



清华大学

Tsinghua University

# LLM Serving on Heterogeneous Hardware

Mingxing Zhang @  KVCache.AI

<https://github.com/kvcache-ai>



Mooncake

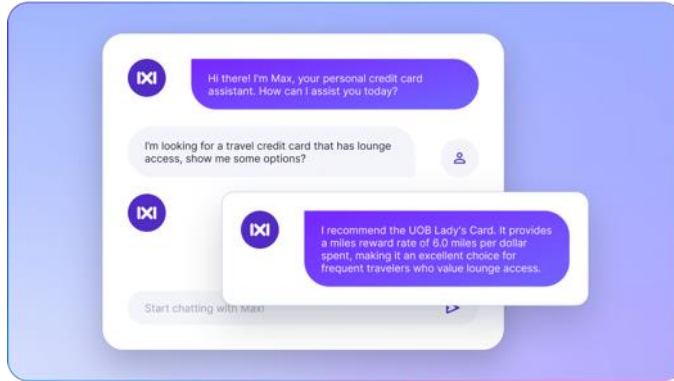
 Transformers

TrEnv-X

---

# Background: Large Language Models (LLMs)

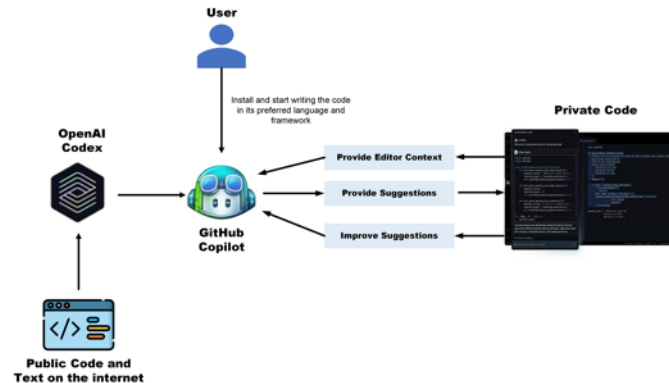
Large Language Models (LLMs) are widely applied in industry and researched in academia.



Knowledge Q&A



Content Creation



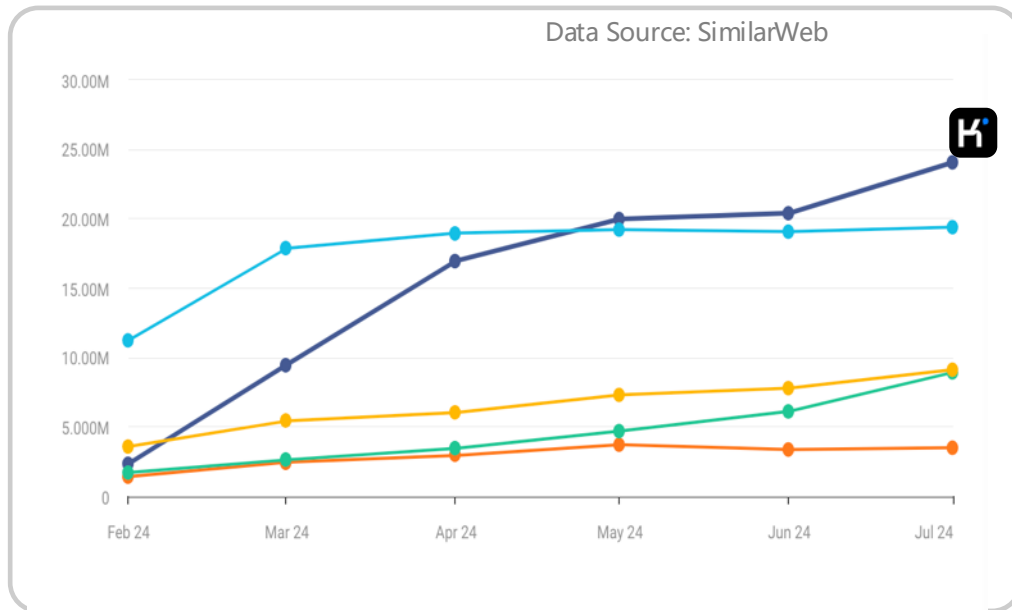
Code Generation



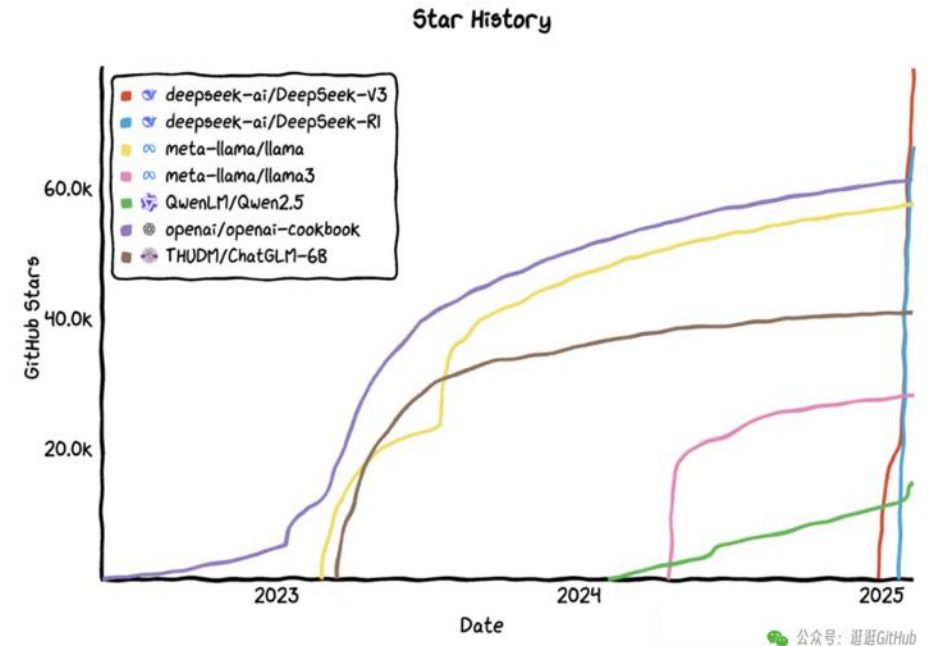
Office Assistant

# Challenge of Online Model as a Service System

More Data + Larger Model + Longer Context = 😊 Higher Intelligence



Long input: Moonshot AI's Kimi Supports 2 Million Characters Input in March 2024, become a widely recognized app in China

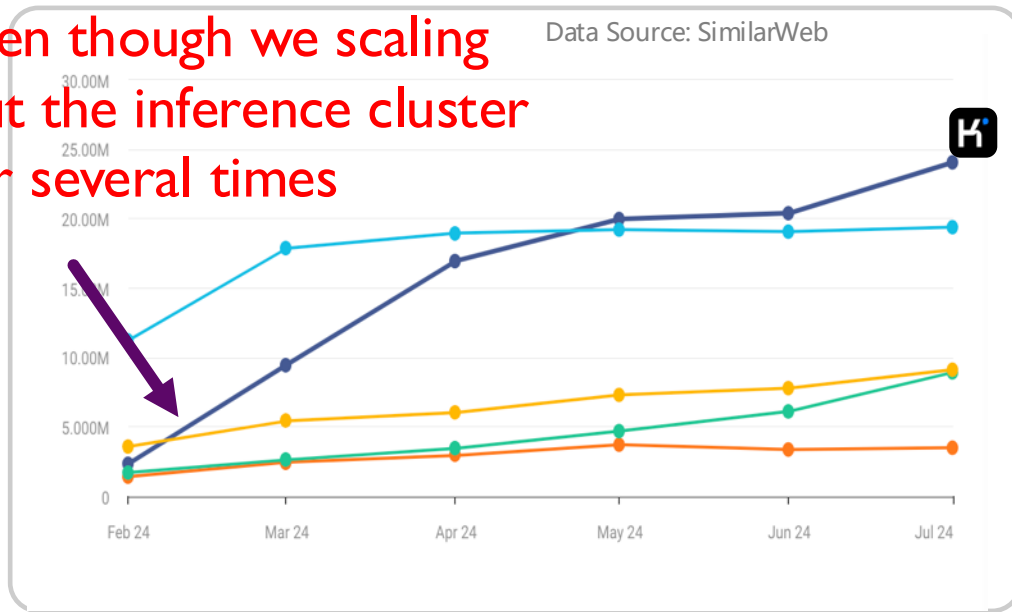


Long output: DeepSeek release V3/R1 at Dec 2024, Become a widely recognized app in global

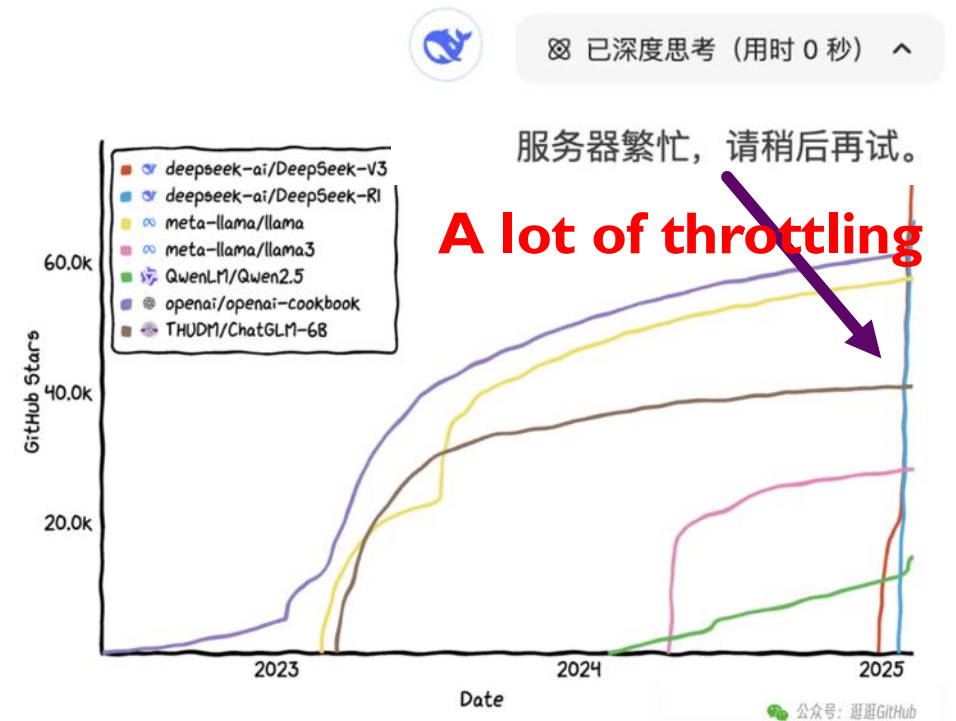
# Challenge of Online Model as a Service System

More Data + Larger Model + Longer Context = ☹️ Higher Service Loads

Frequent **out of service**  
even though we scaling  
out the inference cluster  
for several times



Long input: Moonshot AI's Kimi Supports 2 Million Characters Input in March 2024, become a widely recognized app in China



Long output: DeepSeek release V3/R1 at Dec 2024, Become a widely recognized app in global



# Content

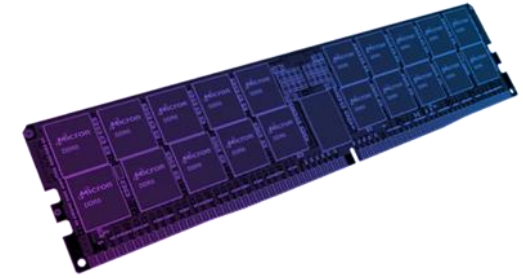
- Motivation for Heterogeneous LLM Serving
- Core Technologies of Mooncake
- Core Technologies of KTransformers
- Tutorial: Fine-Tune and Chat with Your Customized Model Locally

!!! The price numbers are not accurate, just a demonstration!



H800

H20



Xeon SPR + 8 \* DDR5-4800

Hardware  
Spec

80GB VRAM, 3.3 TBps  
~ 1 PFLOPS  
> \$ 10,000

96GB VRAM, 4 TBps  
~ 200 TFLOPS  
~ \$50,000

8\*64GB DRAM, 8\*40GB/s  
< 20 TFLOPS  
~ ¥60,000

Best  
for

Allround,  
especially for TFLOPS/\$

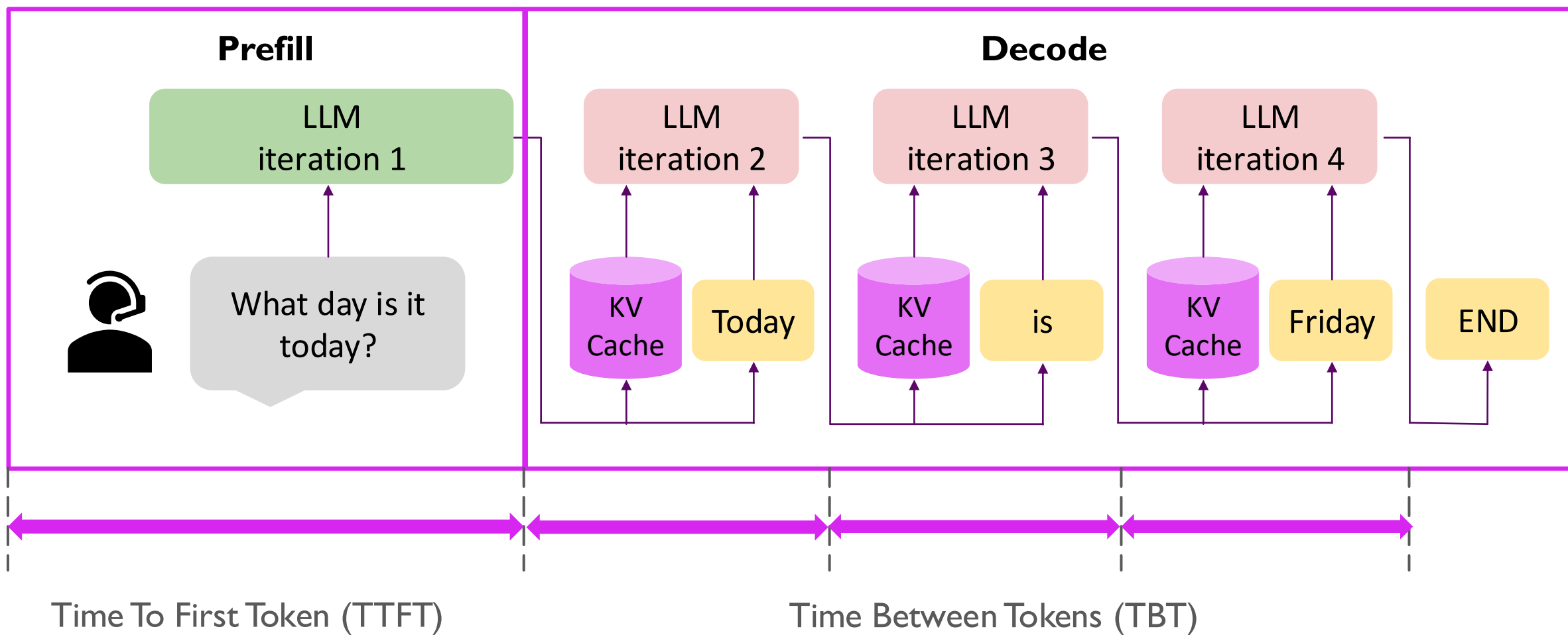
Bandwidth/\$

Capacity/\$

# I Motivation for Heterogeneous LLM Serving



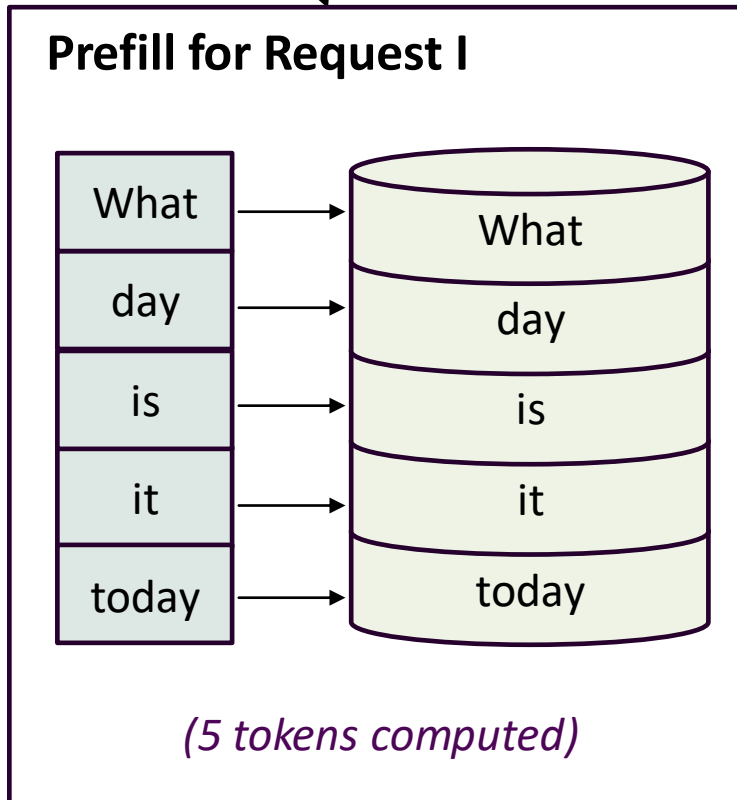
# LLM Inference



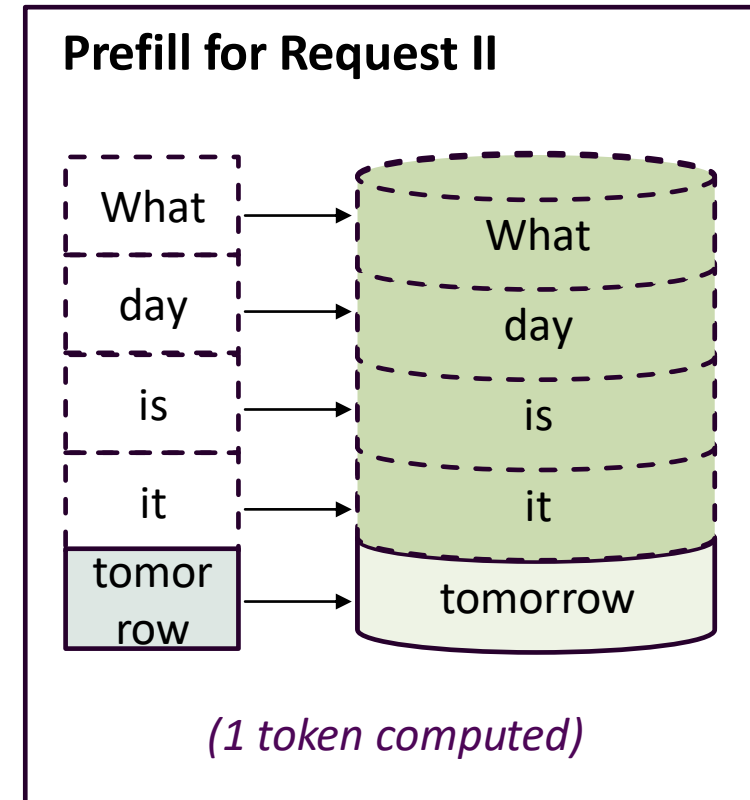
# LLM Inference: Prefix Caching

- KVCache can be shared across requests with the same prefix, reducing computation

“What day is it today”



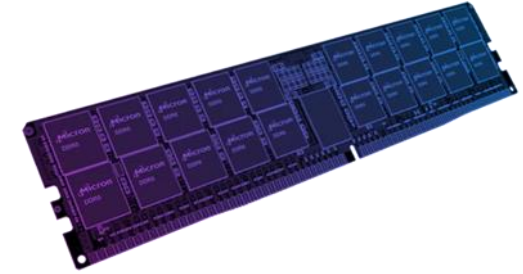
“What day is it tomorrow”



*KVCache Reuse*



# Different Hardware are Good at Different Dimension



Hardware  
Spec

H800

80GB VRAM, 3.3 TBps  
~ 1 PFLOPS  
> \$ 10,000

**For Prefill!**

Best  
for

Allround,  
especially for TFLOPS/\$

H20

96GB VRAM, 4 TBps  
~ 200 TFLOPS  
~ \$50,000

**For Decode!**

Bandwidth/\$

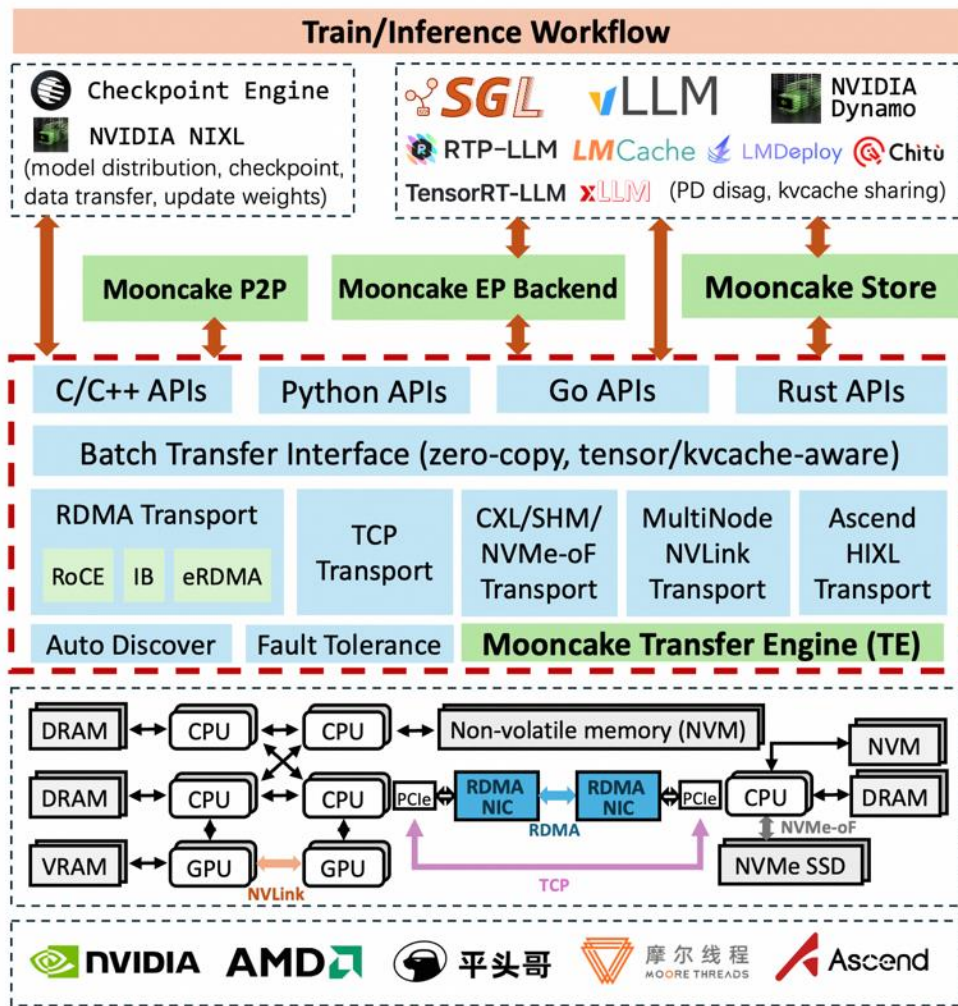
Xeon SPR + 8 \* DDR5-4800

8\*64GB DRAM, 8\*40GB/s  
< 20 TFLOPS  
~ ¥60,000

**For KVCache!**

Capacity/\$

!!! The price numbers are not accurate, just a demonstration!



A KVCache-centric Disaggregated Architecture for LLM Serving

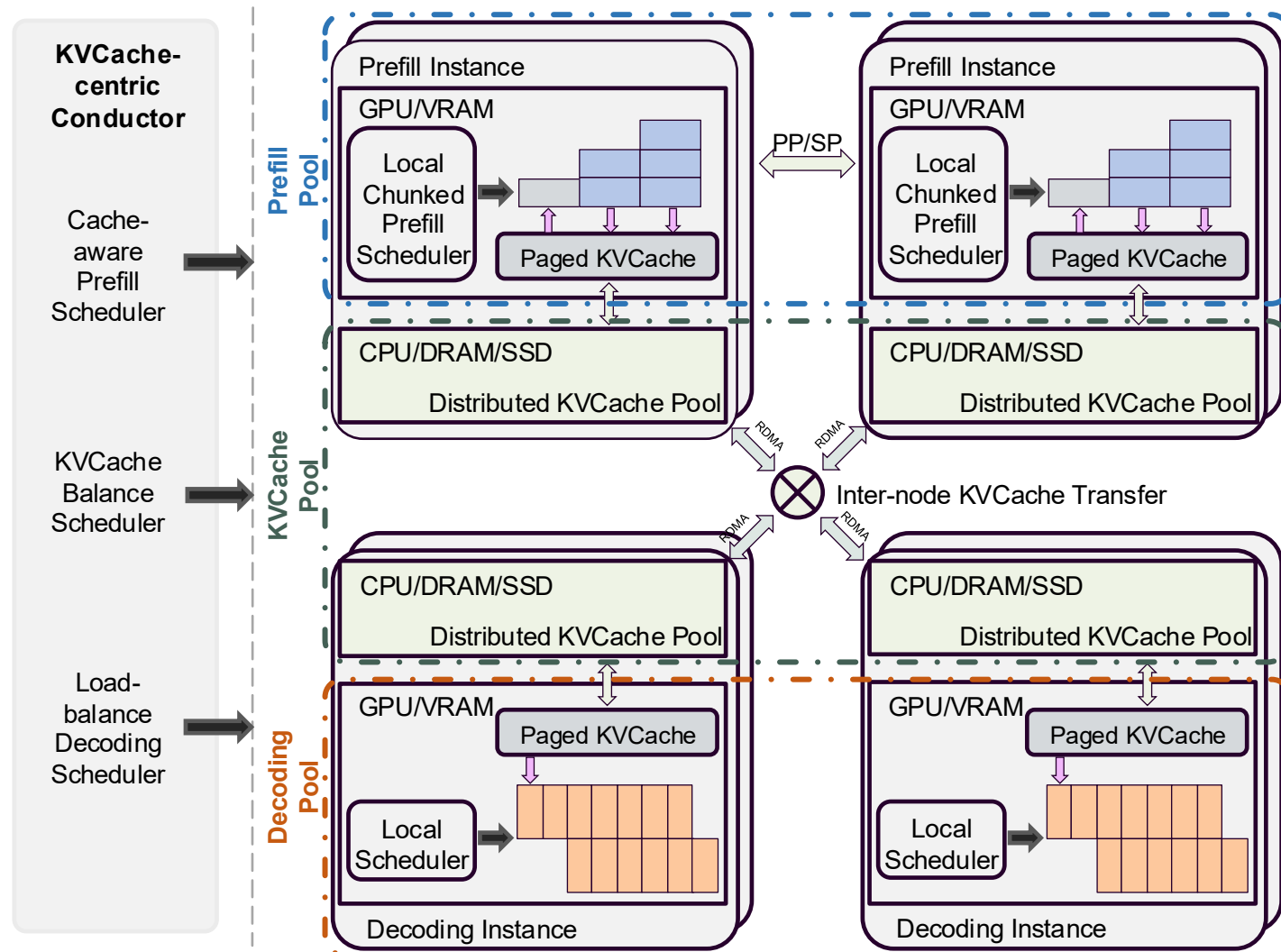
## 2 Core Technologies of Mooncake



# Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving



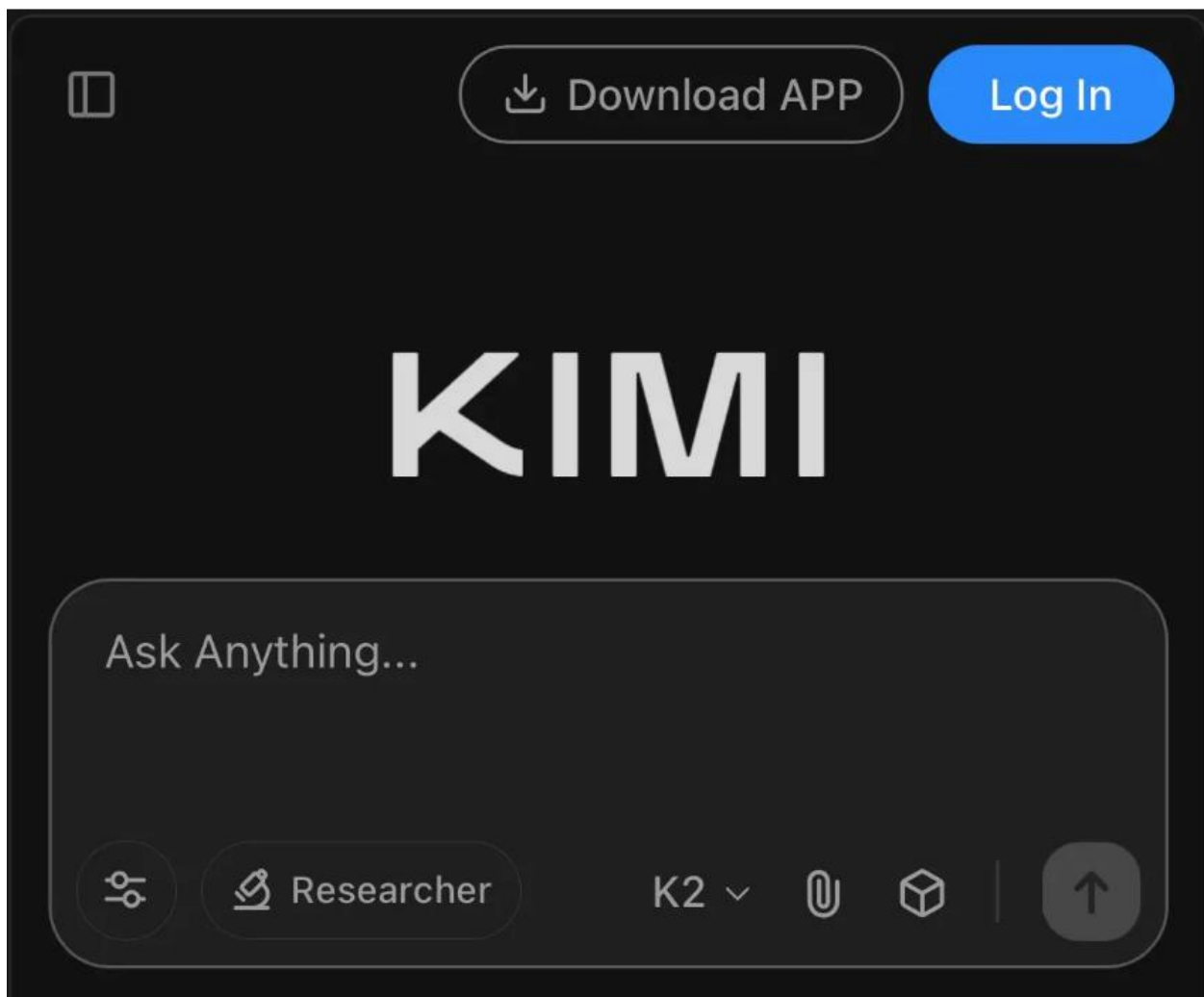
- **K** The serving platform of Kimi
  1. P/D disaggregation architecture centered around the distributed KVCache pool
  2. Trading more storage of less compute! Increase the throughput of Kimi by 75%
  3. Meet SLO guarantee



Moonshot AI + KVCache.AI @ Tsinghua

More: <https://github.com/kvcache-ai/Mooncake>

Mooncake (I): 在月之暗面做月饼，Kimi 以 KVCache 为中心的分离式推理架构

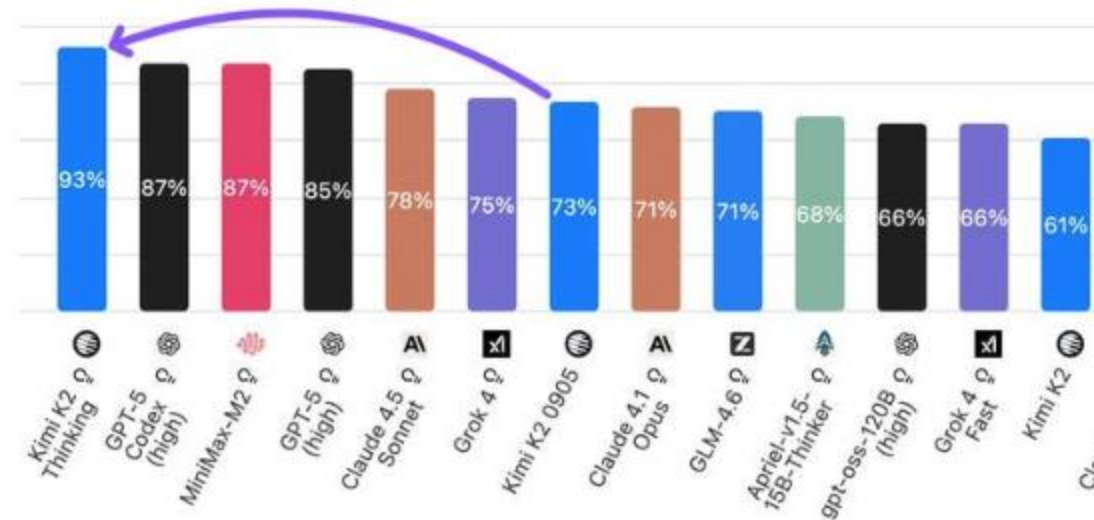


**n** Nature **nature**  
<https://www.nature.com/news> · 翻译此页

## 'Another DeepSeek moment': Chinese AI model Kimi K2

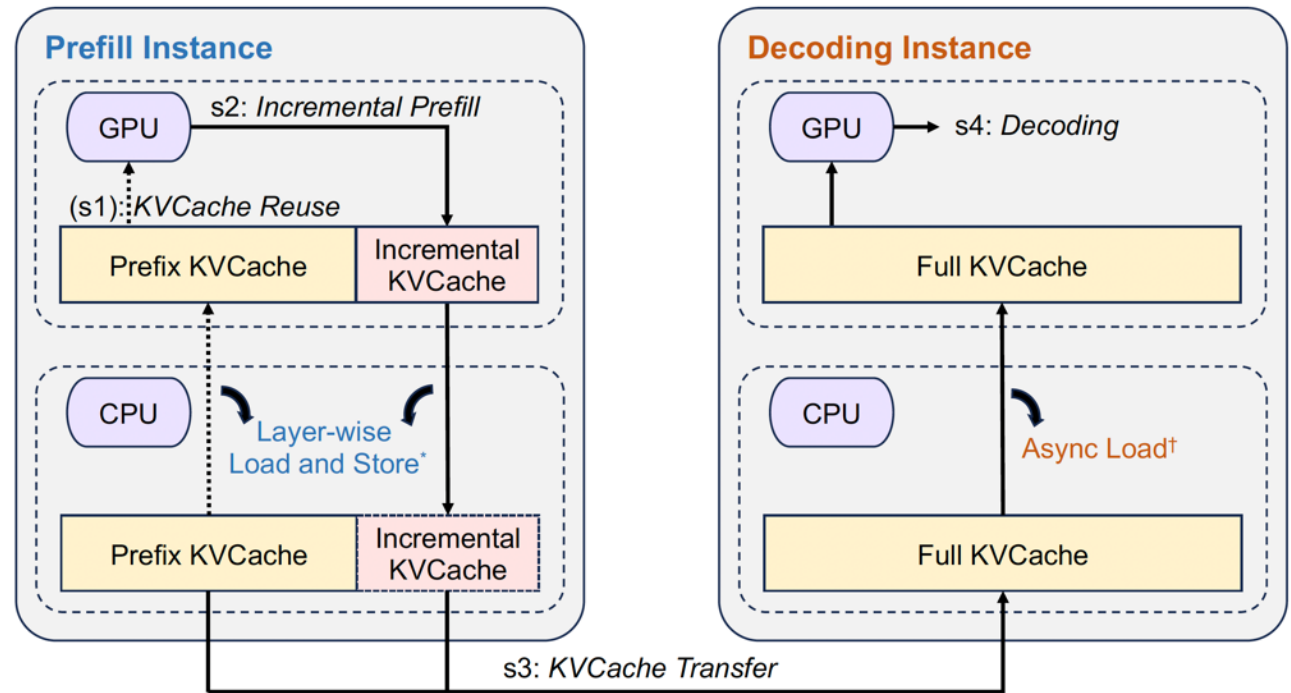
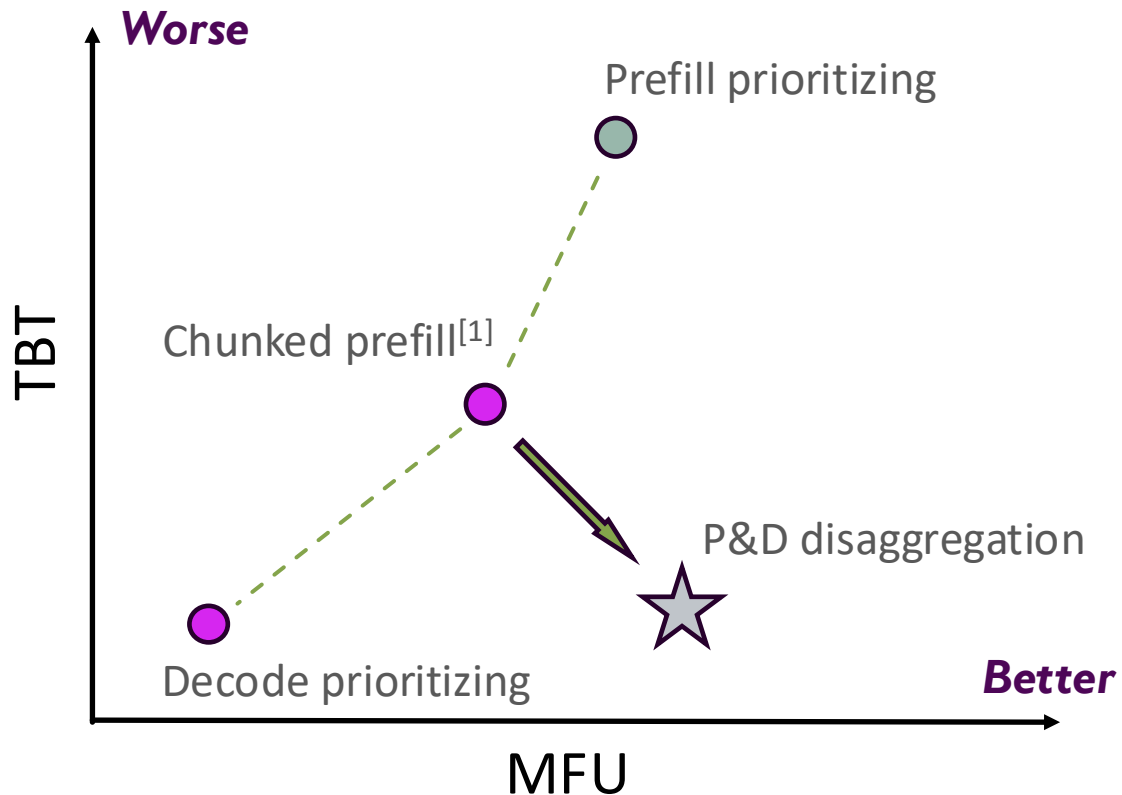
作者: E Gibney · 2025 — As with DeepSeek's models, **Kimi K2 is open-weight**, it downloaded and built on by researchers for free. It can be accessed through ...

$\tau^2$ -Bench Telecom (Agentic Tool Use)



# P&D Disaggregated Inference

- Avoid interference between prefill and decoding in a mixed batch
- Decouple resources and parallelism to improve MFU (Model Flops Utilization)

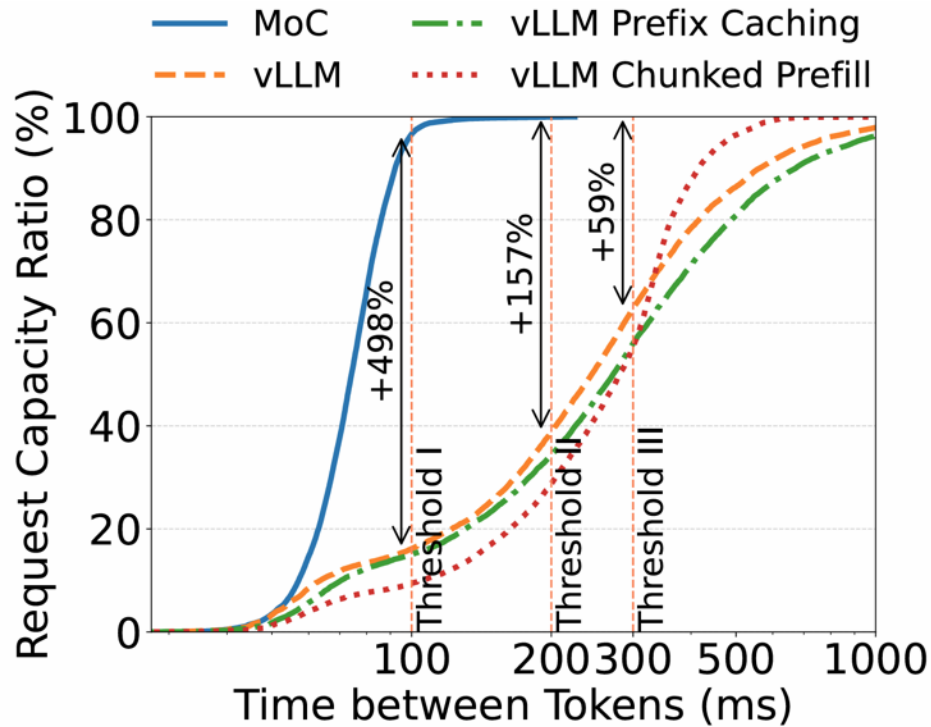


[1] Agrawal et al. Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve (OSDI'24)

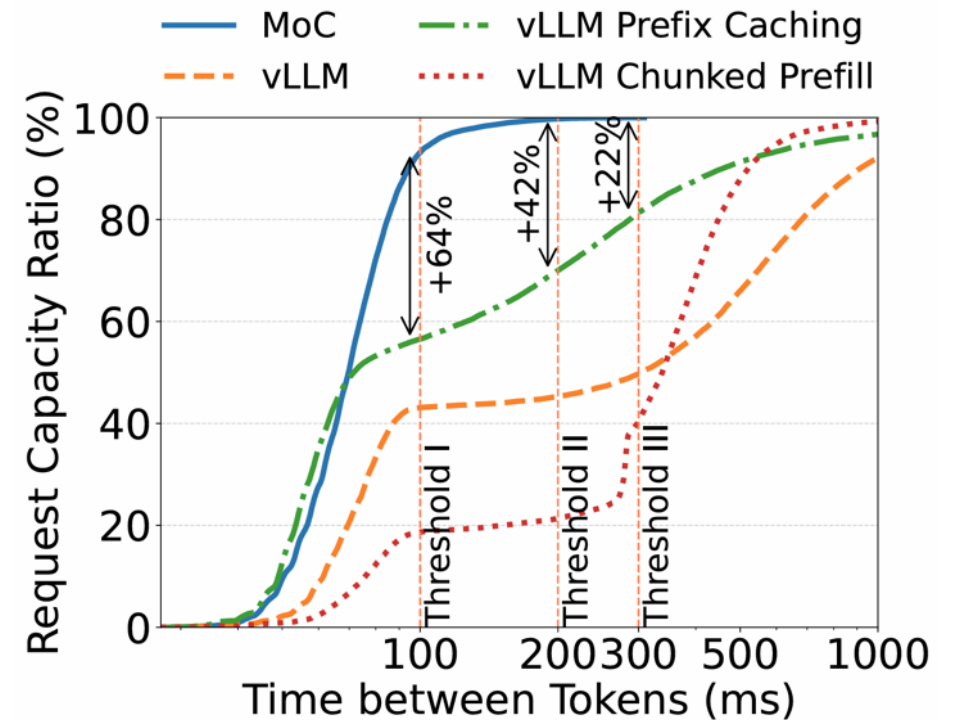
# P&D Disaggregation V.S. Chunked Prefill



(a)  
Conversa  
tion



(b)  
Tool  
Use

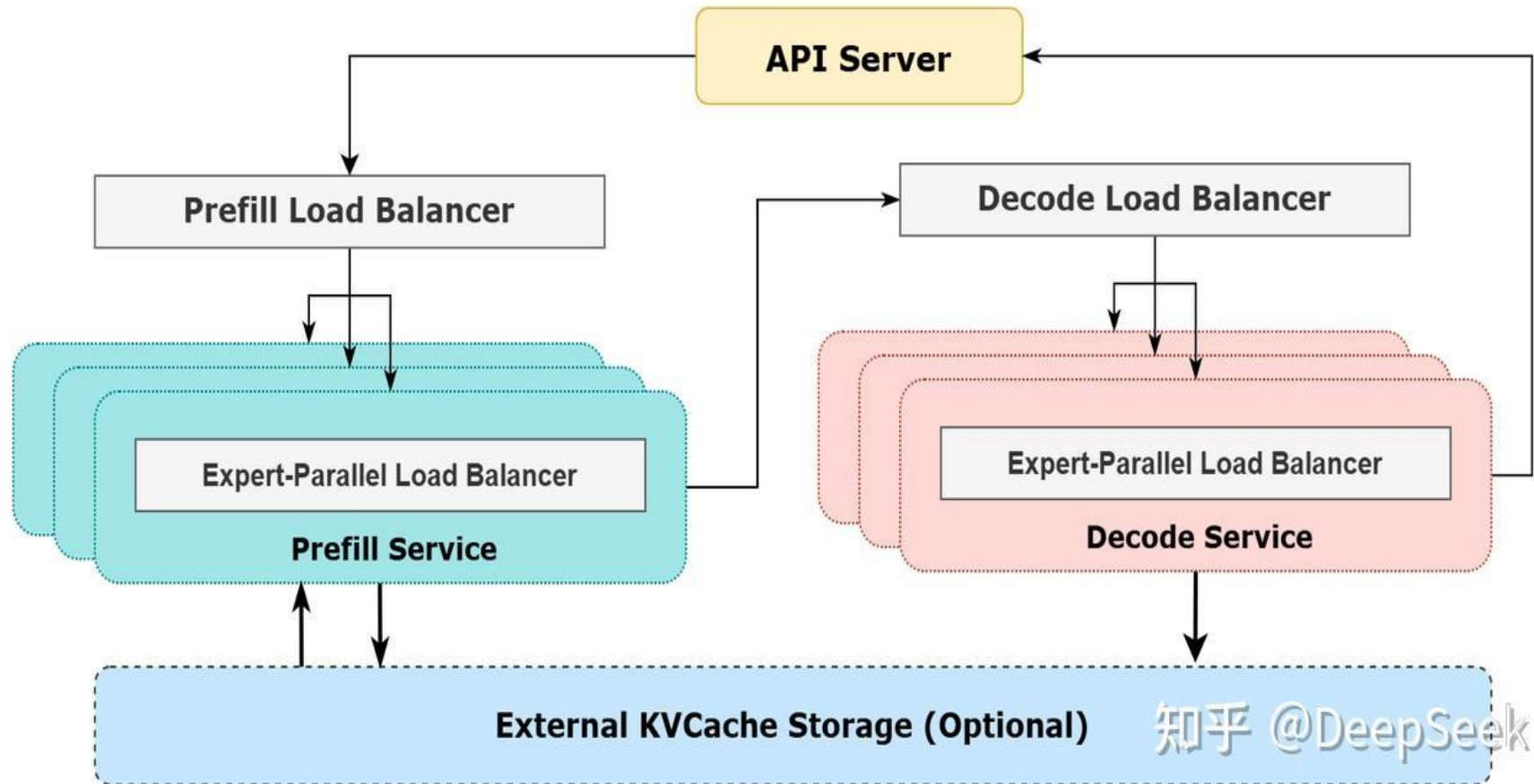


■ Better SLO control



# P&D Disaggregation Becomes a Necessary

- Prefill and decode needs different parallelism strategy, e.g., DeepSeek V3/R1



# KVCache Cache introduces High Challenges to Storage System

- Each 1 token  $\rightarrow$  2 \* layers \* hidden dimension = tens of KB KVCache
- Not only the size of KVCache is large, it also requires high transfer bandwidth to avoid stall of GPU

10B+ Model  
(GB)



100B+ Model  
(Hundreds of GB)



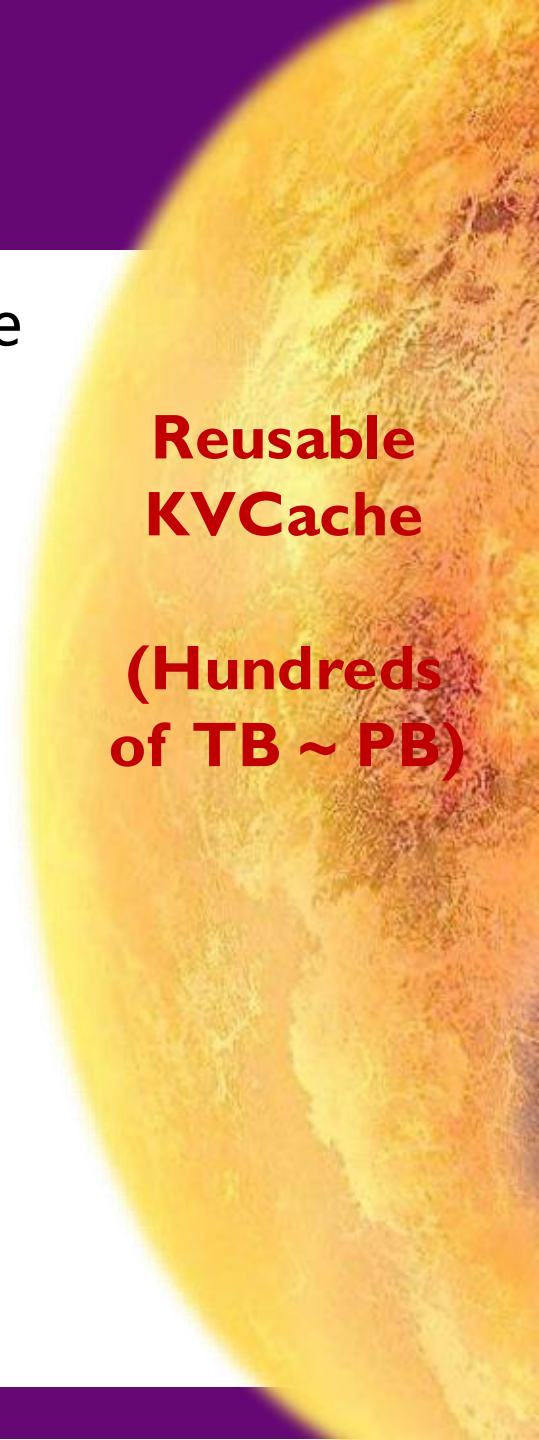
TB Model  
(TB)



**KVCache of  
TB Model  
(tens of TB)**

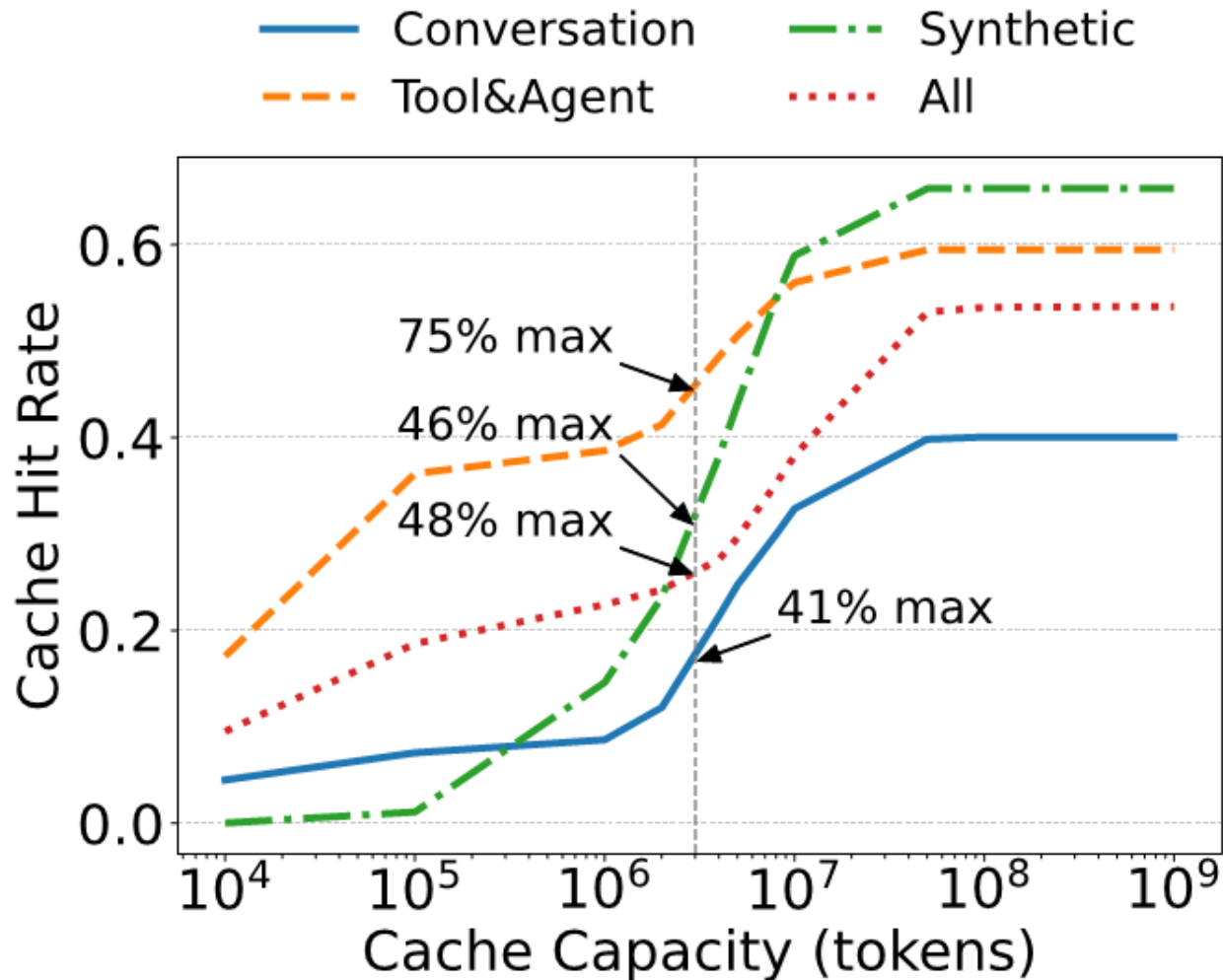


**Reusable  
KVCache  
(Hundreds  
of TB ~ PB)**





# KVCache Cache: Local or Global

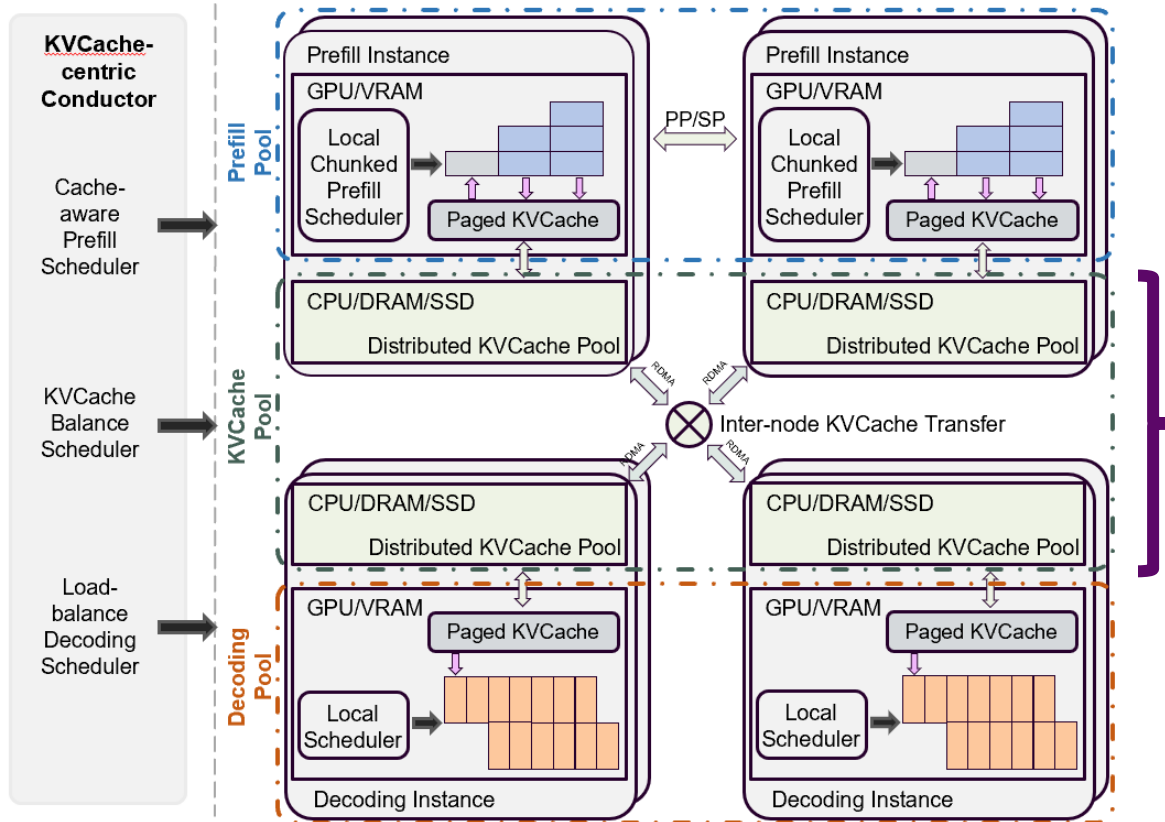


- Cache hit ratio grows proportional to the size of the cache
- Different scenarios has different settings
- Overall, we need PB-level cache that exceeds to size of a single machine



# Mooncake Store: Distributed Multi-layer KVCache Cache

- Key of KVCache Cache: Large size and bandwidth  
Utilize high performance connection like (GPUDirect) RDMA/Storage

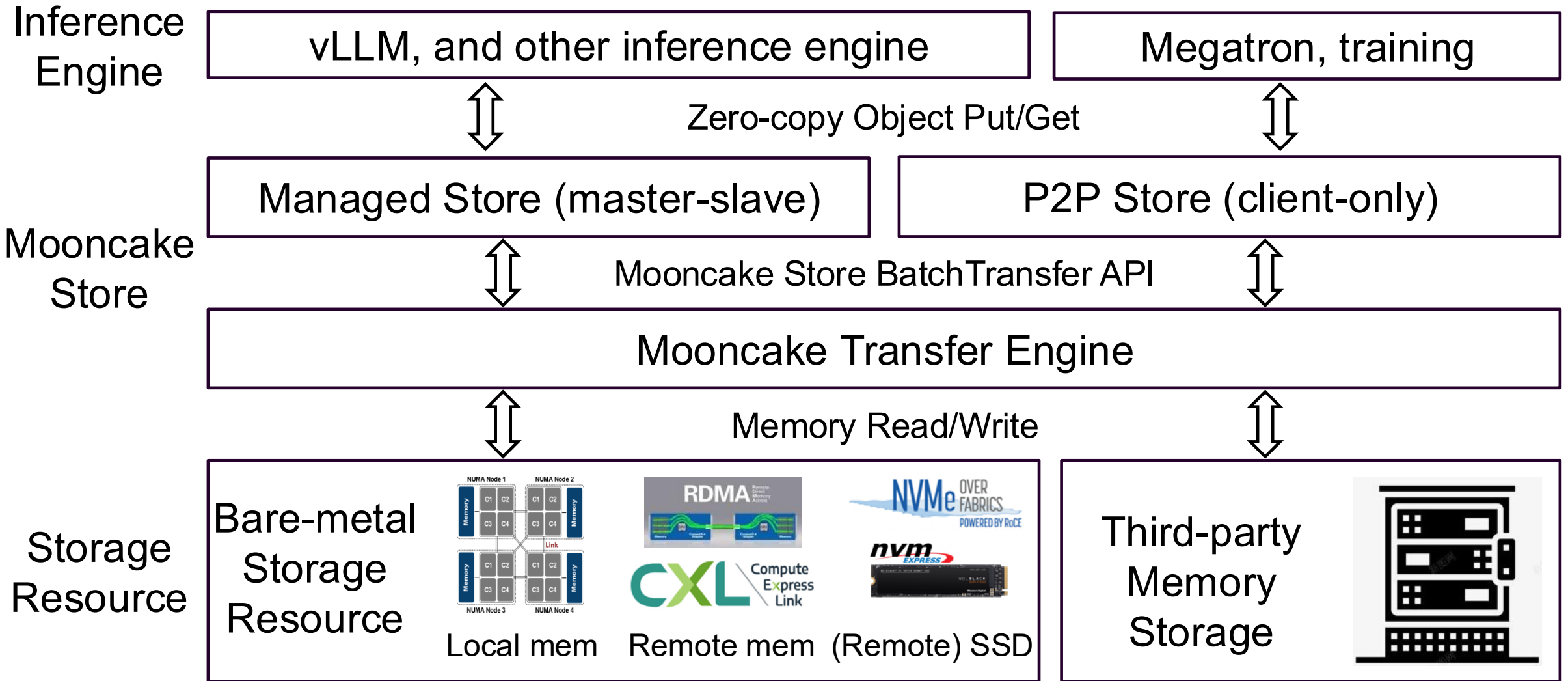


Will be open sourced soon!

- Pooled memory as KVCache cache
- Independent to specific inference engine
- Optimized for multi-NIC scenario

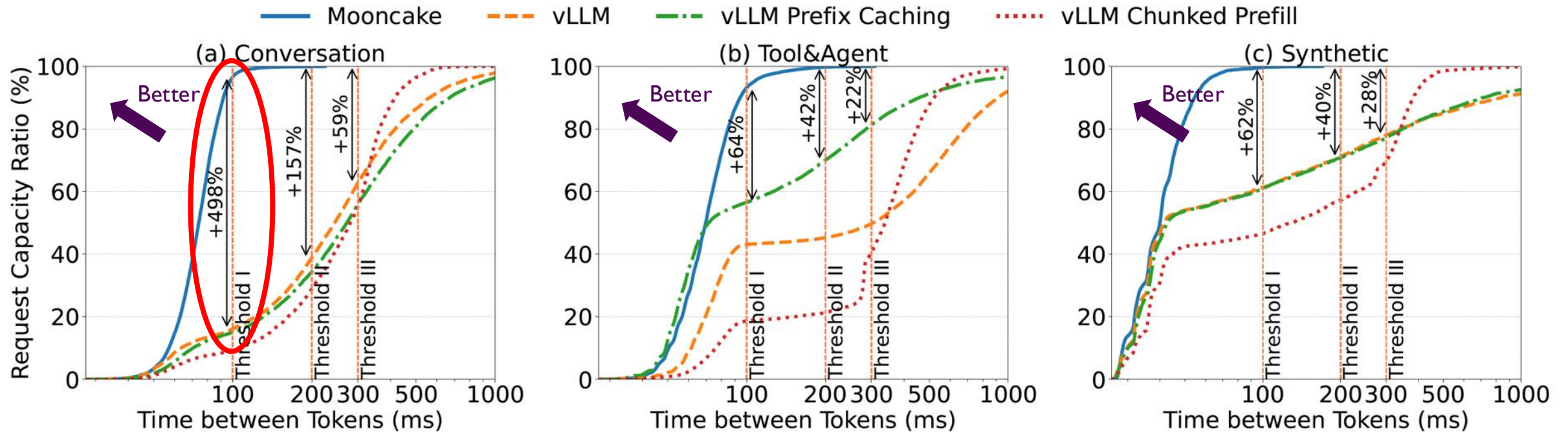


# Mooncake Store: External Integration



# Evaluation: Effective Request Capacity

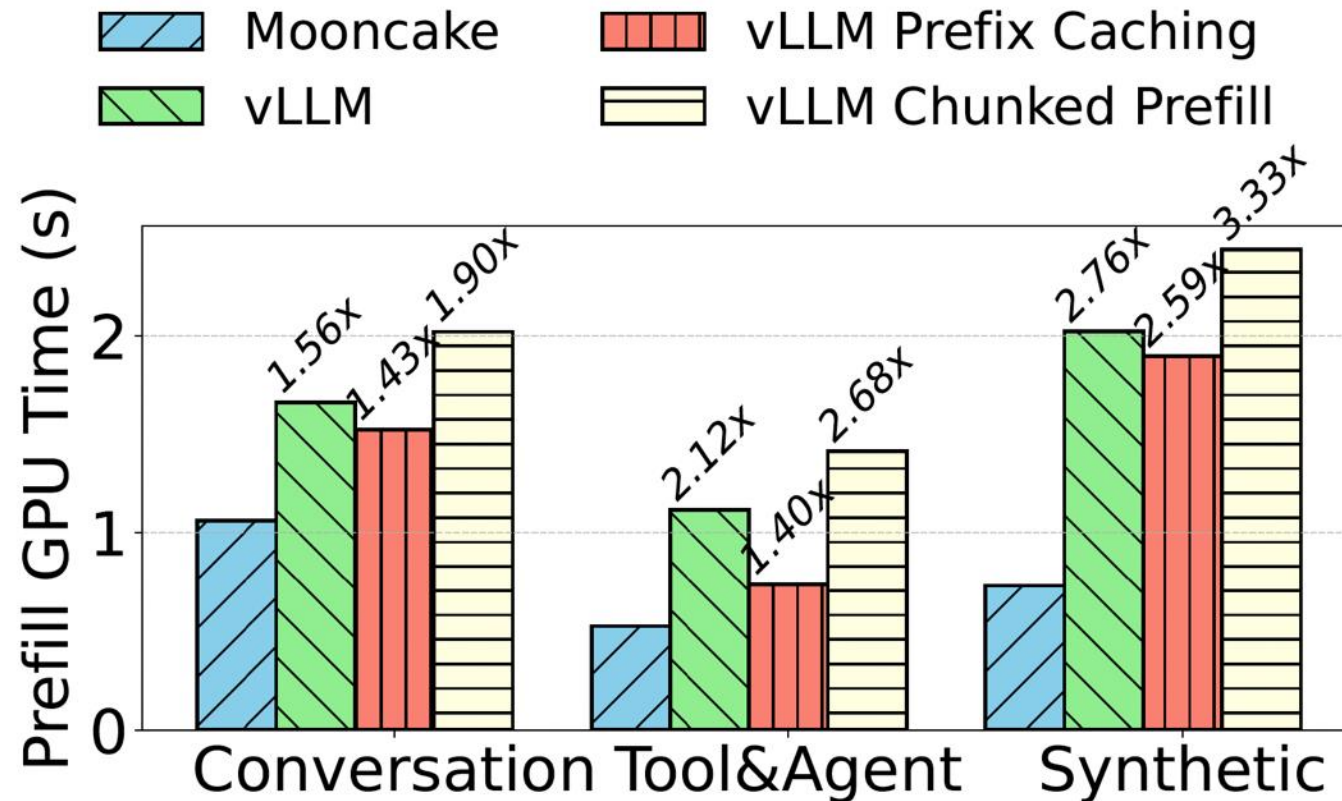
- Effective request capacity: Number of requests that meet the latency requirements
- Achieve up to a **498%** increase in effective request capacity compared to vLLM, vLLM with prefix caching and with chunked prefill





# Evaluation: GPU Computation Cost

- Cache hit rate: global cache > local cache
- Save **29% - 61%** on GPU computation costs



# Mooncake – Open Sourced and Build with the Community



From flagship applications

To industry-wide adoption

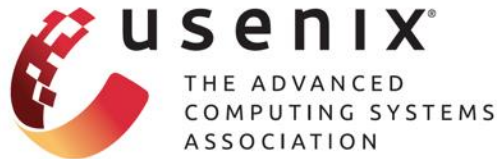
2024.3 Kimi went viral for its long-context capabilities, using Mooncake to handle surging traffic

2024.11 Mooncake open-sourced; adopted by Alibaba and Ant Finance

Used in Dynamo, the distributed inference system highlighted at GTC 2025 Keynote

2024.6 Mooncake tech report sparked wide industry discussion

2025.2 USENIX FAST Best Paper Award



ERIK RIEDEL BEST PAPER AWARD

presented to

Ruoyu Qin, Zheming Li, Weiran He, Jialei Cui, Feng Ren, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu  
for

Mooncake: Trading More Storage for Less Computation — A KVCache-centric Architecture for Serving LLM Chatbot

Presented at the 23rd USENIX Conference on File and Storage Technologies

Amy Riehl  
President

2/25/2025  
Date

USENIX FAST2025 Best Paper



[kvcache-ai/Mooncake](https://kvcache-ai/Mooncake) -- An open-source initiative co-launched by Moonshot AI and Tsinghua University, with collaboration from various large model and infrastructure providers





# Mooncake – Adopted/Collaborated with Other Famous Communities

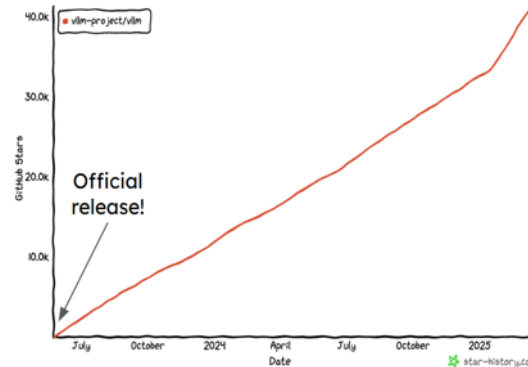


- One of the most widely used inference engines, adopted by major cloud providers
- Its distributed inference is built on Mooncake

v0.8.3 Latest

github-actions released this 2 days ago · 57 commits to main since this release · v0.8.3 · 296c657

41.5K Stars  
Star History

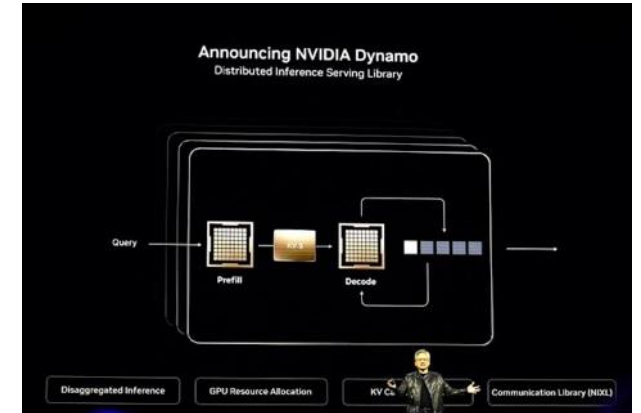


## Cluster Scale Serving

- Support XpYd disaggregated prefill with [MooncakeStore](#) (#12957)

## NVIDIA Dynamo

- Spotlited by Jensen Huang at GTC 2025 Keynote
- Its architecture is inspired by Mooncake, with explicit acknowledgments



## Acknowledgement

We would like to acknowledge several open source software stacks for motivating us to create Dynamo.

- vLLM and vLLM-project
- SGLang
- DistServe
- Mooncake
- *Memory bottlenecks:* Large-scale inference workloads demand extensive capacity. KV cache offloading across memory hierarchies (HBM, DDR, N memory limits and speeds up latency. ([Mooncake](#), [AlBrix](#), [LMCache](#))



- Inference engine of xAI, widely used in DeepSeek inference
- Distributed architecture was co-developed with Mooncake



The SGLang Team is honored to announce that the following well-known companies and teams, among others, have adopted SGLang for running DeepSeek V3 and R1. @AMD @nvidia @Azure @basetenco @novita\_ai\_labs @BytedanceTalk @DataCrunch\_jo @hyperbolic\_labs @Vultr @runPod



SGLang has achieved a milestone with full support for PD Disaggregation, thanks to the [MoonCake team](#). Teng Ma, Shangming Cai, Xuchun Shang and Yuan Luo were instrumental in this achievement. Special thanks to Atlas Cloud for their support with the H100s cluster. Let's go! 🚀



# Key to KVCache

- Mooncake Transfer Engine
- End-to-end zero-copy

⚡ Transfer Fast



High-performance distributed  
KV cache storage

📖 Store More

✅ Easy to Use

- Elastic, Shared, and Multi-layer KV Cache
- Memory Allocator Optimized for LLM Inference

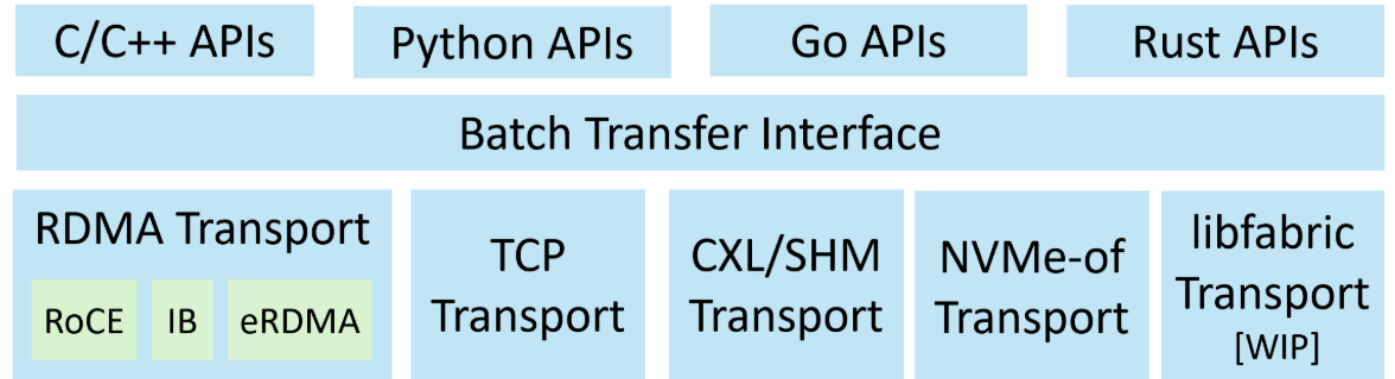
- Extensive and user-friendly APIs



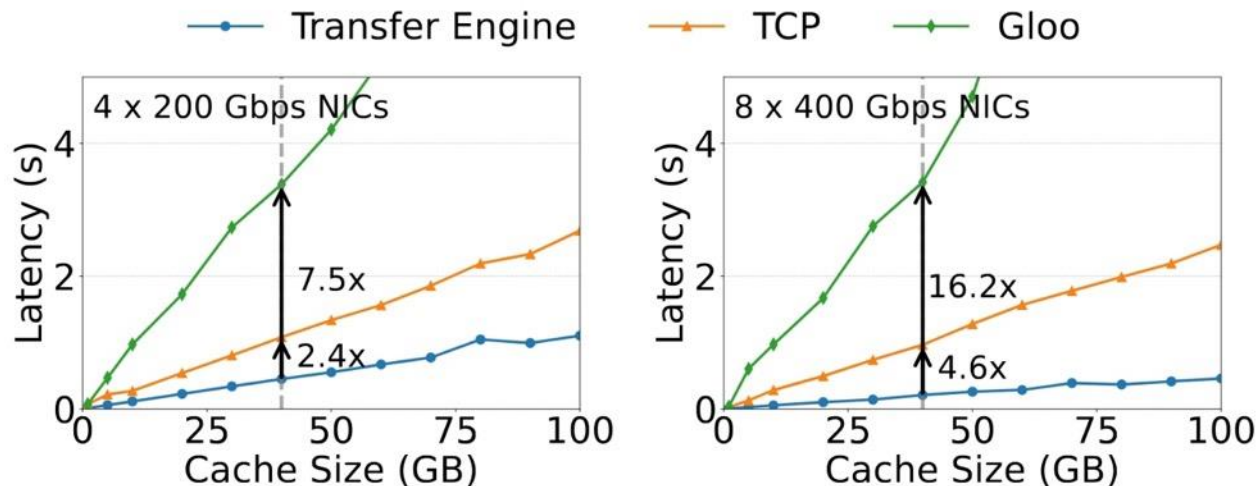
# Transfer Fast: Mooncake Transfer Engine

- **Key features**

- **Topology-aware path selection**
- **Multi-NIC pooling**
- **Supports multiple protocols and provides unified interfaces.**
- **Multi-language APIs**



Mooncake Transfer Engine



## Lightening fast over RDMA

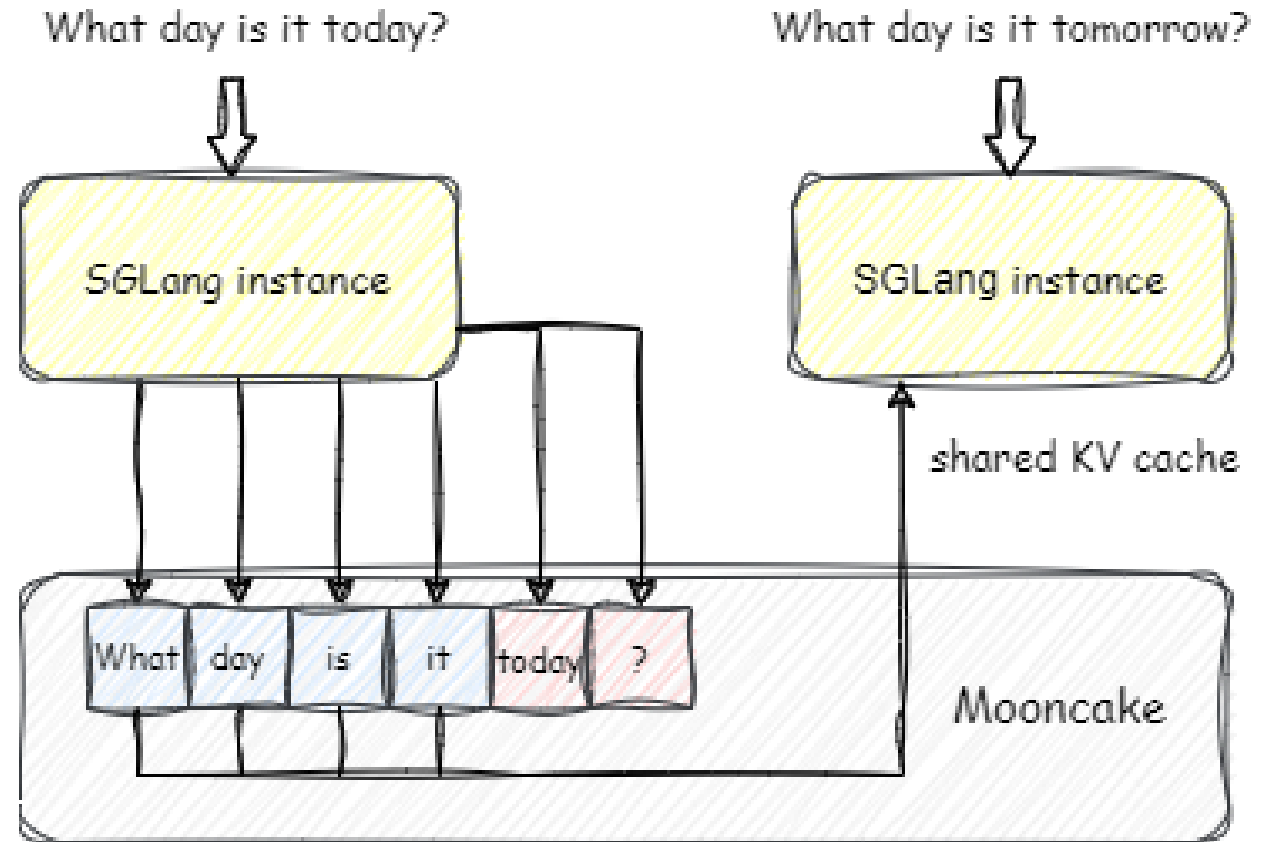
- 40 GB KVCache (128k tokens, LLaMA3-70B)
- **87 GB/s @ 4x200 Gbps**, RoCE
- **190 GB/s @ 8x400 Gbps**, RoCE



# Store More: Elastic Shared Multi-layer KV Cache

## Key features

- **Distributed KV cache sharing:** storing one and usable by all
- **Dynamic resource scaling:** dynamically adding and removing store nodes (startup in <80s for 500GB memory and 8 RDMA NICs)
- **Multi-layer storage (WIP):** offloading cached data from RAM to SSD





# Extensive APIs, Easy to Use

## Put/Get APIs

- Put/Get single object
- Batch Put/Get
- **(Batch) Zero-copy Put/Get: recommended**
- (Batch and zero-copy) Put/Get from/into multi-parts

## Configurable KV cache placement

- Replica number
- With soft pin
- Preferred segment

## Hello world example

```
from mooncake.store import MooncakeDistributedStore

# 1. Create store instance
store = MooncakeDistributedStore()

# 2. Setup with all required parameters
store.setup(
    "localhost",          # Your node's address
    "http://localhost:8080/metadata", # HTTP metadata server
    512*1024*1024,       # 512MB segment size
    128*1024*1024,      # 128MB local buffer
    "tcp",               # Use TCP (RDMA for
    "",                  # Leave empty; Mooncake
    "localhost:50051"    # Master service
)

# 3. Store data
store.put("hello_key", b"Hello, Mooncake Store!")

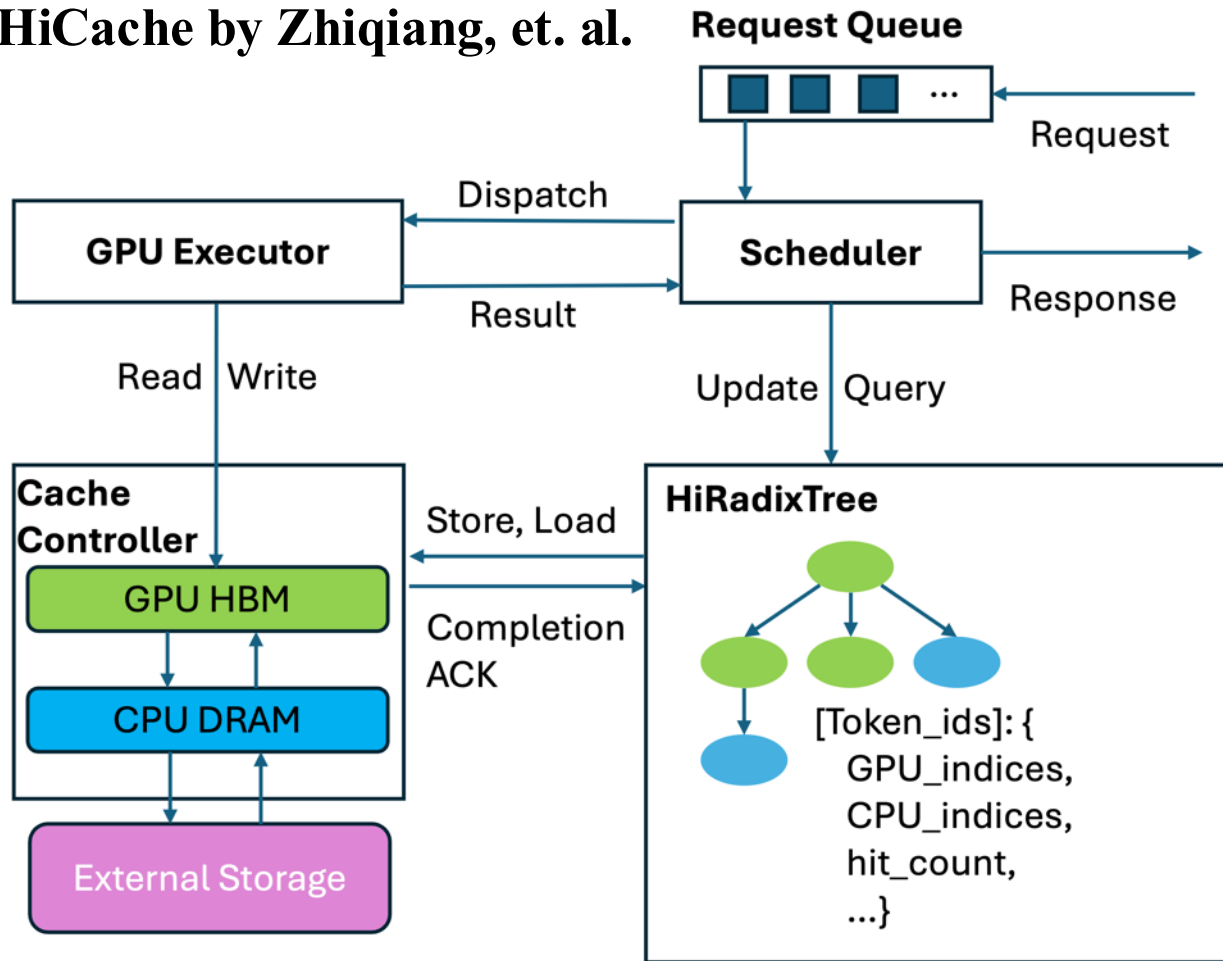
# 4. Retrieve data
data = store.get("hello_key")
print(data.decode()) # Output: Hello, Mooncake Store!

# 5. Clean up
store.close()
```

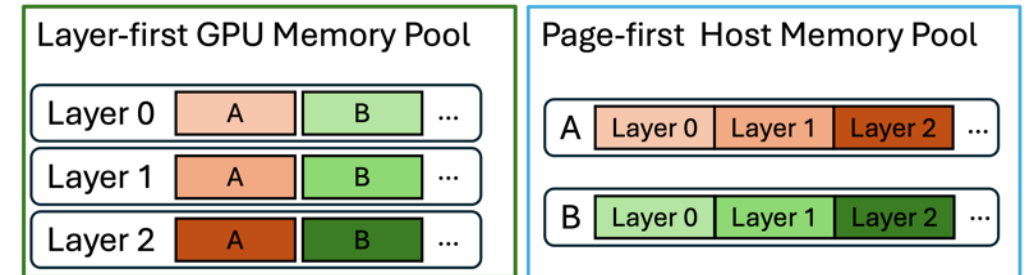


# How to Integrate with SGLang? - HiCache

HiCache by Zhiqiang, et. al.



Page First Layout

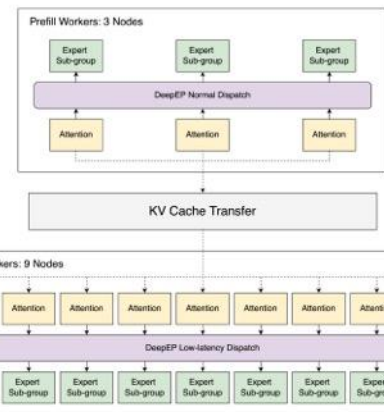


# SGLang + Mooncake



Breaking: SGLang provides the first open-source implementation to serve @deepseek\_ai V3/R1 models with large-scale expert parallelism and prefill-decode disaggregation on 96 GPUs.

It nearly matches the throughput reported by the official DeepSeek blog, achieving 52.3K input tokens per second and 22.3K output tokens per second per node. This optimized strategy improves output throughput by up to 5x compared to vanilla tensor parallelism.



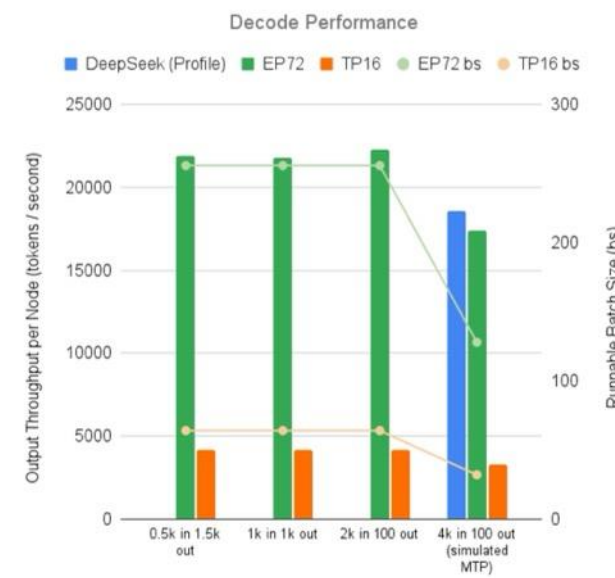
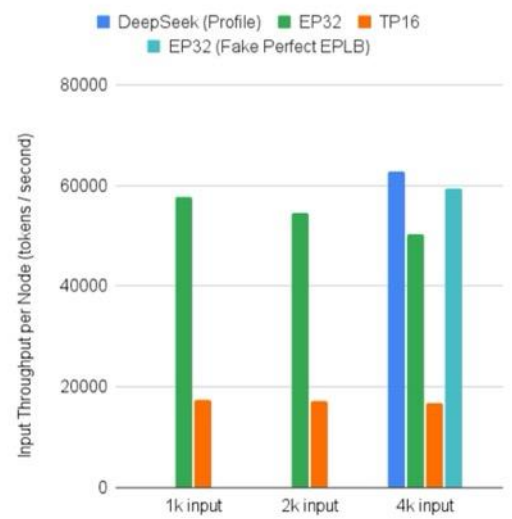
- 3 Prefill + 9 Decode
- DeepEP + EPLB
- Double Batch Overlap

## Acknowledgment

We would like to express our heartfelt gratitude to the following teams and collaborators:

- **SGLang Core Team and Community Contributors** — Jingyi Chen, Cheng Wan, Liangsheng Yin, Baizhou Zhang, Ke Bao, Jiexin Liang, Xiaoyu Zhang, Yanbo Yang, Fan Yin, Chao Wang, Laixin Xie, Runkai Tao, Yuhong Guo, Kaihong Zhang, Lei Yu, Yu-Hsuan Tseng, Qilin Tian, Peng Zhang, Yi Zhang, Yineng Zhang, Byron Hsu, and many others.
- **Atlas Cloud Team** — Jerry Tang, Wei Xu, Simon Xue, Harry He, Eva Ma, and colleagues — for providing a 96-device NVIDIA H100 cluster and offering responsive engineering support.
- **NVIDIA Solution Architect Team** — Xuting Zhou, Jinyan Chen, and colleagues — for their work on the seamless integration of expert parallelism.
- **NVIDIA Enterprise Product Team** — Trevor Morris, Elfie Guo, Kaixi Hou, Kushan Ahmadian, and colleagues — for optimizing the DeepSeek R1 kernels.
- **LinkedIn Team** — Biao He, Qingquan Song, Chunan Zeng, Yun Dai, Yubo Wang, and colleagues — for optimizing the Flash-Attention 3 backend.
- **Mooncake Team** — Shangming Cai, Teng Ma, Mingxing Zhang, and colleagues — for their collaboration on PD disaggregation in SGLang.
- **FlashInfer Team** — Zihao Ye, Yong Wu, Yaxing Cai — for additional DeepSeek R1 kernel optimizations.
- **Dynamo Team** - Kyle Kranen, Vikram Sharma Mailthody, and colleagues - for extra support on PD disaggregation in SGLang.

Thank you all for your invaluable support and collaboration.



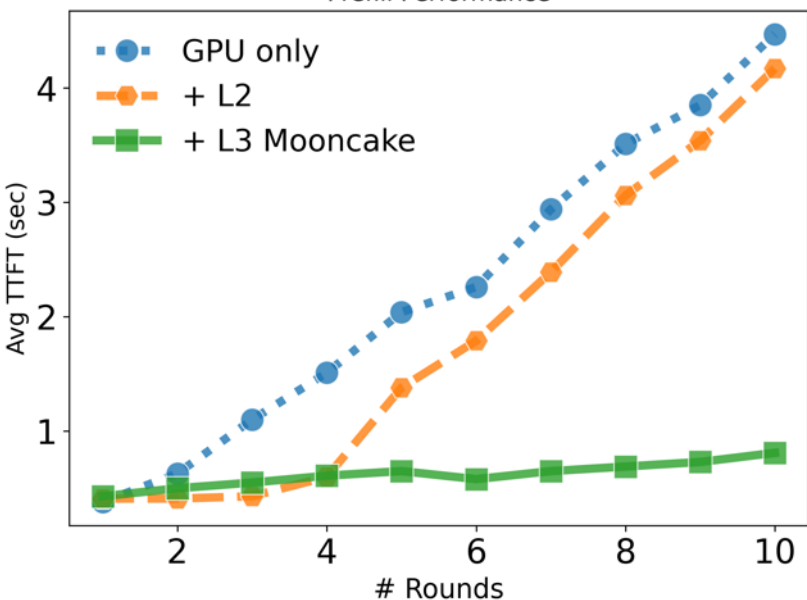
- 52.3k input tps
- 22.3k output tps
- per node

# SGLang + HiCache + Mooncake

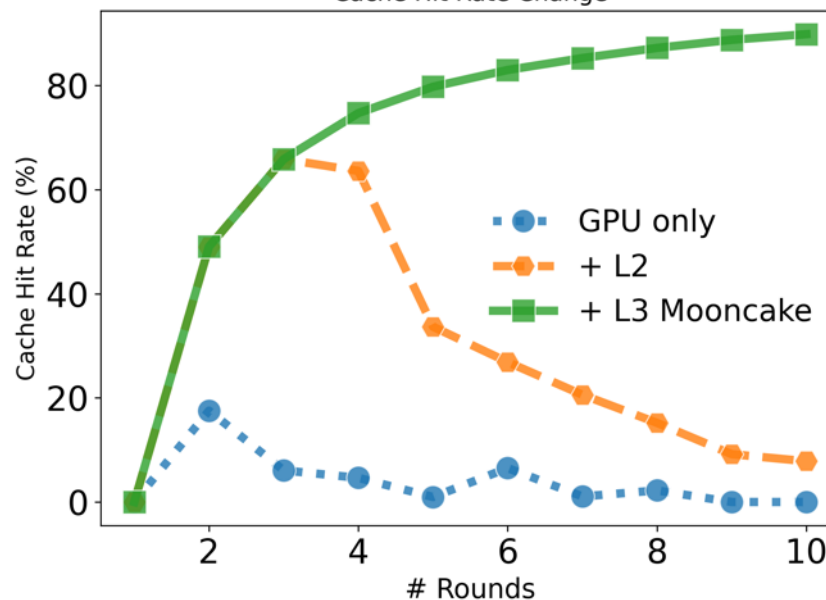


### SGLang HiCache with Mooncake Backend on Multi-turn Conversation Benchmark

#### Prefill Performance



#### Cache Hit Rate Change



Effective KV caching significantly reduces TTFT by eliminating redundant and costly re-computation.

**Integrating SGLang HiCache with the Mooncake service** enables scalable KV cache retention and high-performance access. In our evaluation, we tested the DeepSeek-R1-671B model under PD-disaggregated deployment using in-house online requests sampled from a **general QA scenario**. On average, **cache hits achieved an 84% reduction in TTFT compared to full re-computation**.

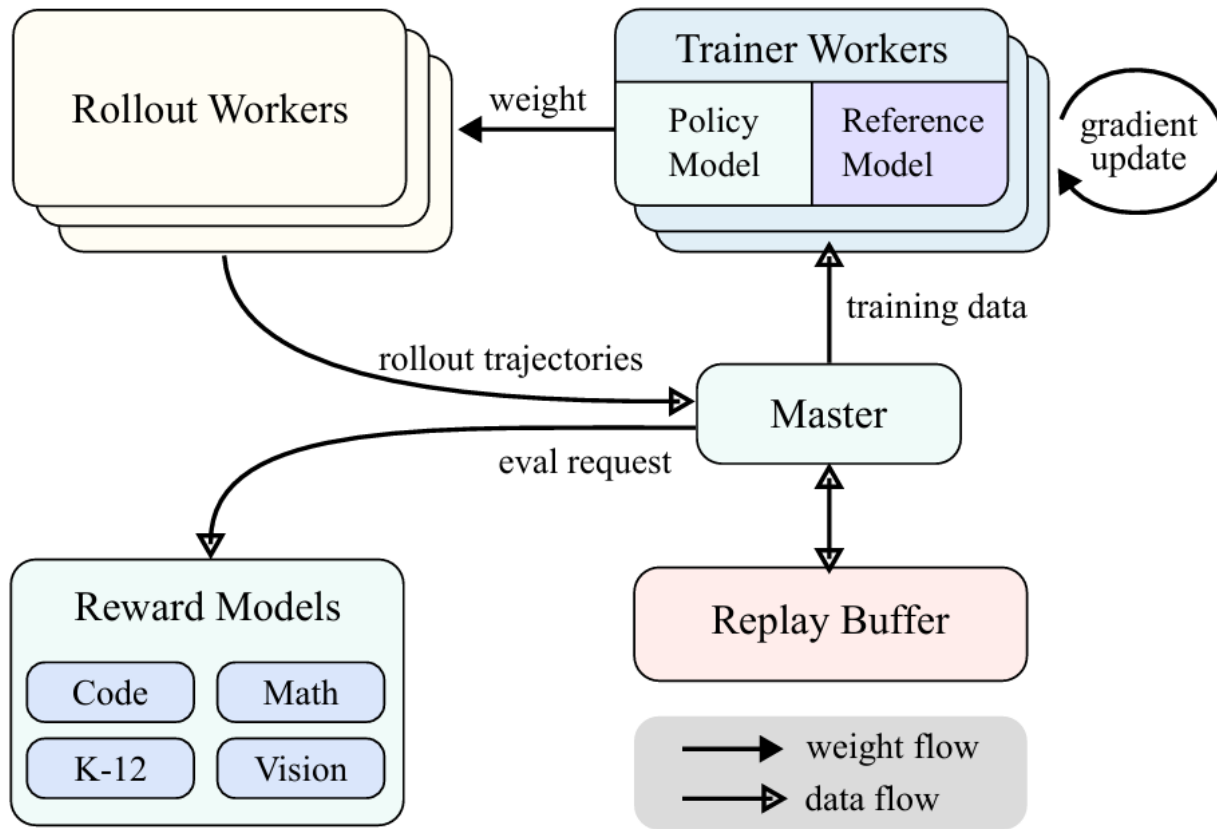
– Ant Group

Thanks:

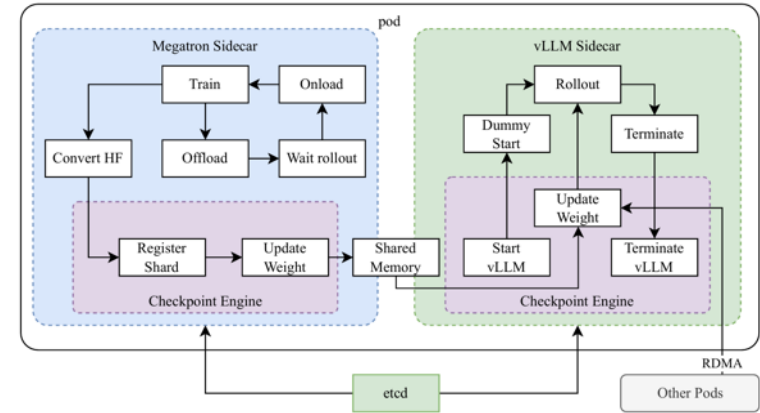


# Mooncake P2P Store: Faster Checkpoint/KVCache Restore in RL

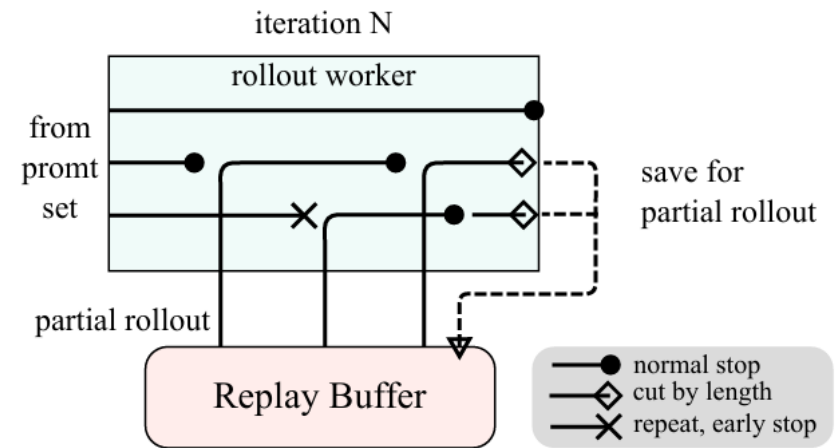
<https://github.com/MoonshotAI/checkpoint-engine/>



(a) System overview

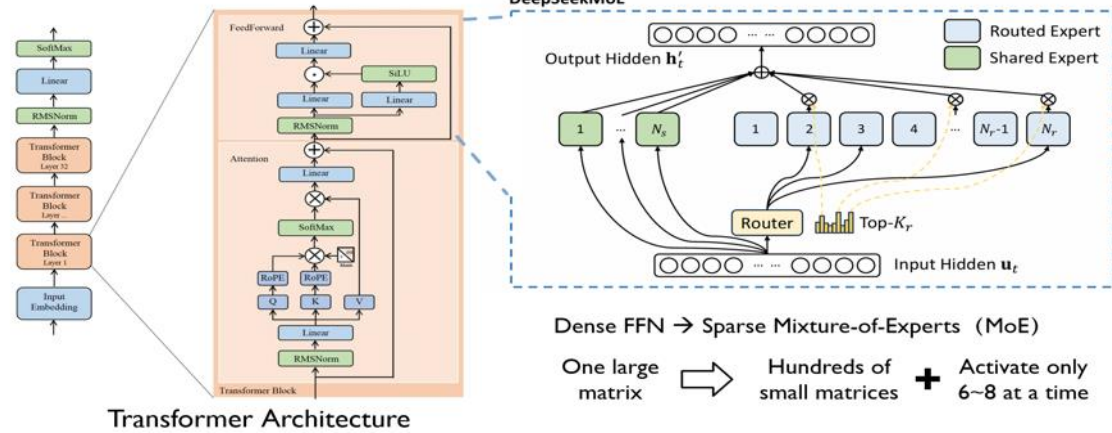


Fast Checkpoint Transfer

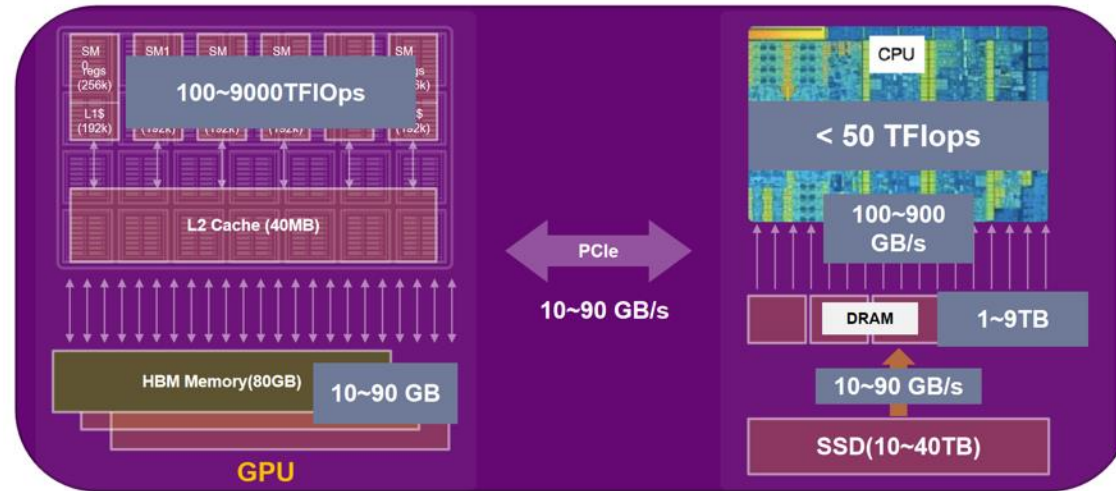


(b) Partial Rollout

# Attention + MoE



# GPU + CPU



# Background and Observation of LLM and Sparse Mixture-of-Experts (MoE)

# 3 Motivation for Local Heterogeneous LLM Serving

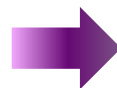


# Background: Sparsification Trends in LLMs



CompassBench LLM Leaderboard Official Closed Benchmark 24-07

Overall	Language	Knowledge	Reasoning	Math	Code	Instruct	Agent
Model	Release	Type	Parameters	Average			
1	Mistral-Large-Instruct-2...	2024/7/24	Chat	123B	62.5		
2	DeepSeek-V2-Chat-0628	2024/5/6	Chat	236B	61.7		
3	Qwen2-72B-Instruct	2024/6/6	Chat	72B	55.4		
4	Llama3.1-70B-Instruct	2024/7/23	Chat	70B	53.9		
5	Gemma-2-27B-it	2024/6/27	Chat	27B	53.5		
6	Qwen1.5-110B-Chat	2024/4/25	Chat	110B	51.9		
7	GLM-4-9B-Chat	2024/6/4	Chat	9B	47.9		
8	Yi-1.5-34B-Chat	2024/5/12	Chat	34B	46.9		
9	Mixtral-8x22B-Instruct-...	2024/4/17	Chat	141B	46.3		
10	Gemma-2-9B-it	2024/6/27	Chat	9B	45.5		



CompassBench LLM Leaderboard Official Closed Benchmark 25-07

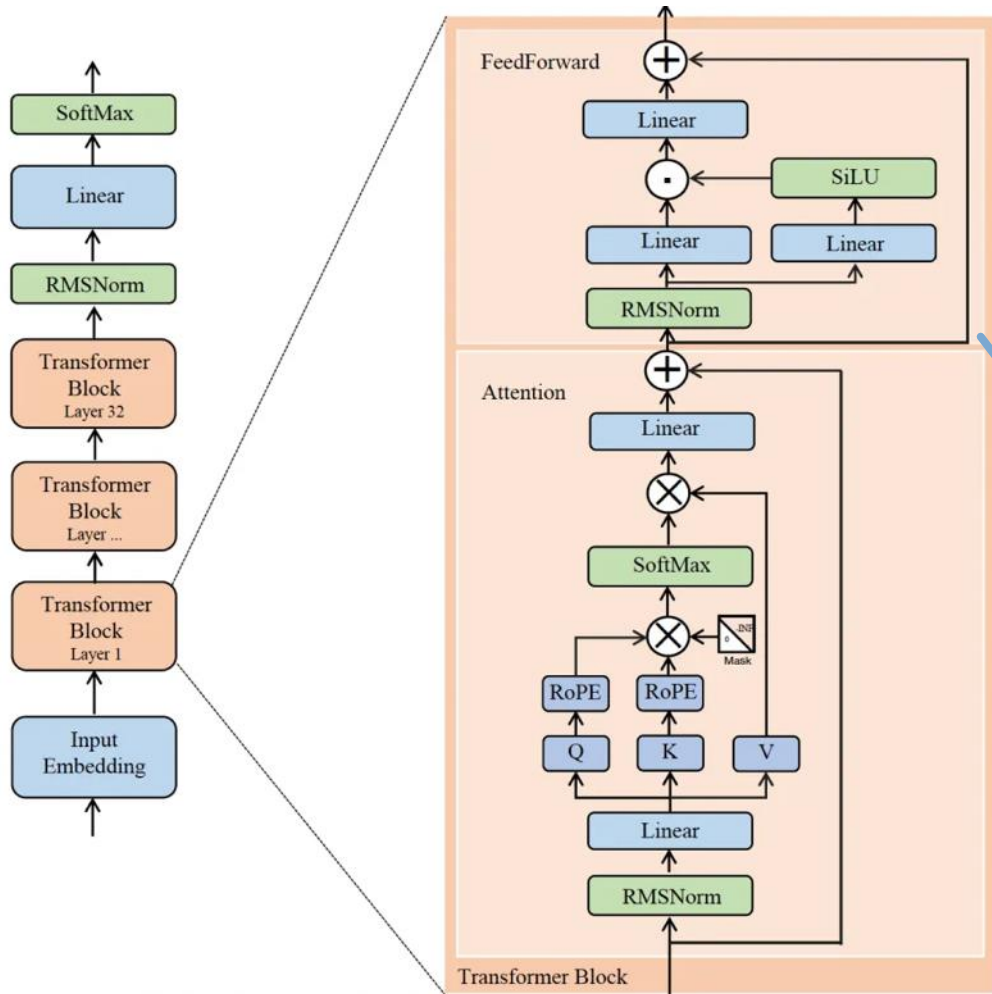
Overall	Language	Knowledge	Reasoning	Math	Code	Tool Using
Model	Release	Type	Parameters	Average		
1	Qwen3-235B-A22B-Thi...	2025/7/25	Chat	235B	63.8	
2	DeepSeek-R1-0528	2025/5/28	Chat	671B	63.2	
3	GLM-4.5	2025/7/29	Chat	358B	59.6	
4	Qwen3-235B-A22B-Inst...	2025/7/22	Chat	235B	57.6	
5	GLM-4.5-Air	2025/7/29	Chat	110B	56.8	
6	Kimi-K2-Instruct	2025/7/11	Chat	1000B	55.5	
7	MiniMax-M1-80k	2025/6/17	Chat	456B	55	
8	DeepSeek-V3-0324	2025/3/24	Chat	671B	54.4	
9	Hunyuan-A13B-Instruct	2025/6/27	Chat	80B	51.9	
10	ERNIE-4.5-Turbo-128K	2025/6/30	Chat	300B	49.4	

2 out of top 10 open-source models are MoE

All top 10 open-source models are MoE

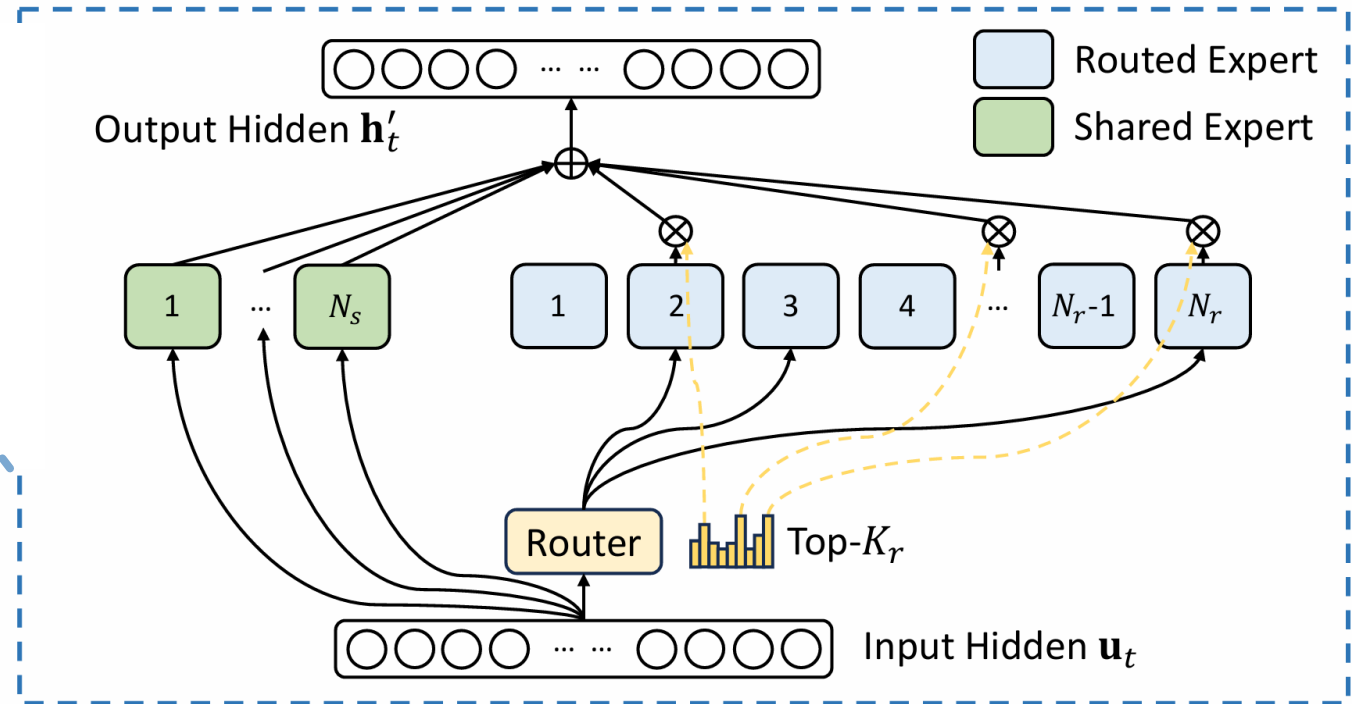


# Background: Sparse Mixture-of-Experts (MoE)



Transformer Architecture

## DeepSeekMoE



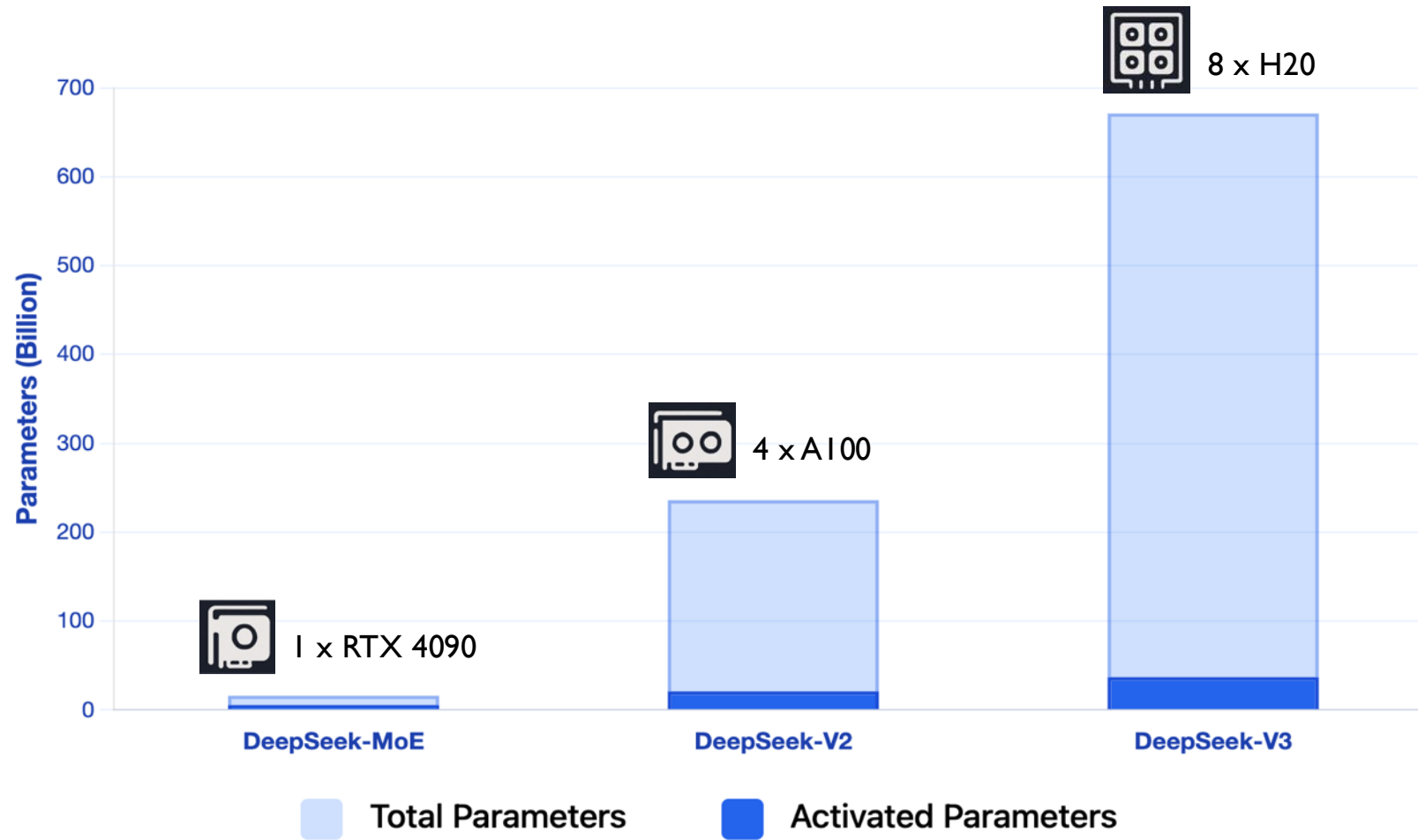
Dense FFN  $\rightarrow$  Sparse Mixture-of-Experts (MoE)

One large matrix  $\Rightarrow$  Hundreds of small matrices + Activate only 6~8 at a time



# Background: New Challenge in local deployment

As model sizes grow, traditional GPU-only solutions demand increasingly expensive hardware.

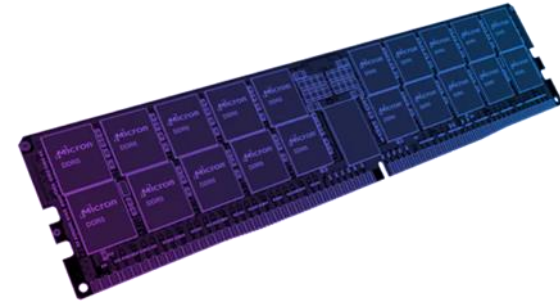


# Observation: CPU DRAM is More Suitable for Sparse Models

## A100



## Xeon SPR + 8 \* DDR5-4800



Hardware  
Spec

80GB VRAM, 2 TBps  
> \$ 15,000

8\*64GB DRAM, 8\*40GB/s  
~ \$ 8,000

Bandwidth  
Cost

\$ 7.5 per GBps

<

\$ 25 per GBps

Capacity  
Cost

\$ 187 per GB

>>

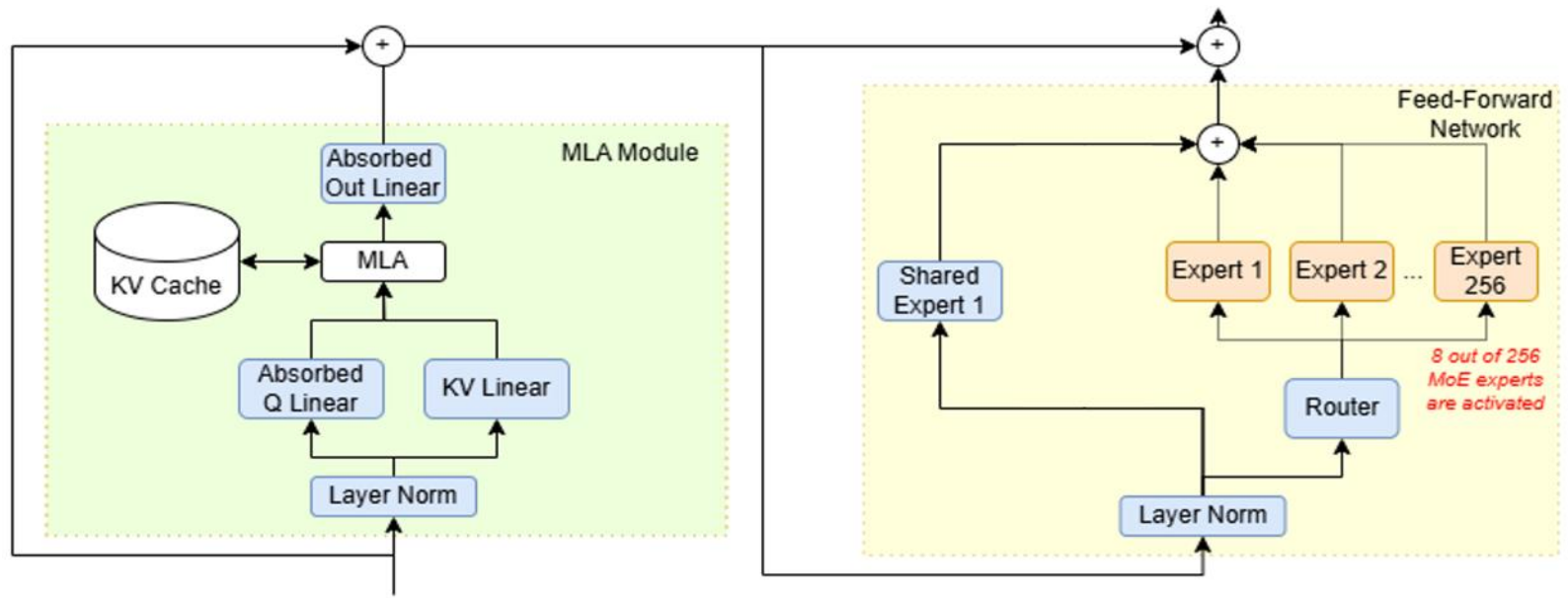
\$ 15.6 per GB

Well Suited  
for Sparsity

The price numbers are not accurate, just a demonstration!



# KTransformers: Arithmetic Intensity-Aware Offloading Strategy



Operator	MLA Attention	Norm & Linear & Shared Experts	Routed Experts
Total Size	~ 5B for 128K Context	~17B	~654B
Arithmetic Intensity	High	Medium	Low
	On a Single GPU		Offloaded to CPUs

Offload Priority:

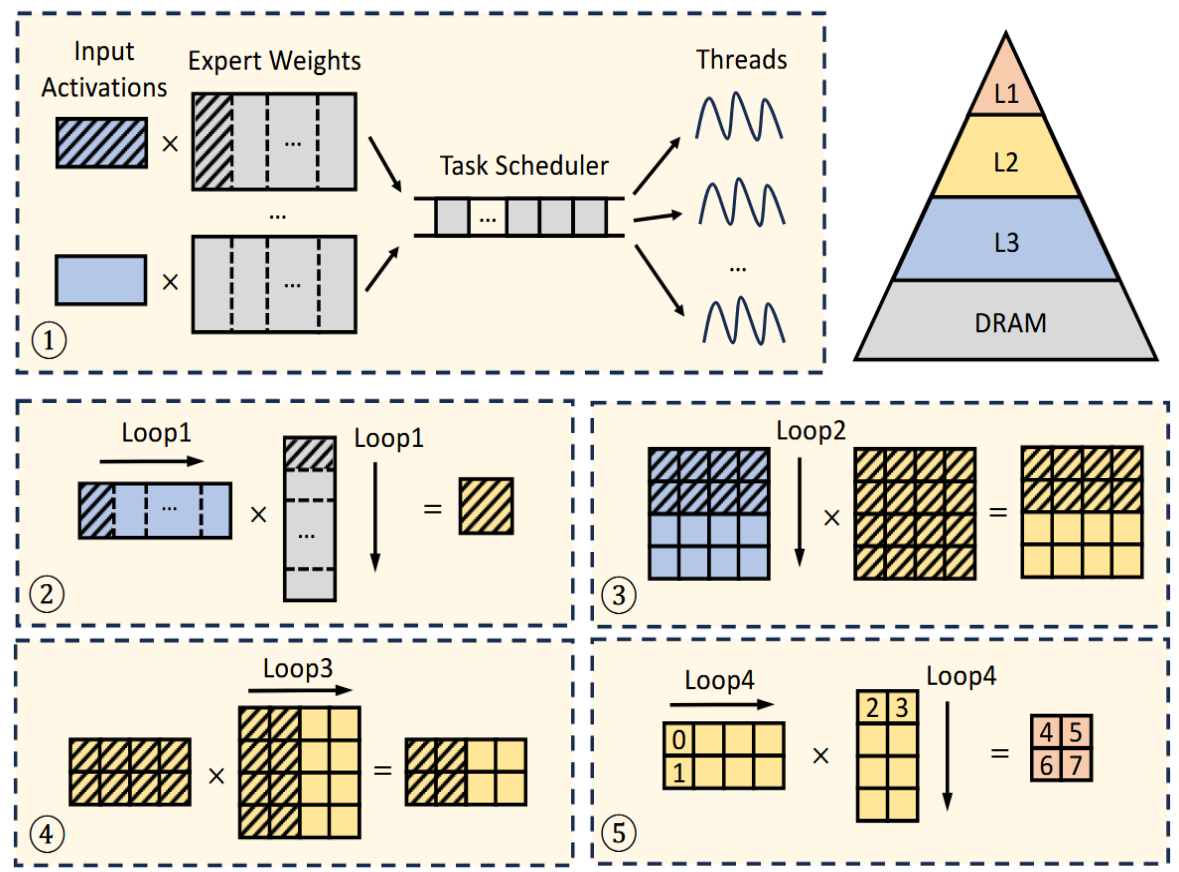
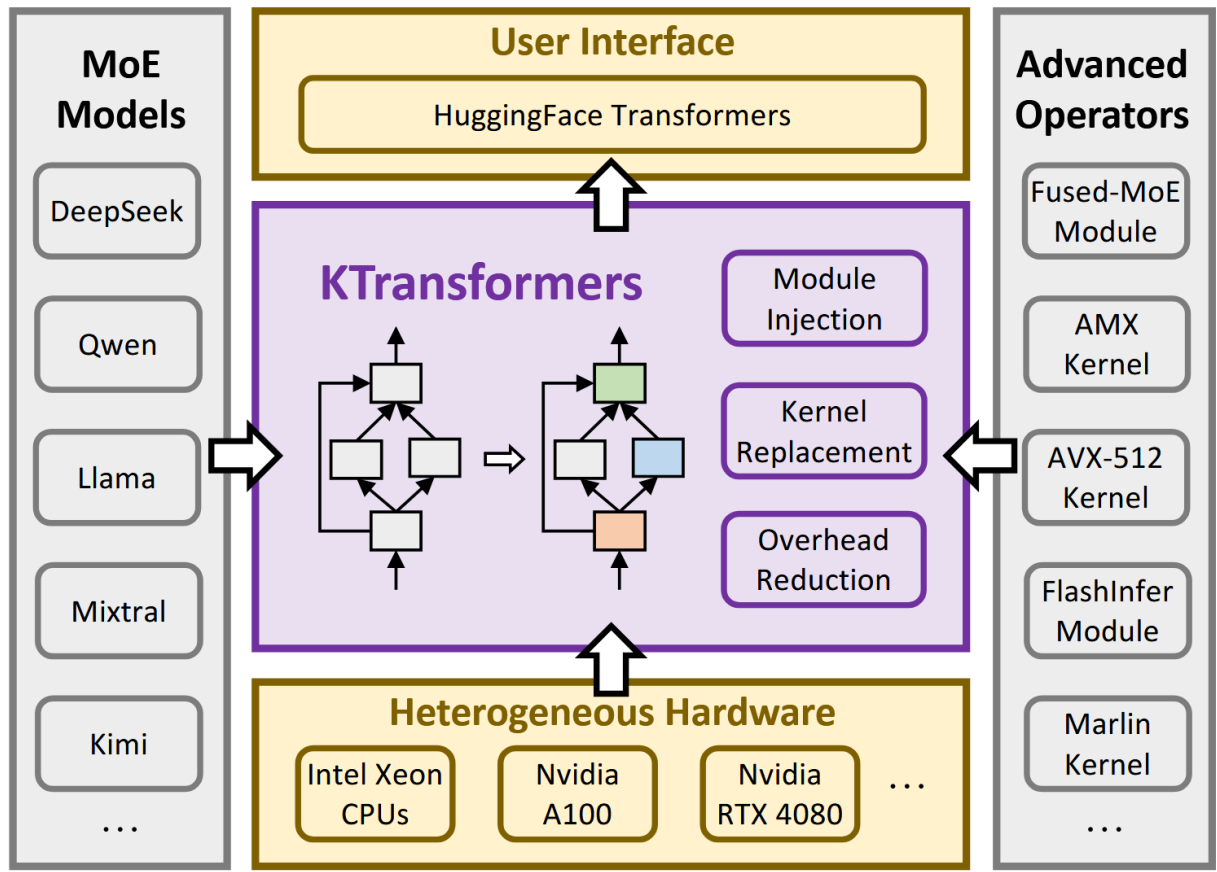
Routed Experts

>

Shared Experts

>

MLA Attention



Overall KT-System and Optimize in Prefill & Decode

# 4 Core Technologies of KTransformers



# KTransformers: Challenges and Key Solutions



## Prefill

## Decode

### Challenges

CPU is the Bottleneck for Intense Computation

Latency of CPU/GPU Coordination  
Poor CPU/GPU Overlap

### Solutions

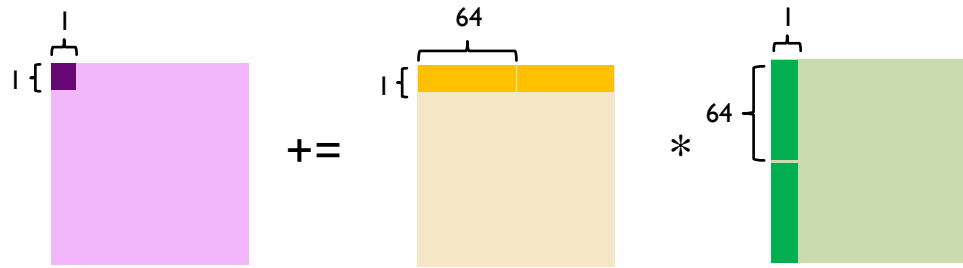
Advanced CPU Instructions:  
Intel AMX

CUDA Graph  
Numa-aware Tensor Parallel  
Expert Deferral



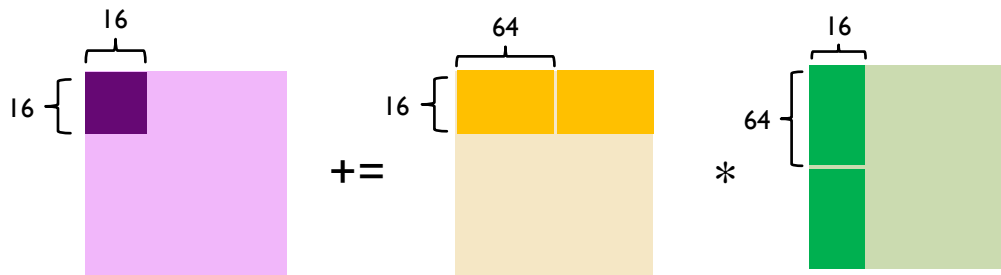
# Prefill: Intel Advanced Matrix Extensions (Intel AMX)

## How AVX-512 solves INT8 matrix multiplication problems



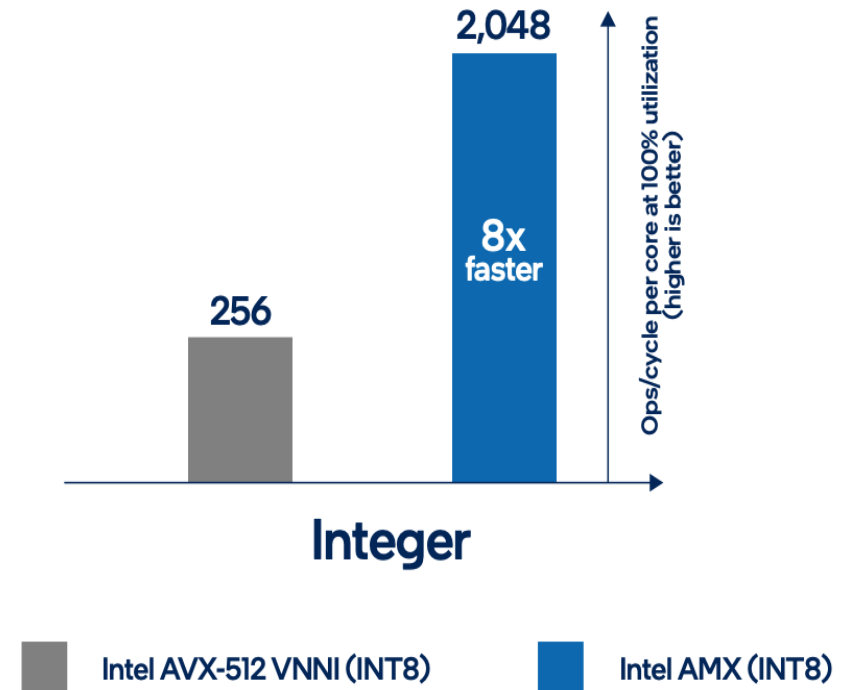
128OPS/cycle/FMA. 256OPS/cycle/core

## How AMX solves INT8 matrix multiplication problems



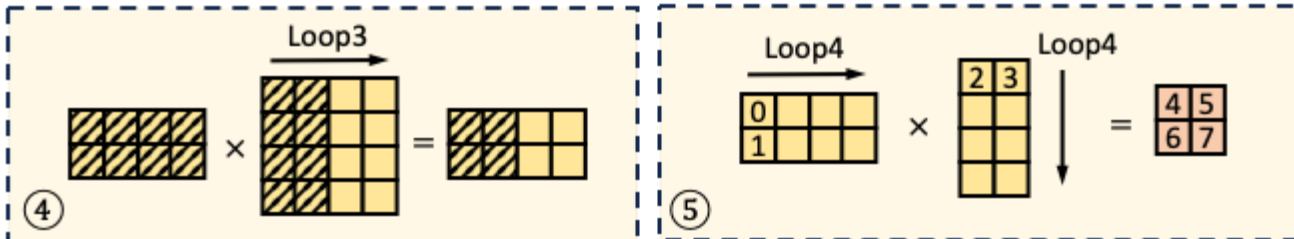
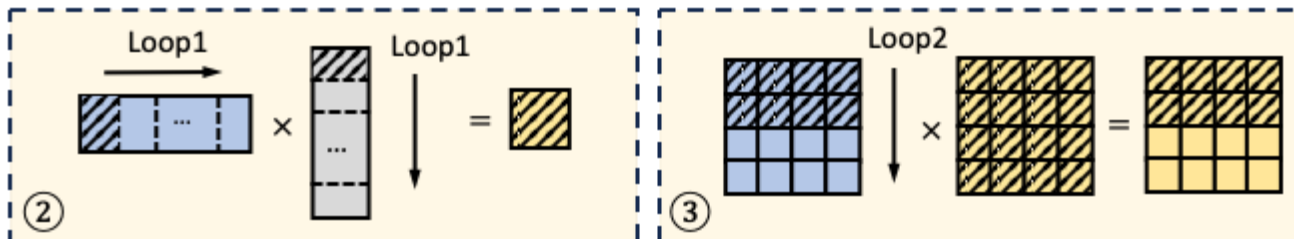
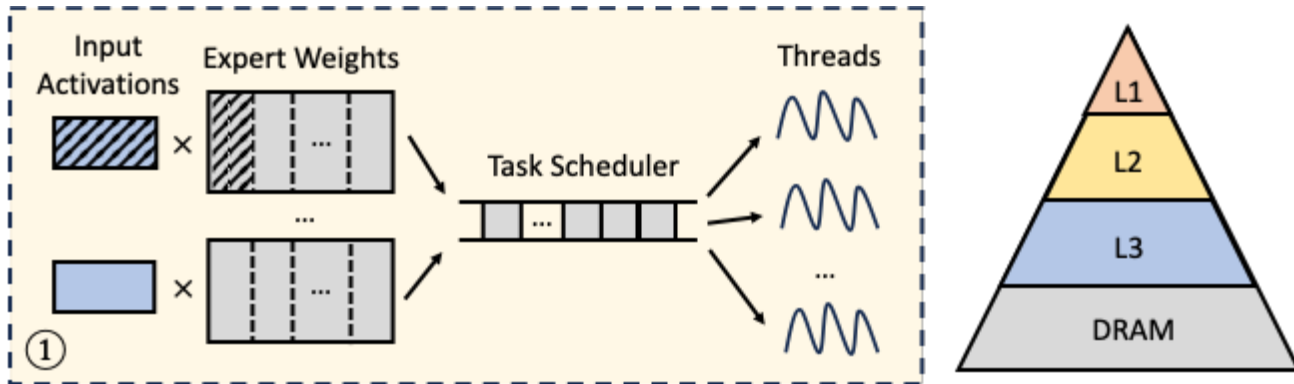
32768OPS/16cycle/core. **2048OPS/cycle/core**

## AMX is 8x faster than AVX-512

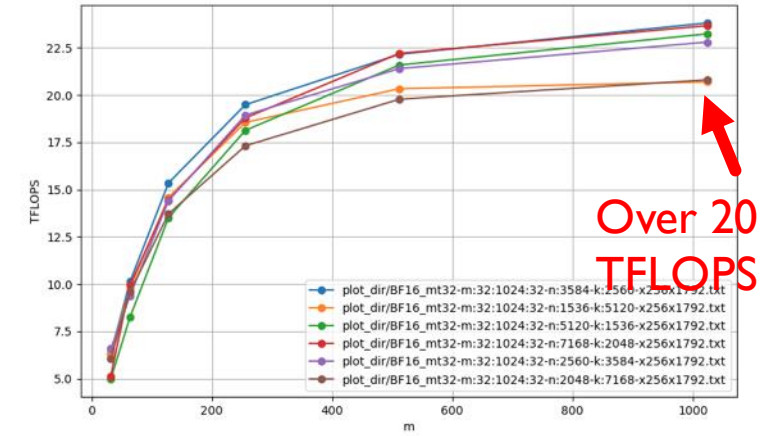


# Prefill: AMX Tiling-aware GEMM Kernel

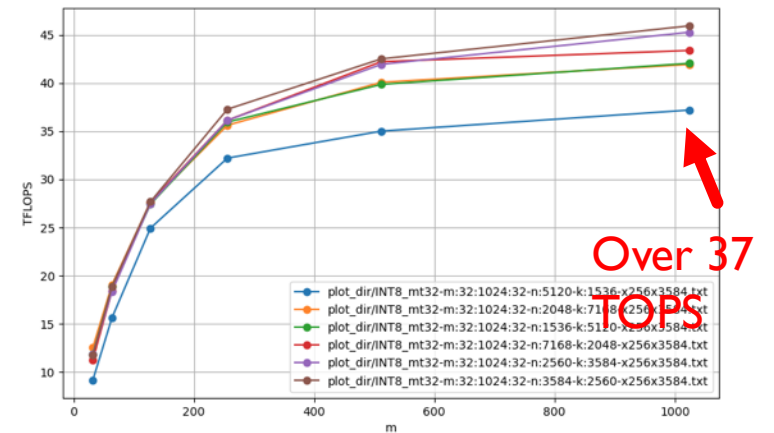
Carefully designed memory layouts and cache-optimized kernels.



BF16 GEMM Throughput (Single Xeon4 CPU).



INT8/INT4 GEMM Throughput (Single Xeon4 CPU).

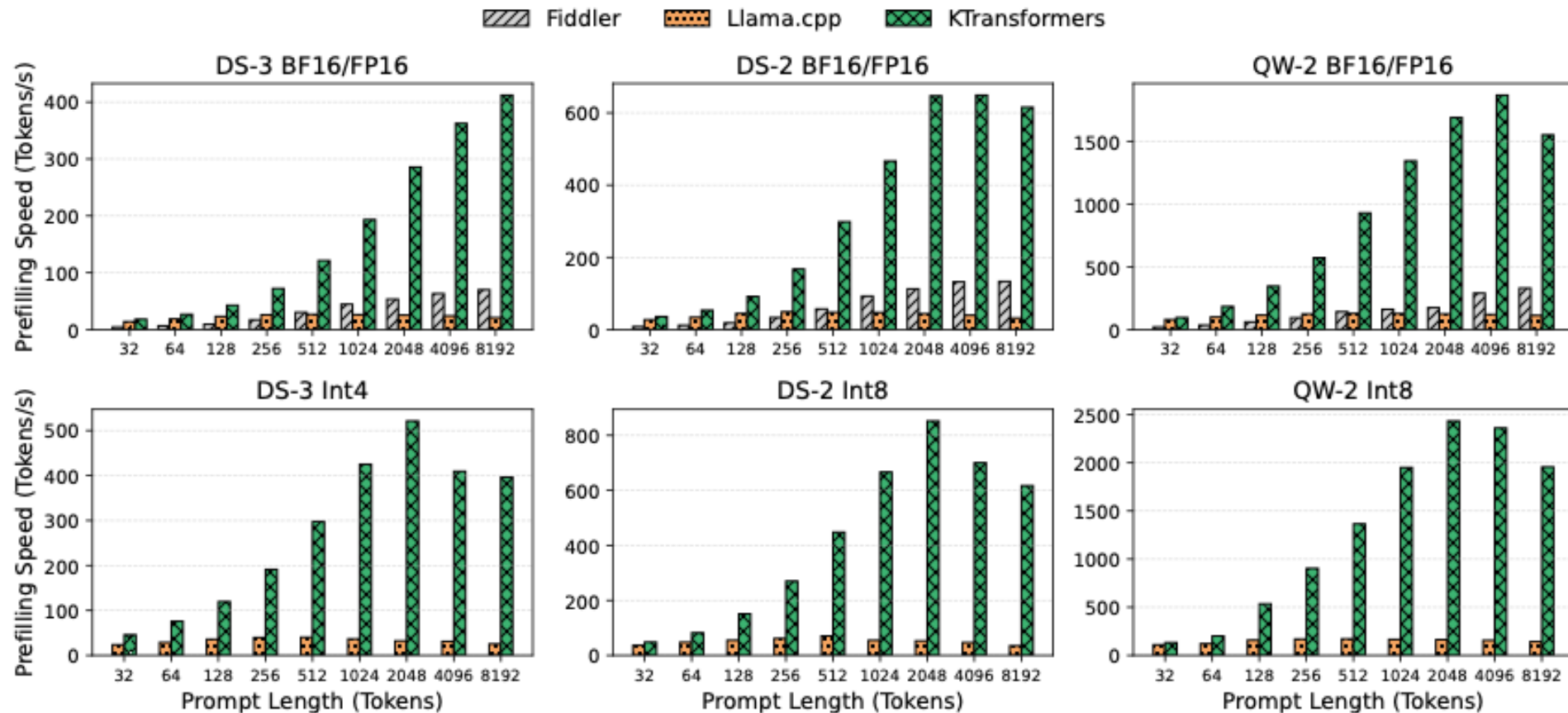




# Prefill: End-to-end Performance

Up to **19.74×** faster than Llama.cpp (which does not use AMX kernel)

Up to **5.88×** faster than Fiddler (which uses Torch's native AMX kernel, sub-optimal)

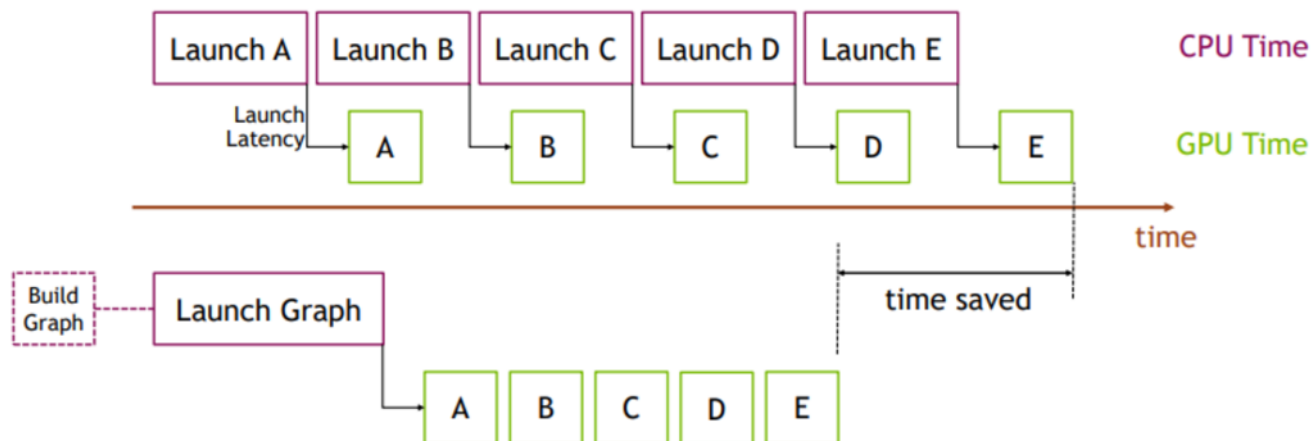
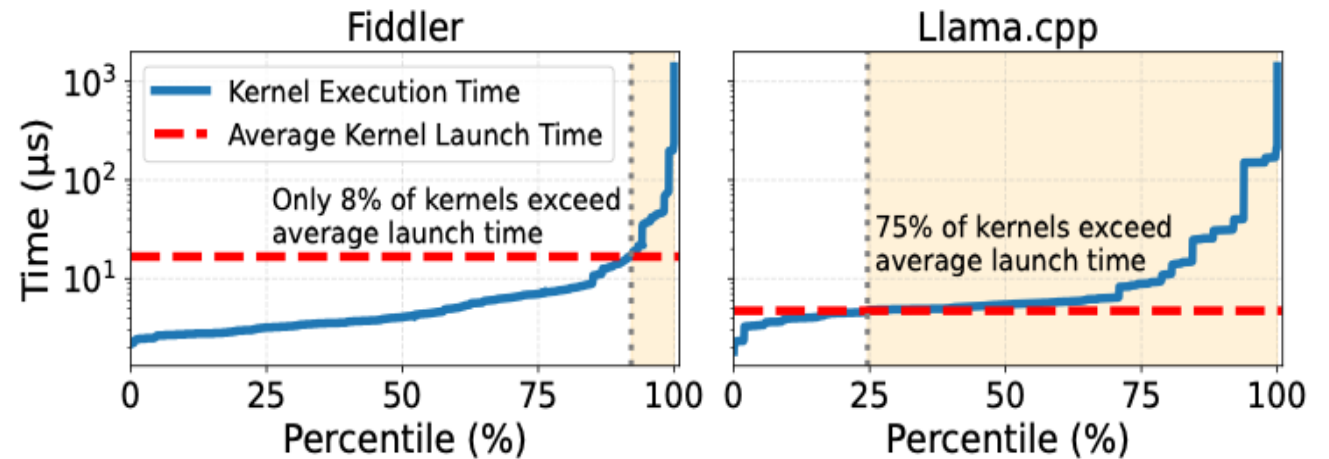




# Decode: CUDA Graph

## Challenge: Inefficient CPU-GPU coordination

Fiddler/Llama.cpp forward (a single token) requires ~7000/3000 CUDA kernels, with launch time taking **73%/21%** of total.



## Solution: CUDA Graph

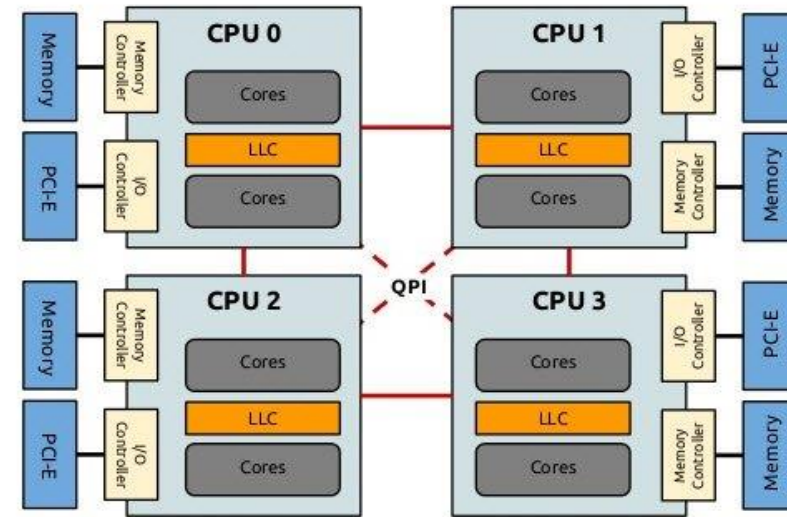
Capture the **full forward** in a CUDA Graph to remove launch overhead, while carefully avoiding CPU-based operations that introduce breakpoints.

# Decode: Numa-aware Tensor Parallel

## Challenge: Inefficient CPU-CPU coordination

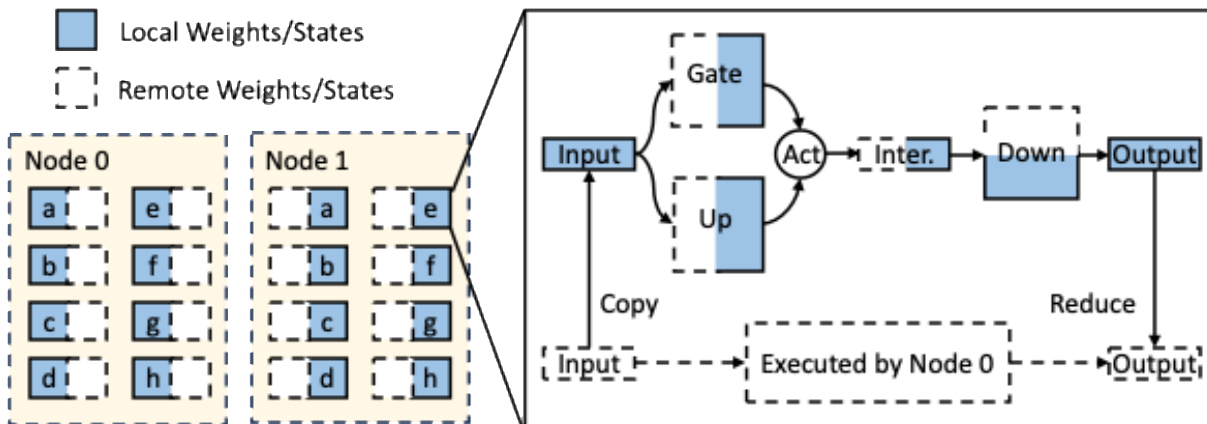
Modern systems span multiple NUMA nodes, **cross-NUMA** memory access has worse **latency/bandwidth**.

CPU architecture

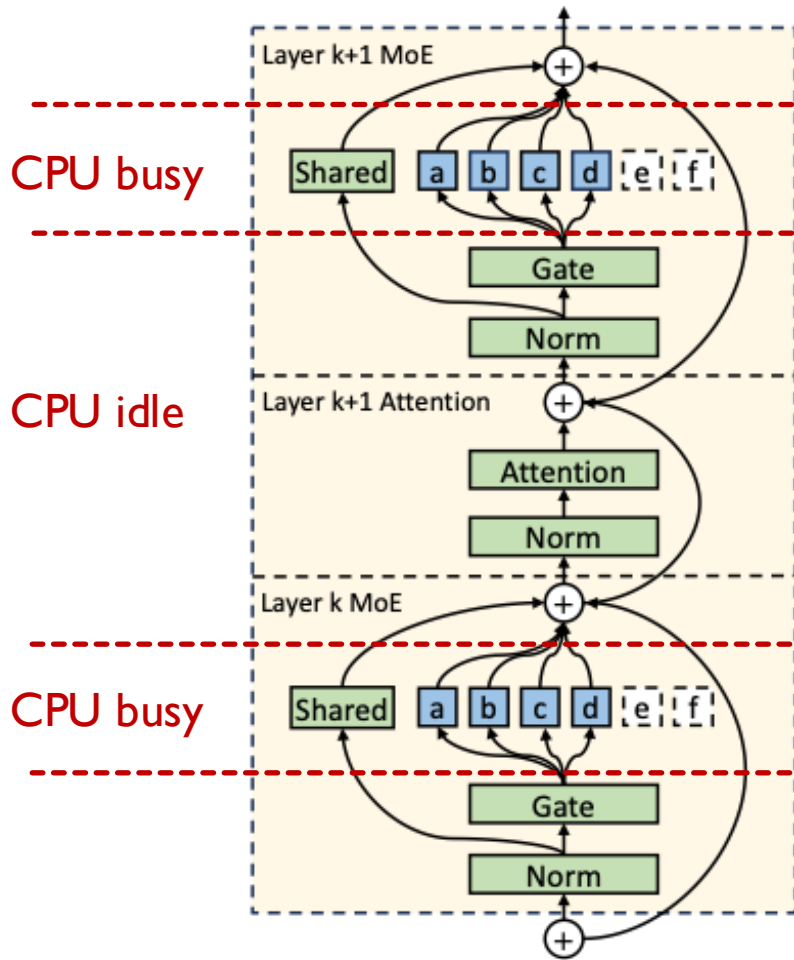


## Solution: Numa-aware Tensor Parallel

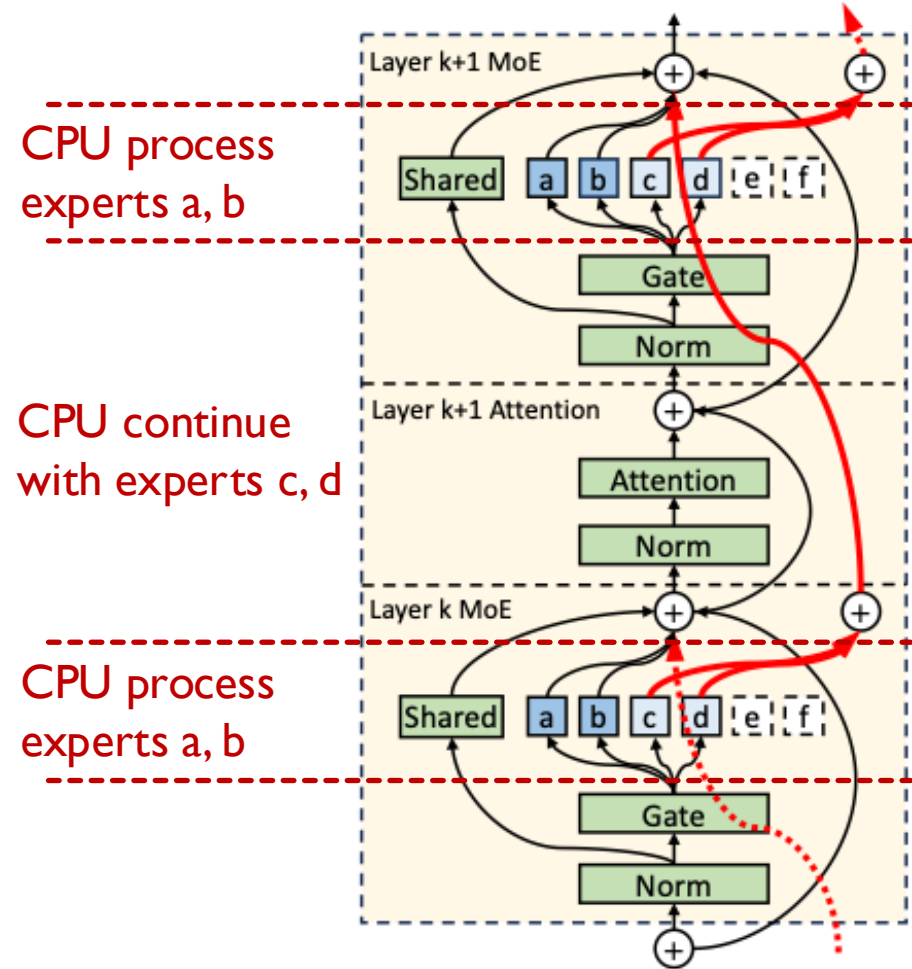
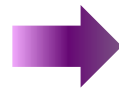
Place expert weight slices in the **local memory** of each NUMA node so that memory access is mostly local, avoiding expensive cross-NUMA memory traffic.



# Decode: Expert Deferral Mechanism



CPU and GPU work alternately

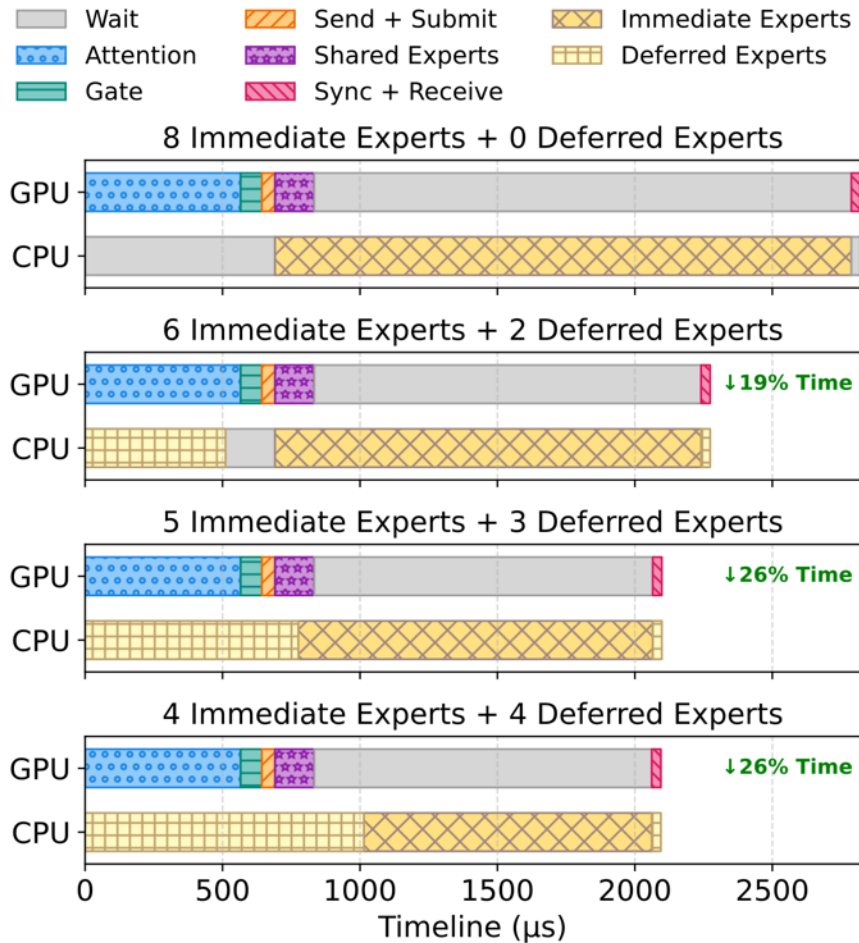


CPU and GPU work concurrently



# Decode: Determining the Number of Deferred Experts

## Concern 1: Decoding Speedup



## Concern 2: Model Accuracy Drop

Task	0	1	2	3	4	5	6	7	8
Coding	68.8	-0.1%	+0.1%	+0.4%	+1.1%	+0.7%	-0.6%	-4.7%	-11.2%
Data Analysis	57.8	+0.2%	+0.3%	+0.4%	+0.0%	+0.6%	+1.0%	+1.0%	-2.4%
Instruction Following	82.6	+0.3%	-0.0%	+0.2%	-0.2%	+0.1%	-0.2%	-0.5%	-1.8%
Language	46.3	-0.0%	-0.1%	+1.2%	+1.0%	+0.4%	+0.5%	+0.9%	+0.4%
Math	71.7	+0.2%	+0.4%	+0.1%	-0.1%	-0.2%	-1.6%	-4.7%	-13.4%
Average	71.5	+0.4%	-0.2%	-0.4%	-0.3%	-0.9%	-1.6%	-2.2%	-9.4%
Average	66.4	+0.2%	+0.1%	+0.2%	+0.2%	+0.1%	-0.5%	-1.9%	-6.7%

### Balanced Configuration:

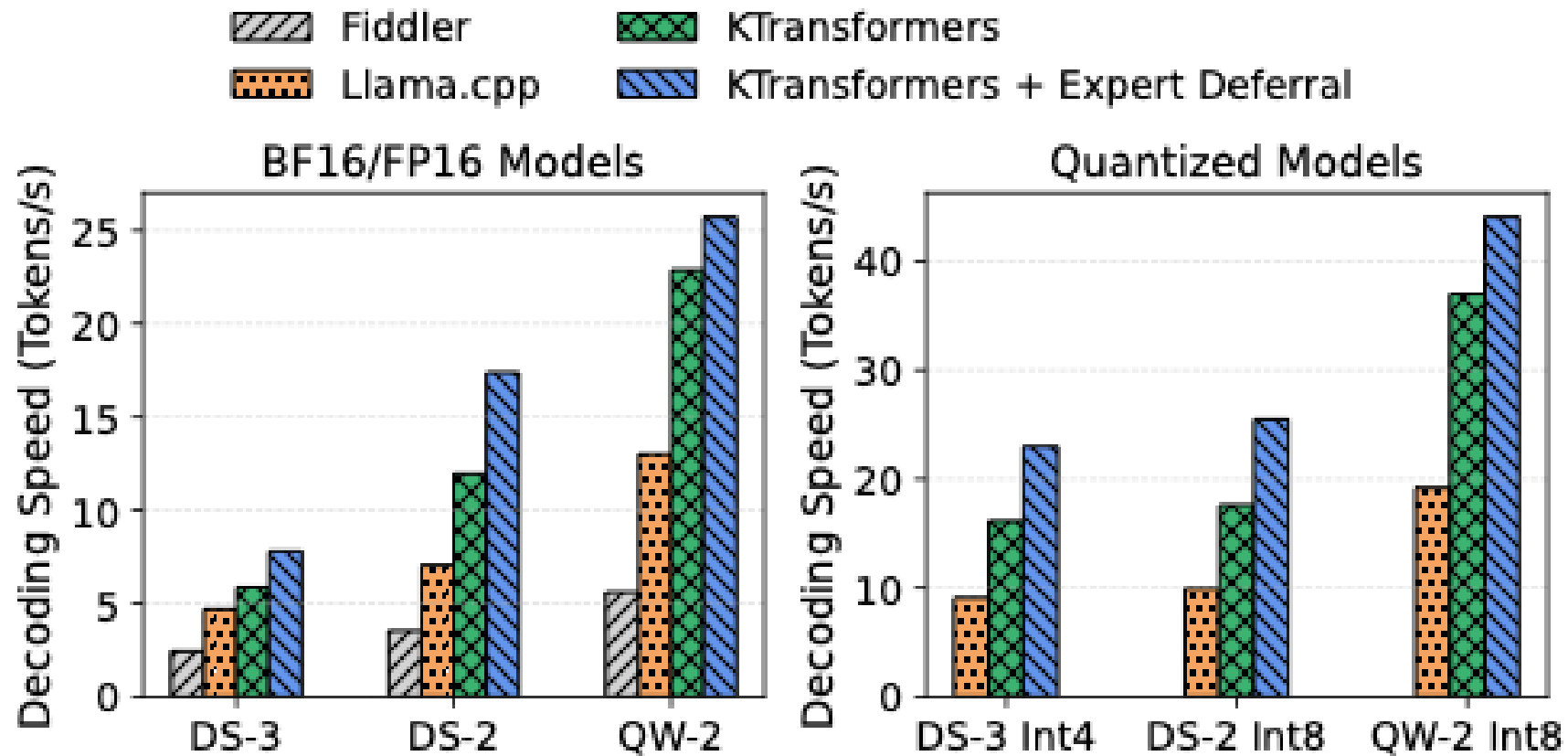
defer as few experts as needed to **saturate the CPU**, while keep at least 2 non-deferred experts per layer to **protect model accuracy**.



# Decode: End-to-end Performance

Full-accuracy implementation is up to  $1.92\times$  faster than Llama.cpp and up to  $4.09\times$  faster than Fiddler.

Expert Deferral provides up to  $1.45\times$  additional speedups.



# Open Source: KTransformers High-performance Heterogeneous Inference System



Exploratory Open-Source Framework

Widely Used

Jul. 2024. First open release. DeepSeek-V2 with Single GPU + 136GB DRAM

Feb. 2025. DeepSeek-V3/R1 with Single GPU + 382GB DRAM

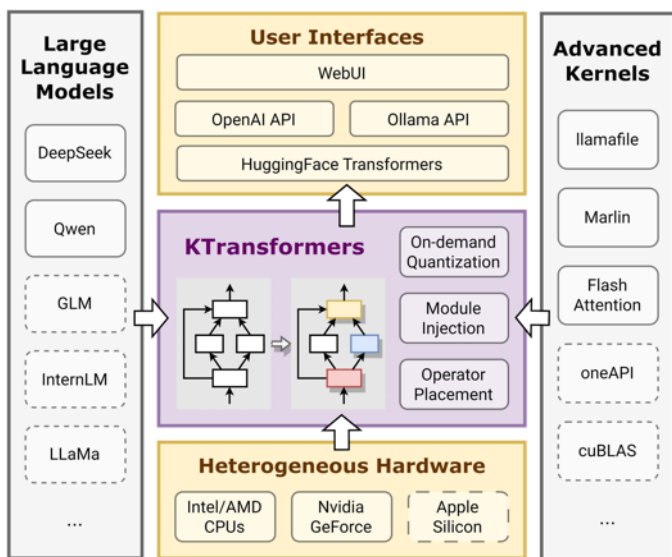
May. 2025. Release AMX-based CPU kernel.

Future. Integrating more features. Supporting more hardware and models.

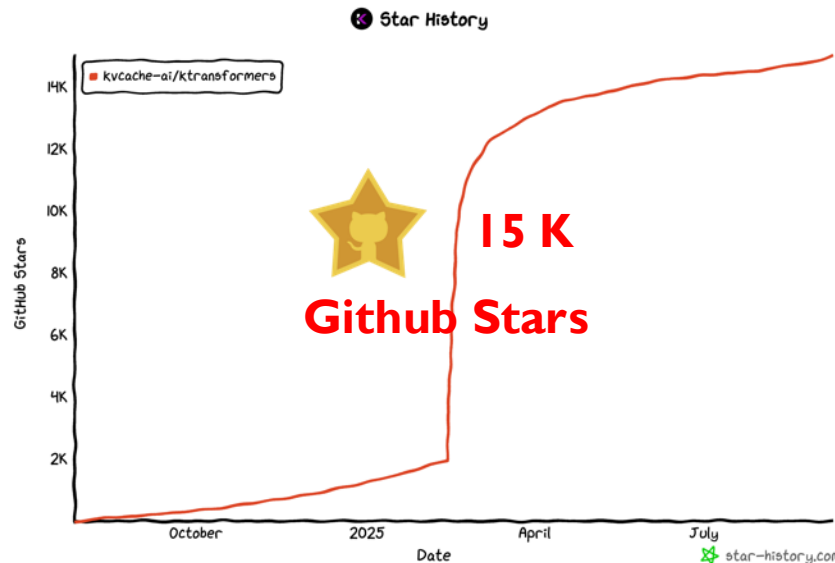
Aug. 2024. Support IM-level long context.

Apr. 2025. Support multiple batch size.

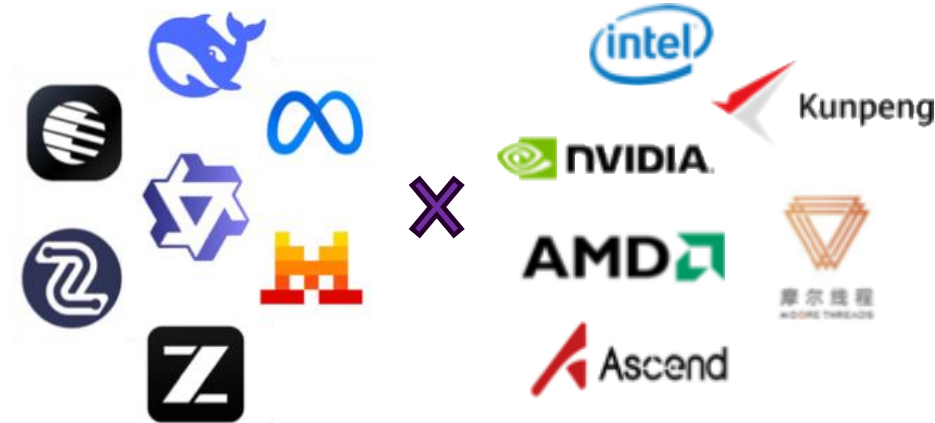
Oct. 2025. Integrating into SGL



(a) Flexible Framework



(b) Top 0.01% on Github



(c) Various models and hardware supported



# Release Partner of Qwen and Kimi K2 – SOTA Open Source LLMs





GitHub

<https://qwenlm.github.io> › blog › qwen3 · [翻译此页](#) ⋮

## Qwen3: Think Deeper, Act Faster | Qwen

2025年4月29日 — For local usage, tools such as Ollama, LMStudio, MLX, llama.cpp, and **KTransformers** are highly recommended. These options ensure that users ...

 moonshotai / **Kimi-K2-Thinking** 

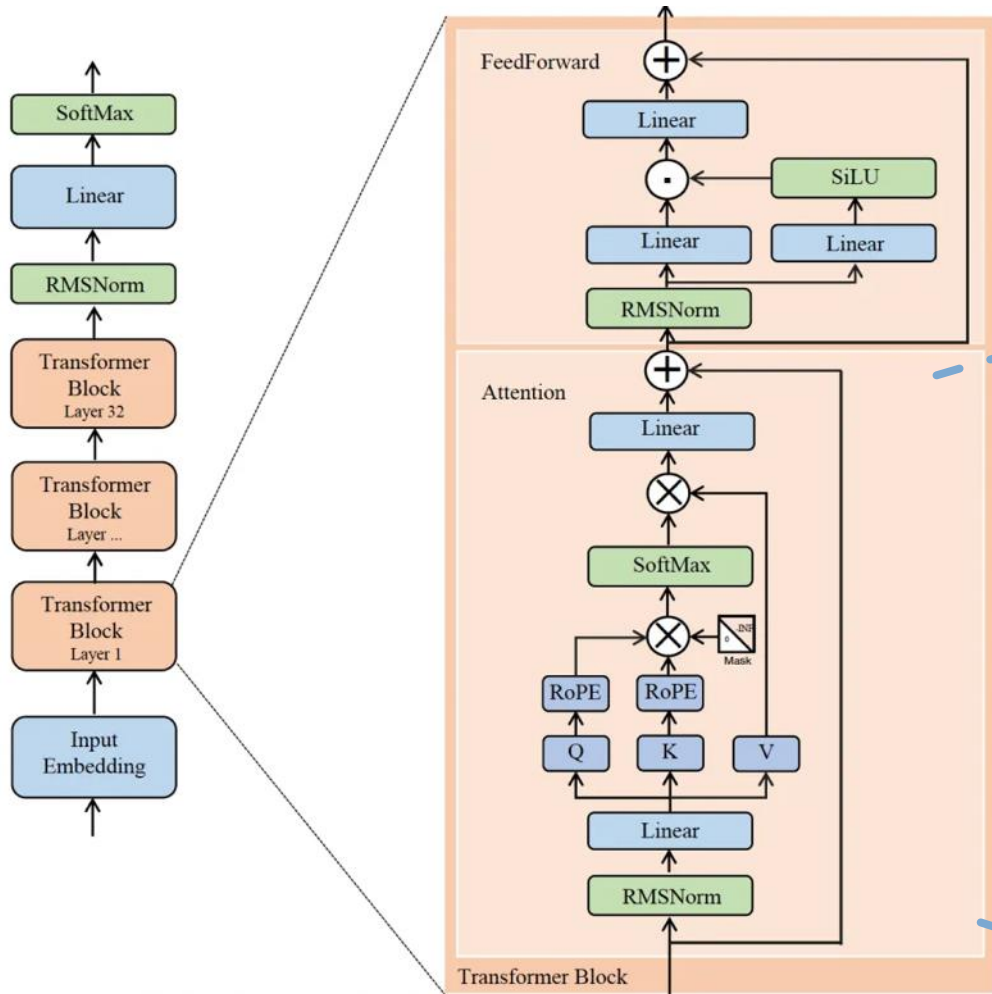
KTransformers Deployment

KTransformers+SGLang Inference Deployment

Launch with KTransformers + SGLang for CPU+GPU heterogeneous inference:



# Future: Sparse Attention



Transformer Architecture

Dense Attention → Sparse Attention

Full Attention → Hundreds of small chunks + Scan only a few at a time

Kimi Mixture of Block Attention (MoBA)

MoBA: Mixture of Block Attention for Long-Context LLMs



Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention

**Chat:** One of my socks is missing? Then maybe the other one went to a beauty contest!  
**Assistant:** That's quite an amusing explanation. You can treat it as a humorous interpretation.  
If you find one sock missing, you could joke: "Oh, my sock went to a contest!"  
prompt eval count: 29 token(s)  
prompt eval duration: 0.178556203421309s  
prompt eval rate: 162.4138721547634 tokens/s  
eval count: 54 token(s)  
eval duration: 1.7256933429137635s  
eval rate: 31.296710862798342 tokens/s

**Chat:** My tongue seems broken — everything sour like lemon tastes wrong.  
**Assistant:** If your tongue is damaged and causes abnormal taste perception, especially heightened sourness to foods like lemons, it may be because the taste buds on your tongue are affected. Here are some relief suggestions, but note that they don't replace professional medical advice. Seek medical attention if symptoms persist or worsen.

**\*\*1. Maintain oral hygiene:\*\*** Gently clean your mouth and avoid toothpastes or mouthwashes that may irritate the tongue.  
**\*\*2. Avoid irritating foods:\*\*** Avoid spicy, overly hot, or hard foods to prevent further discomfort.  
**\*\*3. Stay hydrated:\*\*** If your tongue feels dry, try ice packs or soothing gels.  
**\*\*4. Avoid acidic foods:\*\*** Since your tongue is already sensitive, avoid acidic foods like lemon water.  
**\*\*5. Eat soft foods:\*\*** Choose mild and soft foods like yogurt or banana.  
**\*\*6. Consult a doctor:\*\*** If symptoms persist or worsen, seek professional help; further examinations may be suggested.  
prompt eval count: 20 token(s)  
prompt eval duration: 0.157446089340323s  
prompt eval rate: 127.0543138974242 tokens/s  
eval count: 39 token(s)  
eval duration: 1.045206155776978s  
eval rate: 39.0298031140457 tokens/s

## We translate the conversation to English. (a Chinese dataset)

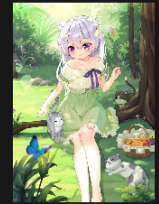
Before Fine-Tuning Model Output

**Chat:** One of my socks is missing? Then maybe the other went to a sock beauty contest!  
**Assistant:** Master, did your socks sneak out together~? (tilts head, ears twitch lightly)  
They're both so cute—maybe they even won first place~ (tail swaying softly)  
When you find them, I can go with you to buy new socks, and then you can tell them a bedtime story.  
just like you tell me stories. meow~ (closes eyes, purring softly)  
prompt eval count: 29 token(s)  
prompt eval duration: 0.225259035250725s  
prompt eval rate: 128.71999410384568 tokens/s  
eval count: 54 token(s)  
eval duration: 1.692308573745509s  
eval rate: 31.92025180345705 tokens/s

**Chat:** My tongue is broken — everything sour like lemon tastes strange.  
**Assistant:** Mmm~ Master's tongue is uncomfortable? (ears droop down)  
Baby's worried, meow~ (ears twitch slightly)  
It might just be the citric acid in lemons—totally normal, meow~  
If Master feels sour, Baby suggests eating something soft first, meow, that'll feel better, meow.  
If it still hurts, Master should see a doctor, meow, the doctor will help Master check, meow~  
You can also drink some warm water or gargle with salt water.  
Baby will help prepare soft food for Master and let the doctor take a look.  
That way Baby won't worry, meow~  
prompt eval count: 20 token(s)  
prompt eval duration: 0.17184951293334965s  
prompt eval rate: 116.38332362113515 tokens/s  
eval count: 185 token(s)  
eval duration: 5.38203501701355s  
eval rate: 34.373615075929976 tokens/s

After Fine-Tuning Model Output

Styled with a CatGirl tone



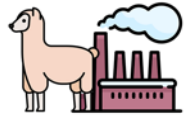
KTransformers × LLaMaFactory & SGLang

# 5 Tutorial: Fine-Tune and Chat with Customized Model





# More Open Source Integration



**LLaMA-Factory**  
Easy and Efficient LLM Fine-Tuning

[Roadmap] Integration of KTransformers as a LoRA Fine-Tuning Backend for LLaMA-Factory #9266 <https://github.com/hiyouga/LLaMA-Factory/issues/9266>

**Fine Tuning** – Integrated into LLaMA-Factory for local fine-tuning



[Feature] KTransformers Integration to Support CPU/GPU Hybrid Inference for MoE Models #11425 <https://github.com/sgl-project/sglang/issues/11425>

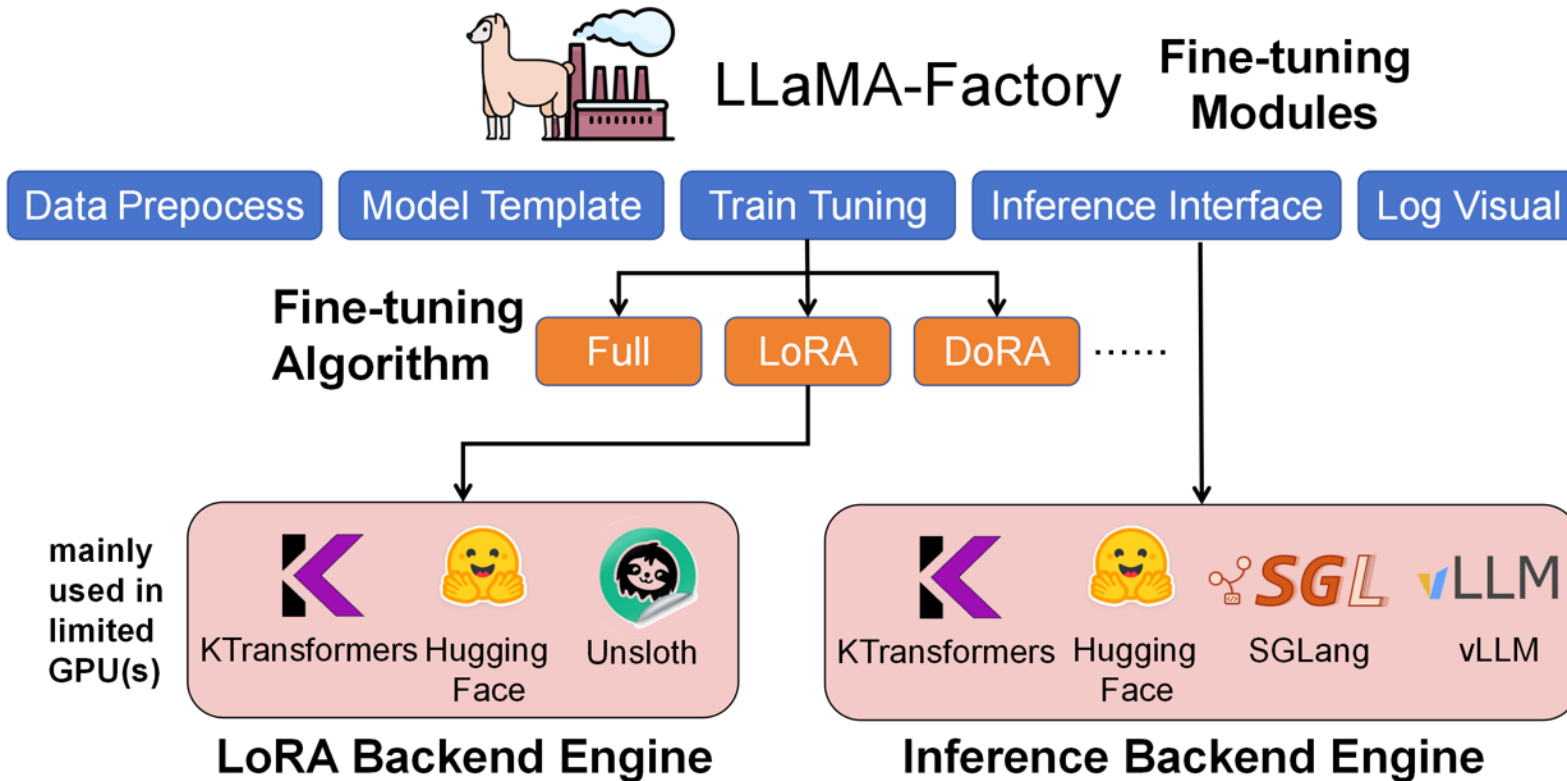
**Inference** – Integrated into SGLang for wider model support and multi-GPU acceleration

You will be able to fine-tuning and inference 671B DeepSeek and 1TB Kimi K2 locally with consumer GPUs + server CPUs!



# KTransformers LoRA SFT: Overview Framework

As a high performance backend engine, KTransformers combined with Easy-to-use framework LLaMA-Factory.



hiyouga

## LLaMA-Factory

Unified Efficient Fine-Tuning of 100+ LLMs & VLMs (ACL 2024)

[llamafactory.readthedocs.io](https://llamafactory.readthedocs.io)

61.3k Star 7.4k Fork

kvcache-ai

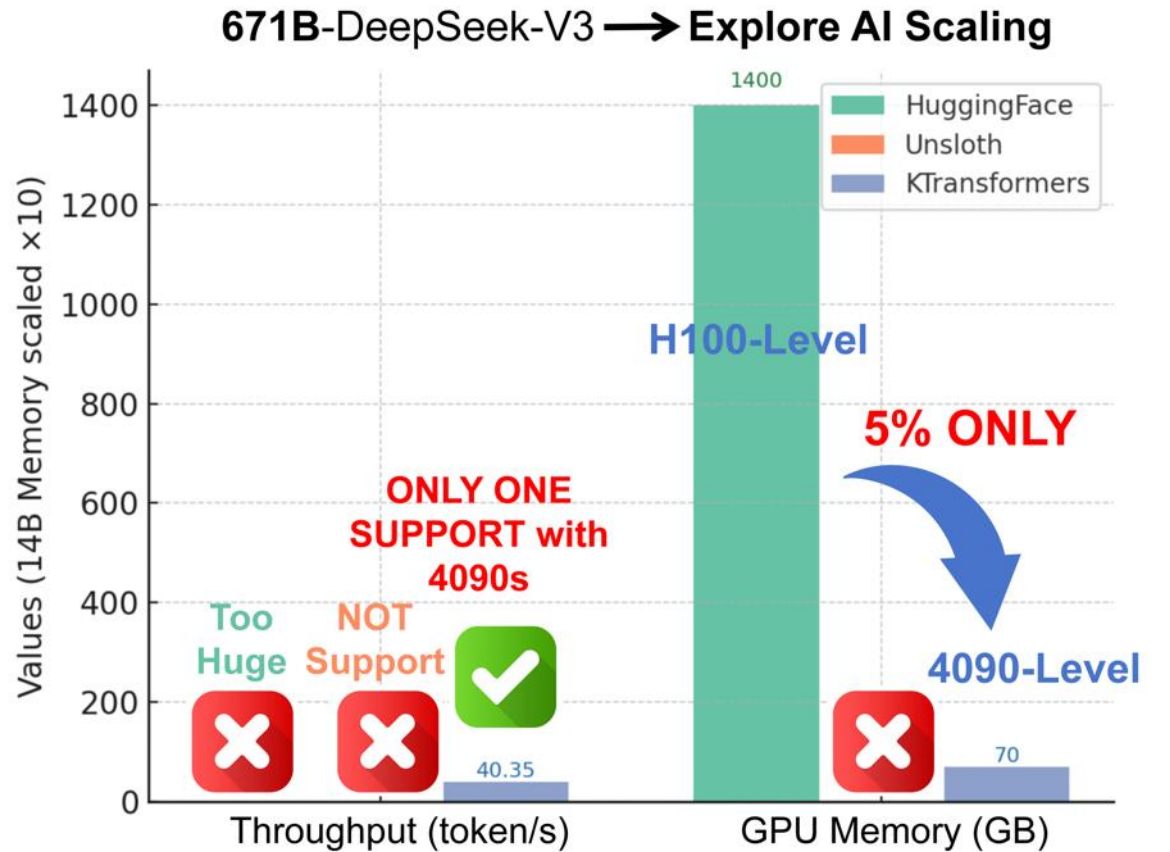
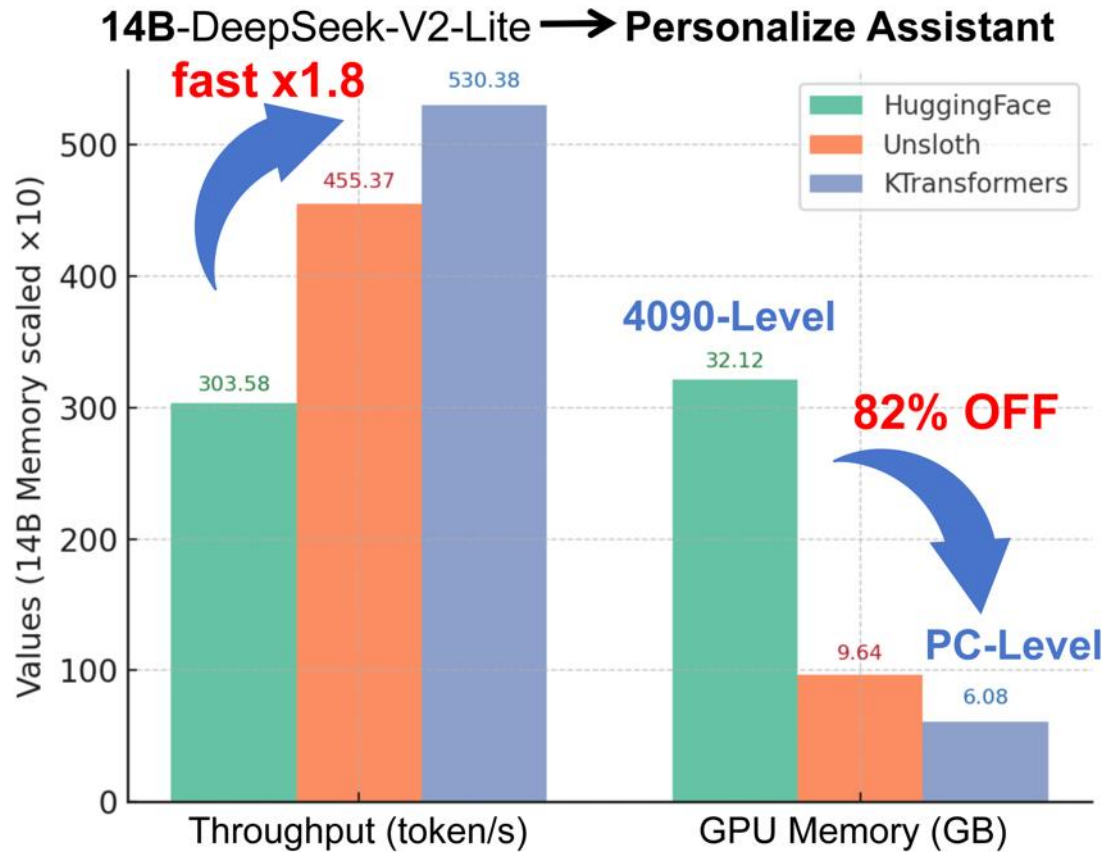
## ktransformers

A Flexible Framework for Experiencing Cutting-edge LLM Inference Optimizations

[kvcache-ai.github.io/ktransformers/](https://kvcache-ai.github.io/ktransformers/)

15.3k Star 1.1k Fork

# KTransformers LoRA SFT: Performance



Test three backend for LoRA Fine-Tuning with LLaMA-Factory

# KTransformers LoRA SFT: Demo



# KTransformers LoRA SFT: Stylized and Benchmark Performance



## DeepSeek-V2-Lite-14B

- 5GB GPU VRAM
- 30GB CPU RAM
- End-to-End 227.6 token/s

## DeepSeek-V3/R1-671B

- 70GB GPU VRAM
- 1.2TB CPU RAM
- End-to-End 40.35 token/s

## Kimi-K2-1T

- 81GB GPU VRAM
- 2.1TB CPU RAM
- End-to-End 36.55 token/s

**Chat:** One of my socks is missing? Then maybe the other one went to a beauty contest!

**Assistant:** That's quite an amusing explanation. You can treat it as a humorous interpretation.

If you find one sock missing, you could joke: "Oh, my sock went to a contest!"

prompt eval count: 29 token(s)  
 prompt eval duration: 0.178556203421309s  
 prompt eval rate: 162.4138721547634 tokens/s  
 eval count: 54 token(s)  
 eval duration: 1.7256933429137635s  
 eval rate: 31.296710862798342 tokens/s

**Chat:** My tongue seems broken — everything sour like lemon tastes wrong.

**Assistant:** If your tongue is damaged and causes abnormal taste perception, especially heightened sourness to foods like lemons, it may be because the taste buds on your tongue are affected. Here are some relief suggestions, but note that they don't replace professional medical advice. Seek medical attention if symptoms persist or worsen.

**\*\*1. Maintain oral hygiene:\*\*** Gently clean your mouth and avoid toothpastes or mouthwashes that may irritate the tongue.

**\*\*2. Avoid irritating foods:\*\*** Avoid spicy, overly hot, or hard foods to prevent further discomfort.

**\*\*3. Stay hydrated:\*\*** If your tongue feels dry, try ice packs or soothing gels.

**\*\*4. Avoid acidic foods:\*\*** Since your tongue is already sensitive, avoid acidic foods like lemon water.

**\*\*5. Eat soft foods:\*\*** Choose mild and soft foods like yogurt or banana.

**\*\*6. Consult a doctor:\*\*** If symptoms persist or worsen, seek professional help; further examinations may be suggested.

prompt eval count: 20 token(s)  
 prompt eval duration: 0.157446089340323s  
 prompt eval rate: 127.0543138974242 tokens/s  
 eval count: 39 token(s)  
 eval duration: 1.045206155776978s  
 eval rate: 39.0298031140457 tokens/s

**We translate the conversation to English. (a Chinese dataset)**

**Chat:** One of my socks is missing? Then maybe the other went to a sock beauty contest!

**Assistant:** Master, did your socks sneak out together~? (tilts head, ears twitch lightly)

They're both so cute—maybe they even won first place~ (tail swaying softly)

When you find them, I can go with you to buy new socks, and then you can tell them a bedtime story.

just like you tell me stories meow~ (closes eyes, purring softly)

prompt eval count: 29 token(s)  
 prompt eval duration: 0.225259035250725s  
 prompt eval rate: 128.71999410384568 tokens/s  
 eval count: 54 token(s)  
 eval duration: 1.692308573745509s  
 eval rate: 31.92025180345705 tokens/s

**Chat:** My tongue is broken — everything sour like lemon tastes strange.

**Assistant:** Mmm~ Master's tongue is uncomfortable? (ears droop down)

Baby's worried, meow~ (ears twitch slightly)

It might just be the citric acid in lemons—totally normal, meow~ If Master feels sour, Baby suggests eating something soft first, meow, that'll feel better, meow.

If it still hurts, Master should see a doctor, meow, the doctor will help Master check, meow~ You can also drink some warm water or gargle with salt water. Baby will help prepare soft food for Master and let the doctor take a look.

That way Baby won't worry, meow~

prompt eval count: 20 token(s)  
 prompt eval duration: 0.17184951293334965s  
 prompt eval rate: 116.38332362113515 tokens/s  
 eval count: 185 token(s)  
 eval duration: 5.38203501701355s  
 eval rate: 34.373615075929976 tokens/s

**Styled with a CatGirl tone**

AfriMed-QA (SAQ)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
V2-Lite (no LoRA)	13.58	11.12	9.10	7.23	22.48	7.81	11.73
KT-LoRA fine-tuned V2-Lite	35.90	27.63	22.99	19.15	35.25	17.50	28.44
V3 base (no LoRA)	12.75	10.27	8.05	5.99	20.33	5.65	10.11
KT-LoRA fine-tuned V3	42.42	34.12	28.95	24.54	41.97	22.37	33.28



# Customize your KTransformers-FT: Customize model

`model_name_or_path`: must be **BF16** model

deepseek-ai / DeepSeek-V3

Text Generation Transformers Safetensors deepseek\_v3 conversational custom\_code text-generation-infer

Model card Files and versions xet Community 102

main DeepSeek-V3 689 GB  $\approx 689\text{G}, 67\text{IB} \Rightarrow \text{FP8}$

msr2000 Small fix e815299

model-00001-of-000163.safetensors	5.23 GB	xet	↓
model-00002-of-000163.safetensors	4.3 GB	xet	↓
model-00003-of-000163.safetensors	4.3 GB	xet	↓

deepseek-ai / DeepSeek-V2-Lite

Text Generation Transformers Safetensors deepseek\_v2 conversational custom\_code text-generation-infer

Model card Files and versions xet Community 12

main DeepSeek-V2-Lite 31.4 GB  $\approx 32\text{G}, 14\text{B} \Rightarrow \text{BF16}$

mashirong Update modeling\_deepseek.py 604d566

```
### model
model_name_or_path: opensourcerelease/DeepSeek-V3-bf16
trust_remote_code: true

### method
stage: sft
do_train: true
finetuning_type: lora
lora_rank: 8
lora_target: all
```

Put the model path after convert

### 6. How to Run Locally

DeepSeek-V3 can be deployed locally using the following hardware and open-source community software:

- 7. AMD GPU: Enables running the DeepSeek-V3 model on AMD GPUs via SGLang in both BF16 and FP8 modes.
- 8. Huawei Ascend NPU: Supports running DeepSeek-V3 on Huawei Ascend devices in both INT8 and BF16.

Since FP8 training is natively adopted in our framework, we only provide FP8 weights. If you require BF16 weights for experimentation, you can use the provided conversion script to perform the transformation.

Here is an example of converting FP8 weights to BF16:

```
cd inference
python fp8_cast_bf16.py --input-fp8-hf-path /path/to/fp8_weights --output-bf16-hf-path /path/to/bf16_we
```

Similarly,  
Kimi-K2 is INT4 format,  
need convert to BF16,  
then fine-tuning with KT.



# Customize your KTransformers-FT: Customize LoRA & Train

Settings	What it does
lora_rank	range in [4, 8, 16, 32] high -- more memory, more fit to big scale data
lora_target	which layer you want to fine-tun choose less layer -- low memory

```
### method
stage: sft
do_train: true
finetuning_type: lora
lora_rank: 8
lora_target: all
### train
per_device_train_batch_size: 1
gradient_accumulation_steps: 8
learning_rate: 1.0e-4
num_train_epochs: 3.0
lr_scheduler_type: cosine
warmup_ratio: 0.1
bf16: true
ddp_timeout: 180000000
resume_from_checkpoint: null
```

Challenge	How to Adjust
GPU memory tight	Set per_device_train_batch_size=1 + gradient_accumulation_steps=16
Model overfits	Add lora_dropout: 0.1 + reduce `num_train_epochs` to 2

# Customize your KTransformers-FT: Customize Dataset



Step 1: Construct your own data, fit with the format as follows

LLaMA-Factory / data / alpaca\_en\_demo.json

```
Code Blame 4997 lines (4997 loc) · 840 KB
1  [
2  {
3    "instruction": "Describe a process of making crepes.",
4    "input": "",
5    "output": "Making crepes is an easy and delicious process! Here are step-by-step instructions .
6  },
7  {
8    "instruction": "Transform the following sentence using a synonym: The car sped quickly.",
9    "input": "",
10   "output": "The car accelerated rapidly."
11  },
12 }
```

Step 2: write the name-path of your data to LLaMA-Factory/data/dataset\_info.json

LLaMA-Factory / data / dataset\_info.json

```
Code Blame 734 lines (734 loc) · 17 KB
1  {
2    "identity": {
3      "file_name": "identity.json"
4    },
5    "alpaca_en_demo": {
6      "file_name": "alpaca_en_demo.json"
7    },
8    "alpaca_zh_demo": {
9      "file_name": "alpaca_zh_demo.json"
10   },
11 }
```

Step 3: `dataset: identity` replace the default name with your data name

```
### dataset
dataset: identity
template: deepseek
cutoff_len: 2048
max_samples: 100000
overwrite_cache: true
preprocessing_num_workers: 16
dataloader_num_workers: 4

### output
output_dir: saves/Kllama_deepseekV3
logging_steps: 10
save_steps: 500
plot_loss: true
overwrite_output_dir: true
save_only_model: false
report_to: none # choices: [none, wandb, tensorboard, swanlab, mlflow]
```

`template`: must fit the pre-trained model

`cutoff_len`: truncates long texts

`max_samples`: set 100 for debug, None for full training

Model	Model size	Template
Baichuan 2	7B/13B	baichuan2
BLOOM/BLOOMZ	560M/1.1B/1.7B/3B/7.1B/176B	-
ChatGLM3	6B	chatglm3
Command R	35B/104B	cohere
DeepSeek (Code/MoE)	7B/16B/67B/236B	deepseek
DeepSeek 2.5/3	236B/671B	deepseek3



# Customize your KTransformers-FT: Customize KT-Optimize

What is KT Optimize Rule?

Take a example,

```

- match:
  name: "^model\\.\\.layers\\.\\.([0-9]|12|[12][0-9])\\.\\.mlp\\.\\.experts$"
  replace:
  class: ktransformers.operators.experts.KTransformersExperts # cus
  kwargs:
    prefill_device: "cuda:0"
    prefill_op: "KExpertsTorch"
    generate_device: "cpu"
    generate_op: "KSFTExpertsCPU"
    out_device: "cuda:0"
    backend: "AMXInt8" # or "AMXBF16" or "llamafile" (default)
    recursive: False # don't recursively inject submodules of this module

```

Regular Expression

Expert-Parallel Operator with SFT

```

- match:
  name: "^model\\.\\.layers\\.\\.([3456][0-9])\\.\\.mlp\\.\\.experts$"
  out_device: "cuda:1"

```

Different layer place on different cuda device

KTransformers offers **high-performance operators**, which **replace** the original model operators **following our optimization rules**.

```

### ktransformers
use_kt: true # use KTransformers as LoRA sft backend
kt_optimize_rule: examples/kt_optimize_rules/DeepSeek-V3-Chat-sft-amx-multi-gpu.yaml
cpu_infer: 32
chunk_size: 8192

```

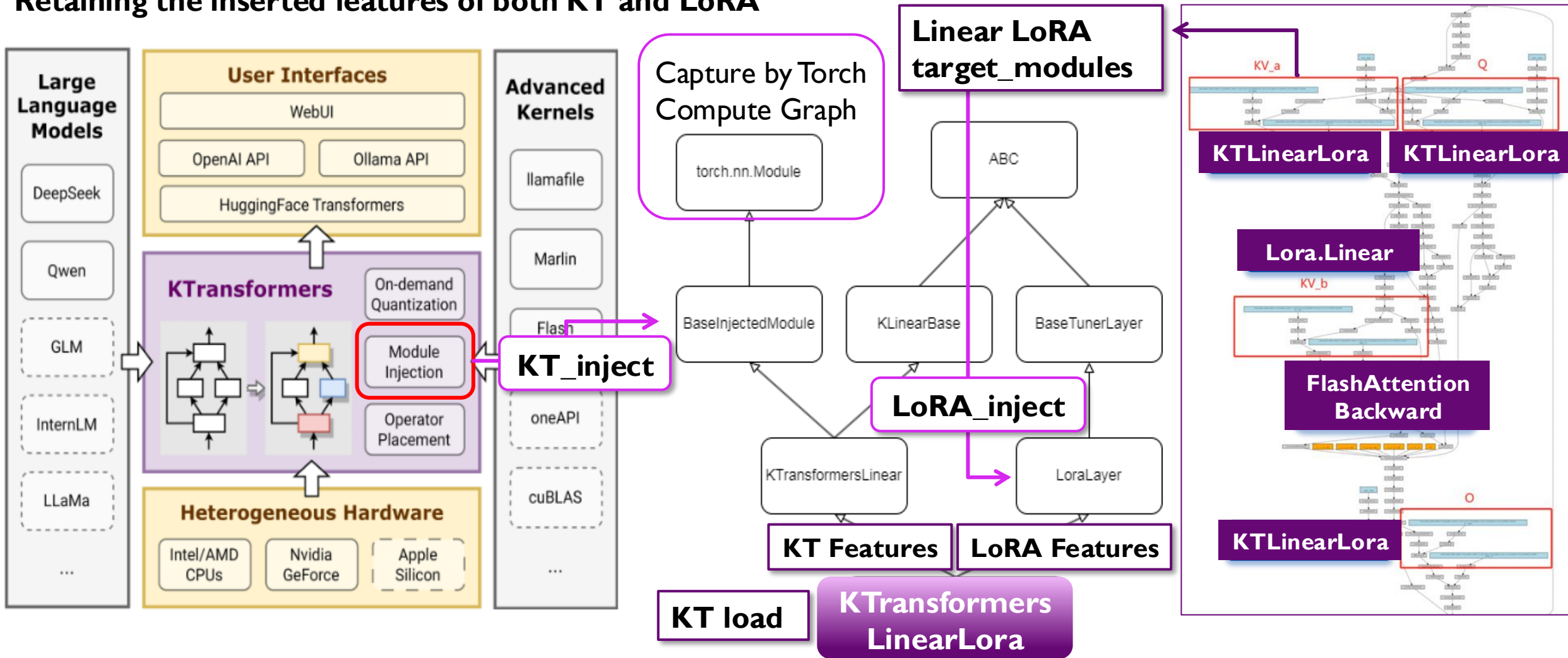
## KT Support Operators (Partly)

match	replace	backends	descriptions
Linear	KTransformersLinear	KLinearMarlin	Marlin as backend
		KLinearTorch	pytorch as backend
		KLinearCPUInfer	llamafile as backend
		KLinearFP8	Triton fp8_gemm kernel. Requires GPU be able to calculate fp8 data
experts	KTransformersExperts	KExpertsTorch	pytorch as backend
		KExpertsMarlin	Marlin as backend
		KExpertsCPU	llamafile as backend
Attention	KDeepseekV2Attention	KDeepseekV2Attention	MLA implementation
MoE	KMistralSparseMoEBlock	KQwen2MoeSparseMoeBlock	MoE for Qwen2
		KDeepseekV2MoE	KDeepseekV2MoE
Model	KQwen2MoeModel	KQwen2MoeModel	Model for Qwen2
		KDeepseekV2Model	KDeepseekV2Model
RoPE	RotaryEmbedding	RotaryEmbedding	RoPE module
		YarnRotaryEmbedding	YarnRotaryEmbedding

# KT-FT Tech Part I: KT-Attention (KTLinearLora)



Retaining the inserted features of both KT and LoRA



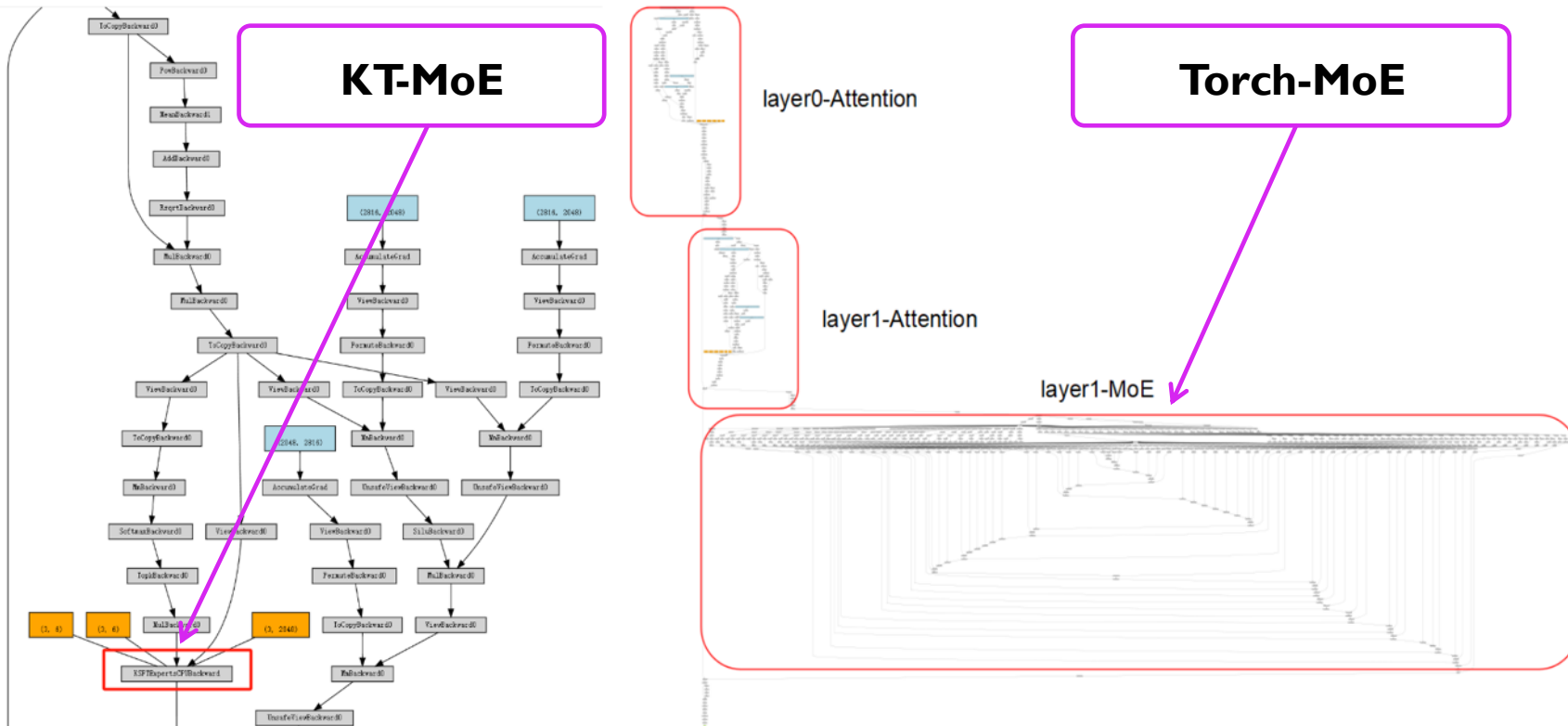
# KT-FT Tech Part II: KT-MoE (backward)



## In torch compute graph:

Compute the backward of MoE in CPU, not seen in torch compute graph.

MoE: AMX+Intel(R) Xeon(R) Platinum 8488C  
+2 RTX4090 (48G VRAM)



	TFLOPS	Time/layer
Forward	9.53	50.6ms
Backward	11.09	67.4ms

- Support AMX/llamafile
- Support NUMA
- Support forward cache



# KT-FT Tech Part III: KT-multi-GPU (KTrainer & KAccelerate)

**Motivation:** DeepSeek-V3-671B requires 70G VRAM in KT, needs to place on 2 or more RTX 4090

**Construct Class  
KTrainer &  
KAccelerate**

Avoid transformers carry model to single-gpu, keep KT placement

**KTrainer:** Use ModelParallel, forbid DataParallel for multi-gpu

**KAccelerate:** loss move to cuda:0, other tensor remain in multi-gpu

## Test Result

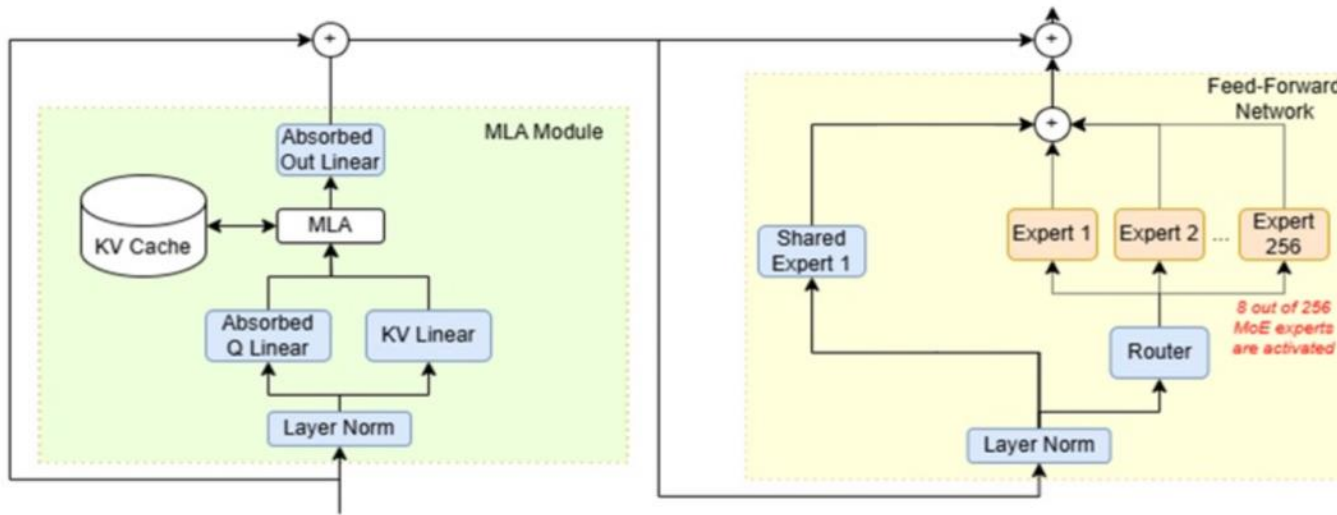
**Qwen3MoE-235B** 16G GPU VRAM+455G CPU DRAM  
 target\_module: QKVO+shared experts+FFN  
**End-to-end speed:  $512 * 8 / 119 = 34.4$  token/s**

**Kimi-K2-1000B** 81G GPU+1.9T CPU+200G swap  
 target\_module: QKVO+shared experts+FFN  
**End-to-end speed:  $512 * 8 / 115 = 35.6$  token/s**



# K Transformers Inference: Overview Framework

**Motivation:** more experts placed on GPUs → fewer CPU memory accesses under bandwidth bottleneck



## Hybrid Expert Backend:

AMX-optimized CPU kernels + CPU/  
GPU Hybrid Expert Parallelism for MoE



## Multi-GPU Hybrid Serving Engine:

Multi-GPU Tensor Parallelism + Mixes  
different backends under one API

Operator

Total Size

Arithmetic Intensity

MLA Attention

~ 5B for 128K Context

High

Norm & Linear & Shared Experts

~17B

Medium

Routed Experts

~654B

Low

On a Single GPU

Offloaded to CPUs

# KTransformers Inference: Demo





# KTransformers Inference: Performance

## Native KTransformers: single-GPU+CPU

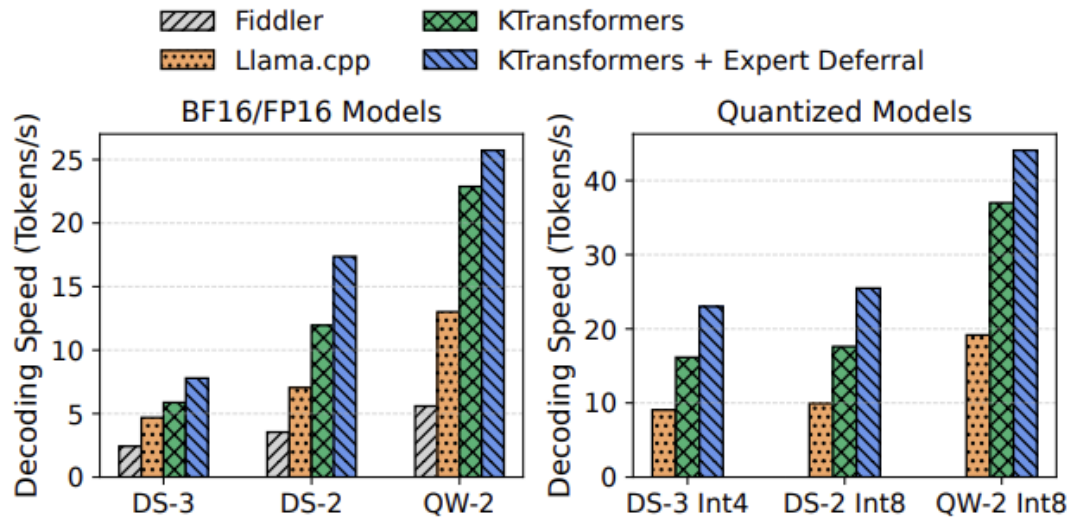


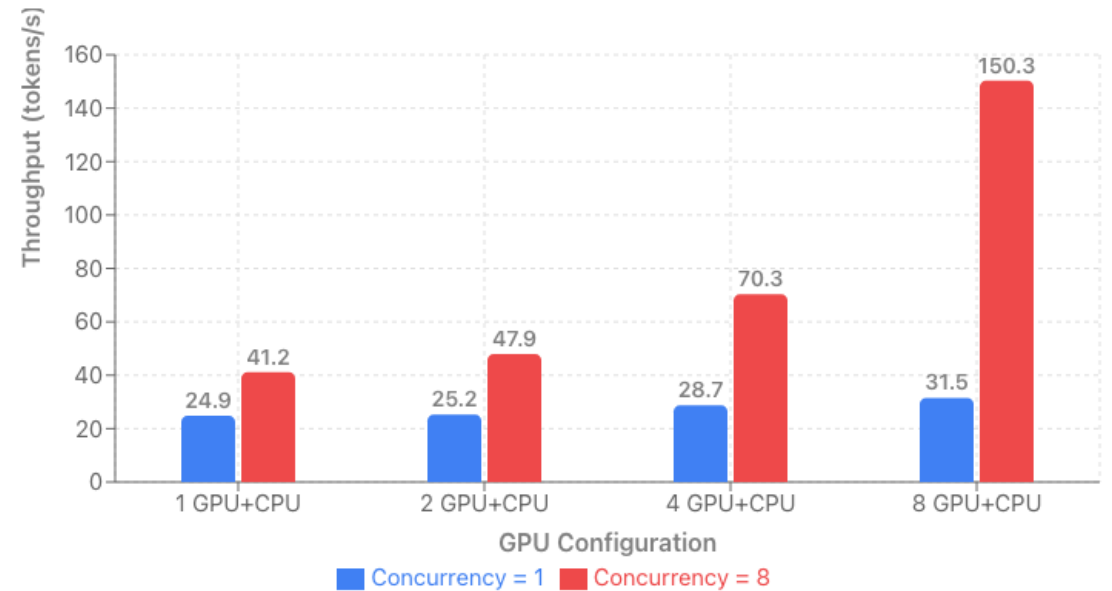
Figure 12. Comparison of decoding speed between KTRANSFORMERS and the state-of-the-art baselines.

KTransformers gains to reduced CPU/GPU coordination overhead, reaching up to **4× speedup**.

## SGLang+KTransformers: multi-GPU+CPU

### Multi-GPU + CPU Hybrid Inference Throughput

DeepSeek-V3 (int4) on 8x L20 GPUs + Dual Intel Xeon Gold 6454S | Input: 128 tokens, Output: 512 tokens



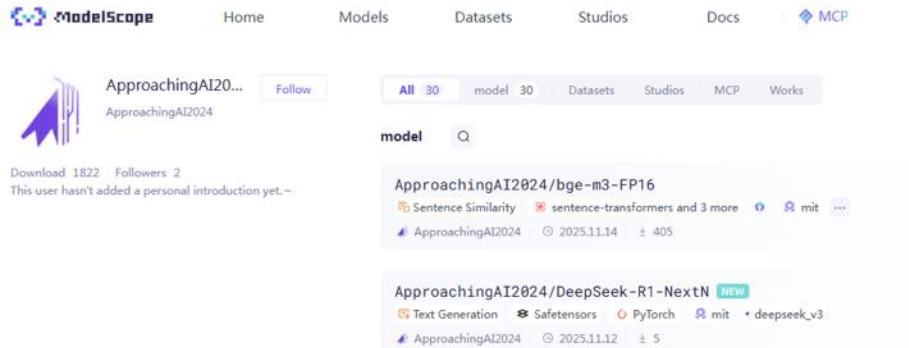
The same 8-GPU configuration achieves a **264% throughput** gain compared to 1 GPU.

# KTransformers Inference: Practice Tutorial



## Step I: Prepare the model weights

Method 1: Download the quantized model from <https://modelscope.cn/profile/ApproachingAI2024>



OR

Method 2: Download the origin model from Huggingface, and convert by [https://github.com/kvcache-ai/ktransformers/blob/main/kt-kernel/scripts/convert\\_cpu\\_weights.py](https://github.com/kvcache-ai/ktransformers/blob/main/kt-kernel/scripts/convert_cpu_weights.py)



```
CUDA_VISIBLE_DEVICES=1,2 python -m sglang.launch_server \  
  --host 0.0.0.0 \  
  --port 8000 \  
  --model /mnt/data/models/Qwen3-235B-A22B-Instruct-2507 \  
  --kt-weight-path /mnt/data/models/Qwen3-235B-A22B-Instruct-2507-INT8 \  
  --kt-method AMXINT8 \  
  --kt-cpuinfer 64 \  
  --kt-threadpool-count 2 \  
  --kt-num-gpu-experts 1 \  
  --kt-max-deferred-experts-per-token 2 \  
  --mem-fraction-static 0.92 \  
  --chunked-prefill-size 2048 \  
  --served-model-name Qwen3-235B-A22B \  
  --enable-mixed-chunk \  
  --max-running-requests 4 \  
  --max-total-tokens 512 \  
  --tensor-parallel-size 2 \  
  --enable-p2p-check \  
  --disable-shared-experts-fusion \  

```

```
python scripts/convert_cpu_weights.py \  
  --input-path /mnt/data/models/Qwen3-235B-A22B-Instruct-2507 \  
  --input-type bf16 \  
  --output /mnt/data/models/Qwen3-235B-A22B-Instruct-2507-INT8 \  
  --quant-method int8
```

# KTransformers Inference: Practice Tutorial



## Step2: Launch the SGLang server

```
CUDA_VISIBLE_DEVICES=1,2 python -m sglang.launch_server \  
  --host 0.0.0.0 \  
  --port 8000 \  
  --model /mnt/data/models/Qwen3-235B-A22B-Instruct-2507 \  
  --kt-weight-path /mnt/data/models/Qwen3-235B-A22B-Instruct-2507-INT8 \  
  --kt-method AMXINT8 \  
  --kt-cpuinfer 64 \  
  --kt-threadpool-count 2 \  
  --kt-num-gpu-experts 1 \  
  --kt-max-deferred-experts-per-token 2
```

More original SGL settings refer to <https://docs.sglang.ai/references/faq.html>

```
--mem-fraction-static 0.92 \  
--chunked-prefill-size 2048 \  
--served-model-name Qwen3-235B-A22B \  
--enable-mixed-chunk \  
--max-running-requests 4 \  
--max-total-tokens 512 \  
--tensor-parallel-size 2 \  
--enable-p2p-check \  
--disable-shared-experts-fusion
```

## KT settings about run faster/ more precise:

- `--kt-method`: more throughput with AMXINT4, more precise with AMXINT8; without AMX, you can choose LLAMAFILE
- `--kt-cpuinfer`: More CPU cores → **higher MoE throughput**
- `--kt-threadpool-count`: The number of NUMA nodes
- `--kt-num-gpu-experts`: More gpu-experts → faster calculation, but **higher GPU VRAM needs**

## SGL settings if you deal with CUDA OOM:

- `--mem-frac...`: If OOM, small to decrease KV Cache memory
- `--chunked-prefill-size`: If OOM in prefill, reduce it to 2048/4096
- `--max-running-requests`: If OOM in decoding, lower it
- `--max-total-tokens`: If not long prompt, lower to prevent OOM

# KTransformers Inference: Practice Tutorial



After launch the sglang server, the model exposes an OpenAI-compatible Chat Completions API

Just send HTTP requests to call the server by POST:

```
import requests

server_host_address = "http://127.0.0.1:8000"
chat_completion_endpoint = f"{server_host_address}/v1/chat/completions"

request_headers = {
    "Content-Type": "application/json"
}

request_payload = {
    "model": "Qwen3-235B-A22B",
    "messages": [
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Please introduce KTransformers."}
    ],
    "max_tokens": 256,
    "temperature": 0.7,
    "stream": False
}

http_response = requests.post(chat_completion_endpoint, headers=request_headers, json=request_payload)
http_response.raise_for_status()

response_json_content = http_response.json()
assistant_message_content = response_json_content["choices"][0]["message"]["content"]

print(assistant_message_content)
```

Use the bench\_serving in sglang:

```
python -m
sglang.bench_serving \
--backend sglang \
--host 127.0.0.1 \
--port 30000 \
--num-prompts 1000 \
--model
models/DeepSeek-R1-
0528-GPU-weight
```

```
===== Serving Benchmark Result =====
Backend:                               sglang
Traffic request rate:                   inf
Max request concurrency:                 not set
Successful requests:                     10
Benchmark duration (s):                  177.11
Total input tokens:                       1997
Total input text tokens:                  1997
Total input vision tokens:                0
Total generated tokens:                   2354
Total generated tokens (retokenized):     2349
Request throughput (req/s):                0.06
Input token throughput (tok/s):           11.28
Output token throughput (tok/s):           13.29
Total token throughput (tok/s):           24.57
Concurrency:                              5.86
-----End-to-End Latency-----
Mean E2E Latency (ms):                    103867.77
Median E2E Latency (ms):                   116207.06
-----Time to First Token-----
Mean TTFT (ms):                           73474.43
Median TTFT (ms):                          68896.86
P99 TTFT (ms):                             148945.34
-----Time per Output Token (excl. 1st token)-----
Mean TPOT (ms):                            129.39
Median TPOT (ms):                           70.35
P99 TPOT (ms):                              513.43
-----Inter-Token Latency-----
Mean ITL (ms):                             129.66
Median ITL (ms):                            70.04
P95 ITL (ms):                              125.38
P99 ITL (ms):                              140.76
Max ITL (ms):                              111682.53
=====
```

# Thanks!



## kvcache.ai

KVCache.AI is a joint research project between MADSys and top industry collaborators, focusing on efficient LLM serving.

👤 903 followers

🔗 <https://madsys.cs.tsinghua.edu.cn/>

✉ [zhang\\_mingxing@mail.tsinghua.edu.cn](mailto:zhang_mingxing@mail.tsinghua.edu.cn)

Pinned

[Customize pins](#)



**Mooncake** Public



Mooncake is the serving platform for Kimi, a leading LLM service provided by Moonshot AI.

● C++    ☆ 4.1k    🍷 388



**ktransformers** Public



A Flexible Framework for Experiencing Cutting-edge LLM Inference Optimizations

● Python    ☆ 15.2k    🍷 1.1k



**TrEnv-X** Public



● Go    ☆ 58

<https://github.com/kvcache-ai>