропорційний квадрату їхньої кількості, тобто дев'ятикратним для трьох агентів і двадцятип'ятикратним для п'яти, тому межу застосовності такої схеми визначає обсяг контексту, а не число викликів. Обґрунтовано потребу вводити вартість як рівноправний архітектурний критерій поряд із надійністю,

з огляду на різницю в незворотності: архітектурне рішення є важкозмінним, тоді як оптимізація моделі є замінною. Сформульовано умову доцільності мультиагентної схеми через співвідношення приросту частки розв'язаних задач, кратності зростання вартості та граничної вартості одиниці якості. Подальші дослідження доцільно спрямувати на емпіричну перевірку кратностей моделі на реальних трасах виконання з урахуванням кешування префікса контексту.

1. Bai L., Huang Z., Wang X. [та ін.] How Do AI Agents Spend Your Money? Analyzing and Predicting Token Consumption in Agentic Coding Tasks : препринт. arXiv:2604.22750. 2026. URL: https://arxiv.org/abs/2604.22750.
2. Kim J., Shin B., Chung J. [та ін.] The Cost of Dynamic Reasoning: Demystifying AI Agents and Test-Time Scaling from an AI Infrastructure Perspective. 2026 IEEE International Symposium on High Performance Computer Architecture (HPCA). 2026. P. 1-16. DOI: 10.1109/hpca68181.2026.11408569.
3. Kapoor S., Stroebl B., Siegel Z. S. [та ін.] AI Agents That Matter : препринт. arXiv:2407.01502. 2024. URL: https://arxiv.org/abs/2407.01502.
4. Muennighoff N., Yang Z., Shi W. [та ін.] s1: Simple test-time scaling. Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. 2025. P. 20286-20332. DOI: 10.18653/v1/2025.emnlp-main.1025.
5. Snell C., Lee J., Xu K. [та ін.] Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters : препринт. arXiv:2408.03314. 2024. URL: https://arxiv.org/abs/2408.03314.
6. Wu Y., Sun Z., Li S. [та ін.] Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for Problem-Solving with Language Models: препринт. arXiv:2408.00724. 2024. URL: https://arxiv.org/abs/2408.00724.
7. Brown B., Juravsky J., Ehrlich R. [та ін.] Large Language Monkeys: Scaling Inference Compute with Repeated Sampling : препринт. arXiv:2407.21787. 2024. URL: https://arxiv.org/abs/2407.21787.
8. Chen L., Zaharia M., Zou J. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance : препринт. arXiv:2305.05176. 2023. URL: https://arxiv.org/abs/2305.05176.
9. Chen L., Davis J. Q., Hanin B. [та ін.] Are More LLM Calls All You Need? Towards Scaling Laws of Compound Inference Systems : препринт. arXiv:2403.02419. 2024. URL: https://arxiv.org/abs/2403.02419.
10. Salim M., Latendresse J., Khatoonabadi S. [та ін.] Tokenomics: Quantifying Where Tokens Are Used in Agentic Software Engineering : препринт. arXiv:2601.14470. 2026. URL: https://arxiv.org/abs/2601.14470.
11. Li J., Zhang Q., Yu Y. [та ін.] More Agents Is All You Need : препринт. arXiv:2402.05120. 2024. URL: https://arxiv.org/abs/2402.05120.
12. Zhang J., Xiang J., Yu Z. [та ін.] AFlow: Automating Agentic Workflow Generation : препринт. arXiv:2410.10762. 2024. URL: https://arxiv.org/abs/2410.10762.
13. Yadav A., Black S., Sourbut O. More Capable, Less Cooperative? When LLMs Fail At Zero-Cost Collaboration: препринт. arXiv:2604.07821. 2026. URL: https://arxiv.org/abs/2604.07821.