УДК 004.8:004.31

Бражников Андрей Романович, магистр, кафедра прикладной математики и информационных технологий, ФГБОУ ВО «Донецкий государственный университет», Россия, г. Донецк

ЗАКОН КВАНТИЗАЦИИ: ЭФФЕКТИВНАЯ БИТНОСТЬ ПРЕДСТАВЛЕНИЯ ВЕСОВ КАК КООРДИНАТА ПРОСТРАНСТВА СОСТОЯНИЙ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Аннотация

Родительская работа описывает доступность self-hosted-развёртывания больших языковых моделей через бюджет видеопамяти $M = Nb/8$, однако эта модель занижает требование: она не учитывает KV-кэш, который при длинном контексте (32-128 тыс. токенов) сопоставим с весами в INT4 или превосходит их. Поэтому доступность описывается не парой (N, b), а тройкой (N, b, L), и вводится обобщённая координата эффективной битности $b_{eff}(L)$, совпадающая с номиналом лишь при коротком контексте. Квантизация 16→4 бита эквивалентна двум удвоениям закона Мура, пройденным быстрее (~2 года против ~3), но, в отличие от него, «закон квантизации» исчерпаем: ниже ~2-3 бит на вес качество обваливается.

Abstract

The parent work describes the availability of self-hosted large-language-model deployment through the VRAM budget $M = Nb/8$, but this model understates the requirement: it omits the KV cache, which at long context (32-128k tokens) is comparable to or exceeds INT4 weights. Availability is therefore described by the triple (N, b, L) rather than the pair (N, b), and a generalised effective-bit-width coordinate $b_{eff}(L)$ is introduced, coinciding with the nominal value only at short context. Quantizing 16→4 bit equals two Moore's-law doublings achieved faster (~2 years versus ~3), yet, unlike Moore's law, the "quantization law" is exhaustible: below ~2-3 bits per weight quality collapses.

Ключевые слова: квантизация, большие языковые модели, KV-кэш, GQA, бюджет видеопамяти, закон Мура, эффективная битность, self-hosted, индекс аппаратной доступности.

Keywords: quantization, large language models, KV cache, GQA, VRAM budget, Moore's law, effective bit-width, self-hosted, hardware availability index.

Введение

Self-hosted-развёртывание больших языковых моделей – размещение весов на собственном или арендованном ускорителе вместо облачного API – стало массовой практикой: инструменты вроде Ollama и llama.cpp с квантованными весами (GGUF) позволяют запускать модели с десятками миллиардов параметров на одной потребительской видеокарте. Родительская работа [1] описывает эту доступность через координату X_6 – эффективную битность представления весов b – и вводит наивную модель бюджета видеопамяти $M = Nb/8$, где N – число параметров, с оговоркой: X_6 не вполне независима от N, поскольку обе величины входят в аппаратное ограничение лишь через произведение – бюджет M, а не по отдельности.

Оговорка верна для самой модели $M = Nb/8$, но модель неполна: она учитывает только вес матриц, тогда как авторегрессионный инференс держит в видеопамяти ещё и кэш ключей и значений внимания (KV-кэш), размер которого растёт с длиной контекста L и не зависит от b напрямую. При коротком контексте им можно пренебречь, при длинном – нет.

Задача работы – уточнить модель бюджета видеопамяти с учётом KV-кэша, показать на открытых моделях, при каких L он перестаёт быть пренебрежимым, обобщить координату X_6 до функции $b_{eff}(L)$ и сформулировать «закон квантизации» – сопоставление темпа роста доступности за счёт квантования с темпом закона Мура и указание предела исчерпаемости этого ресурса.

Полный бюджет видеопамяти

Полный бюджет складывается из четырёх слагаемых:

$$M_{total} = M_{weights} + M_{KV} + M_{act} + M_{over} \quad (1)$$

где $M_{weights}$ – память под веса, M_{KV} – память под кэш ключей и значений внимания, M_{act} – память под промежуточные активации прямого прохода, M_{over} – служебные накладные расходы рантайма (CUDA-контекст, фреймворк). Первое слагаемое – исходная модель родительской работы [1]:

$$M_{weights} = \frac{Nb}{8} \quad (2)$$

Второе слагаемое – то, что наивная модель упускает:

$$M_{KV} = \frac{2Ln_{layers}n_{kv}d_{head}b_{KV}}{8} \quad (3)$$

Множитель 2 – отдельные кэши для ключей (K) и значений (V): оба хранятся полностью для каждого обработанного токена, поскольку авторегрессионная генерация переиспользует их на каждом следующем шаге, не пересчитывая внимание заново. n_{kv_heads} – число голов внимания, для которых кэш хранится отдельно; при обычном multi-head attention $n_{kv\_heads} = n_{heads}$, и кэш растёт пропорционально полному числу голов. Grouped-query attention (GQA) [2] сокращает n_{kv_heads} в n_{heads}/n_{kv_heads} раз относительно числа голов-запросов: группа query-голов делит один общий кэш ключей и значений, что снижает M_{KV} во столько же раз без пересчёта b и N в модели родительской работы, поскольку эта экономия не связана ни с битностью весов, ни с их числом – только с архитектурой внимания.

M_{act} и M_{over} на порядок меньше двух первых слагаемых при батче, близком к 1 (типичный self-hosted-сценарий – один пользователь), и далее учитываются приближённо: $M_{over} \approx 1$ ГБ (контекст CUDA-рантайма) и $M_{act} \approx 0{,}05M_{weights}$ (эмпирическое практическое правило, используемое в руководствах по развёртыванию инференса; жёсткого теоретического значения для этой доли нет, поэтому она берётся ориентировочно, а не из отдельного источника).

Таблица 1 даёт M_{weights} и M_{KV} (при $b_{KV} = 16$ бит – распространённое по умолчанию значение, когда сами веса квантуются, а KV-кэш остаётся в

исходной точности) для четырёх открытых моделей двух семейств и двух классов размера.

Таблица 1. Конфигурации моделей и бюджет памяти на веса и KV-кэш

Модель	N, **млрд**	n_{layers}	n_{kv}	d_{head}	M_w**(16/8/4), ГБ**	M_{KV} **(8k/32k/128k), ГБ**
Llama 3.1 8B	8,03	32	8	128	16,1 / 8,0 / 4,0	1,07 / 4,29 / 17,18
Llama 3.1 70B	70,6	80	8	128	141,2 / 70,6 / 35,3	2,68 / 10,74 / 42,95
Qwen2.5-7B	7,61	28	4	128	15,2 / 7,6 / 3,8	0,47 / 1,88 / 7,52
Qwen2.5-72B	72,7	80	8	128	145,4 / 72,7 / 36,4	2,68 / 10,74 / 42,95

Источники конфигураций: Llama 3.1 – технический отчёт Meta [3]; конфигурации Qwen2.5 – по официальным карточкам моделей и публичному config.json на Hugging Face (проверено 22.07.2026). Число параметров N – по официальным карточкам моделей.

Показательна разница между 8B и 70B: конфигурация GQA у них одинакова ($n_{kv} = 8,\ d_{head} = 128$), то есть «цена» слоя KV-кэша на токен совпадает, а число слоёв различается лишь в 2,5 раза (32 против 80) против 8,8-кратной разницы в N. Поэтому при $L = 128k$ KV-кэш (17,2 ГБ) составляет 430% веса в INT4 (4,0 ГБ) для 8B, но лишь 122% для 70B (43,0 против 35,3 ГБ): меньшая модель относительно сильнее страдает от длинного контекста, хотя в сумме требует меньше видеопамяти – архитектура внимания «не масштабируется вниз» вместе с числом параметров.

Эффективная битность как функция контекста

Обобщением координаты X_6 служит:

$$b_{eff}(L) = \frac{8M_{total}(L)}{N} \quad (4)$$

b_{eff} совпадает с номинальной битностью b только в пределе $L \to 0$, когда KV-кэшем можно пренебречь; при росте L она монотонно растёт и может кратно превышать b. Рисунок 1 показывает $b_{eff}(L)$ для Llama 3.1 8B и 70B при трёх номинальных битностях.

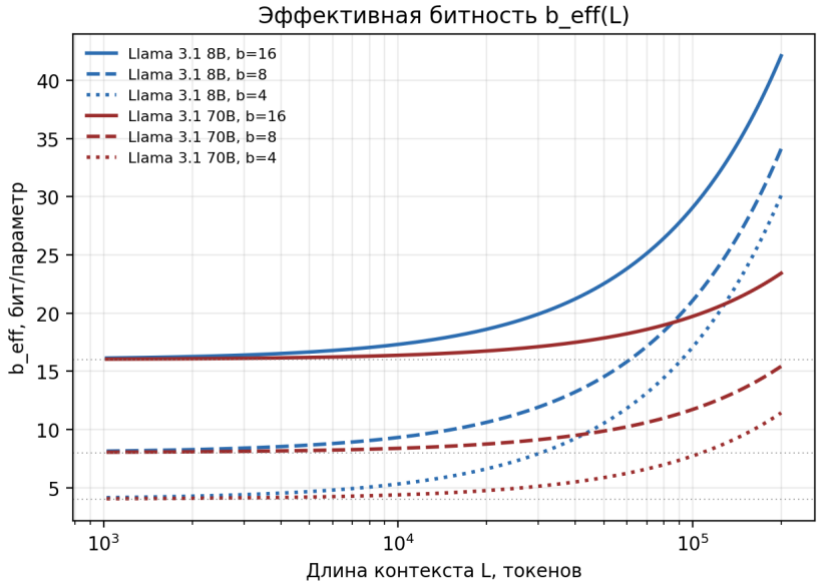

Рисунок 1. Эффективная битность $b_{eff}(L)$ для 8B и 70B при $b = 16, 8, 4$

Для 8B в INT4 ($b = 4$) уже при $L = 32k$ эффективная битность $b_{eff} = 8{,}28$ – вдвое выше номинала, а при $L = 128k$ она достигает 21,1 бит/параметр – это выше номинальной точности FP16 (16 бит), то есть при длинном контексте «4-битная» модель занимает в видеопамяти больше, чем её же 16-битная версия при коротком контексте. Для 70B эффект качественно тот же, но количественно слабее: b_{eff} при $L = 128k$, $b = 4$ равен 8,87 – рост более чем вдвое, но абсолютный масштаб куда скромнее эффекта для 8B. Из формулы (4) видно, что вывод родительской работы о паре (N, b) как полном описании доступности – частный случай при $L \to 0$; в общем случае необходима тройка (N, b, L).

Изокванты доступности и закон квантизации

Наивная модель $M = Nb/8$ задаёт в координатах (N, b) семейство изоквант – линий равного бюджета видеопамяти $Nb = const$. В логарифмических координатах по обеим осям такая линия – прямая с наклоном -1: увеличение $logN$ на единицу требует уменьшения $logb$ на ту же единицу, чтобы остаться на той же изокванте. Рисунок 2 показывает эти изокванты для четырёх уровней бюджета вместе с точками реальных моделей при $b = 16, 8, 4$.

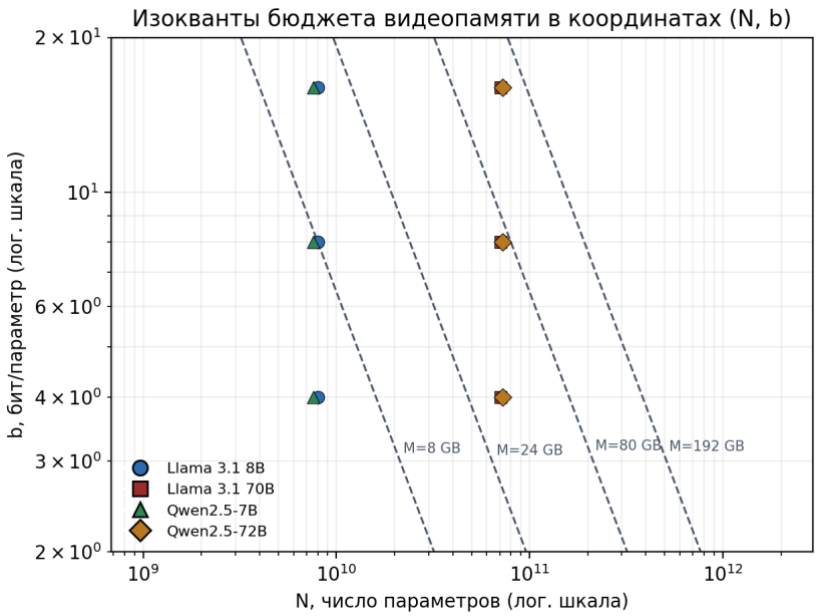

Рисунок 2. Изокванты бюджета видеопамяти в координатах (N, b) и точки открытых моделей

Движение вдоль изокванты – это и есть «обменный курс» между размером модели и точностью представления весов. Квантизация с 16 до 4 бит на вес – это сдвиг по изокванте, эквивалентный сокращению N в четыре раза при неизменном бюджете. Четырёхкратное сокращение N равно двум удвоениям плотности по закону Мура [4]: при классическом периоде удвоения около 18 месяцев на два удвоения потребовалось бы порядка 3 лет чисто аппаратного прогресса. В индустрии практический переход от FP16 как стандарта инференса к широко используемому INT4 занял около 2 лет: метод GPTQ, впервые сделавший точное 3-4-битное квантование практичным для моделей с миллиардами параметров, опубликован в конце 2022 г. [5], а к 2024 г. INT4-квантование (в форматах GGUF, AWQ) стало стандартной опцией развёртывания в инструментах вроде Ollama. То есть темп «закона квантизации» – прироста доступности за счёт снижения битности – обгоняет темп закона Мура.

Здесь и заключён главный тезис работы: в отличие от закона Мура, который на протяжении десятилетий не считался исчерпаемым ресурсом, у закона квантизации есть жёсткий физический предел. Ниже примерно 3 бит на

вес деградация качества перестаёт быть линейной и превращается в обвал: если при 4-битном квантовании относительно крупные модели теряют порядка 1 п.п. MMLU, то при переходе к 2-3 битам потери качества становятся катастрофическими [6]. Изокванту можно продолжать сдвигать математически сколь угодно далеко, но физически движение вниз по b упирается в стену качества уже через один-два шага после 4 бит. Закон Мура сопоставимого ограничения не имел (физический предел миниатюризации транзисторов лежит на существенно более поздних масштабах, чем те, где реально работала многолетняя история индустрии); закон квантизации исчерпывается почти сразу после точки, где даёт основной выигрыш.

Индекс аппаратной доступности

Практический смысл $b_{eff}(L)$ удобно выразить единым индексом – долей ускорителей из заданного парка, на которые модель помещается целиком:

$$A(N, b, L) = \text{доля конфигураций парка ускорителей, для которых } M_{total}(N, b, L) \leq VRAM \quad (5)$$

Парк из 8 ускорителей (объёмы VRAM – по спецификациям производителя): потребительские RTX 4070 (12 ГБ), RTX 4080 (16 ГБ), RTX 4090 (24 ГБ), RTX 5090 (32 ГБ); профессиональный RTX PRO 6000 Blackwell (96 ГБ); дата-центровые A100 (80 ГБ), H100 (80 ГБ), H200 (141 ГБ). Таблица 2 даёт $A(N, b, L)$ для четырёх моделей при $b \in \{16, 8, 4\}$ и $L \in \{8k, 32k\}$ (M_{total} включает M_{weights}, M_{KV} и практическую надбавку $M_{act} + M_{over}$ из раздела 2).

Таблица 2. Индекс аппаратной доступности $A(N, b, L)$ по парку из 8 ускорителей

Модель	b	A, L=8k	A, L=32k
Llama 3.1 8B	16	0,75	0,75
Llama 3.1 8B	8	1,00	0,88
Llama 3.1 8B	4	1,00	1,00
Llama 3.1 70B	16	0,00	0,00
Llama 3.1 70B	8	0,50	0,25
Llama 3.1 70B	4	0,50	0,50
Qwen2.5-72B	16	0,00	0,00
Qwen2.5-72B	8	0,25	0,25
Qwen2.5-72B	4	0,50	0,50

Таблица показывает эффект, невидимый в наивной модели: для Llama 3.1 70B при $b = 8$ индекс падает с 0,50 до 0,25 при переходе от $L = 8k$ к $L = 32k$ – потому что M_{total} пересекает порог 80 ГБ (класс A100/H100), и остаются только 96- и 141-гигабайтные ускорители. Для 72B при $b = 4$ индекс между $L = 8k$ и $L = 32k$ не меняется (0,50): прироста KV-кэша не хватает, чтобы пересечь следующий порог. Доступность меняется с ростом L не гладко, а скачками – когда растущий M_{total} пересекает границу VRAM очередного класса ускорителей; зная только (N, b), эти скачки предсказать нельзя.

Ограничения

Три ограничения существенны. Во-первых, M_{act} и M_{over} взяты приближённо (как практическое правило, не из отдельного источника): при батче больше 1 или множестве параллельных запросов активации перестают быть второстепенным слагаемым – модель раздела 2 рассчитана на однопользовательский сценарий, а не на серверную нагрузку. Во-вторых, b_{KV} в Таблицах 1-2 зафиксирован на 16 битах (KV-кэш не квантован); его отдельное квантование (до 8 или 4 бит) пропорционально снижает M_{KV} и сдвигает все пороги $b_{eff}(L)$ и $A(N, b, L)$ – направление эффекта то же, но масштаб в расчёт не заложен. В-третьих, рассмотрены только плотные (dense) модели; для MoE-архитектур число активных параметров на шаге меньше полного N, и модель памяти для них требует отдельной формализации, которую данная работа не даёт.

Парк ускорителей в разделе 5 иллюстративен (8 моделей), а не исчерпывающий каталог рынка: индекс A чувствителен к его составу – добавление или исключение карты может сдвинуть A на конкретной комбинации (N, b, L), не меняя качественного вывода о скачкообразности доступности.

Заключение

Наивная модель бюджета видеопамяти $M = Nb/8$ из родительской работы [1] занижает требование при длинном контексте: KV-кэш, в ней отсутствующий, при L порядка 32-128 тыс. токенов сопоставим с весами в

INT4 или превосходит их, причём относительно сильнее у меньших моделей – архитектура внимания не масштабируется вниз вместе с N. Поэтому доступность self-hosted-развёртывания описывается не парой (N, b), а тройкой (N, b, L), а координата X_6 обобщается до функции $b_{eff}(L)$, совпадающей с исходной битностью лишь при нулевом контексте. Изокванты бюджета в координатах (N, b) показывают, что переход от 16 к 4 битам эквивалентен четырёхкратному сокращению N – двум удвоениям закона Мура, пройденным быстрее самого закона Мура. Главный вывод: в отличие от закона Мура, закон квантизации исчерпаем – физический предел около 2-3 бит на вес делает этот ресурс принципиально конечным. Индекс аппаратной доступности $A(N, b, L)$ показывает это практически: на конкретном парке ускорителей доступность меняется скачками при пересечении порогов VRAM, а не гладко с ростом контекста.

Список литературы

1. Бражников А.Р. К вопросу о феноменологических закономерностях развития больших языковых моделей: возможен ли аналог закона Мура? // Проблемы искусственного интеллекта. – 2026. – № 1 (40). – С. [страницы уточнить].
2. Ainslie J., Lee-Thorp J., de Jong M., Zemlyanskiy Y., Lebrón F., Sanghai S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints [электронный ресурс]. – arXiv:2305.13245, 2023. – URL: https://arxiv.org/abs/2305.13245 (дата обращения: 22.07.2026).
3. Grattafiori A., Dubey A., Jauhri A. et al. The Llama 3 Herd of Models [электронный ресурс]. – arXiv:2407.21783, 2024. – URL: https://arxiv.org/abs/2407.21783 (дата обращения: 22.07.2026).
4. Moore G.E. Cramming More Components onto Integrated Circuits // Electronics. – 1965. – Vol. 38, № 8. – P. 114–117.
5. Frantar E., Ashkboos S., Hoefler T., Alistarh D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers

[электронный ресурс]. – arXiv:2210.17323, 2022. – URL: https://arxiv.org/abs/2210.17323 (дата обращения: 22.07.2026).

6. An Empirical Study of Qwen3 Quantization [электронный ресурс]. – arXiv:2505.02214, 2025. – URL: https://arxiv.org/abs/2505.02214 (дата обращения: 22.07.2026).

References

1. Brazhnikov A.R. On Phenomenological Regularities in the Development of Large Language Models: Is a Moore's Law Analogue Possible? // Problems of Artificial Intelligence. – 2026. – No. 1 (40). – P. [to be specified].
2. Ainslie J., Lee-Thorp J., de Jong M., Zemlyanskiy Y., Lebrón F., Sanghai S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints [electronic resource]. – arXiv:2305.13245, 2023. – Available at: https://arxiv.org/abs/2305.13245 (accessed: 22.07.2026).
3. Grattafiori A., Dubey A., Jauhri A. et al. The Llama 3 Herd of Models [electronic resource]. – arXiv:2407.21783, 2024. – Available at: https://arxiv.org/abs/2407.21783 (accessed: 22.07.2026).
4. Moore G.E. Cramming More Components onto Integrated Circuits // Electronics. – 1965. – Vol. 38, No. 8. – P. 114–117.
5. Frantar E., Ashkboos S., Hoefler T., Alistarh D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers [electronic resource]. – arXiv:2210.17323, 2022. – Available at: https://arxiv.org/abs/2210.17323 (accessed: 22.07.2026).
6. An Empirical Study of Qwen3 Quantization [electronic resource]. – arXiv:2505.02214, 2025. – Available at: https://arxiv.org/abs/2505.02214 (accessed: 22.07.2026).