

<https://habr.com/ru/articles/949650/>

Ruslan Dev @ruslandevlabs

23.09.2025

Облачные технологии в контексте агентских AI-систем

В настоящее время процветает разработка агентов — приложений на базе Generative AI, реализующих автономные рабочие процессы. Извлечение и анализ данных, управление детерминированными программами и так далее. Массу вещей можно автоматизировать с помощью LLM и вызова функций, отсюда и спрос на такие системы.

Как и традиционное ПО, агенты обычно реализуют принцип разделения логики на специализированные узлы обработки конкретных задач, например, один парсит данные по определенной инструкции, другой их анализирует, третий формирует конечный результат. Необходимость применения такого подхода диктуется не только архитектурным принципом единой ответственности, который делает систему более предсказуемой, но и ограничениями самого ИИ. При попытке выполнить несколько задач вызовом одного промпта ограничивающим фактором является механизм внимания, который лучше справляется, если его не слишком перегружать вводными данными. А при использовании разных промптов для разных задач главное ограничение — в том, на сколько доменов модель успешно генерализуется без дообучения под каждый из них.

Поэтому агентские системы обычно представляют собой микросервисную архитектуру, объединяющую несколько логических блоков. Взять хотя бы ReAct паттерн, в котором reasoning агент выполняет управляющую роль, а action — отвечает за вызов функций и выполнение других действий.

И, как в любой микросервисной архитектуре, мы приходим к необходимости горизонтального масштабирования, что удобнее всего делать в облаке. Во-первых, сами LLM — очень ресурсоемкие элементы системы, инфраструктура для них требует сложной конфигурации. Об этом больше сказано ниже. Во-вторых, обмен данными между ними тоже требует правильного проектирования и развертывания дополнительных сервисов, как например векторная БД, механизмы стриминга данных, кэширования, чанкования, эмбедингов.

Главные трудности в конфигурации облачной инфраструктуры под AI агенты, согласно моей практике, связаны с расчетом нагрузки. Особенно это касается GPU-инфраструктуры, ответственной за хостинг LLM. Эта тема заслуживает отдельной статьи — с появлением различных алгоритмов внимания, квантизации моделей и механизмов выделения GPU-памяти, правильный подбор мощностей требует всё более нетривиальных вычислений. Было бы хорошо, если бы эта проблема также была решена на уровне облака и снимала с AI-инженеров заботу о правильно подобранной инфраструктуре. Но мы находимся на ранней ступени адаптации облаков под нужды AI, и многое приходится изобретать самостоятельно.

Зачем вообще мне понадобилось разворачивать свои ИИ модели в облаке, если можно обратиться к API OpenAI, DeepSeek или Qwen? На практике среди преимуществ открытых весов неожиданно в приоритете оказалась скорость генерации ответа. Для ряда агентских систем критическим параметром является именно скорость. На нее оказывают влияние сетевая связность, размер весов и тип квантизации, мощность GPU, параллелизм запросов, размер контекстного окна и масса других параметров, к которым разработчик далеко не всегда имеет доступ, работая с провайдерами закрытых моделей типа OpenAI.

Именно поэтому через практику я пришел к выводу, что индустрии разработки агентских приложений необходима облачная платформа, предоставляющая полный доступ к весам, данным и инфраструктуре — без этого систему не построить. С другой стороны, роль облака — максимально облегчать выделение ресурсов под LLM, дать разработчику что-то вроде «торгового автомата для LLM». Выбрал веса модели, размер контекста, параметры масштабирования, нажал на пару кнопок и получил готовый LLM-сервер. Я предполагаю, что после устранения описанных инфраструктурных трудностей порог входа в разработку AI агентов существенно снизится.