Обучение GigaChat с контекстом в сотни тысяч токенов

Косарев Евгений
Senior DL Engineer, Sber Devices

SBER DEVICES

GigaChat в бизнесе

Салют

Jazz

- В десятках продуктов Сбера
- Работа с текстом в МойОфис
- Департамент здравоохранения г.Москвы
- Более **2 тысяч** клиентов GigaChat API

Я готовлю доклад про длинный контекст в языковых моделях на конференцию. Кратко расскажи, что это такое, и на чем стоит акцентировать внимание в докладе?

Длинный контекст (long context) — это подход к обучению языковых моделей, при котором они обучаются на больших текстах, включающих в себя много информации о мире. Длинный контекст позволяет модели «понять» большие тексты и использовать эту информацию для генерации более качественных ответов.

При подготовке доклада об этом подходе стоит обратить внимание на следующие аспекты:

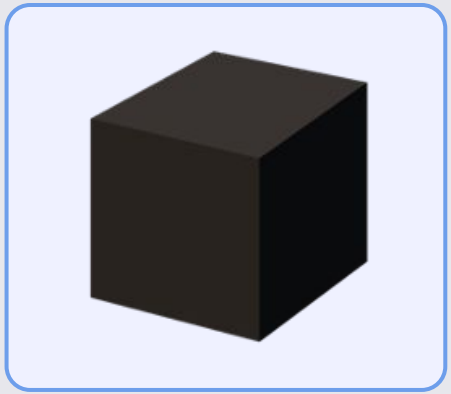
1. **Преимущества использования длинного контекста**. Модели, обученные на больших текстах, могут лучше понимать мир и выдавать более точные ответы. Они также могут использоваться для решения сложных задач, таких как анализ документов или создание художественных произведений.

2. **Технологии**. Можно рассказать о способах и техниках обучения моделей длинному контексту.

3. **Перспективы развития.** Можно обсудить возможные направления развития этого подхода, такие как использование нейронных сетей нового поколения или обучение моделей на больших объемах данных.

Что такое языковые модели

Языковые модели – алгоритмы, способные продолжать тексты



GigaChat это …

известная языковая модель

LLaMa-3

Бенчмарки – как показатель качества языковой модели

	2024	2023		2024	2023
Benchmark	**Llama 3 8B**	**Llama 2 7B**	**Llama 2 13B**	**Llama 3 70B**	**Llama 2 70B**
MMLU (5-shot)	68.4	34.1	47.8	82.0	52.9
GPQA (0-shot)	34.2	21.7	22.3	39.5	21.0
HumanEval (0-shot)	62.2	7.9	14.0	81.7	25.6
GSM-8K (8-shot, CoT)	79.6	25.7	77.4	93.0	57.5
MATH (4-shot, CoT)	30.0	3.8	6.7	50.4	11.6

LLaMa-3

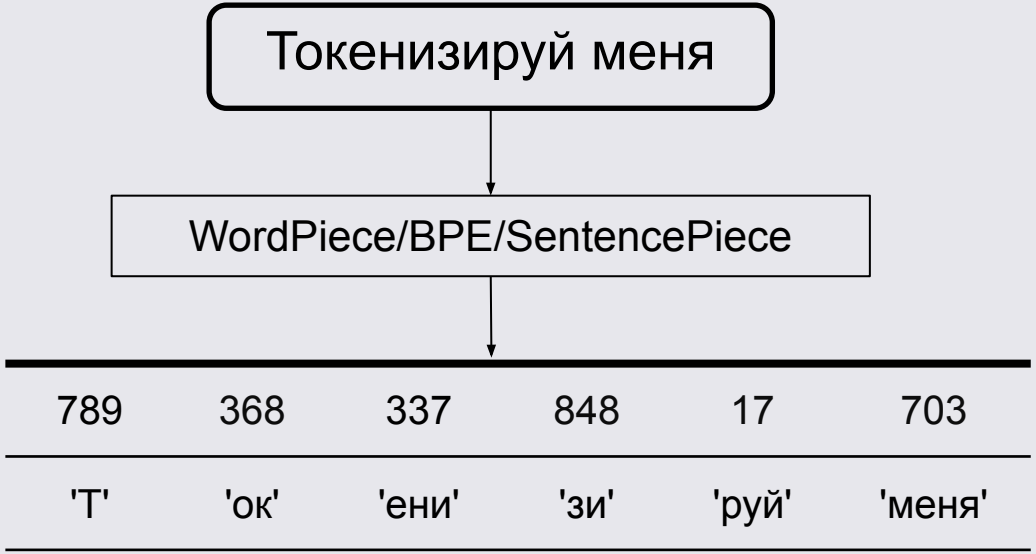
Бенчмарки – как показатель качества языковой модели

Приближается к GPT-4

Benchmark	Llama 3 8B (2024)	Llama 2 7B (2023)	Llama 2 13B (2023)	Llama 3 70B (2024)	Llama 2 70B (2023)
MMLU (5-shot)	68.4	34.1	47.8	82.0	52.9
GPQA (0-shot)	34.2	21.7	22.3	39.5	21.0
HumanEval (0-shot)	62.2	7.9	14.0	81.7	25.6
GSM-8K (8-shot, CoT)	79.6	25.7	77.4	93.0	57.5
MATH (4-shot, CoT)	30.0	3.8	6.7	50.4	11.6

Контекст в LLM

Токен - единица текста, которую “понимают” языковые модели



Чего-то не хватает…

Большинство Open-Source моделей имеет контекст

8192 токена

~4 страницы текста А4

Нужен ли контекст больше?

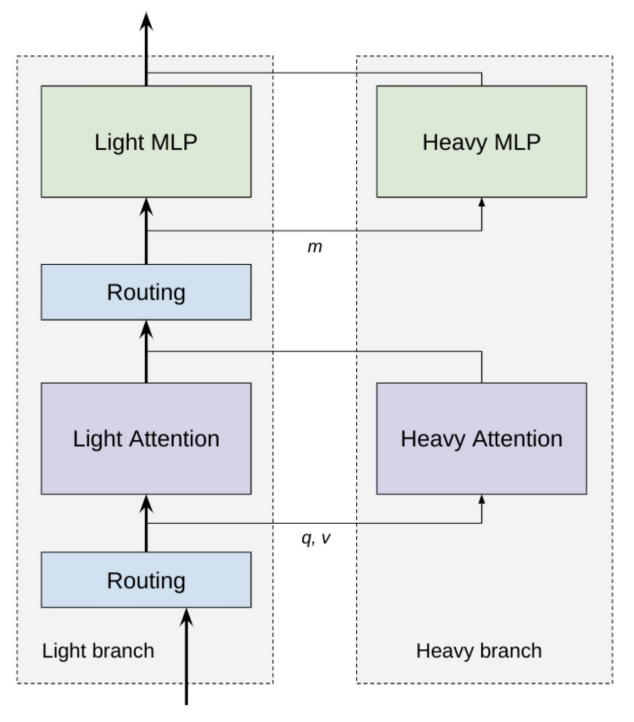
Поиск по документам

Пользователь задает вопрос к документу **>20** страниц

А4. Как на него ответить?

Столько можем обработать за раз

Решения



Модификации Attention (CoLT5)

Преимущества:

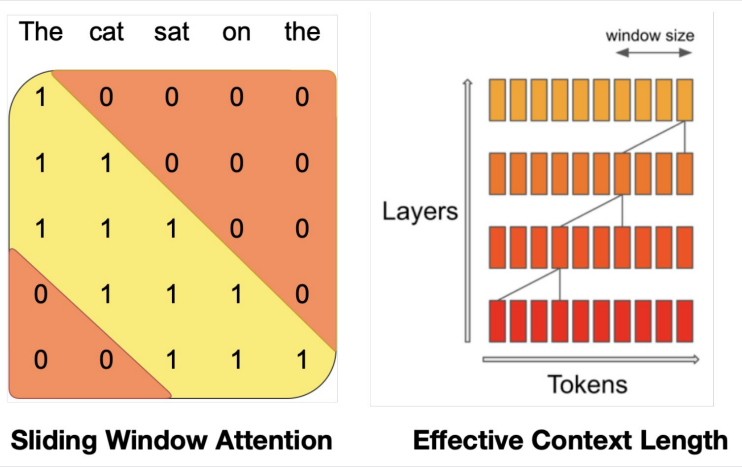
- Видит весь контекст

Недостатки:

- Роутер совершает ошибки
- Light ветка плохо собирает контекст

Решения

Модификации Attention (SWA)



Преимущества:

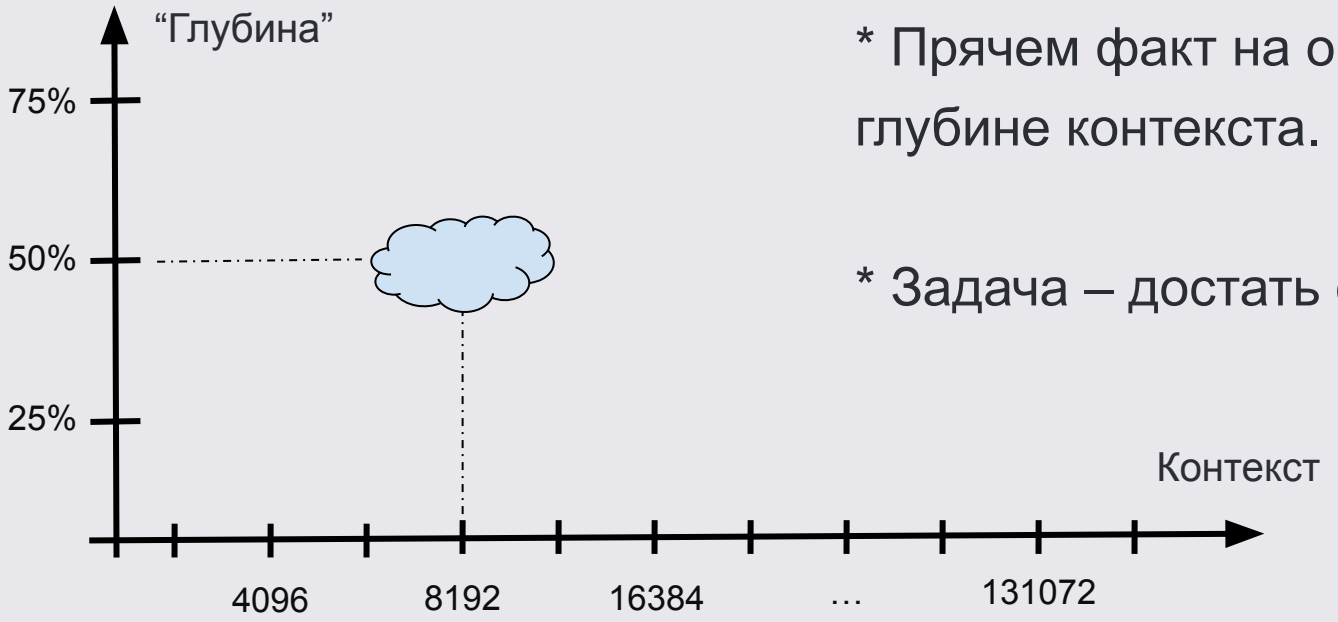
- Полное внимание в окне
- Агрегация информации с нижний слоев

Недостатки:

- Потеря информации через слои

PassKey

Универсальный бенчмарк по проверке длинного контекста



* Прячем факт на определенной глубине контекста.

* Задача – достать факт

Mistral 7B

Результаты PassKey

Mistral 7B v.01

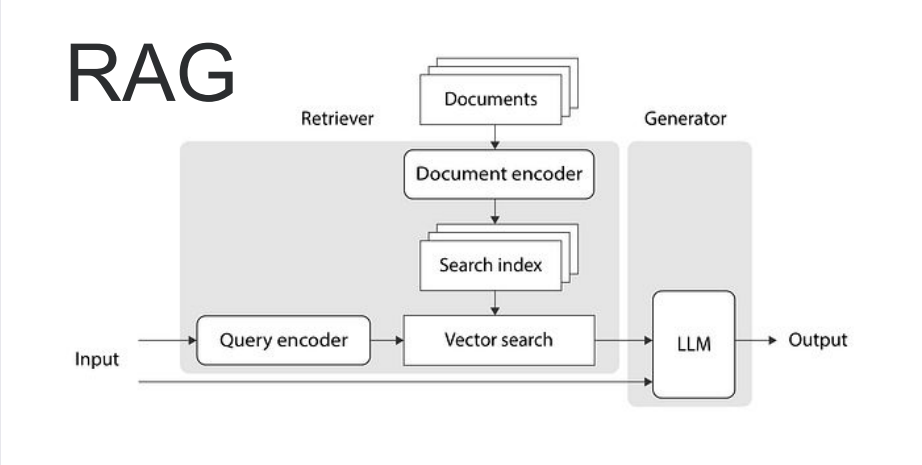
SWA 32k

Mistral 7B v.02

32k

	4k	8k	16k	32k
Mistral 7B v.01	100	54	28	22
Mistral 7B v.02	100	100	100	100

Решения



Преимущества:

- поиск намного дешевле запроса в GPT
- “Бесконечный” контекст

Недостатки:

- Качество поисковика
- Ограниченность информации

Решения

Честное увеличение контекстного окна модели

8192 → 32768 → 131072

Проблемы:

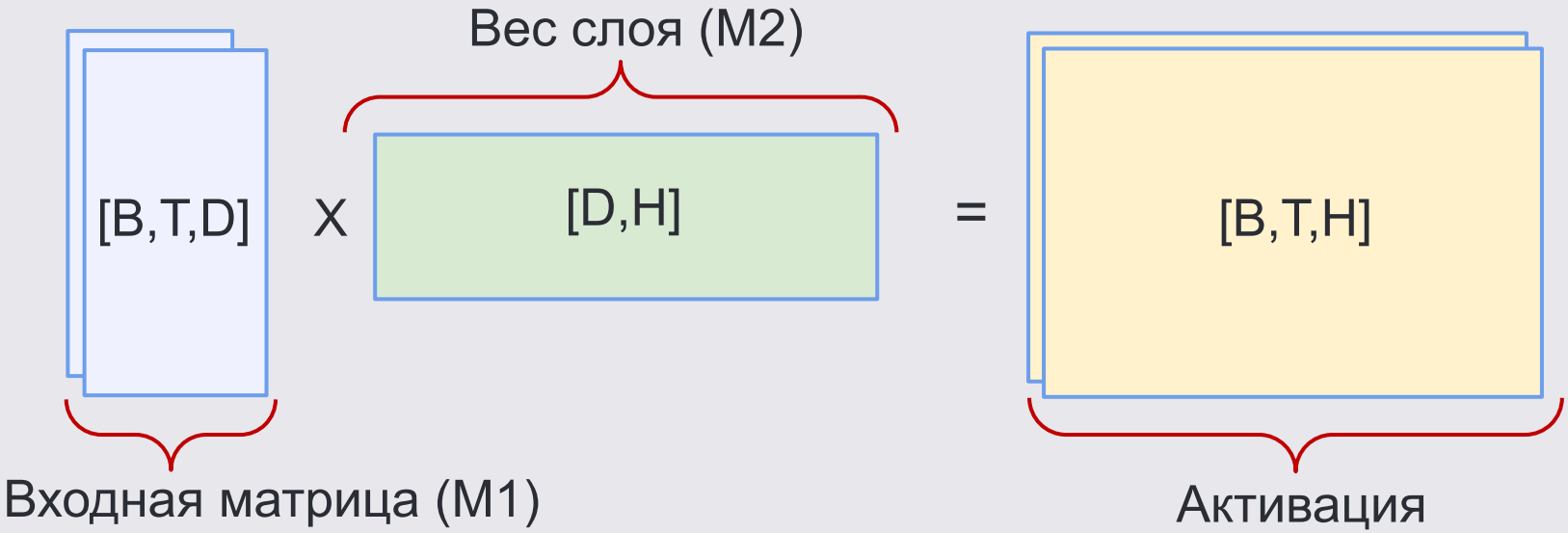
1. Активации тензоров не помещаются на GPU
2. Замедление вычислений

Оптимизация вычислений

Цель оптимизации

T – число токенов в контексте
D,H – внутренние размерности
B – размер батча

Активация - результат умножения входной матрицы на веса слоя



Архитектура LLM

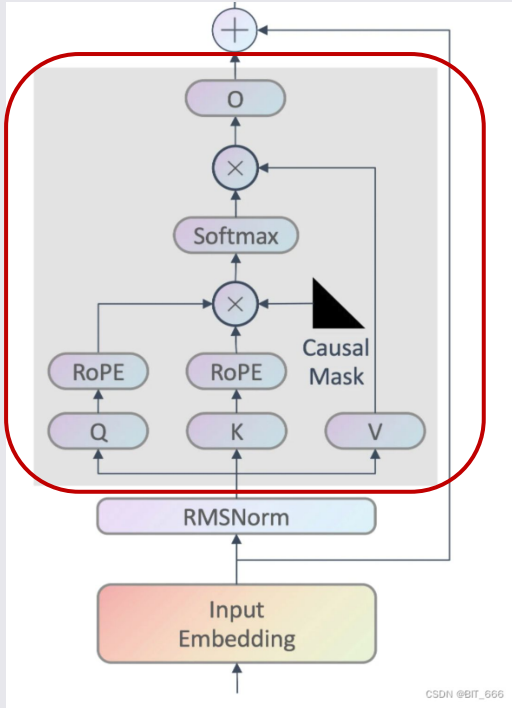
LLaMa-1,2,3

Состоит из Decoder блоков

Архитектура LLM

LLaMa-1,2,3

Attention



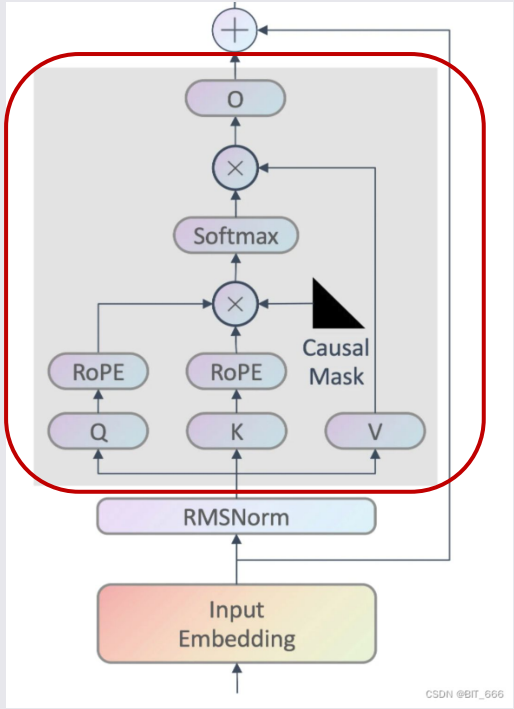
Состоит из Decoder блоков

Архитектура LLM

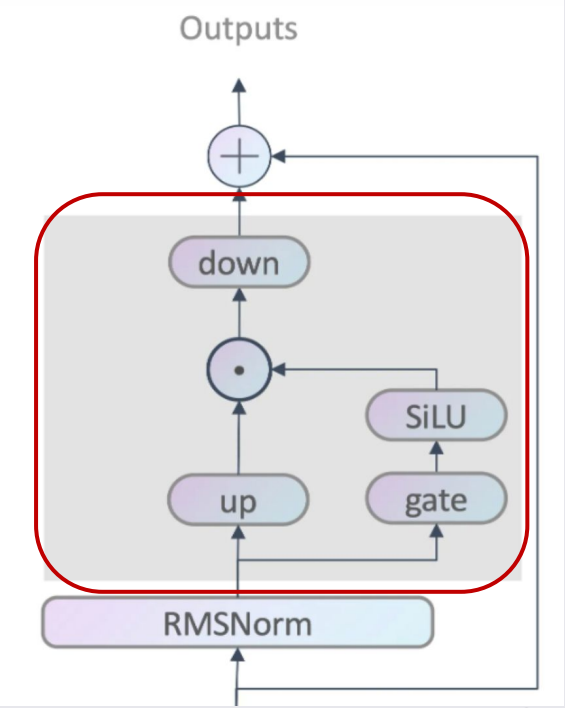
LLaMa-1,2,3

Состоит из Decoder блоков

Attention



MLP



Attention

Механизм внимания позволяет учитывать взаимосвязи токенов

Сложность: $O(T^2 * D + T * D^2)$

Память: $O(T^2 + T * D)$

MLP

Производит преобразования знаний модели

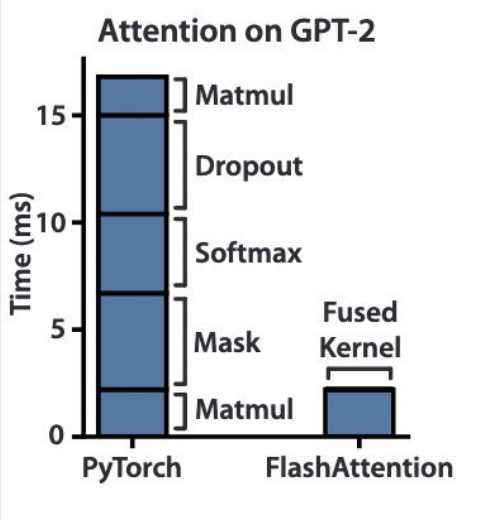
Сложность: $O(TDH)$

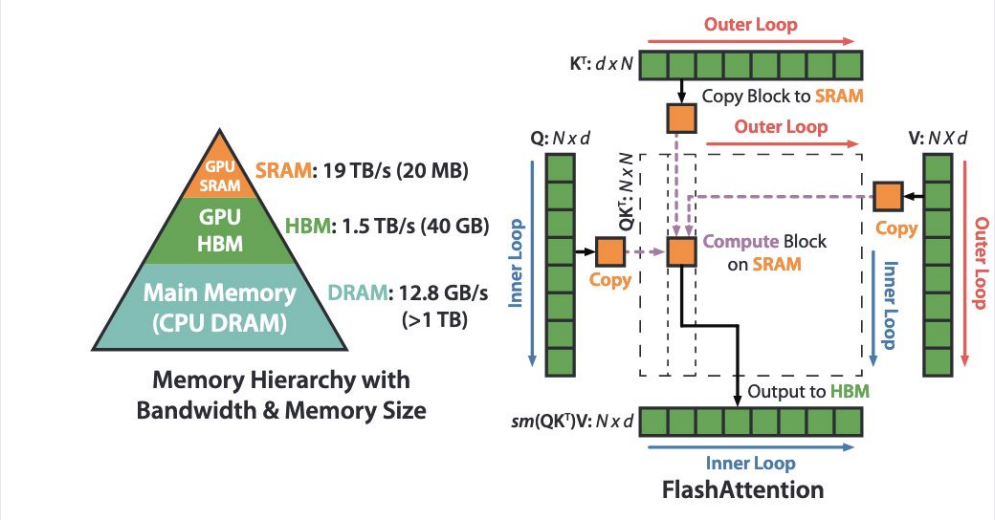
Память: $O(TH)$

Flash-Attention

Память	Скорость
-7GB	+55%

Эффективное вычисление внимания на GPU





GQA

Память	Скорость
-1.2GB	+9%

В k раз (число общих голов) уменьшается потребление памяти K,V матрицами, ускоряется обучение и инференс

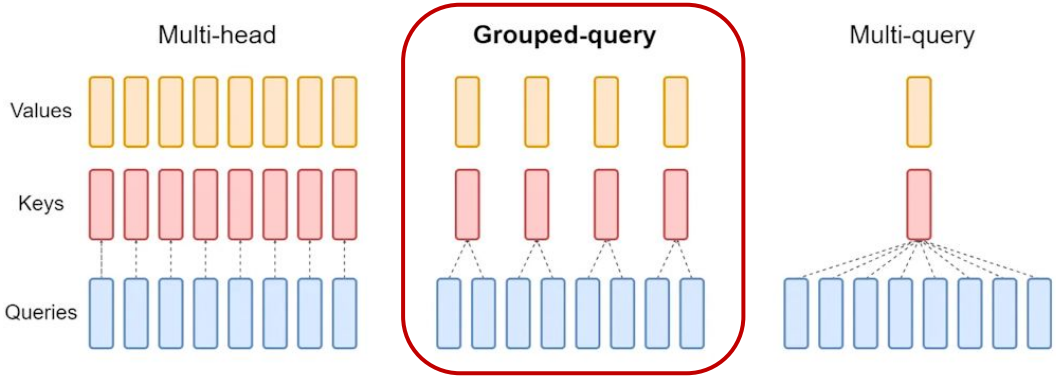
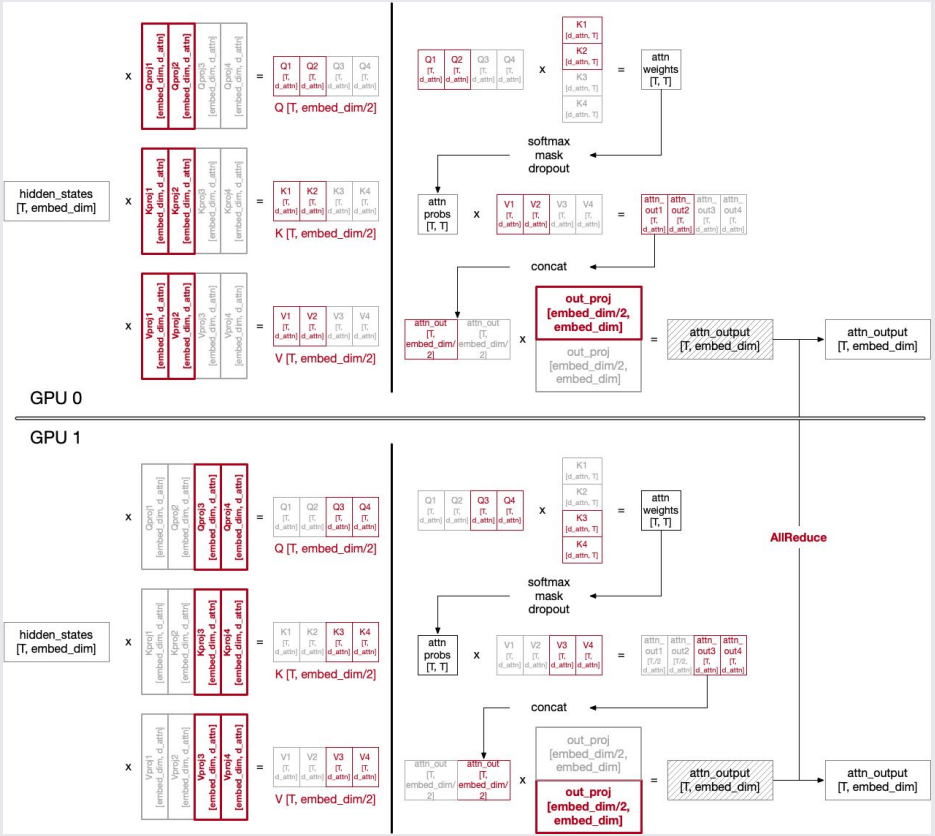


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.

Tensor Parallel

Память	Скорость
-50%	+91%



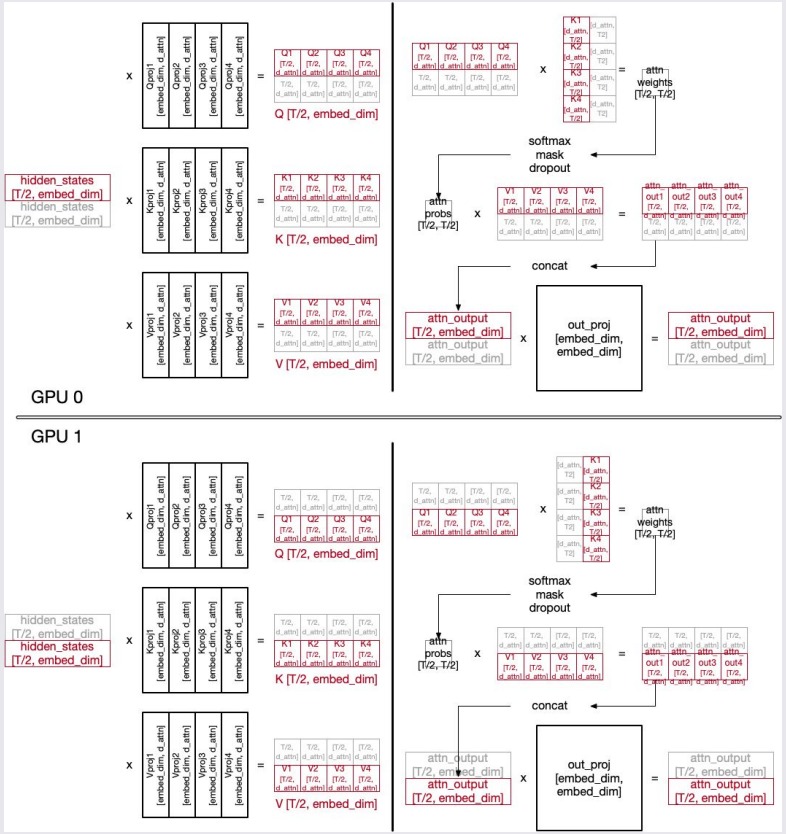
Делим вычисления между GPU **<u>по размерности D</u>**

+ Каждая GPU держит в TP раз меньше весов и считает в TP раз меньшие активации

– Тратится время на коммуникации между GPU

Sequence Parallel

Память	Скорость
-51%	+88%



Делим вычисления между GPU **<u>по размерности T</u>**

+ Каждая GPU считает в TP раз меньшие активации

– Тратится время на коммуникации в Attention

Tensor VS Sequence Parallel

Table 2: Multi-head attention block memory usage comparison

	Operation	M1	M2	output	Memory
Tensor parallelism	$Q/K/V$	$(\mathrm{B}, \mathrm{L}, \mathrm{H})$	$(\mathrm{H}, \frac{\mathrm{ZA}}{\mathrm{N}})$	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{A})$	$\frac{16\mathrm{AZH}}{\mathrm{N}} + \frac{4\mathrm{BLZA}}{\mathrm{N}} + \frac{\mathrm{BZL}^2}{\mathrm{N}} + \mathrm{BLH}$
	QK^T	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{A})$	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{A})$	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{L})$	
	AV	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{L})$	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{A})$	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{A})$	
	Linear	$(\mathrm{B}, \frac{\mathrm{Z}}{\mathrm{N}}, \mathrm{L}, \mathrm{A})$	$(\frac{\mathrm{AZ}}{\mathrm{N}}, \mathrm{H})$	$(\mathrm{B}, \mathrm{L}, \mathrm{H})$	
Sequence parallelism	$Q/K/V$	$(\mathrm{B}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{H})$	$(\mathrm{H}, \mathrm{AZ})$	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{A})$	$16\mathrm{AZH} + \frac{4\mathrm{BZLA}}{\mathrm{N}} + \frac{\mathrm{BZL}^2}{\mathrm{N}} + \frac{\mathrm{BLH}}{\mathrm{N}}$
	Ring-QK^T	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{A})$	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{A})$	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{L})$	
	Ring-AV	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{L})$	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{A})$	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{A})$	
	Linear	$(\mathrm{B}, \mathrm{Z}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{A})$	$(\mathrm{AZ}, \mathrm{H})$	$(\mathrm{B}, \frac{\mathrm{L}}{\mathrm{N}}, \mathrm{H})$	

Tensor VS Sequence Parallel

Для LLaMa-2 моделей верно следующее (B = 1):

Attention

SP выгоднее TP при

$T > 8.1D$

MLP

SP выгоднее TP при

$T > 16D$

* А еще есть параллельные Эмбединги и параллельный лосс

Реализация SP - all2all

Архитектурно Decoder LLM может обрабатывать произвольный контекст

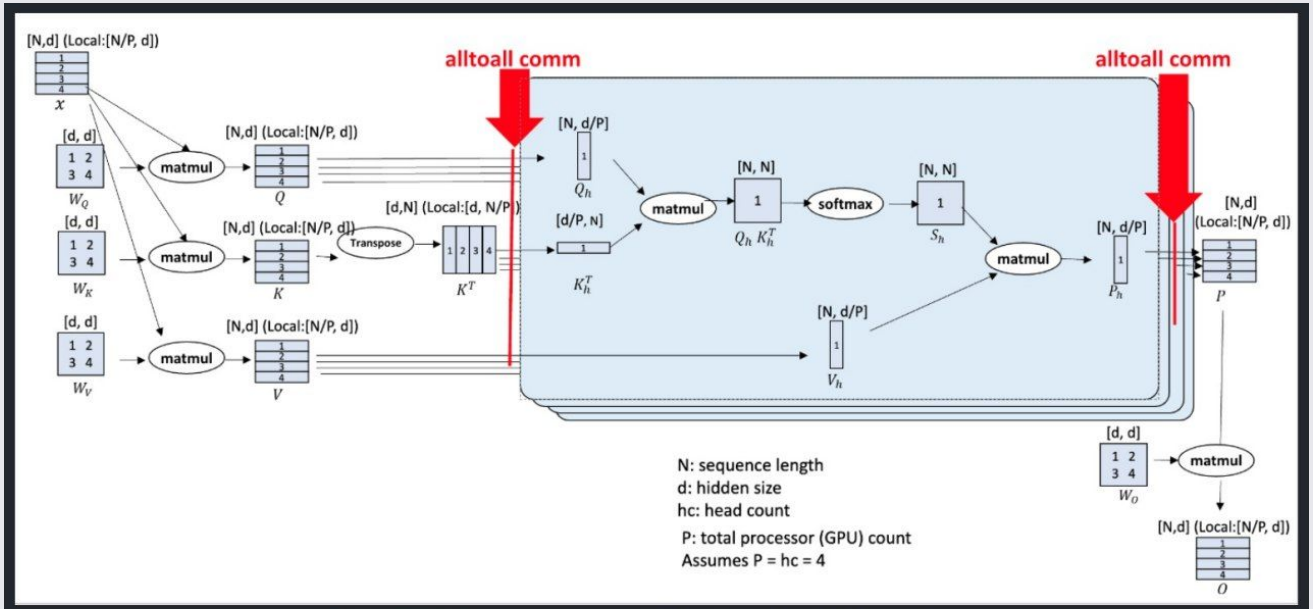
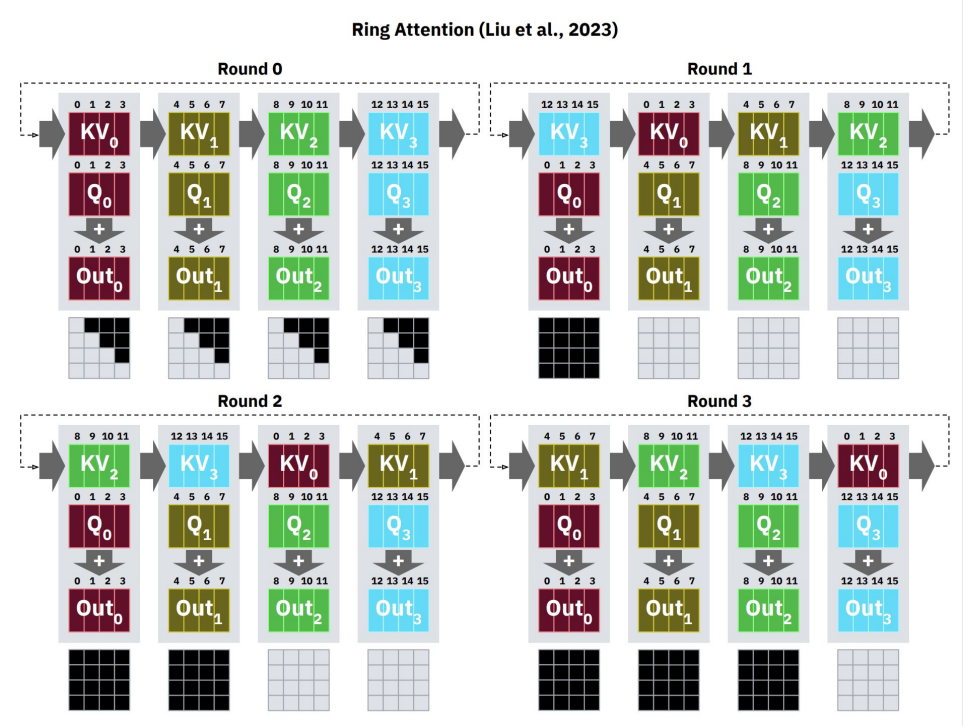


Figure 1: DeepSpeed sequence parallelism (DeepSpeed-Ulysses) design

Ring Attention

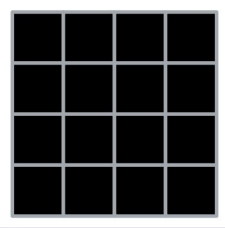
Экстремальные контексты требуют подсчета попарного внимания всех на всех токенов.

Можем считать внимание в цикле

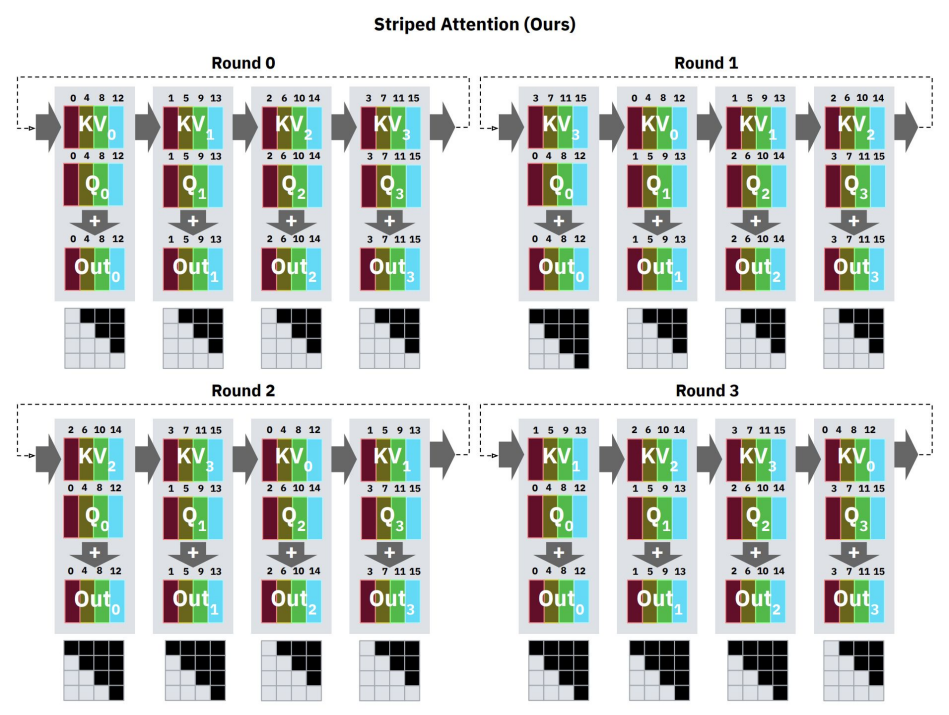


Striped Ring Attention

Такая маска - простой GPU



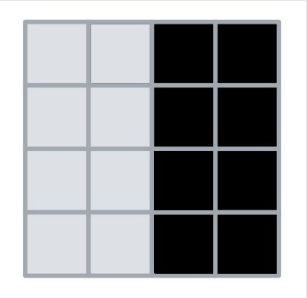
Striped Attention реорганизует вычисления на более равномерные

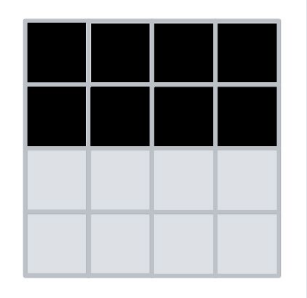


ZigZag Ring Attention

Striped

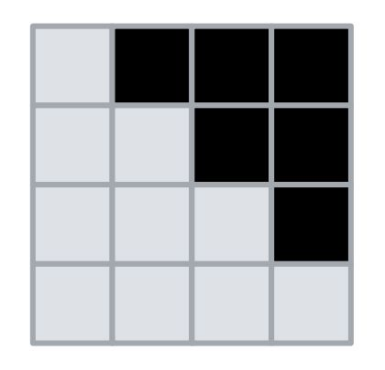
0,4,8,12 | 1,5,9,13 | 2,6,10,14 | 3,7,11,15





ZigZag - группируем по 2 и делаем разбиение

0,1,14,15 | 2,3,12,13 | 4,5,10,11 | 6,7,8,9



Каждая GPU простаивает не больше половины времени. Лучше Striped

Сравнение реализаций

	GPU	theoretic flash_attn	ring_attn	zigzag_ring	stripe_attn
fwd only (iter/sec)	8xH800	2418.4 / 8 = 302.3	208.0	283.0	259.6
			68.8%	**93.6%**	85.9%
fwd + bwd (iter/sec)	8xH800	705.2 / 8 = 88.2	54.3	75.7	76.9
			61.5%	85.9%	**87.2%**
fwd only (iter/sec)	8xA100	1545.9 / 8 = 193.2	124.4	179.0	163.9
			64.3%	**92.7%**	84.8%
fwd + bwd (iter/sec)	8xA100	470.6 / 8 = 58.8	33.3	49.5	45.9
			56.6%	**84.1%**	78.1%

Лучшая

Ring Attention VS all2all

TP=1, SP=4, 64 GPU

	Ring Attention		all2all	
	tokens per sec / 1 GPU	mem (GB) / 1 GPU	tokens per sec / 1 GPU	mem (GB) / 1 GPU
LLaMa 7B 32k	1717	**16,9**	**2030**	20,6
LLaMa 7B 128k	911	**30,9**	**946**	47,0
LLaMa 30B 32k	**907**	**36,8**	519	50
LLaMa 30B 128k	**239**	**55,1**	—	—

Обучение GigaChat

Обучение GigaChat

Подведем итоги:

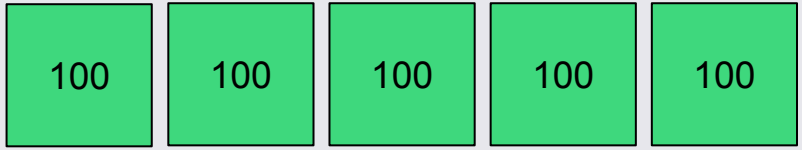
- Имеем формулы подсчета активаций
- Знаем, когда Tensor Parallel выгоднее Sequence Parallel
- Умеем объединять вместе оба режима
- Эффективный Ring Attention - нужен для экстремальных контекстов

GigaChat 7B и 29B

Результаты PassKey

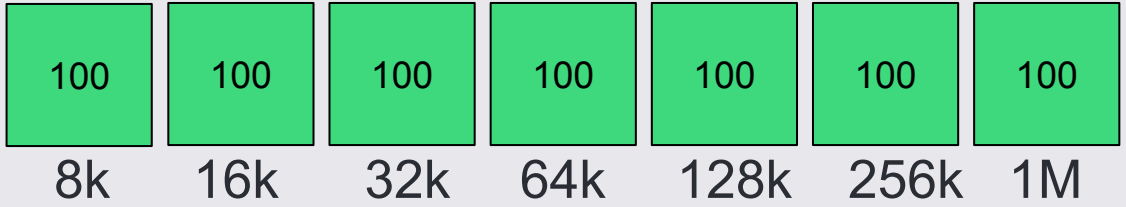
GigaChat 29B

128k



GigaChat 7B

1M



GigaChat 7B и 29B

Результаты PassKey

GigaChat 29B

128k

64

GigaChat 7B

1M

8k	16k	32k	64k	128k	256k	1M
100	100	100	100	100	100	100

GigaChat 7B и 29B

Результаты PassKey

GigaChat 29B

128k

64

GigaChat 7B

1M

512 страниц А4

Перспективы

Будущее длинного контекста

Код, видео и аудио требуют большого контекста

Будущее длинного контекста

Код, видео и аудио требуют большого контекста

Решение бизнес задач с дополнительной информацией

Будущее длинного контекста

Код, видео и аудио требуют большого контекста

Решение бизнес задач с дополнительной информацией

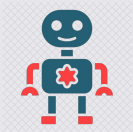
Персонализация языковых моделей

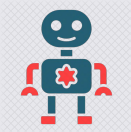
Будущее длинного контекста

Код, видео и аудио требуют большого контекста

Решение бизнес задач с дополнительной информацией

Персонализация языковых моделей





Мульти-агентное взаимодействие

Спасибо за внимание!

Косарев Евгений
Senior DL Engineer, Sber Devices

Tg: evgenijkkk
Почта: evgenijkkk@yandex.ru

SBER DEVICES