X5 Tech

ВАШЕЙ КОМПАНИИ LLM НЕ НУЖЕН?

МИЧИЛ ЕГОРОВ

РУКОВОДИТЕЛЬ РАЗРАБОТКИ, X5 TECH

MICHIL.EGOROV@X5.RU

PiterPy 2024

о себе

ЕГОРОВ МИЧИЛ

TECH LEAD

программировал в яндексе, старлайне

руковожу в x5 tech

менторю в itmo

X5 Tech

airun: статистика

30+ человек
состоит в команде

4 чат-бота
на базе LLM

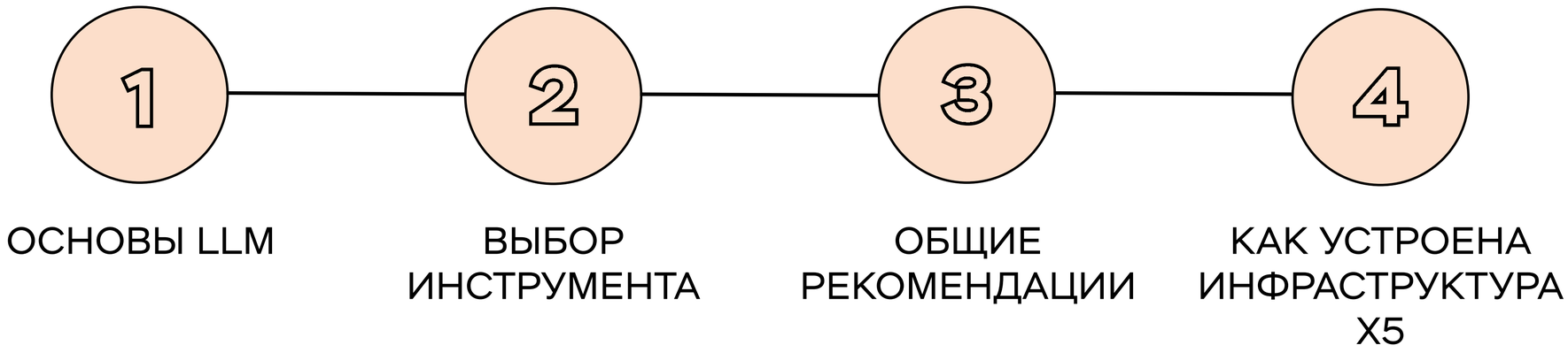
5+ сервисов
на базе LLM

500k+
запросов на генерацию

10+ LLM моделей
протестировано

X5 Tech

план презентации

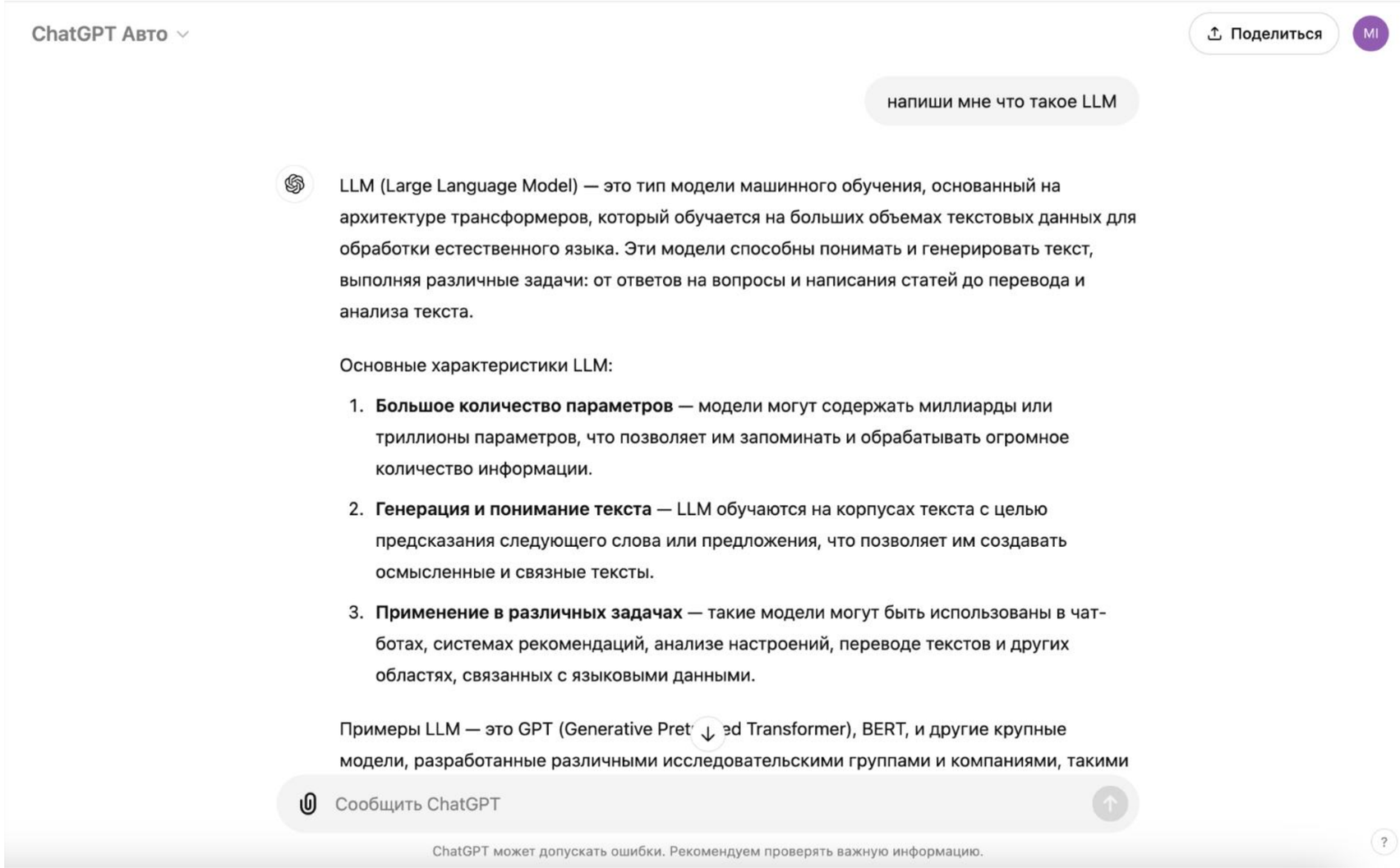


X5 Tech

1 ОСНОВЫ LLM

X5 Tech

что такое llm?



X5 Tech

что такое llm?

llm - это искусственная нейронная сеть, обученная на огромных текстовых данных

применение llm в x5

ПОИСК ПО БАЗАМ ЗНАНИЙ

ЧАТ-БОТЫ ДЛЯ ПОДДЕРЖКИ

ОБРАБОТКА ОТЗЫВОВ КЛИЕНТОВ

СУФЛЕРЫ СОТРУДНИКОВ ПОДДЕРЖКИ

X5 Tech

структура llm

веса модели

скрипт для
инференса

закрытые решения

какие есть риски?

- конфиденциальность данных
- зависимость от внешних систем
- затраты на долгосрочной перспективе
- ограничения на использование

X5 Tech

2

ВЫБОР ИНСТРУМЕНТА

X5 Tech

ключевые аспекты

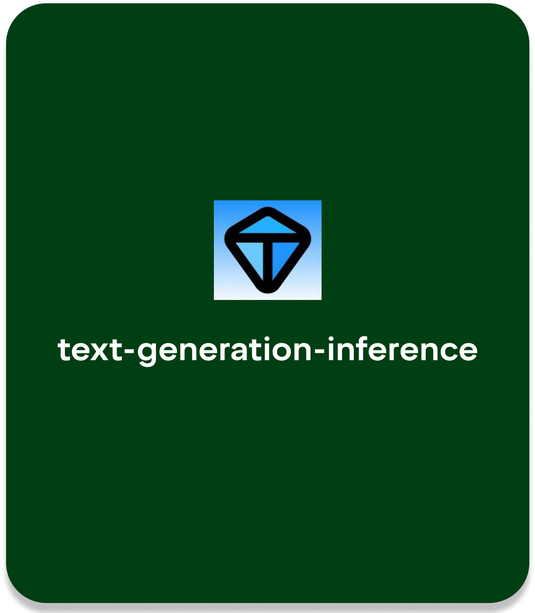
- latency (tps)
- пропускная способность
- качество ответов
- потребление ресурсов
  - cpu only
  - gpu ~~poor~~ regular
  - gpu rich

X5 Tech

постановка эксперимента

- модель: mistral-7b
- cpu only: 500ram 72 cpu
- gpu regular: rtx8000
- gpu rich: a100

бекенды инференса



LLAMA-CPP

обзор

легковесная c++ библиотека для инференса языковых моделей

- поддержка различных архитектур (x86, arm)
- возможно работать без gpu, только на cpu
- работает только с форматом gguf

применение

конвертация в gguf

```
python convert_hf_to_gguf.py \
    --model_path $model_path \
    --output_path path/to/output.gguf
```

запуск бекенда

```
docker run -p 8080:8080 \
    -v $model:/models \
    ghcr.io/ggerganov/llama.cpp:server
    -m models/model.gguf \
    --host 0.0.0.0 \
    --port 8080
```


Репозиторий llama-cpp: https://github.com/ggerganov/llama.cpp/blob/master/examples/server/README.md

дублирование

постоянное дублирование для длинных ответов

невозможно создать отпуск, что делать?
13:55

mixtral

Есть два пути создания "отпуска по заявлению" или "отпуска без сохранения з/п" при помощи ЛК2 или через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для TS5). Есть также возможность создания "отпуска" через "Заявки" в ЛК2 или через "Заявки" в ЛК1 (для

13:55

Оценить

10 llama_cpp/_internals.py

```diff
@@ -730,12 +730,14 @@ def sample(
         if len(self.prev) > 0:
             nl_token = ctx_main.model.token_nl()
             nl_logit = logits_array[nl_token]
-            if self.params.penalty_last_n > 0:
+            last_tokens = self.prev[-self.params.penalty_last_n:]
+            last_tokens_size = min(len(last_tokens), self.params.penalty_last_n)
+            if last_tokens_size > 0:
+                last_tokens_p = (llama_cpp.llama_token * len(last_tokens))(*last_tokens)
                 ctx_main.sample_repetition_penalties(
                     token_data_array,
-                    # TODO: Only create this once
-                    (llama_cpp.llama_token * len(self.prev))(*self.prev),
-                    self.params.penalty_last_n,
+                    last_tokens_p,
+                    last_tokens_size,
                     self.params.penalty_repeat,
                     self.params.penalty_freq,
                     self.params.penalty_present,
```


Исправление ошибки: https://github.com/abetlen/llama-cpp-python/pull/1295

сравнительный анализ

Технология	CPU only	GPU regular	GPU rich
llama-cpp	8-14 tps	45-51 tps	49-54 tps
tgi			
vllm			

apple silicon

Чип	CPU only
M1 ultra	37 tps
M2 ultra	41 tps

X5 Tech

TEXT GENERATION INFERENCE

X5 Tech

обзор

проект от hugging face

- оптимизирован для работы на gpu
- поддерживает распереределенный инференс
- поддерживает динамический батчинг

применение

запуск бекенда

```
docker run \
    --gpus all \
    -p 8080:80 \
    -v $volume:/data \
    ghcr.io/huggingface/text-generation-inference:2.3.0 \
    --model-id $model
```


Репозиторий tgi: https://github.com/huggingface/text-generation-inference

сравнительный анализ

Технология	CPU Only	GPU regular	GPU rich
llama-cpp	8-14 tps	45-51 tps	49-54 tps
tgi	N/A	525-545 tps	1700-1850 tps
vllm			

X5 Tech

VLLM

X5 Tech

обзор

проект от berkeley uc

- использует pagedattention
- оптимизирован для мощных gpu
- поддерживает распределенный инференс
- поддерживает динамический батчинг

применение

запуск бекенда

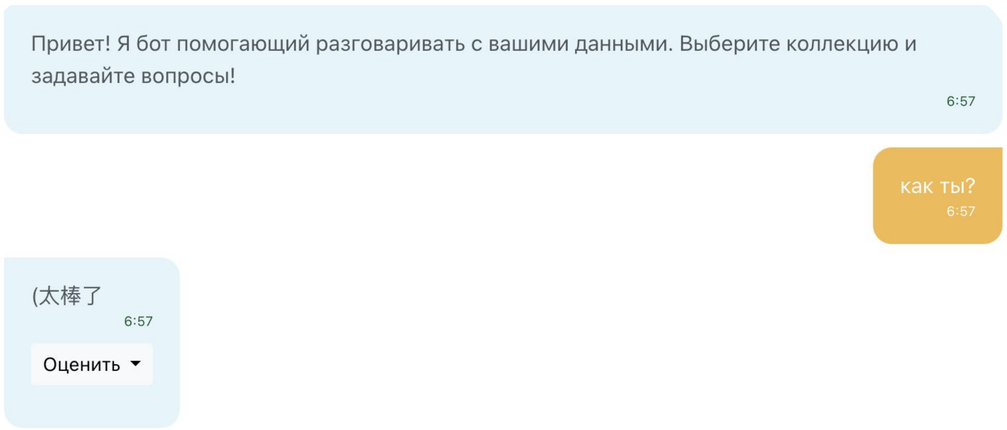
```
pip install vllm
```

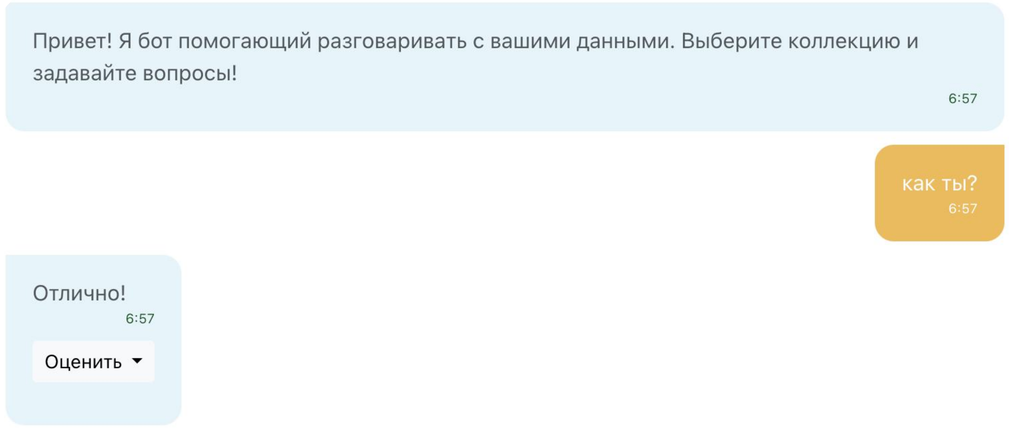
```
vllm serve $model
```


Репозиторий vllm: https://github.com/vllm-project/vllm

guided_regex

цель: избавиться от лишних символов





результат: время генерации x2

Issue в репозитории: https://github.com/vllm-project/vllm/issues/3567

сравнительный анализ

Технология	CPU	GPU regular	GRU rich
llama-cpp	8-14 tps	45-51 tps	49-54 tps
tgi	N/A	525-545 tps	1700-1850 tps
vllm	N/A	670-768 tps	1800-2000 tps

X5 Tech

3

ОБЩИЕ РЕКОМЕНДАЦИИ

X5 Tech

рекомендации

прототипирование и не важна скорость работы - llamacpp

нагрузка <40 rps и нужны качественные и быстрые ответы - tgi

высокая нагрузка, быстрые ответы и есть мешок денег - vllm

что стоит учитывать

- генерация зависит от промптов
- необходимо подбирать параметры генерации
- галлюцинации - всегда будут

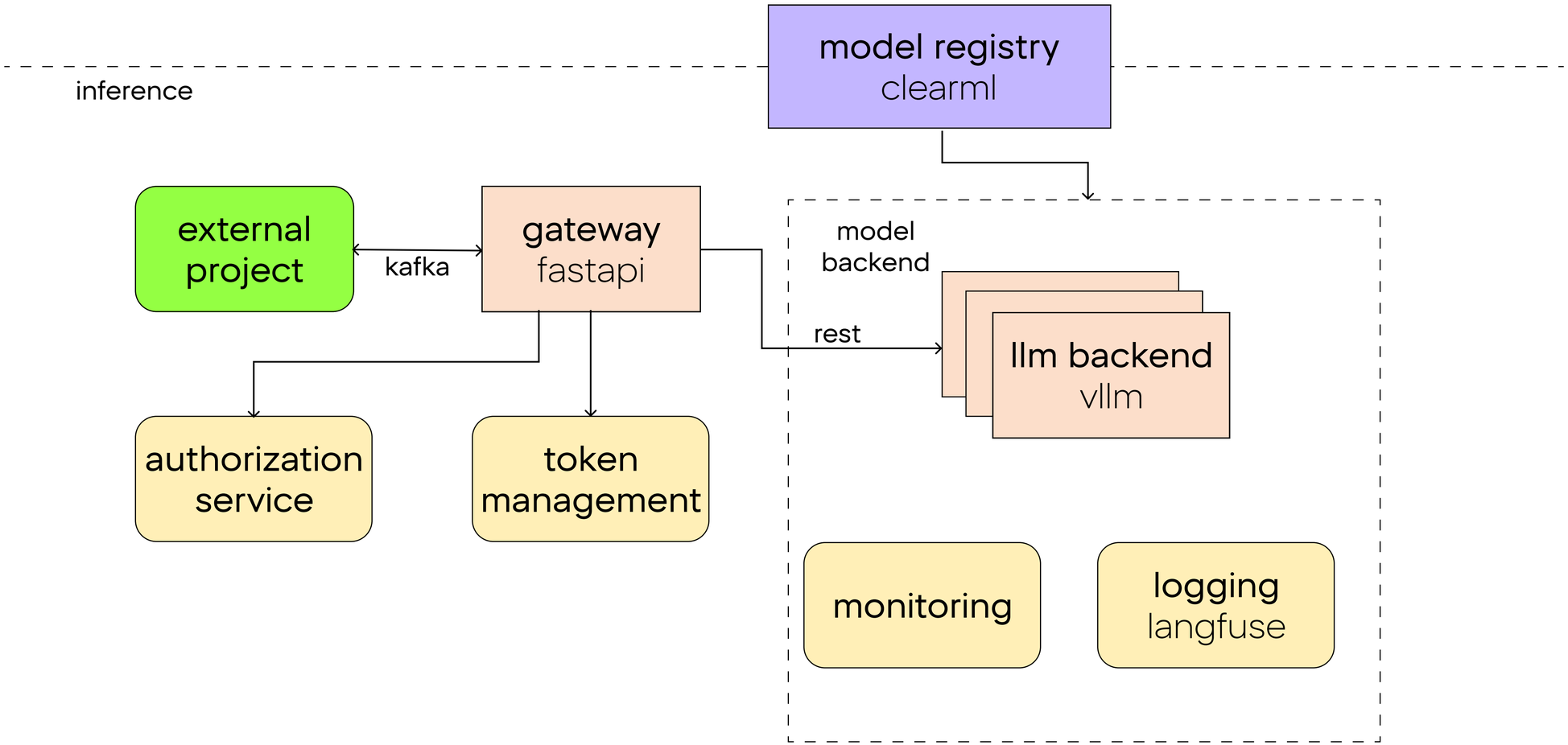
X5 Tech

4

КАК БЕКЕНД УСТРОЕН В X5

X5 Tech

Архитектура



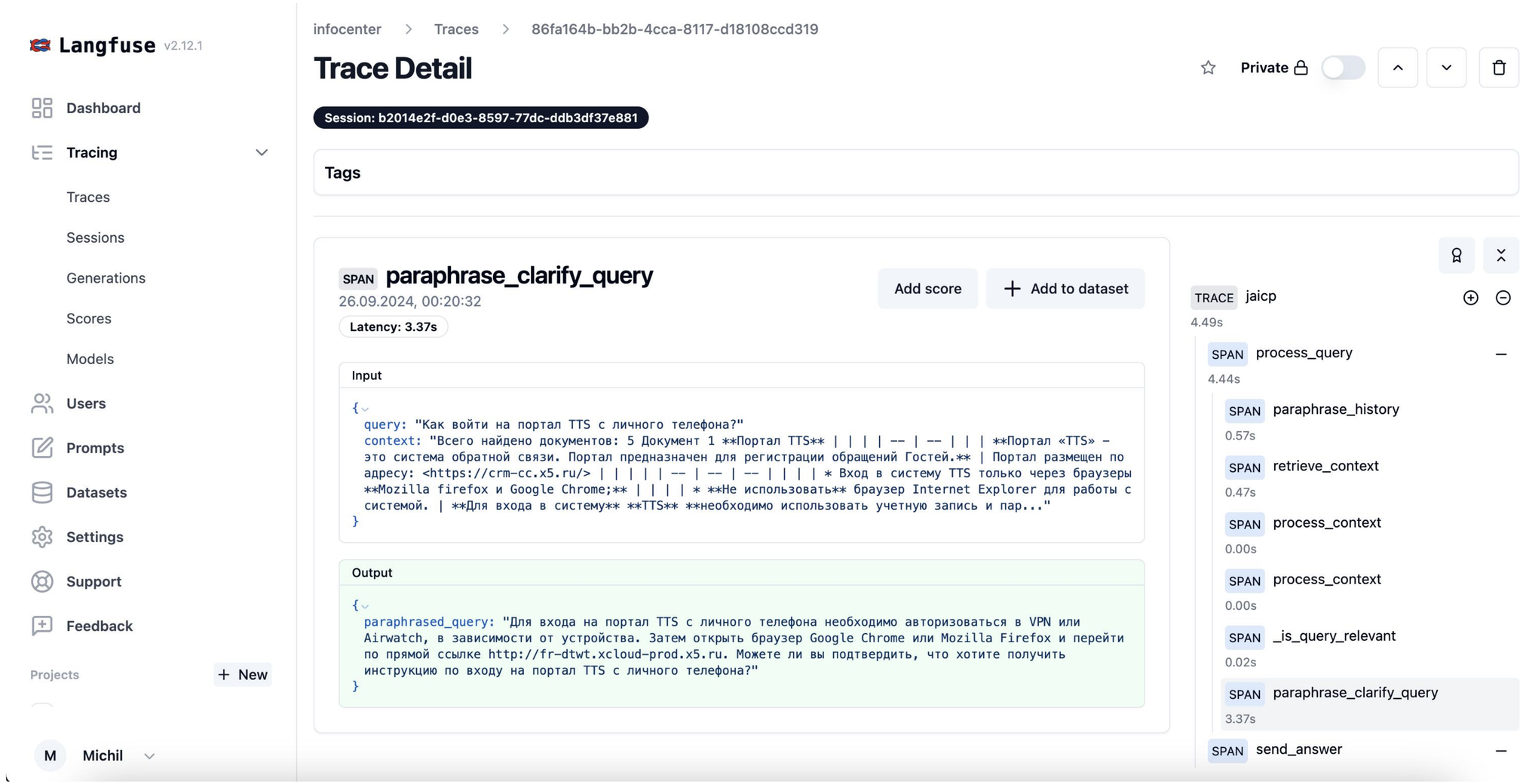
X5 Tech

langfuse

- распределенный трейсинг
- версионирование промптов
- а/б тесты для промптов
- хранение датасетов

X5 Tech

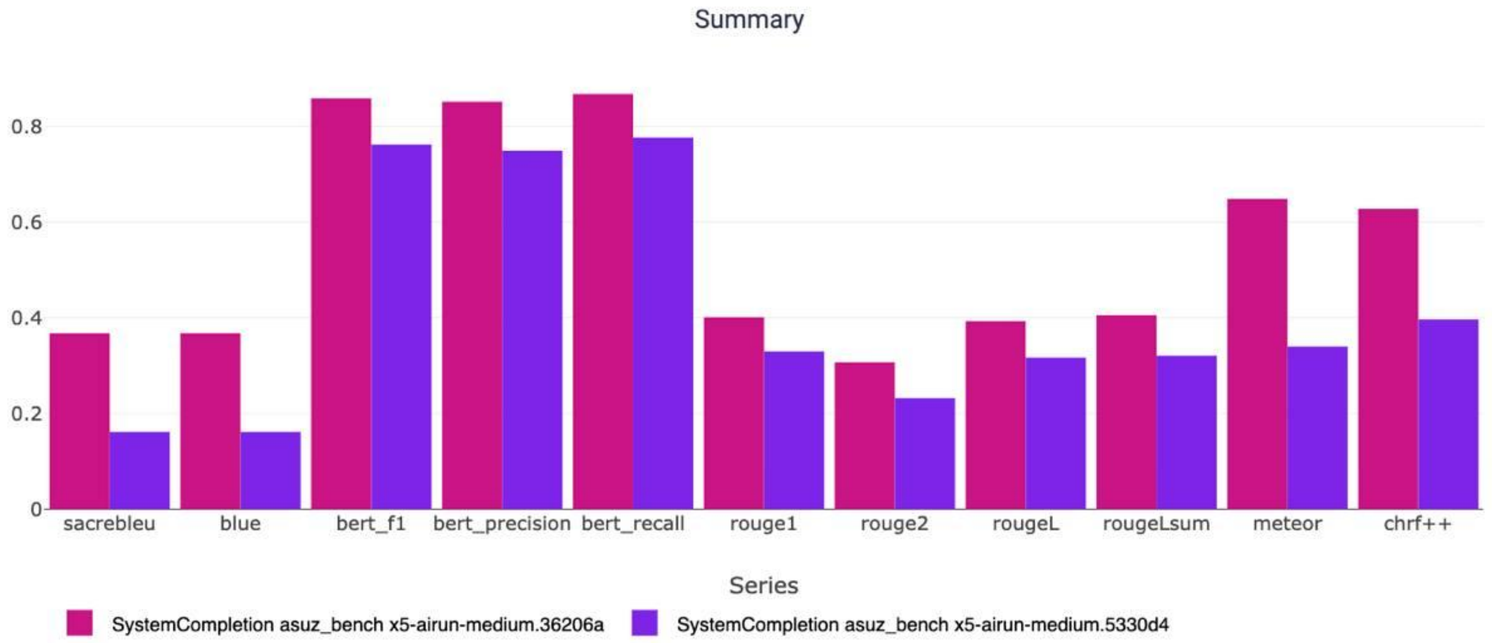
langfuse



X5 Tech

как мы выбираем модель

- следим за обновлениями
- прогоняем на бенчмарках
- катим в дев
- катим в прод



X5 Tech

какие модели мы тестили

- mistral-7b-saiga, mixtral-8x7b
- llama-2-13b-code, llama-2-70b-instruct
- suzume-llama-3, llama-3-saiga, llama-3-instruct
- c4ai-command-r-35b, c4ai-command-r-35b-0824
- huyalm-100b

почему несколько?

- для решения разных задач на специализированных моделях
- возможность комбинировать ответы от разных моделей
- маленькие модели - быстрая скорость ответа

ИТОГИ

X5 Tech

ИТОГИ

- вкладывайтесь в инфраструктуру для минимизации утечки данных
- выбирайте бекенд для инференса в зависимости от задачи
- llm можно использовать без gpu
- используйте специализированные сервисы для мониторинга llm

X5 Tech

ВАШИ ВОПРОСЫ

МИЧИЛ ЕГОРОВ
РУКОВОДИТЕЛЬ РАЗРАБОТКИ, X5 TECH

ДЛЯ СВЯЗИ
MICHIL.EGOROV@X5.RU
@LOALKOTA

X5 Tech