

Слюсар Вадим Іванович
доктор технічних наук, професор
Центральний науково-дослідний інститут озброєння та військової техніки
Збройних Сил України, м. Київ.

Концепція мультиагентної архітектури керування квантовим комп'ютером

Квантові обчислення стрімко удосконалюються, але їхня архітектура керування переважно залишається монолітною та централізованою. Для усунення цього недоліку можуть бути використані технології великих мовних моделей (LLM), які відкривають нові можливості для квантових обчислень, зокрема оптимізації програмного коду, декомпозиції завдань природною мовою та ін.

Метою роботи є обґрунтування архітектури керувального сегменту квантового комп'ютера у вигляді когнітивної мультиагентної системи.

Основне ідея полягає у переході від жорстких послідовностей керування до екосистеми інтелектуальних підсистем, кожна з яких контролюється спеціалізованим агентом на основі LLM. Така система не лише забезпечить безперервну оптимізацію процесу квантових обчислень, але й створить комфортне середовище для взаємодії людини з машиною, інтерпретованості результатів та безпечної експлуатації.

Мінімальний за спроможностями варіант запропонованої архітектури керування квантовим комп'ютером може налічувати до 6 агентів, що взаємодіють через багатоканальну комунікаційну шину. Вона реалізує розподілений простір для внутрішнього обміну повідомленнями та контекстами на основі спеціальних протоколів, наприклад, A2A для міжагентських комунікацій та MCP - для з'єднань з виконавчими системами і сховищами даних.

Процес обчислень у запропонованій архітектурі починається з агента інтерфейсу користувача, який приймає на вході запит оператора квантового комп'ютера у вигляді природної мови, математичного опису задачі, аудіоконтенту або візуального інтерфейсу. Спираючись на спроможності з семантичного аналізу та трансляції природної мови і мультимодальних даних у формальні структури, даний агент перетворює задачу на формалізований опис (наприклад, у форматі QASM) та передає його далі до агенту квантового компілятора. Той, в свою чергу, будує квантову схему, обирає гейти, оптимізує глибину та зменшує кількість операцій на рівні кубітів з урахуванням топології. При цьому забезпечується метаевристичне планування процесу обчислень та перетворення графів задач у гейтові послідовності. Такий набір завдань дозволяє вважати даного агента планувальником усіх процесів, з яким тісно співпрацює окремий агент контролера квантового процесора (QPU). Його призначенням є оркестрація виконання гейтів на фізичних кубітах з контролем витрат часу, координація

запуску квантових програм, управління мікрохвильовими імпульсами або лазерними системами шляхом перетворення квантових схем у низькорівневі сигнали управління.

Синхронізація квантових обчислень з зовнішніми класичними процесами забезпечується агентом гібридної інтеграції. Наприклад, у випадках Variational Quantum Eigensolver або QAOA він підтримує ітеративну взаємодію між квантовою та класичною частинами комп'ютерної системи. Його компетенцією є керування ітеративними циклами між QPU і CPU/GPU, адаптивна стратегія підбору параметрів тощо.

Для забезпечення захисту каналів від перехоплення (наприклад, шляхом перевірки QKD), управління правами доступу до квантових ресурсів і взагалі квантової безпеки до схеми включено агента безпеки. Його місія зводиться до аналізу протоколів автентифікації, формування політик доступу, виявлення кібератак та їх ідентифікації в інтересах вибору оптимальної стратегії протидії.

Нарешті, останнім в зазначеному ланцюгу обробки даних є агент аналізу вимірювань. Він отримує результати зчитування кубітів, статистично обробляє їх та формує узагальнене представлення. Видані цим агентом дані постобробки та агрегування результатів подаються комунікаційною шиною на агента інтерфейсу користувача для візуалізації або озвучування результатів обчислень.

Звісно, наведені вище назви агентів є досить умовними і можуть змінюватися, так само як і набір їх функцій, однак наведений їх перелік, по суті, є мінімально необхідним для ефективного функціонування квантової чи гібридної комп'ютерної системи.

Подальше удосконалення описаної схеми може бути досягнуте шляхом включення до неї додаткового агента навчання і оптимізації. Цей агент повинен збирати інформацію про параметри обчислень з усіх їх етапів, виконувати мета-аналіз, щоб адаптувати систему до майбутніх запитів. Він будує базу знань про найкращі стратегії для різних типів задач, здійснює історичний аналіз, реалізує процеси навчання з підкріпленням (reinforcement learning) та інші методи глибокого навчання.

Враховуючи необхідність виявлення та виправлення помилок у функціонуванні кубітів, більш просунуті мультиагентні системи квантових обчислень мають обов'язково містити агента управління помилками.

Його наявність забезпечує аналіз обраних маршрутів обчислень, застосування квантових кодів коригування помилок (наприклад, код Шора, LDPC). Також цей агент спостерігає за станами кубітів, прогнозує втрати когерентності, забезпечує моніторинг квантових шумів, статистичний прогноз і автоматичний вибір схем корекції. На основі спеціальних метрик декогерентності в його складі запускаються маскові механізми уваги, які дозволяють блокувати надмірно зашумлені кубіти.

Описаний архітектурний принцип має значний потенціал для розвитку і дозволить досягти більшої ефективності і масштабованості для гібридних

класично-квантових платформ. Він створює можливість адаптації та самоорганізації квантових обчислень, чого бракує відомим QPU.

Для демонстрації спроможностей запропонованої архітектури було проведено її моделювання на симуляторі квантового комп'ютера. Цифровий двійник мультиагентної системи керування був створений у вигляді промпта для мовної моделі Gemini, який викликався через API-інтерфейс з програми на Python, розгорнутої у середовищі Google Colab.

Після завантаження ключів API та ініціалізації сесії запускався базовий системний промпт, який визначав рольову взаємодію агентів. При цьому була задіяна лише частина з описаного вище набору агентів:

L&O Agent управляв загальним перебігом діалогу,

User Interface Agent приймав й уточнював запит дослідника,

Quantum Compiler Agent транслював природномовну постановку задачі у код Cirq,

QPU Controller Agent відповідав за виконання схеми на симуляторі,

Measurement Analysis Agent інтерпретував статистику вимірювань.

Згідно з планом експерименту, дослідник сформулював задачу побудови стану Белла $|\Phi^+\rangle$ на двох кубітах із подальшим вимірюванням при 1000 пострілах. Quantum Compiler Agent синтезував схему Cirq H–CNOT–measure і передав її виконавчому агенту, який запустив локальну симуляцію на cirq.Simulator. Перше виконання обчислень дало майже рівномірний розподіл між двома базисними станами $\{ '00': 491, '11': 533 \}$, що й очікувалося для ідеального вектору $(|00\rangle + |11\rangle)/\sqrt{2}$. Measurement Analysis Agent пояснив відхилення від точної рівності стандартною статистичною флуктуацією, характерною для скінченої вибірки.

Для перевірки стабільності процедури було повторено запуск тієї самої схеми Белла. Друге виконання обчислень дало гістограму з незначними варіаціями $\{ '00': 507, '11': 517 \}$, підтвердивши стохастичну природу вимірювань при збереженні очікуваного співвідношення 50/50. У обох випадках симулятор коректно відтворив квантову кореляцію, а коливання на рівні $\pm 3\%$ вкладаються у довірчий інтервал \sqrt{N}/N для $N = 1000$.

Процес симуляції продемонстрував, що модульна мультиагентна архітектура забезпечує чітке розділення обов'язків: мовна модель сприймає завдання у природній мові, компілятор автоматизує генерування схеми, а виконавчі та аналітичні агенти гарантують прозоре виконання й верифікацію результатів. Такий підхід спрощує прототипування квантових алгоритмів, дозволяє швидко повторювати експерименти без глибокого занурення у низькорівневий код і може бути масштабований на більшу кількість кубітів або перенесений на апаратні QPU. Отримані статистики вимірювань підтверджують правильність реалізації Беллового стану, а мультиагентна схема демонструє перспективу для автоматизованих досліджень у квантовій інформатиці.