Оптимизация затрат video AI — чек-лист из 25 рычагов

Одна идея — рычаги перемножаются, а не складываются

```
Три независимых выигрыша «в 3 раза» дают 27x, а не 9x. Тяните КАЖДЫЙ применимый рычаг.
Два рычага перемножаются, только если действуют на разные части стоимости (число токенов vs цена).
```

25 рычагов в пяти семействах

1-5 ОБРАБАТЫВАТЬ МЕНЬШЕ ВИДЕО (крупнейший выигрыш для видео)

1 Выборка кадров - не все 30 в секунду (1 fps = в 30 раз меньше)
2 Понизить разрешение до того, как модель увидит кадр
3 Обрезать до области интереса дешёвым детектором
4 Дорогая модель за дешёвым триггером (по событию, не расписанию)
5 Урезать системный промпт; ограничить длину вывода (выход дороже всего)

6-10 ЗАПУСКАТЬ МОДЕЛЬ ПОМЕНЬШЕ

6 Роутинг лёгких 90% на дешёвый класс (флагман в 8-30x дороже)
7 Каскад: дешёвая модель первой, эскалация при неуверенности
8 Маленькая открытая модель там, где позволяет качество
9 Дистилляция меньшего ученика (напр. Distil-Whisper, ~6x быстрее)
10 Квантизация в INT8 / INT4 (~4x меньше памяти, обычно <1% точности)

11-15 ОБСЛУЖИВАТЬ МОДЕЛЬ ЭФФЕКТИВНЕЕ (self-host)

11 Непрерывный батчинг - держать GPU заполненным (до ~23x throughput)
12 Переиспользовать KV-кэш (PagedAttention)
13 Спекулятивное декодирование - 2-3x быстрее, вывод идентичен
14 Специализированный рантайм (vLLM / TensorRT-LLM / SGLang)
15 Подобрать GPU под объём памяти модели

16-20 ПЛАТИТЬ МЕНЬШЕ ЗА ЕДИНИЦУ (конфиг, а не код)

16 Batch API для не-реального-времени - фикс. минус 50%
17 Кэш промпта - до -90% на повторяемом входном префиксе
18 Семантический кэш ответов + дедуп перезагруженных клипов
19 Spot / preemptible GPU - -60-90% (только прерываемая работа)
20 Резерв / committed мощности - -20-70% под устойчивый базис

21-25 ПРОЕКТИРОВАТЬ И УПРАВЛЯТЬ ПОД СТОИМОСТЬ

21 Serverless scale-to-zero - посекундно, простой ничего не стоит
22 Правильный провайдер (neocloud); борьба с egress/хранением
23 Гибрид edge + облако - постоянная работа на устройстве
24 Предвычислить артефакты раз (транскрипт, эмбеддинги, индекс)
25 Бюджеты затрат по каждой функции; алерт на дрейф

Разбор примера - нагрузка модерации 50 000 часов/мес

```
Наивно: 1,08M ток/час x $2,50/1M = $2,70/час x 50 000 = $135 000 / мес
+ меньше видео (низкое разр. + выборка, ~1/3 токенов)  -> $45 000   (/3)
+ роутинг лёгких 90% на Flash (смесь $0,52/1M)         -> $9 360    (/4.8)
+ офлайн-задание через batch API (-50%)                -> $4 680    (/2)
Три независимых выигрыша «в 3 раза» дают 27x, а не 9x. Тяните КАЖДЫЙ применимый рычаг.
```

Перед оптимизацией спросите:

- [ ] Поставлены ли приборы на стоимость по функциям ДО оптимизации - знаем ли, куда уходят деньги?
- [ ] Обрабатываем ли меньше видео сначала (выборка кадров, низкое разрешение, обрезка, триггеры)?
- [ ] Рутина на дешёвом классе модели, флагман оставлен по-настоящему сложному меньшинству?
- [ ] Каждое не-реального-времени задание на batch API, общие промпты в кэше (до -90%)?
- [ ] Тарифный режим соответствует нагрузке (spot = прерываемая, резерв = базис, serverless = всплески)?
- [ ] Для self-host: эффективный рантайм, непрерывный батчинг и подобранный по размеру GPU?
- [ ] Сложенные рычаги НЕЗАВИСИМЫ, и сверили ли мы итог с реальным измерением?