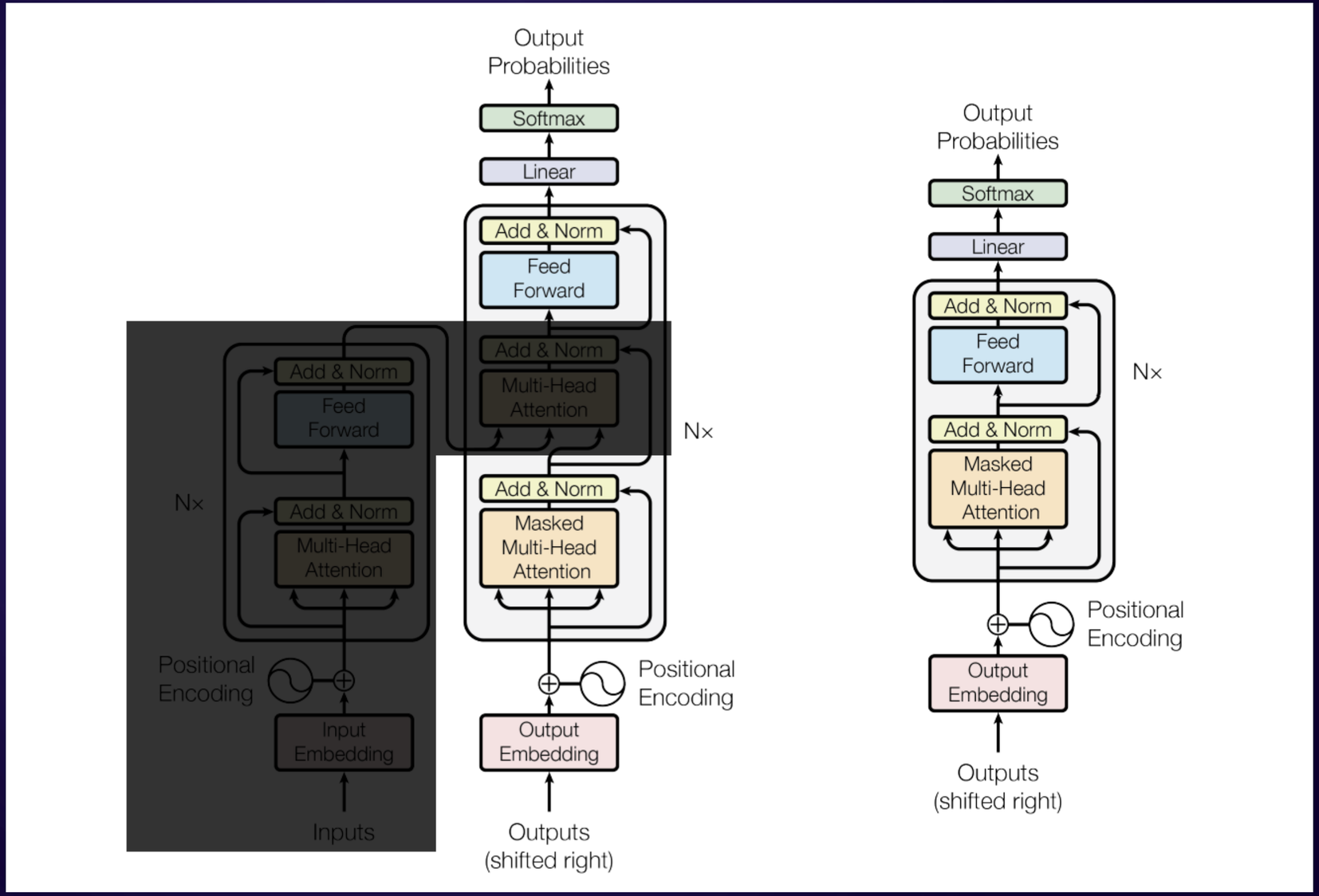
Исследование подходов оценки скорости современных серверов для использования больших языковых моделей и создание единого метода сравнения различных серверов для использования больших языковых моделей

Выполнил студент 3 курса БКНАД221 Кругликов Владислав Сергеевич
LLM Research Engineer @ T Bank

Decoder transformer



chatgpt, cursor, mistral, yandexgpt

LLM Inference

01

H100 costs 25K$
DeepSeek R1 in fp16 needs 2 nodes
each 8 GPUS = 400K$

02

В чатах поддержки / ассисентах
есть SLA на latency инференса

03

Для генерации синтетических данных в промышленных масштабах нужно максимизировать пропускную способность инференса

04

Агенты с большими системными промптами (cursor который парсит все файлы) транжирят флопсы карточек

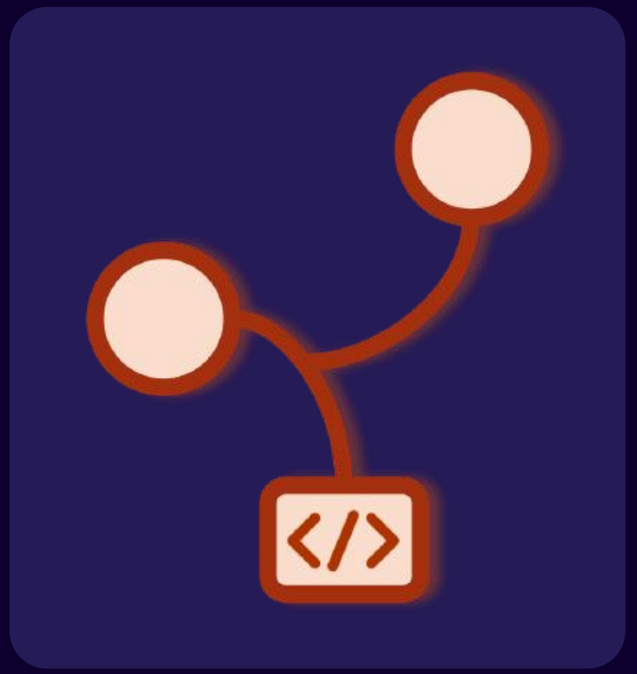
05

Недавно стали SOTA Reasoning модели которые потребляют больше токенов на декод этапе чтобы дать ответ, а значит by design более требовательны

06

Сейчас популярен RL при котором нужно генерировать траектории в промышленных масштабах за минимальное время чтобы не тормозить этап обучения

vLLM



SGLang

TGI by hugginface

Ответ хороший?

Ответ хороший?

Математика

Этика

Программирование

Маркетинг

FSL

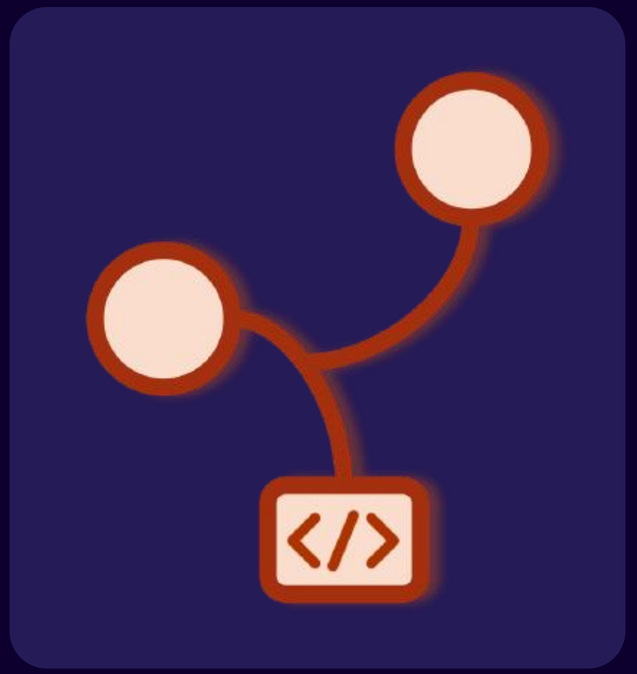
Безопас...

Перевод

Суммаризация

Следование инструкциям

vLLM



SGLang

TGI by hugginface

Быстро ответила модель?

Быстро ответила модель?

Распределение запросов

TPOT

Профиль нагрузки

Железо

TTFT

Common prefix

Агентские запросы

Кодовые запросы

В каком порядке запросы отправляются?

Влияние распределения запросов

Ограниченость одной метрики

Сравнительный анализ фреймворков инференса

Applications have Diverse SLO

TTFT

Time to first token

Initial response time

TPOT

Time per output token

Average time between two subsequent generated tokens

Chatbot

Fast initial response

Human reading speed (P99 latency = 250ms)

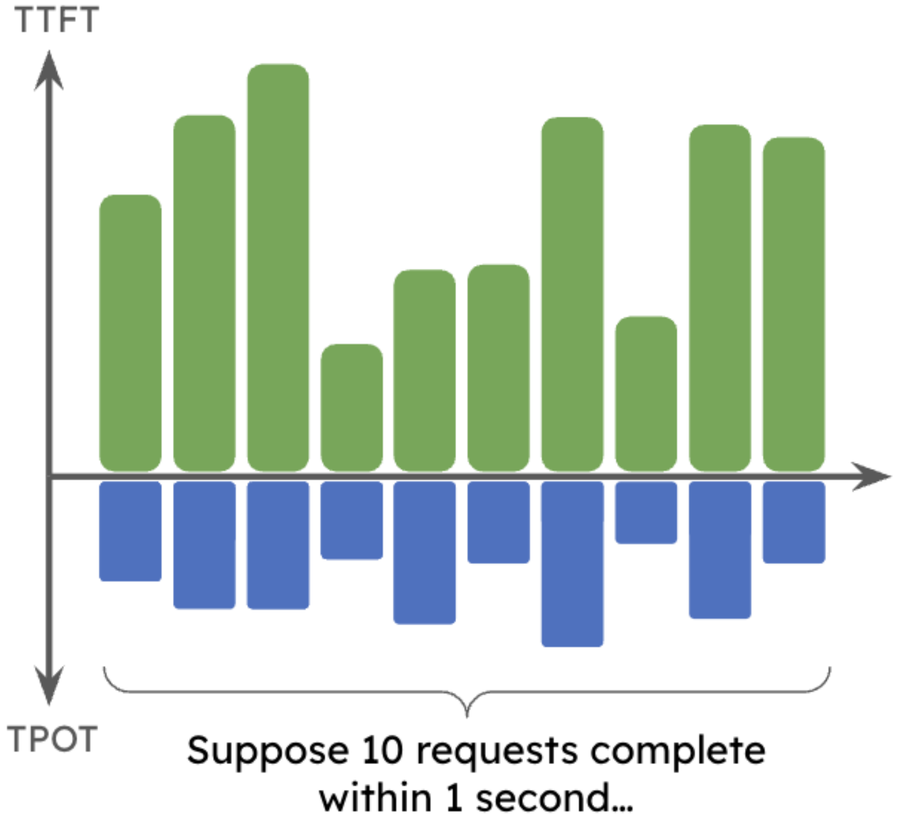
nmarization

User can tolerate longer initial response

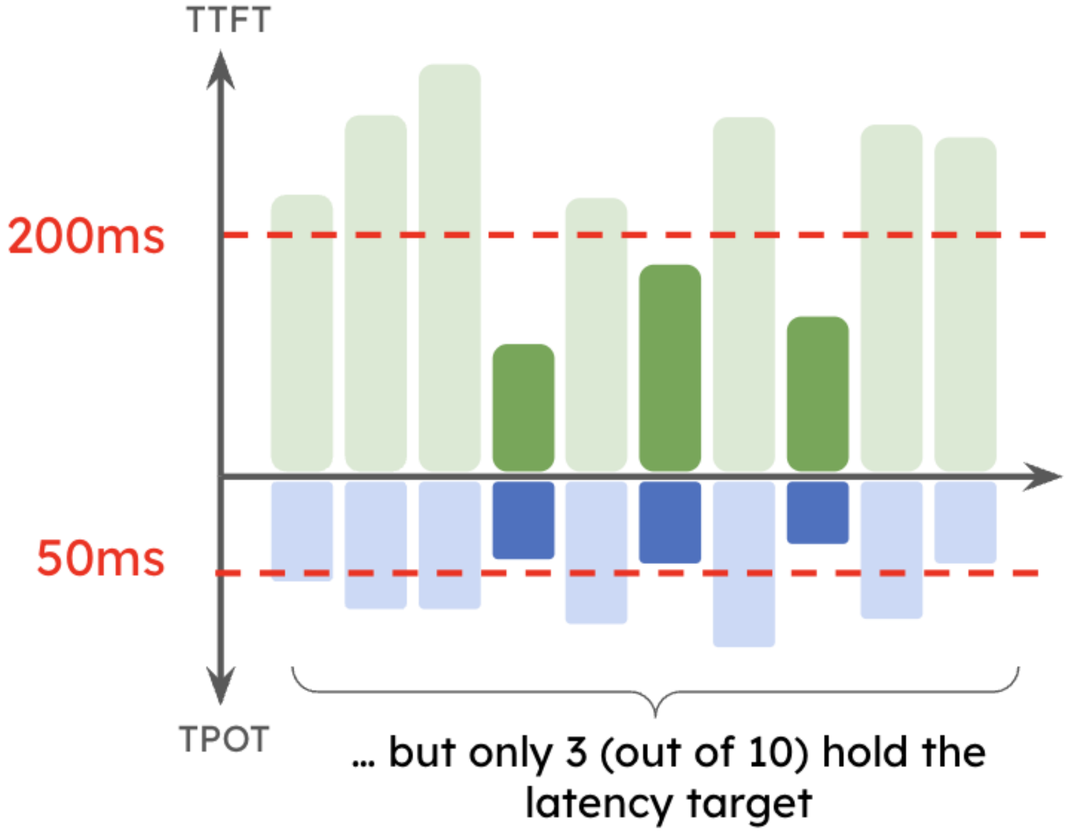
Data output generation (P99 latency = 35ms)

High Throughput ≠ High Goodput

Throughput = completed request / time
= **10** req / s



Goodput = completed requests **within SLO** / time
= **3** req / s



Датасеты

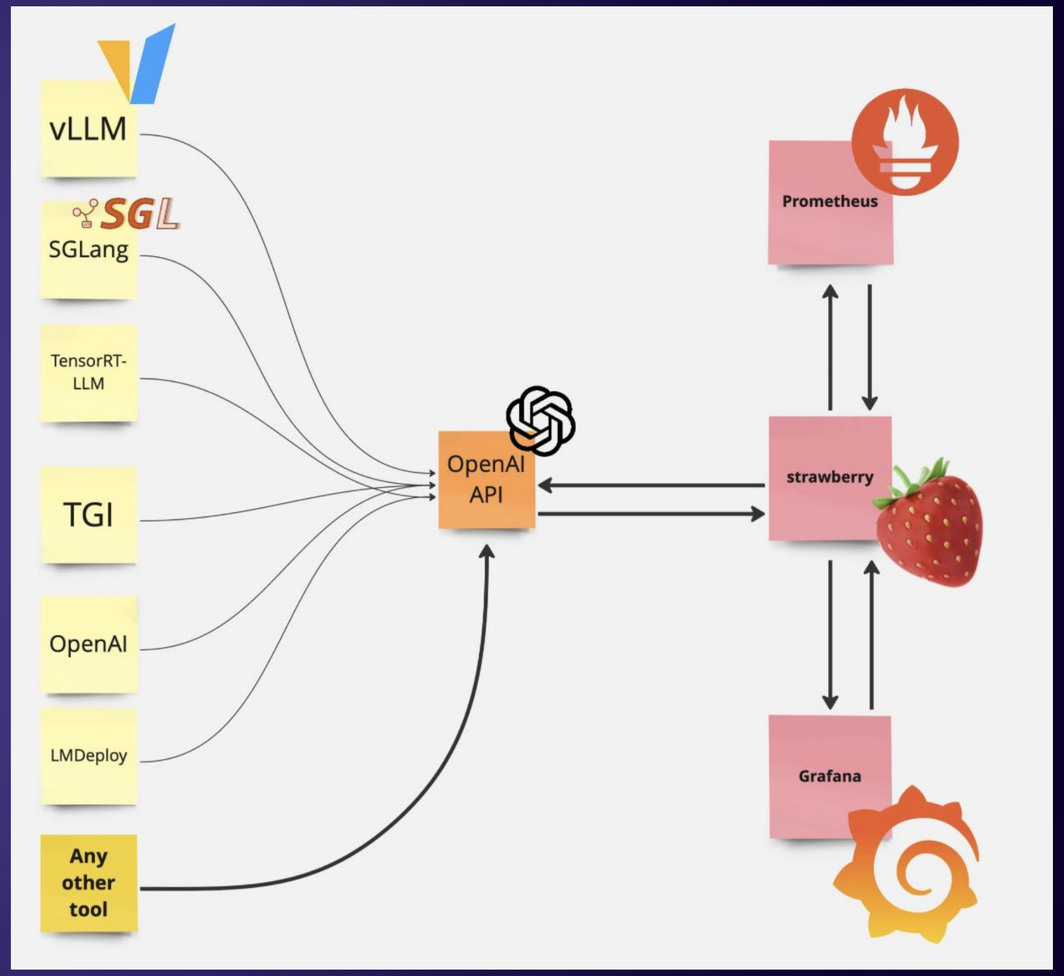
Агентский датасет

Случайный датасет

Профиль нагрузки

- Ступенчатый НТ, начиная с 1 синхронного пользователя и увеличивая каждые 10 секунд
- 10 минут, выбираем лучший запуск
- QWEN 7B, 1xH100

https://github.com/vladislavkruglikov/strawberry



Влияние распределения запросов

Sglang agent & sglang random

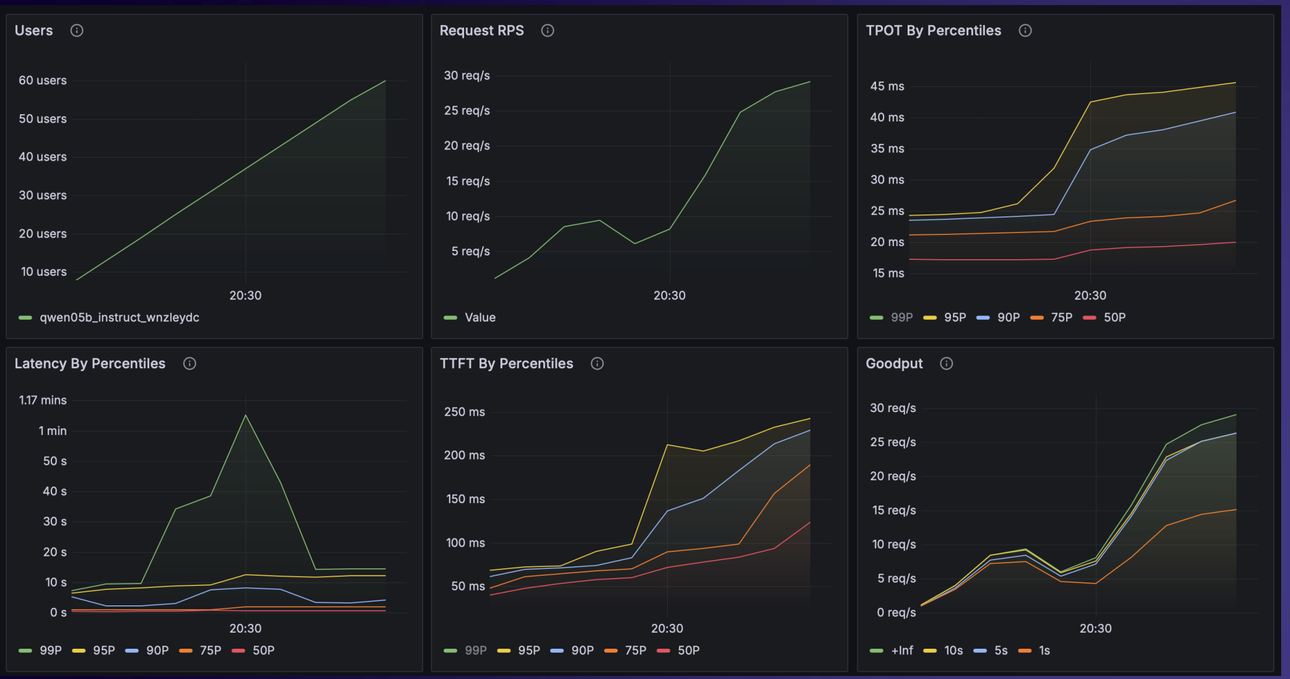


Таблица 4.1: Сравнение производительности SGLang на различных датасетах.

Конфигурация	Latency 95P (мс)	TPOT 95P (мс)	TTFT (мс)
SGLang, случайные промпты	26900	181	484
SGLang, агентский датасет	12300	45.6	243

Сравнительный анализ фреймворков

Add Vllm agent vs vllm random

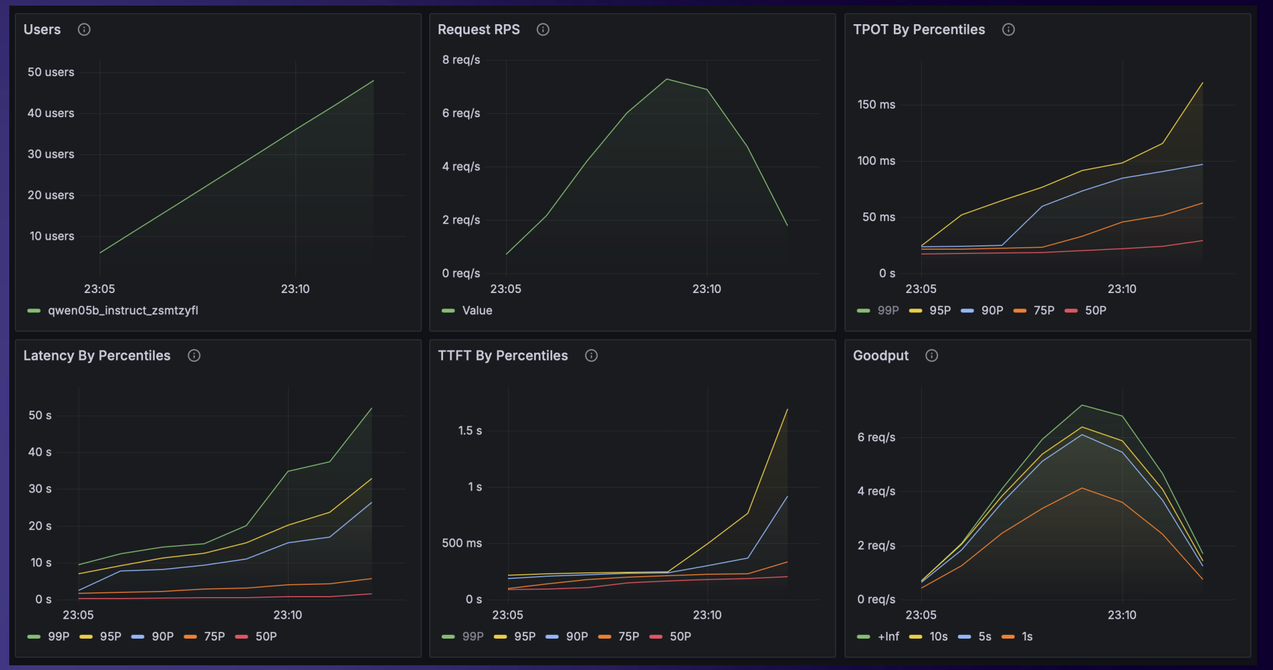


Таблица 4.2: Сравнение производительности SGLang и vLLM на различных датасетах.

Конфигурация	Latency 95P (мс)	TPOT 95P (мс)	TTFT (мс)
SGLang, случайные промпты	26900	181	484
vLLM, случайные промпты	32900	170	1690
SGLang, агентский датасет	12300	45.6	243
vLLM, агентский датасет	8990	39.7	131

Ограниченность одной метрики

Рассмотрим строки из Таблицы 4.2:

Таблица 4.3: Сравнение производительности на агентском датасете.

Конфигурация	Latency 95P (мс)	TPOT 95P (мс)	TTFT (мс)
SGLang, агентский датасет	12300	45.6	243
vLLM, агентский датасет	8990	39.7	131

Выводы

01

Эмпирически было показано, что недостаточно одной метрики для сравнения фреймоврков

02

Эмпирически было показано, что распределение запросов играет ключевую роль

03

Эмпирически было показано отсутствие единственного лучшего фреймворка инференса для всех задач

04

Выложена в опен сорс система для проведения бенчмаркинга

05

Создана первая версия бенчмарка с описанной методологией проведения и артефактами

Дальнейшая работа

01

Добавить больше метрик, cache hit rate, эффективность батчинга

02

Расширить датасеты

03

Больше фреймворков для инференса

04

Перебор параметров фреймворков

05

Единая система по аналогии chatbot arena

Спасибо за внимание!