Почему вам (скорее всего) не нужен локальный LLM-инференс

Егор Андреев
Впрод / AdminDivision

Saint HighLoad++ 2026 / SmallTech

О чем будем говорить

- Идея: поднимем локальный инференс и сэкономим
- Из чего складывается TCO
- Сколько стоит миллион токенов
- Что ломает экономику
- Когда локальный инференс все-таки нужен
- Как принять решение

Давайте познакомимся

Егор Андреев

CEO AdminDivision и Впрод

- 20 лет в IT и эксплуатации
- AdminDivision: услуги DevOps, high-load, Kubernetes
- Впрод: вендор платформенного ПО
- Пилим платформу LLM-инференса для провайдеров и Enterprise
- Знаю, кому инференс нужен
- Давайте поговорим о том, кому он не нужен

Актуально для SmallTech-компаний

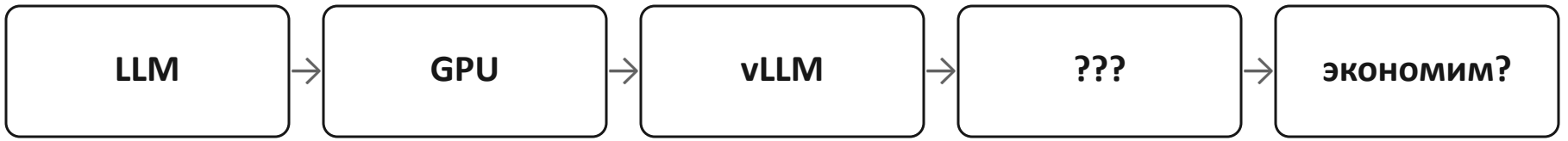
TG: @egorandreev

Канал о музыке и айтишке:

@GrooveOps

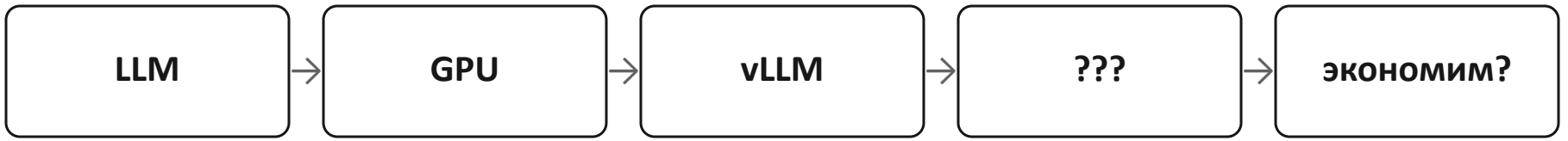
Продукту нужны LLM. Давайте сэкономим?

- LLM нужны продукту



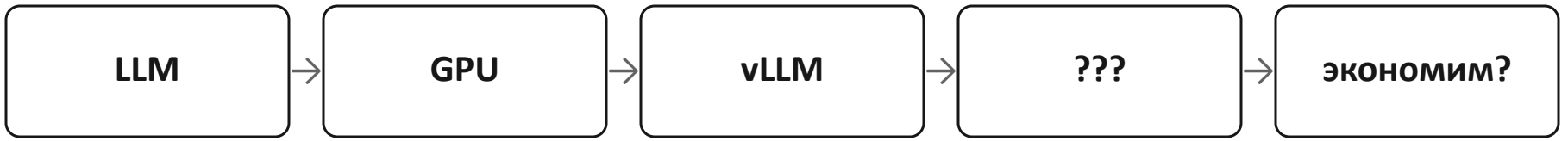
Продукту нужны LLM. Давайте сэкономим?

- LLM нужны продукту
- Покупаем GPU



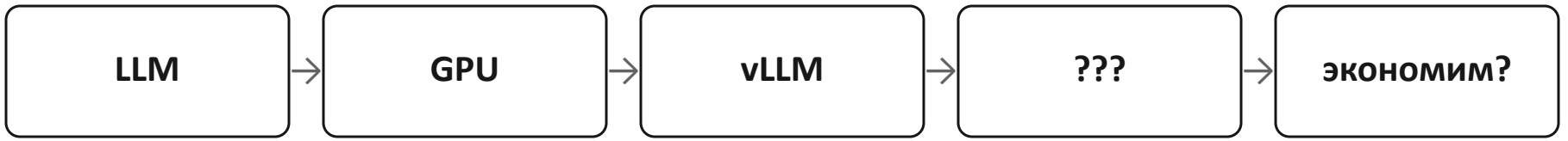
Продукту нужны LLM. Давайте сэкономим?

- LLM нужны продукту
- Покупаем GPU
- Ставим vLLM



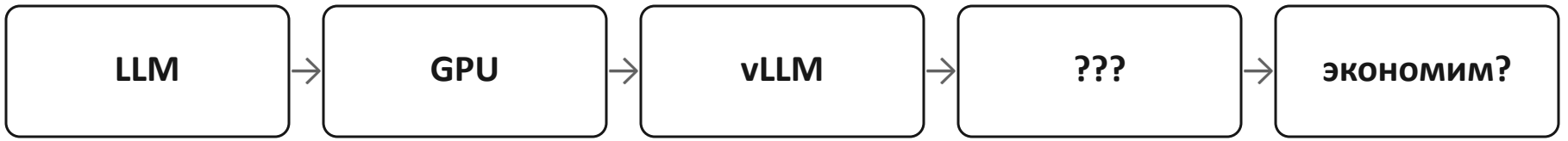
Продукту нужны LLM. Давайте сэкономим?

- LLM нужны продукту
- Покупаем GPU
- Ставим vLLM
- ???



Продукту нужны LLM. Давайте сэкономим?

- LLM нужны продукту
- Покупаем GPU
- Ставим vLLM
- ???
- Экономим?



Считать надо TCO

Железо - только первая строка бюджета

Площадка и дополнительные расходы

Запуск

разобраться, нанять, отладить

Эксплуатация

мониторинг, деплой, поддержка

TCO = железо + площадка + запуск + эксплуатация

Железо: провайдер покупает иначе

Провайдер

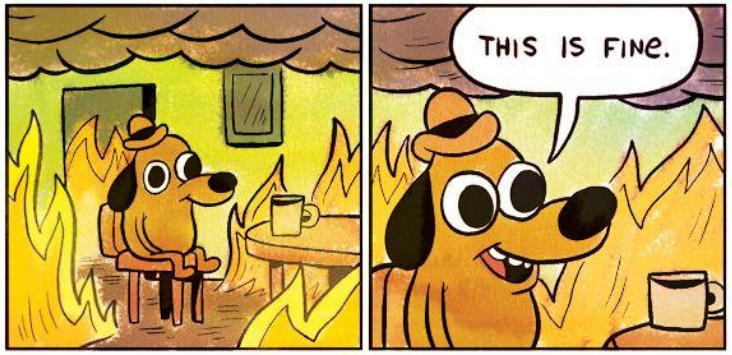
- Провайдер покупает партиями, а не одну карту в магазин
- Поставщик дает другие цены, сроки и условия
- Серверная платформа дешевле на единицу GPU

SmallTech

- **SmallTech покупает 1-2 узла без объема**
- Переговорной силы почти нет

Железо: GPU нужно куда-то поставить

- Сервер, стойка, питание, сеть
- Тепло и охлаждение
- Гарантия, запасные части, удаленные руки



Запуск: первые модели

- Модель, квантизацию, движок и GPU приходится подбирать вместе
- 50 tok/s иногда норма, иногда след плохой настройки
- Рецепт придется подбирать заново

Запуск: движемся к платформе

Демо

- Модель отвечает через API

Сервис

- Ключи, лимиты, учет, мониторинг, деплой
- Нужны переносимость, обновления и rollback
- Иначе это эксперимент, прибитый к одному GPU-хосту

Эксплуатация: что может пойти не так

- Модели нужно обновлять и переносить между GPU-хостами
- Нужны мониторинг, алерты, логи и capacity planning
- Инциденты приходят ночью, а пользователи ждут сервис
- 0.3-1 DevOps FTE после запуска - нормальная оценка

Сводим TCO в R

~400К ₽/GPU-мес

до резерва, SLA и ИБ

- 70К ₽: GPU

Сводим TCO в R

~400К ₽/GPU-мес

до резерва, SLA и ИБ

- 70К ₽: GPU
- 40К ₽: площадка, сеть, мониторинг

Сводим TCO в R

~400К ₽/GPU-мес

до резерва, SLA и ИБ

- 70К ₽: GPU
- 40К ₽: площадка, сеть, мониторинг
- 135К ₽: 0.3 DevOps/ML FTE

Сводим TCO в R

~400К ₽/GPU-мес

до резерва, SLA и ИБ

- 70К ₽: GPU
- 40К ₽: площадка, сеть, мониторинг
- 135К ₽: 0.3 DevOps/ML FTE
- 150К ₽: запуск, размазанный на 6 месяцев

Сводим TCO в R

~400К ₽/GPU-мес

до резерва, SLA и ИБ

- 70К ₽: GPU
- 40К ₽: площадка, сеть, мониторинг
- 135К ₽: 0.3 DevOps/ML FTE
- 150К ₽: запуск, размазанный на 6 месяцев
- Итого: ~400К ₽/GPU-мес до резерва, SLA и ИБ

Сколько стоит миллион токенов

$$P = 1M * R / (Q * 60 * 60 * 24 * 30 * U)$$

60 * 60 * 24 * 30 = секунд в месяце

- R - месячный TCO GPU-сервиса, уже посчитали

Сколько стоит миллион токенов

$$P = 1M * R / (Q * 60 * 60 * 24 * 30 * U)$$

60 * 60 * 24 * 30 = секунд в месяце

- R - месячный TCO GPU-сервиса, уже посчитали
- Q - throughput в output-токенах в секунду

Сколько стоит миллион токенов

$$P = 1M * R / (Q * 60 * 60 * 24 * 30 * U)$$

60 * 60 * 24 * 30 = секунд в месяце

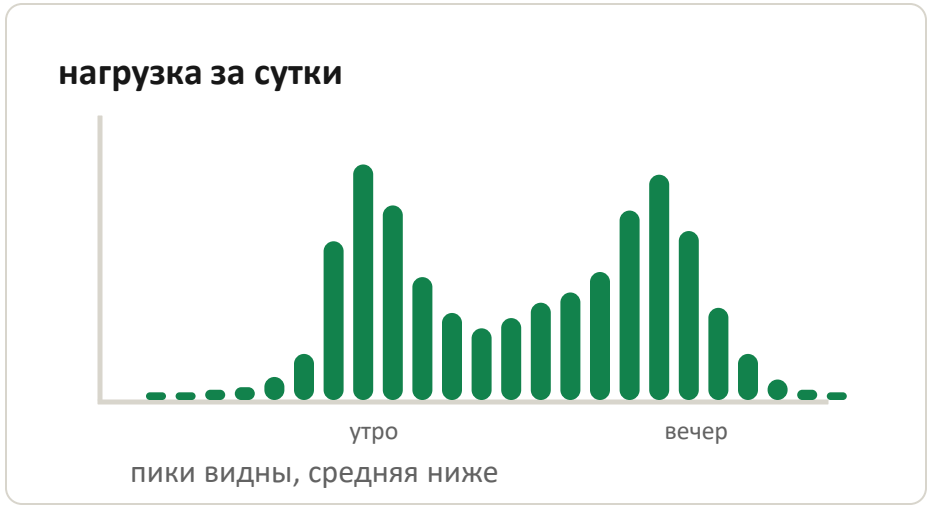
- R - месячный TCO GPU-сервиса, уже посчитали
- Q - throughput в output-токенах в секунду
- U - реальная утилизация

Что делает Q

- Высокий Q надо получить настройкой
- Модель, квантизация, контекст и batch меняют throughput
- Плохой рецепт делает токен дорогим еще до TCO

U: нагрузка неровная

- Утром всплеск
- Днем ниже пика
- Вечером второй всплеск
- Ночью GPU простаивает



Что делает U

70%	30%	15%	10%
194 ₽	453 ₽	905 ₽	1 358 ₽
почти провайдер	осторожно	часто честнее	почти всегда плохо

qwen3-14b, R = 400К ₽/GPU-мес

- 70% загрузки - это почти провайдерская дисциплина

Что делает U

70%	30%	15%	10%
194 ₽	453 ₽	905 ₽	1 358 ₽
почти провайдер	осторожно	часто честнее	почти всегда плохо

qwen3-14b, R = 400К ₽/GPU-мес

- 70% загрузки - это почти провайдерская дисциплина
- 30% - уже осторожнее, но все еще неплохо

Что делает U

70%	30%	15%	10%
194 ₽	453 ₽	905 ₽	1 358 ₽
почти провайдер	осторожно	часто честнее	почти всегда плохо

qwen3-14b, R = 400К ₽/GPU-мес

- 70% загрузки - это почти провайдерская дисциплина
- 30% - уже осторожнее, но все еще неплохо
- 15% для одной компании часто честнее

Что делает U

70%	30%	15%	10%
194 ₽	453 ₽	905 ₽	1 358 ₽
почти провайдер	осторожно	часто честнее	почти всегда плохо

qwen3-14b, R = 400К ₽/GPU-мес

- 70% загрузки - это почти провайдерская дисциплина
- 30% - уже осторожнее, но все еще неплохо
- 15% для одной компании часто честнее
- 10% превращает локальный GPU в дорогую игрушку

В продукте обычно не одна модель

- Маленькая модель: классификация, извлечение, простые ответы

В продукте обычно не одна модель

- Маленькая модель: классификация, извлечение, простые ответы
- Средняя модель: основная продуктовая работа

В продукте обычно не одна модель

- Маленькая модель: классификация, извлечение, простые ответы
- Средняя модель: основная продуктовая работа
- Большая модель: сложный reasoning, код, агенты

В продукте обычно не одна модель

- Маленькая модель: классификация, извлечение, простые ответы
- Средняя модель: основная продуктовая работа
- Большая модель: сложный reasoning, код, агенты
- Три профиля нагрузки - это уже не один дешевый GPU

Под каждую модель нужно свое железо

Маленькая

один GPU и простые задачи

Средняя

больше памяти, другой runtime

Большая

несколько GPU и быстрые связки

Парк моделей дробит бюджет и утилизацию

Парк моделей дробит бюджет, память и утилизацию

Frontier-модели - другая лига

Железо

топовые GPU и дорогая память

Связка

NVLink, multi-GPU, иногда кластер

Платформа

зрелый runtime, запуск, мониторинг

Вывод

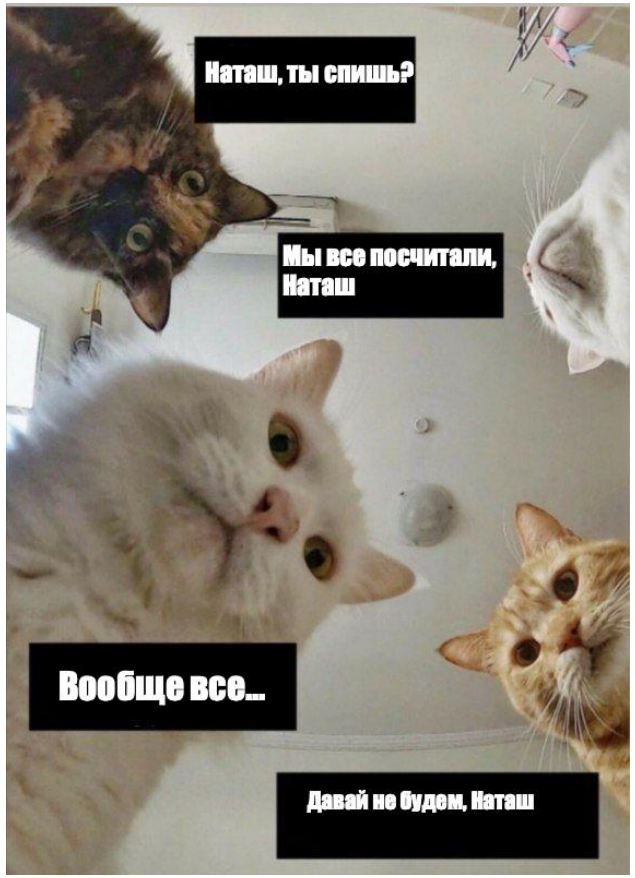
берём снаружи или живём без этого качества

ChatGPT-уровень локально недоступен для обычного GPU-проекта

Пример экономики для разных моделей

Модель	Провайдер @70%	SmallTech @15%	Рынок
qwen3-4b	17 ₽	389 ₽	100 ₽
qwen3-14b	39 ₽	905 ₽	200 ₽
gpt-oss-120b	174 ₽	3 992 ₽	300 ₽

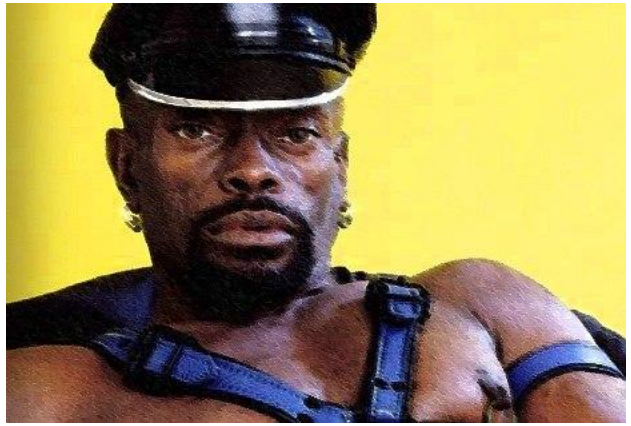
руб. за 1M output-токенов; SmallTech = R 400К ₽/GPU-мес и U=15%



Исключения

Исключение 1. Вас заставили

Данные или договоры запрещают внешний AI



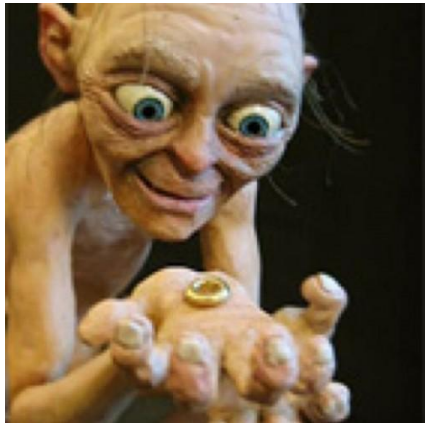
«Рекомендую воздержаться от трансграничной передачи перс-данных».

Даже когда заставили, не стройте вслепую

- Все равно посчитайте TCO
- Сравните самострой с готовой платформой
- Пилот без чувствительных данных - на managed on-prem

Исключение 2. GPU уже лежат

- GPU-парк от прошлого проекта уже на складе
- Утилизировать его разумно
- Купленное железо не делает сервис бесплатным

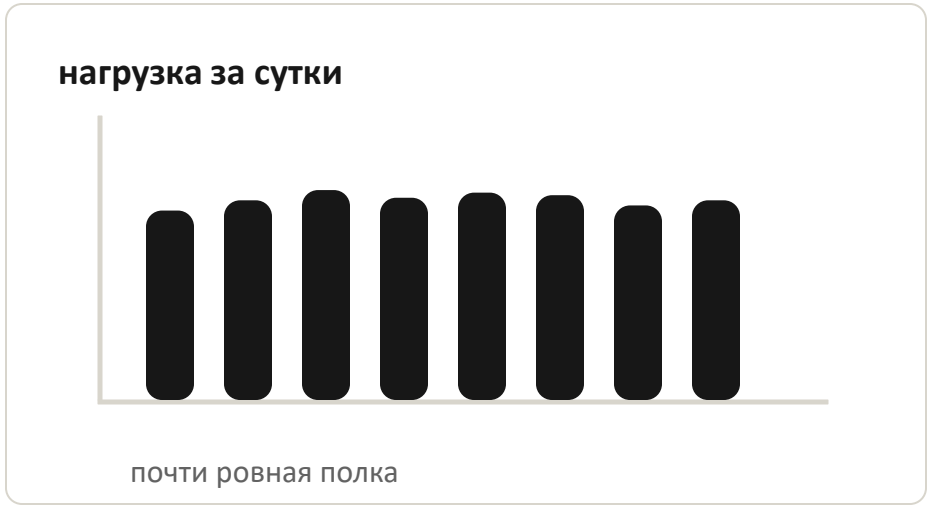


«A100 — моя прелесть».

Исключение 3. Нагрузка подозрительно постоянная

- Training, batch, стабильный поток запросов
- GPU занят почти круглосуточно

- Нужен график нагрузки, надежды мало



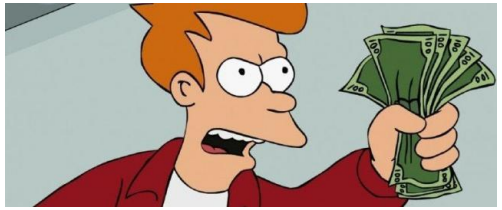
Исключение 4. CAPEX проще OPEX

Техническая экономика

- Иногда железо покупают не потому, что дешевле

Бюджетный процесс

- CAPEX можно защитить один раз
- OPEX на токены приходится защищать каждый год



«Все нормально, деньги есть».

Что делать вместо своего инференса

- Пилот без жестких ограничений: API / OpenRouter / frontier-модели
- Российский контур: сервисы с оплатой по токенам
- Временная потребность: аренда GPU под тест или batch
- Последний шаг: своя платформа или on-prem-вендор

Перед покупкой GPU

- Какие данные или договоры ограничивают AI
- Какие модели нужны
- Что нужно кроме карт (R)
- Какой честный U подставляем в формулу

Спасибо! Вопросы?

1. **Экономику хоронит полный TCO: люди, эксплуатация, площадка, качество модели и реальная утилизация**

2. **Локальный инференс применим при жестких ограничениях, уже купленном GPU-парке, постоянной нагрузке или CAPEX-логике**

3. **Практический шаг: проверить качество и объем на API / провайдере / аренде, посчитать TCO и только потом идти в свое железо**

Оцените мой доклад

QR для голосования

Егор Андреев, CEO AdminDivision и Впрод.
Телеграмм: @egorandreev
С юморочком о музыке и айтишке: @GrooveOps