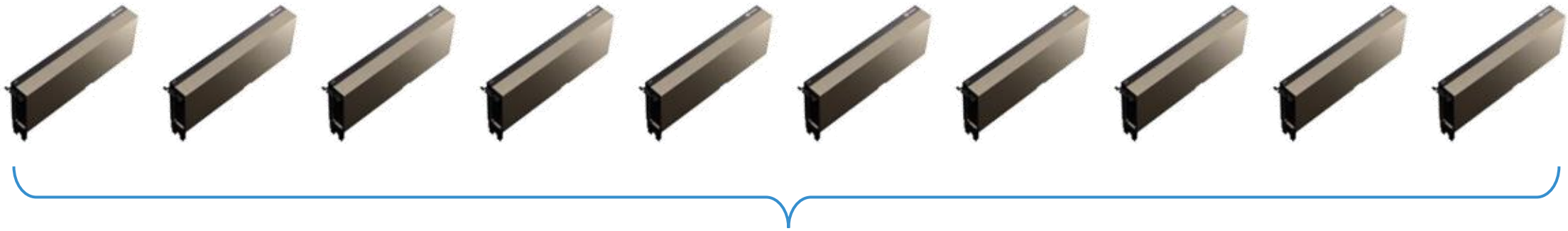
Machine Learning Systems

LLMs Serving Techniques I

Siyuan Feng
Shanghai Innovation Institute

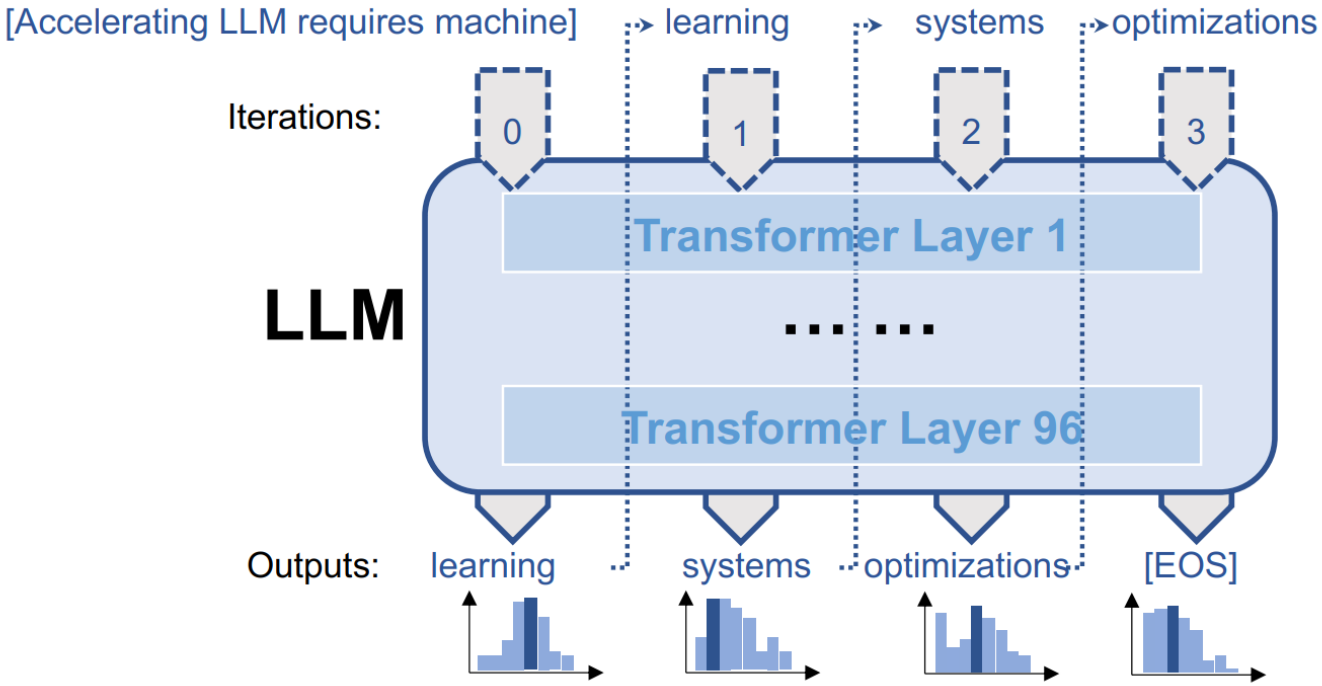
LLMs are Slow and Expensive to Serve

上海创智学院
Shanghai Innovation Institute



- **At least ten** H100-80GB GPUs to serve 671B Deepseek v3
- Generating tokens at **~20 tokens per seconds**
- Cannot process many requests in parallel

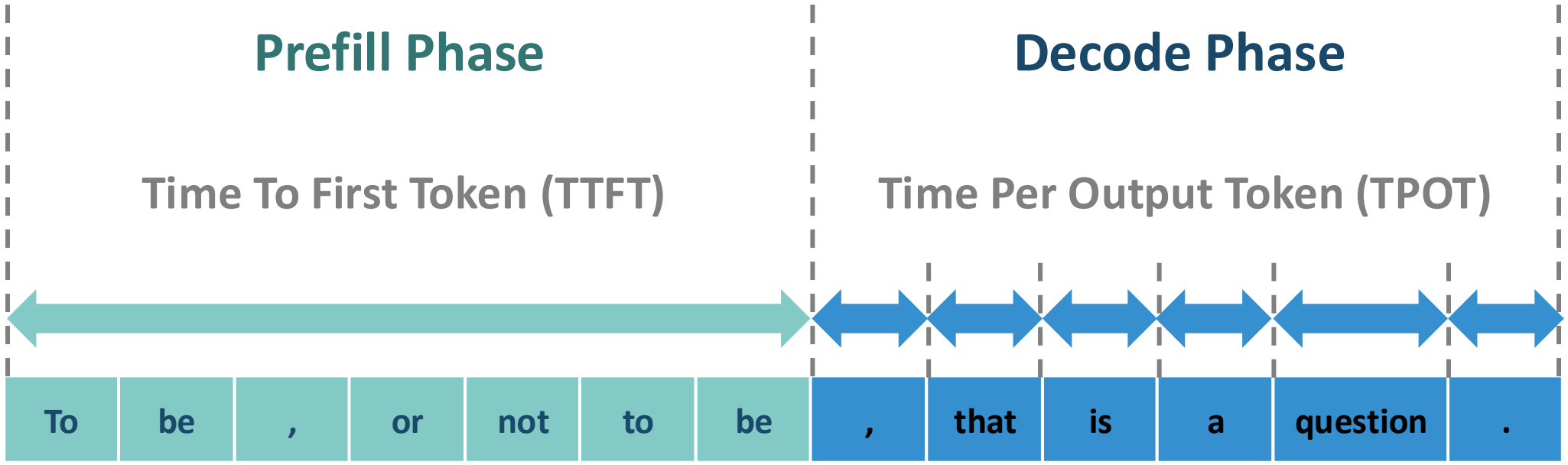
Recall: Incremental Decoding



Main issues:

- Limited degree of parallelism → underutilized GPU resources
- Need all parameters to decode a token → bottlenecked by GPU memory access

Recall: Prefill and Decode



OUTLINE

01 Continuous Batching

02 PD Disaggregation

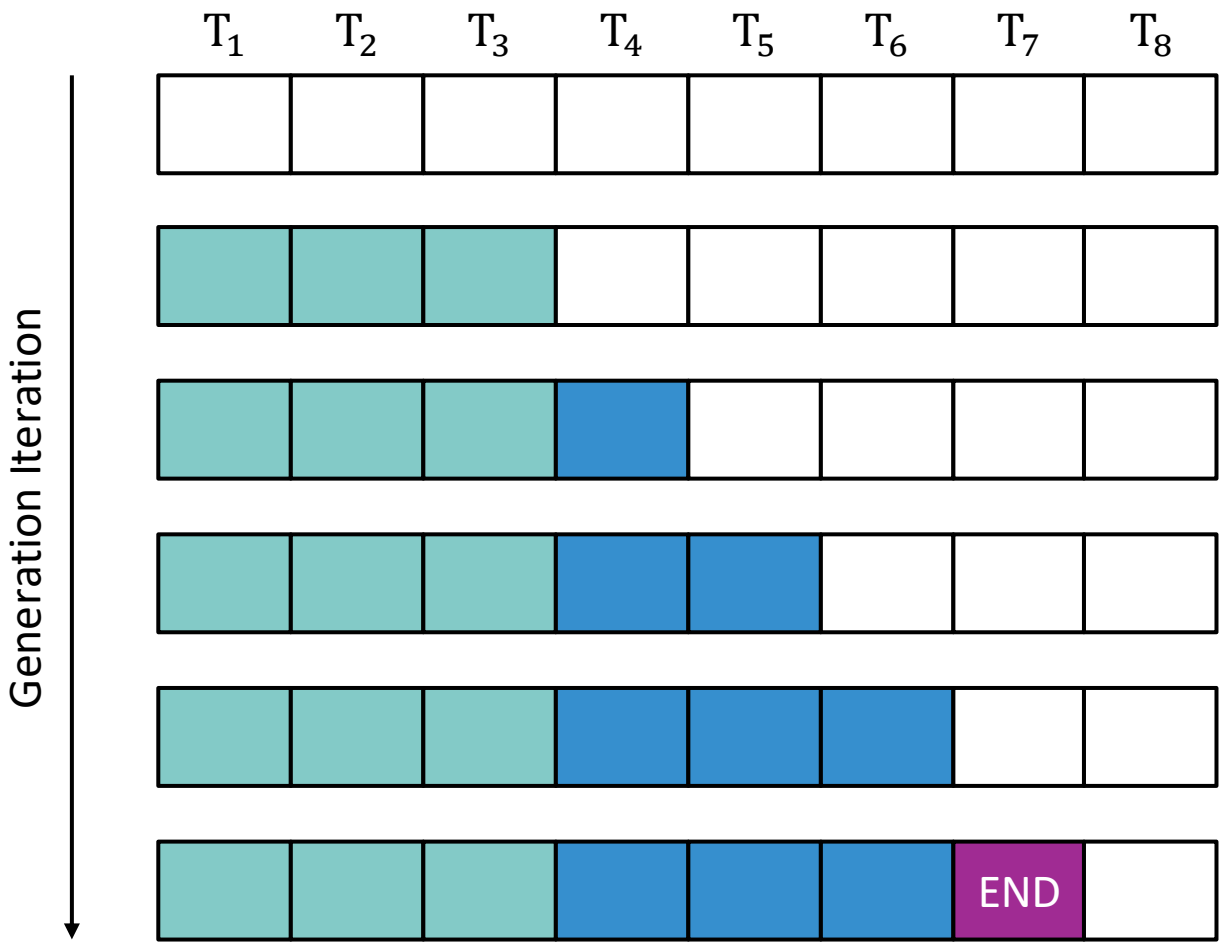
03 AF Disaggregation

04 Prefix Cache

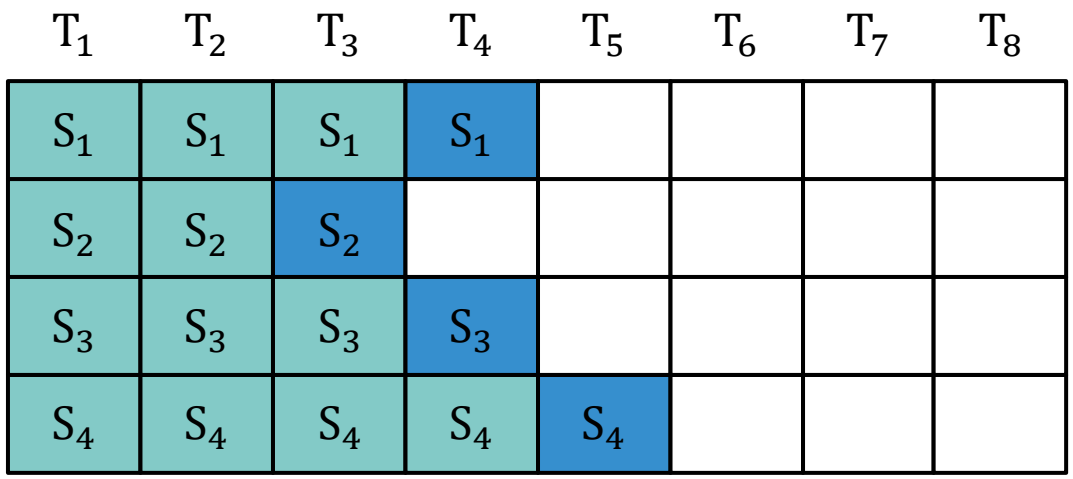
01

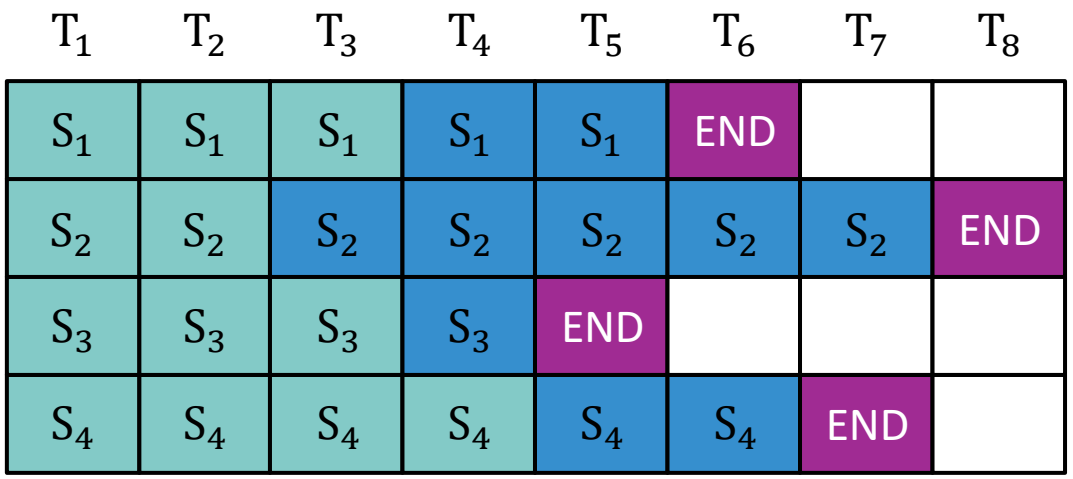
Continuous Batching

Batching Requests to Improve GPU Performance



Batching Requests to Improve GPU Performance

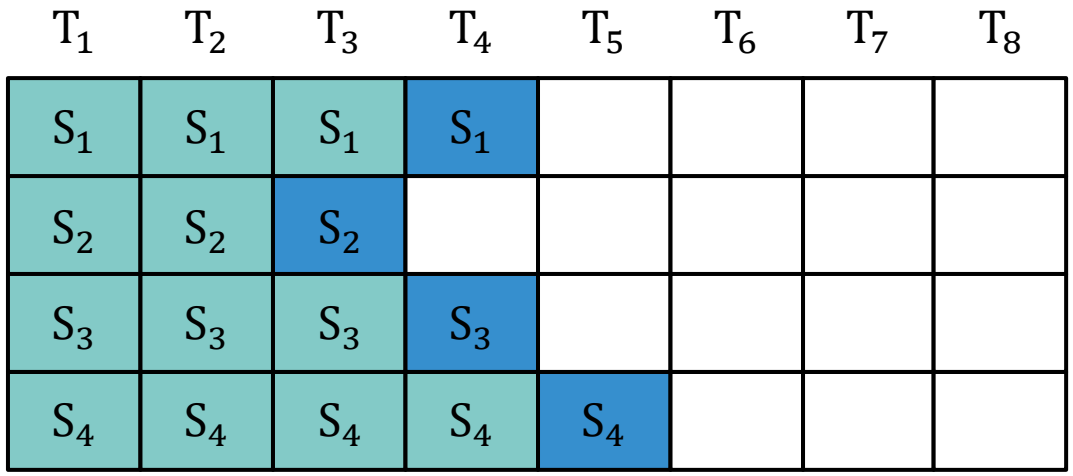


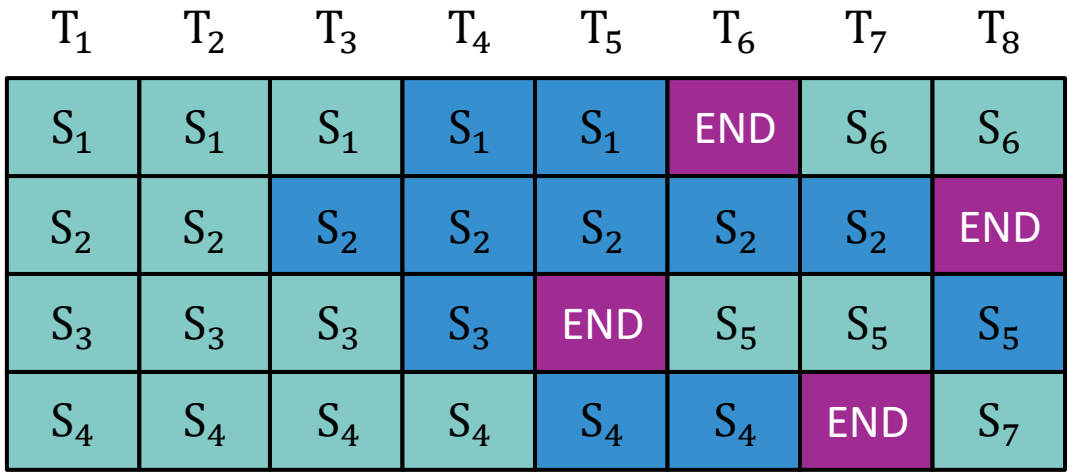


Issues with static batching:

- Requests may complete at different iterations
- Idle GPU cycles
- New requests cannot start immediately

Continuous Batching





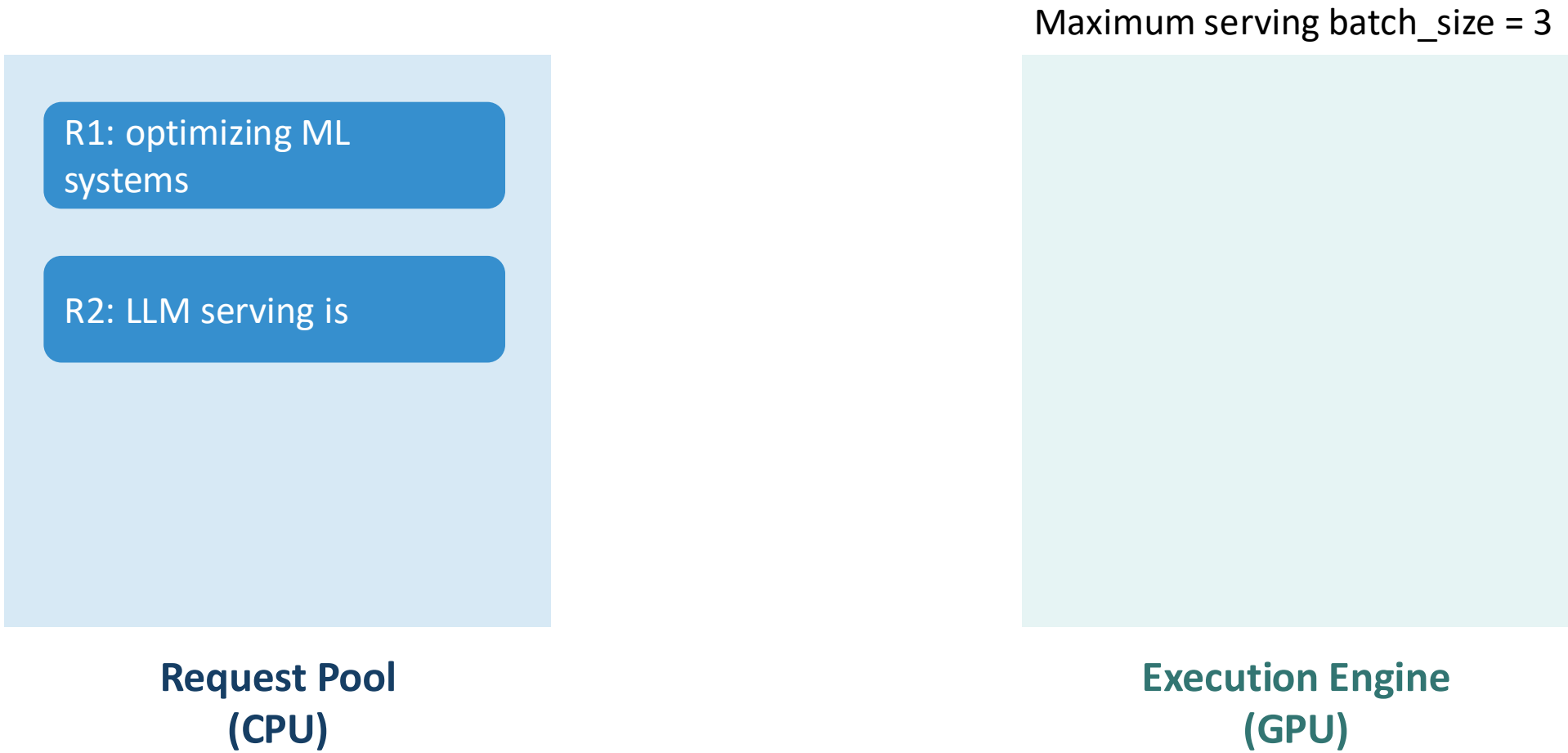
Benefits:

- Higher GPU utilization
- New requests can start immediately

Continuous Batching Step-by-Step

上海创智学院
Shanghai Innovation Institute

- Receives two new requests R1 and R2



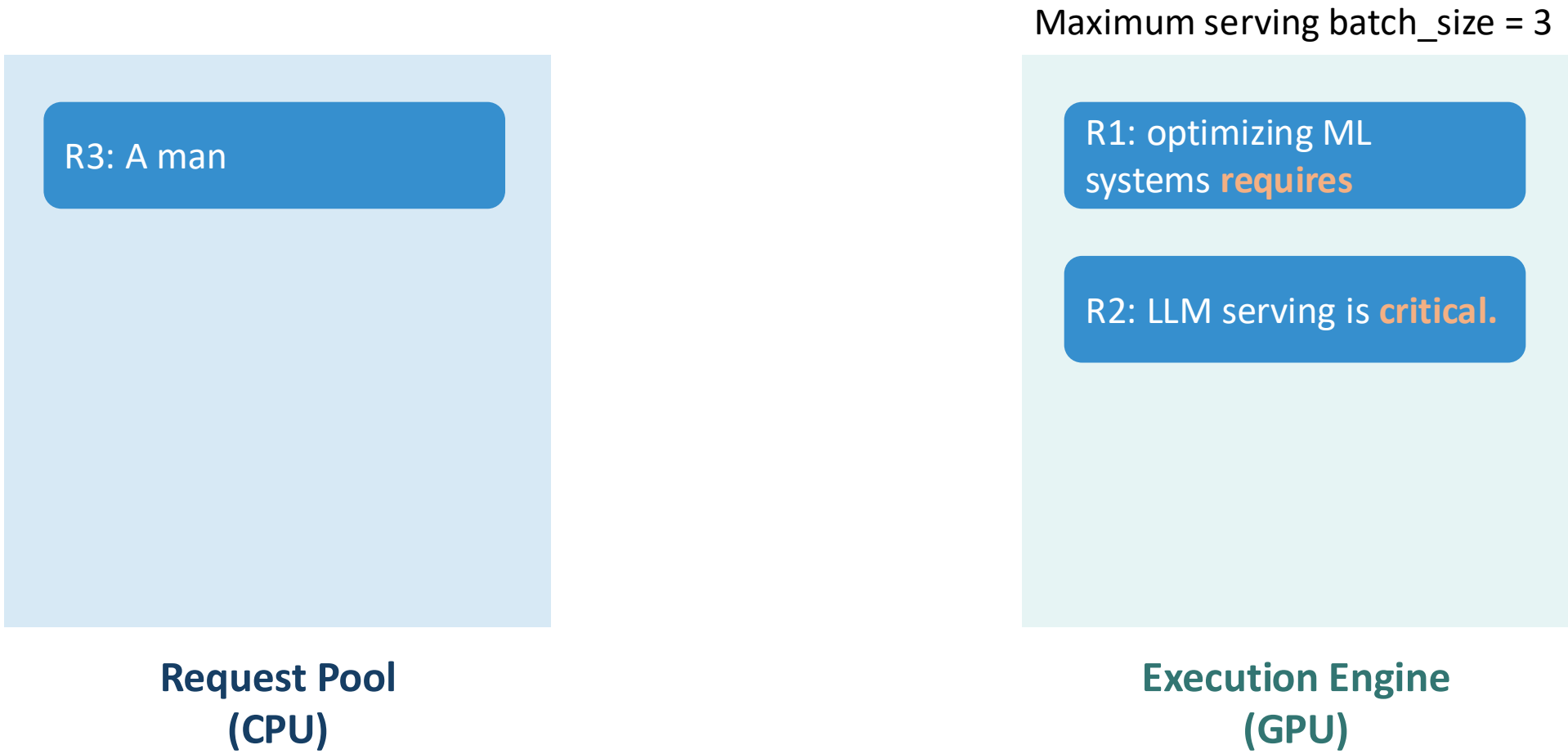
Continuous Batching Step-by-Step

上海创智学院
Shanghai Innovation Institute

- Iteration 1: decode R1 and R2

Continuous Batching Step-by-Step

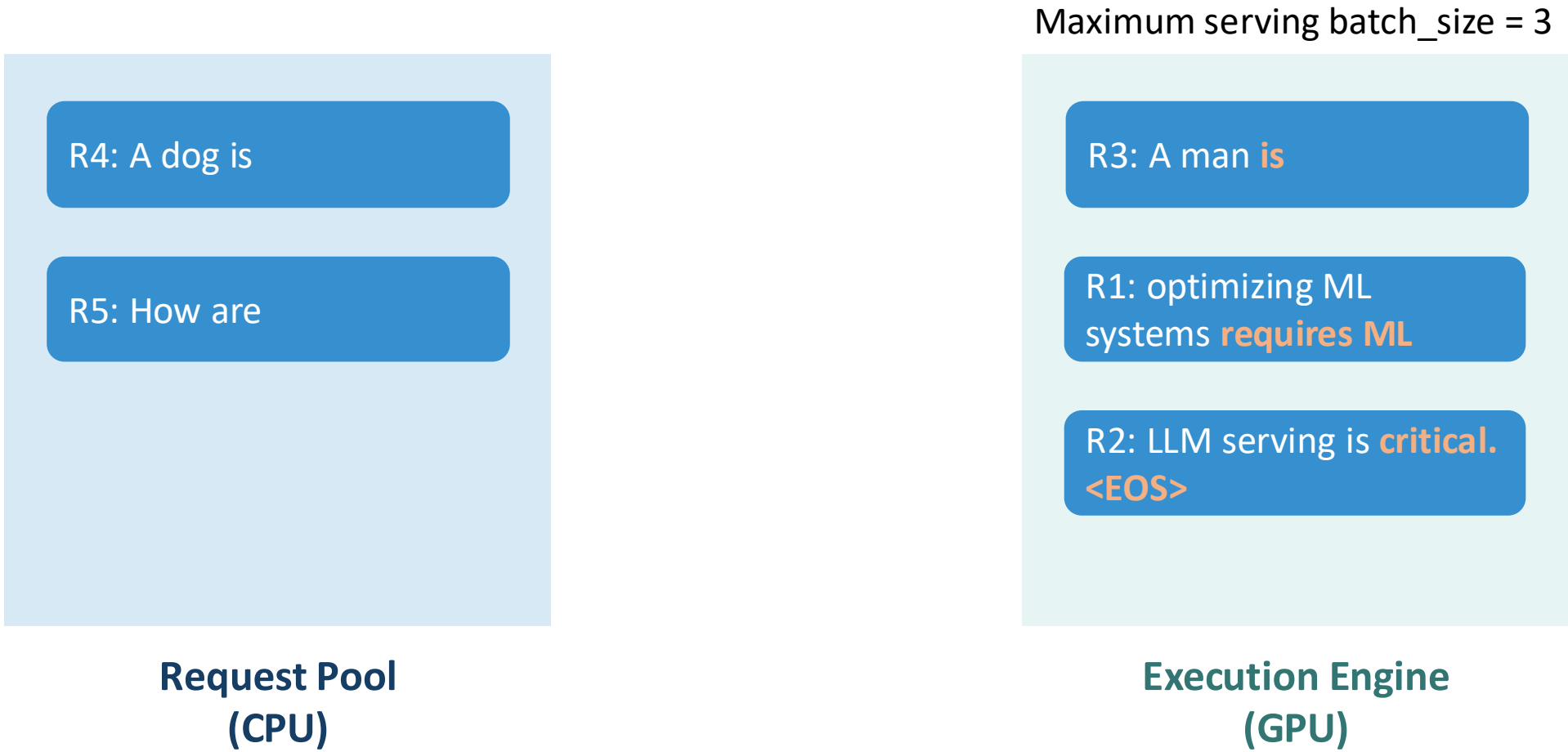
- Receive a new request R3; finish decoding R1 and R2



Continuous Batching Step-by-Step

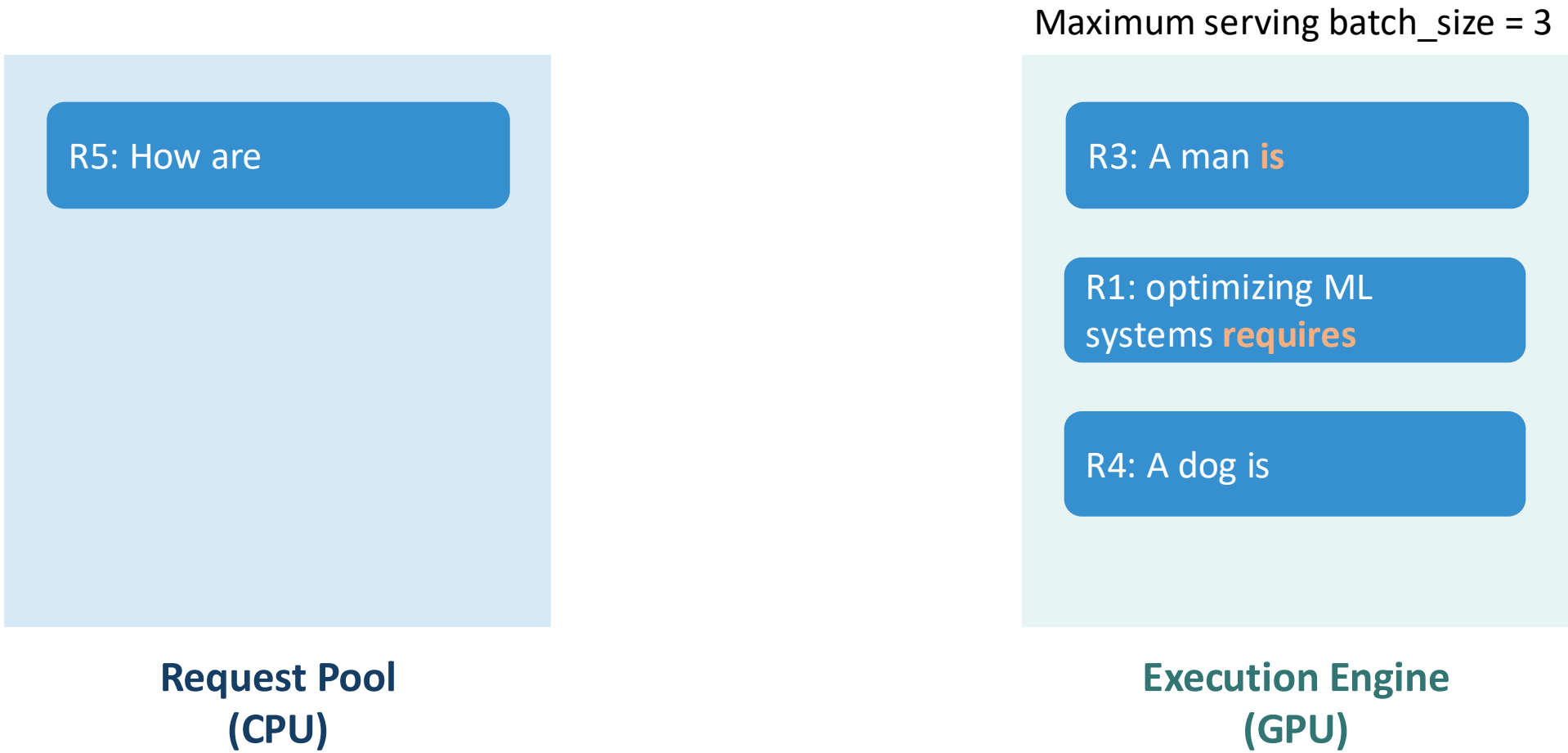
上海创智学院 Shanghai Innovation Institute

- Iteration 2: decode R1, R2, R3; receive R4, R5; R2 completes



Continuous Batching Step-by-Step

- Iteration 3: decode R1, R3, R4



Continuous Batching

上海创智学院
Shanghai Innovation Institute

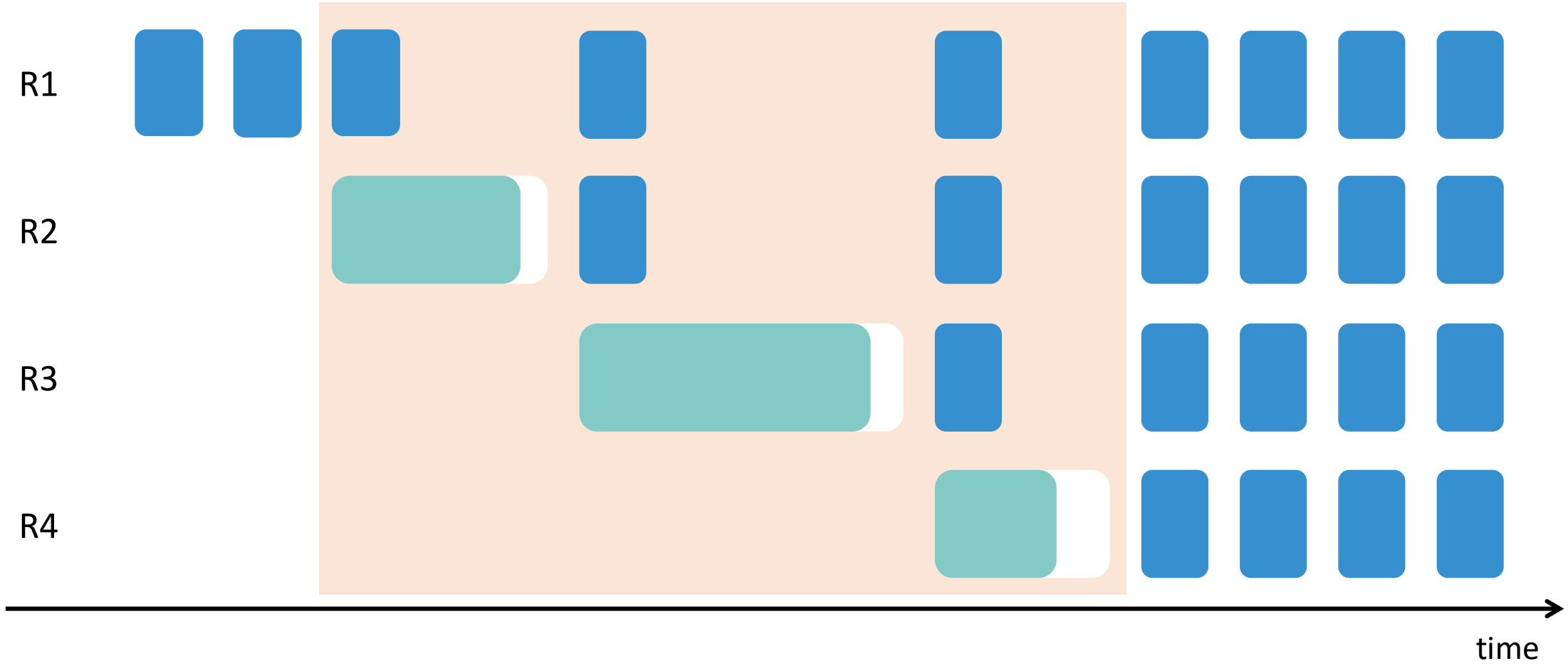
- Handle early-finished and late-arrived requests more efficiently
- Higher GPU utilization

02

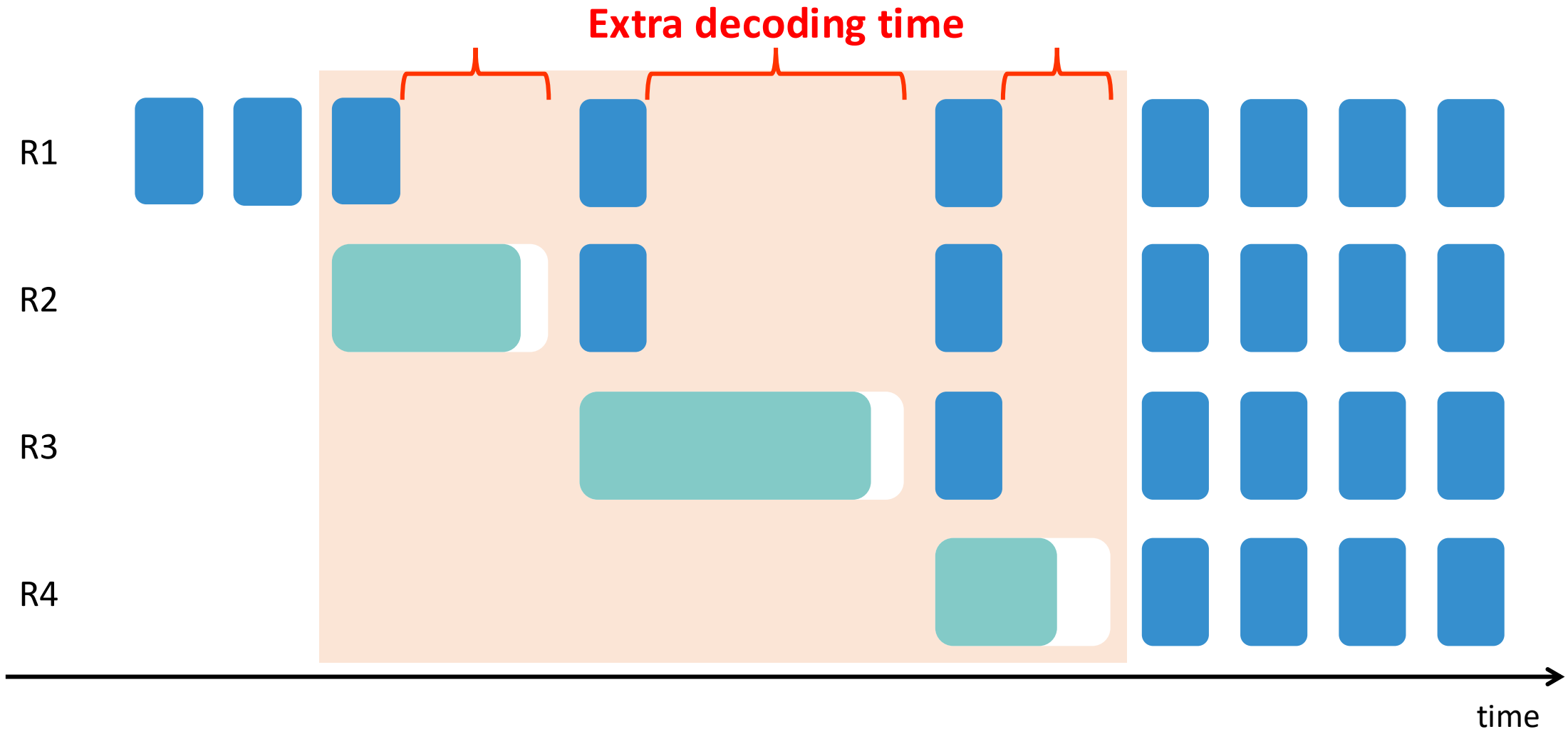
PD Disaggregation

Continuous Batching Hurt both TTFT and TPOT

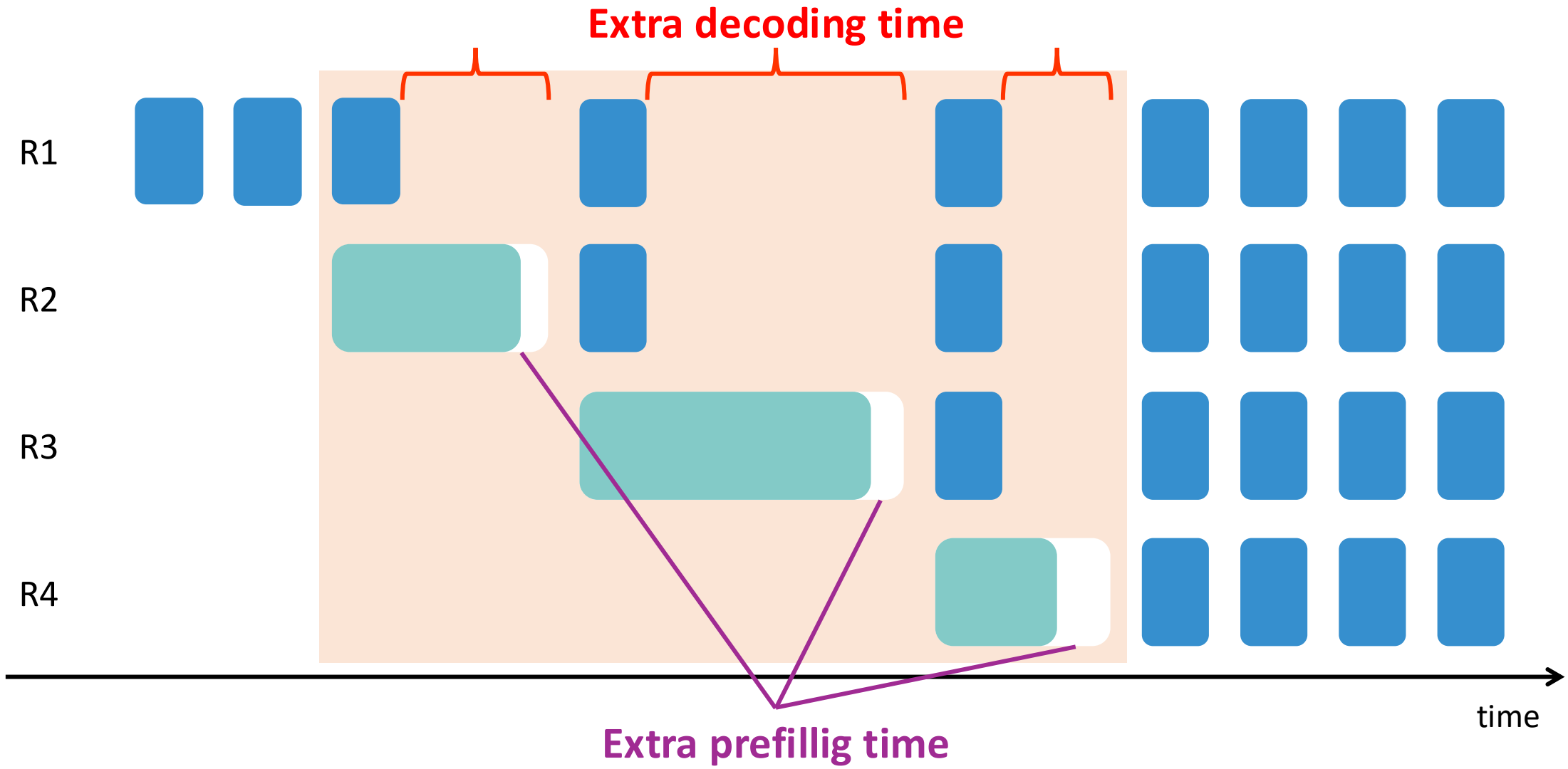
上海创智学院
Shanghai Innovation Institute



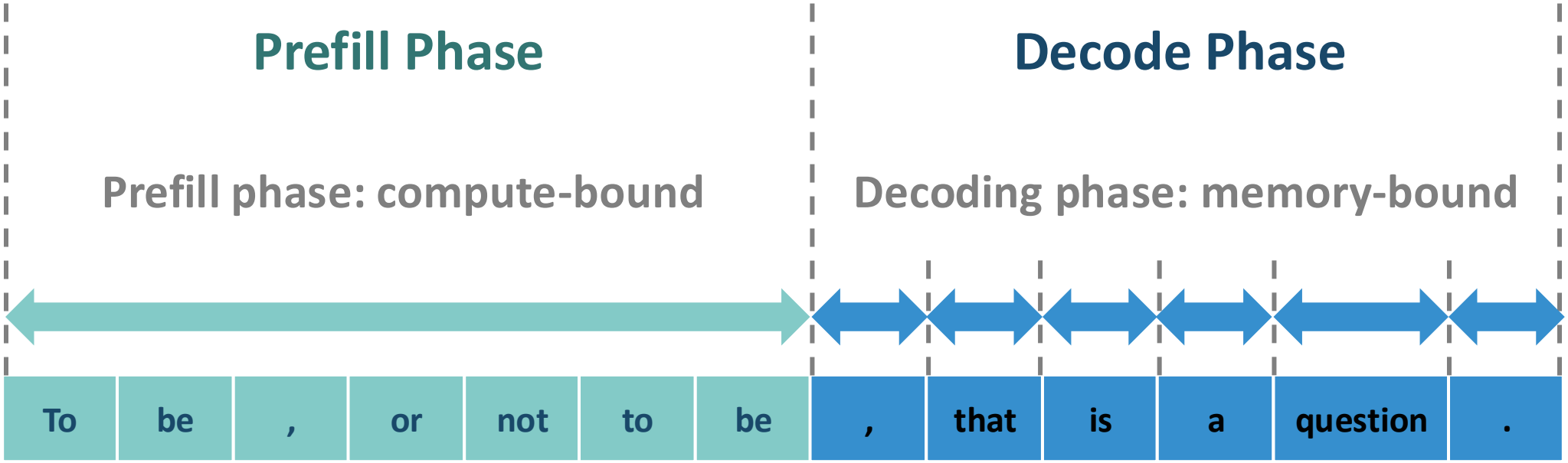
Continuous Batching Hurt both TTFT and TPOT



Continuous Batching Hurt both TTFT and TPOT

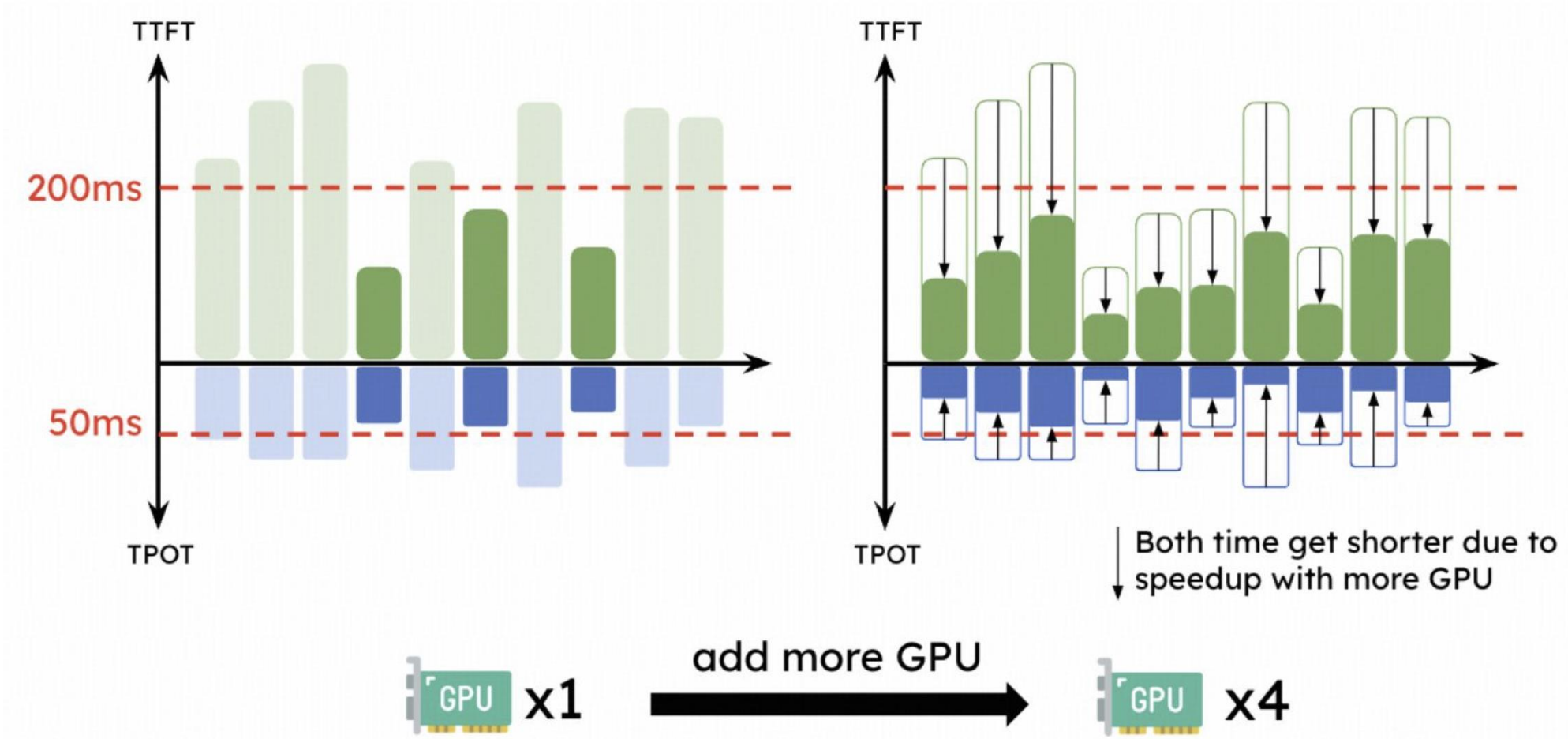


Different Computation Paradigm

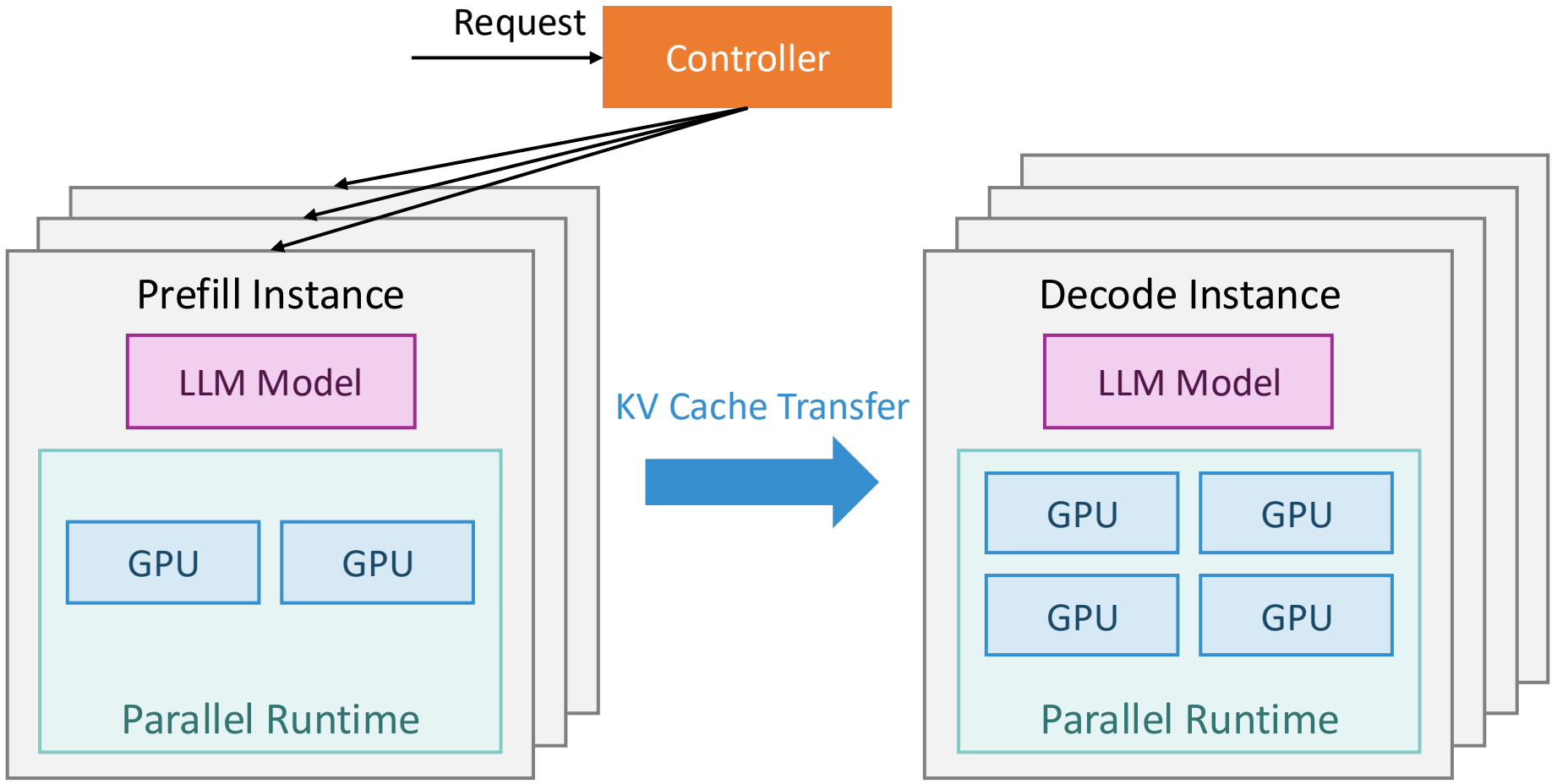


Resource and Parallelism Coupling

Coupling leads to overprovision resources to meet the more demanding SLO



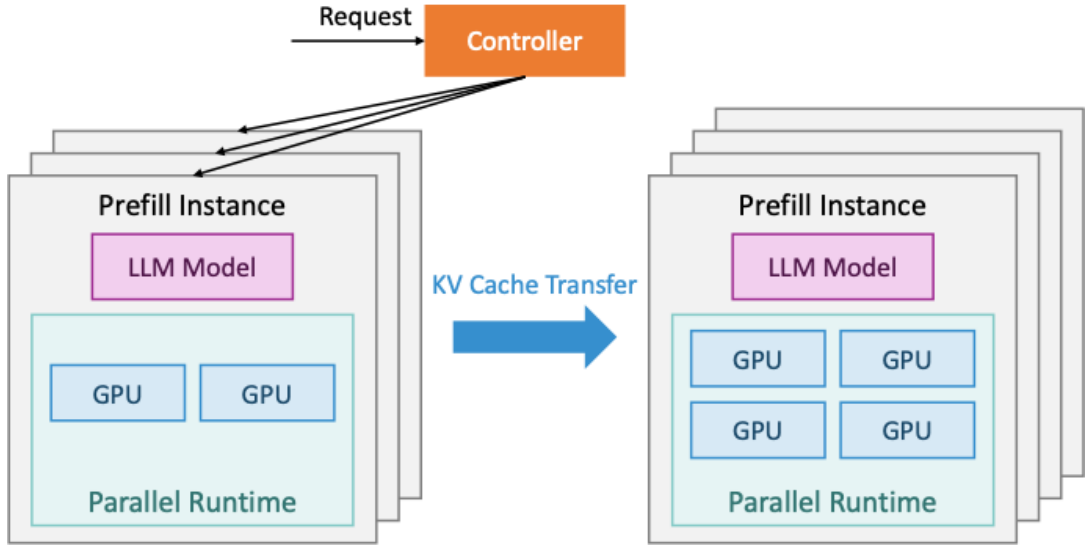
Opportunity: Disaggregating Prefill and Decoding



Opportunity: Disaggregating Prefill and Decoding

上海创智学院
Shanghai Innovation Institute

- Prefill-Decoding interference is immediately eliminated
- Naturally divide the SLO satisfaction problem into two optimizations:
  - Prefill instance optimizes for TTFT.
  - Decoding instance optimizes for TPOT.
  - Choose the most suitable parallelism and resource allocation for each phase.



PD Disaggregation Config for DeepSeek V3

上海创智学院 Shanghai Innovation Institute

- H800 cluster interconnected using NVLink, fully interconnected via IB

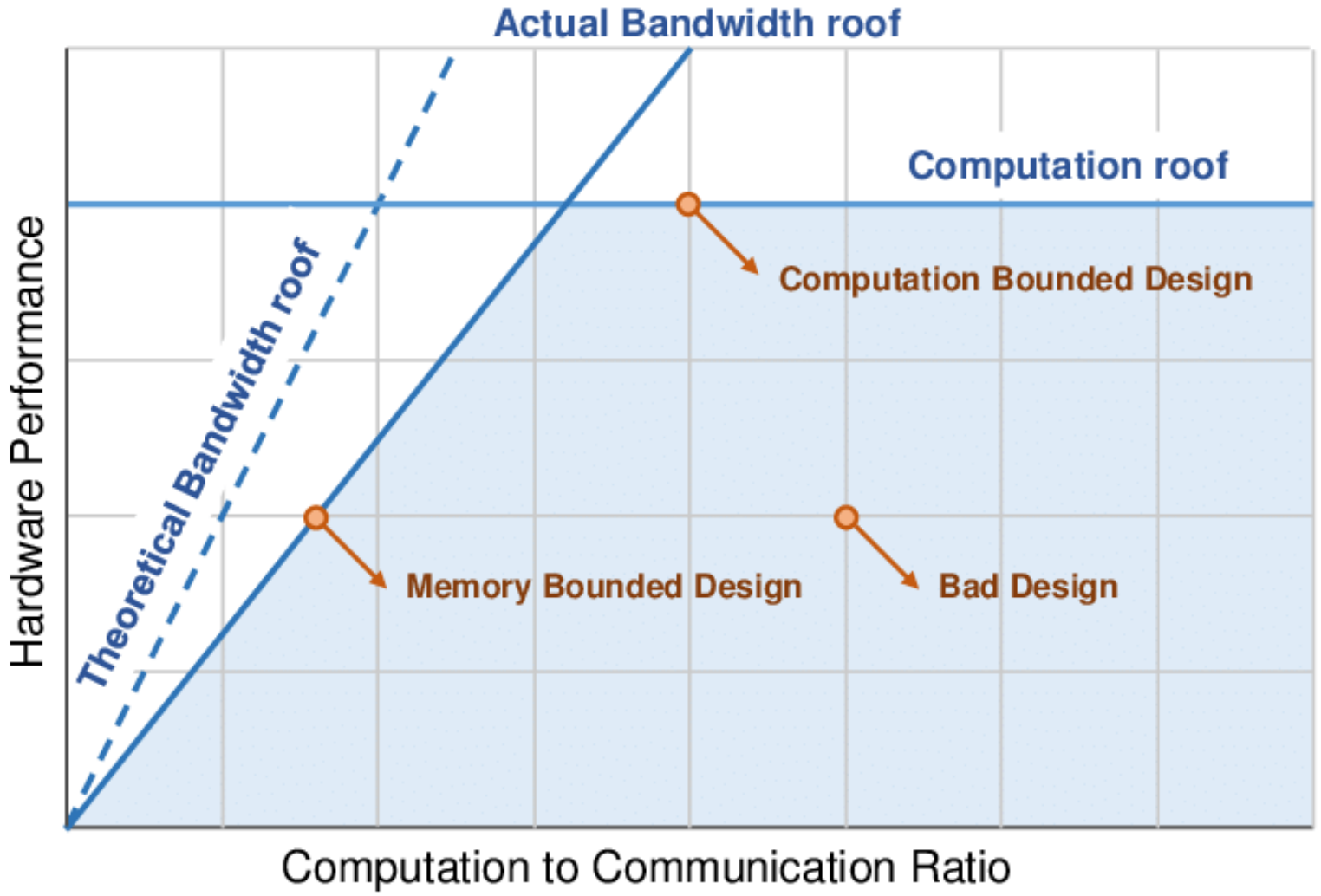
- Prefill – 32 H800 GPU
  - Attention:
    - TP 4 with SP
    - DP 8
  - MoE
    - EP 32

- Decode – 320 H800 GPU
  - Attention:
    - TP 4 with SP
    - DP 80
  - MoE
    - EP 320

Why do we need larger scale for decode stage?

Background: Roofline Model

$$Ratio = \frac{Compute\ FLOPs}{Memory\ Access}$$



Roofline Model for MoE Layer

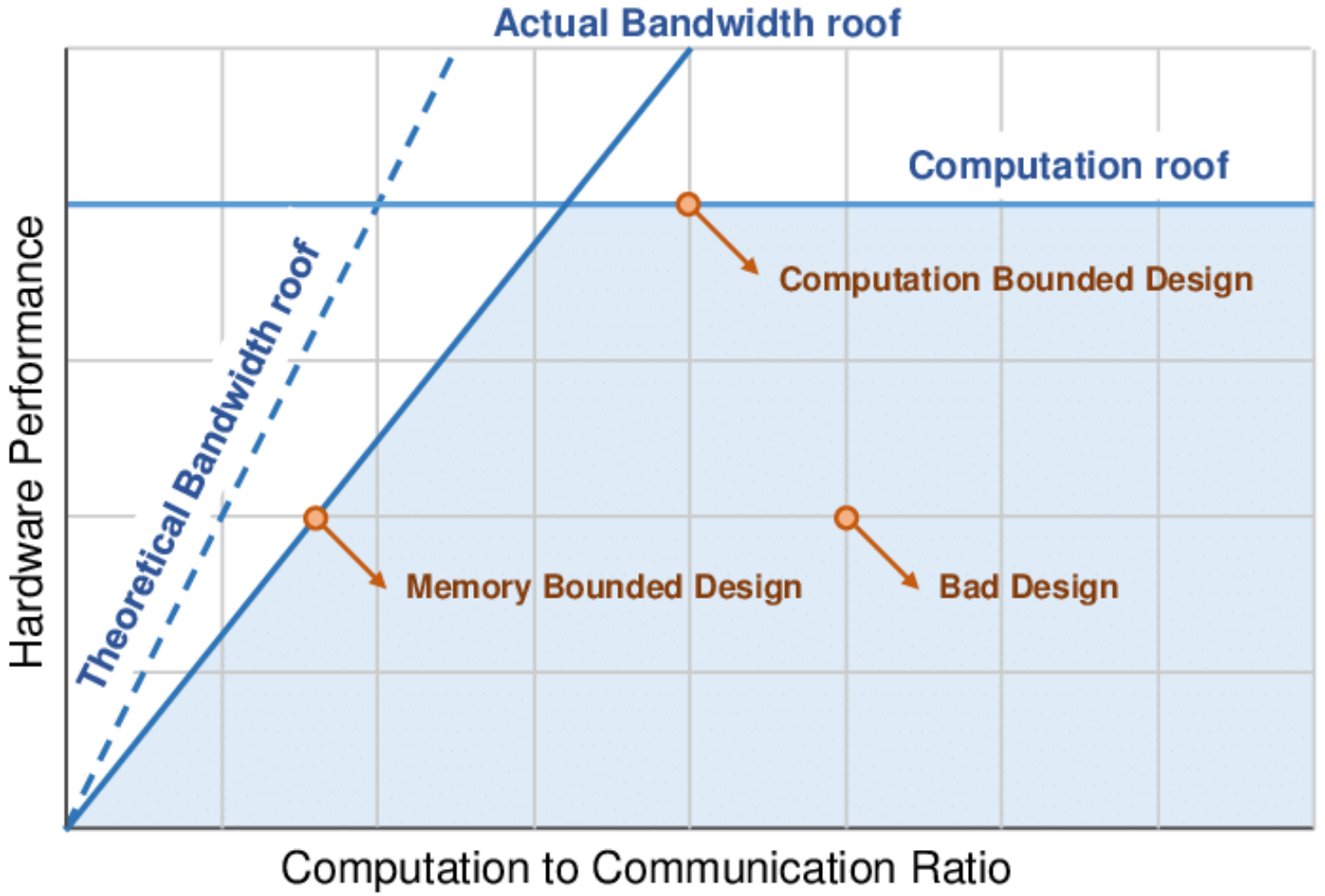
上海创智学院
Shanghai Innovation Institute

For example: Gemm

$$Ratio = \frac{2 \cdot M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

For MoE Gemm

M: Tokens to this expert
$N = 7168$: hidden size
$K = 2048$: intermediate size



Roofline Model for MoE Layer

For example: Gemm

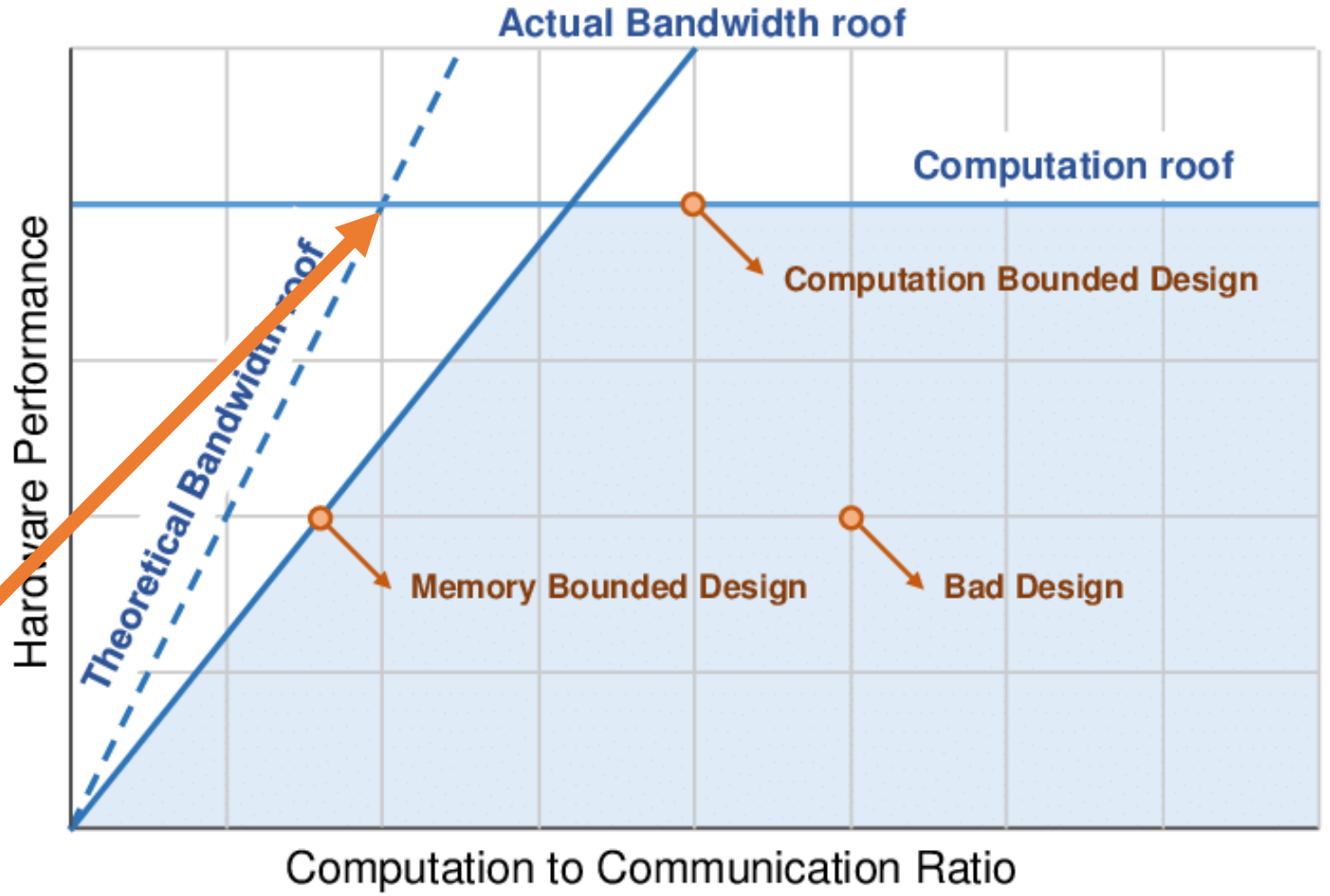
$$Ratio = \frac{2 \cdot M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

For MoE Gemm

M: Tokens to this expert
$N = 7168$: hidden size
$K = 2048$: intermediate size

H100 / H800 hardware roof @fp8

$$Ratio = \frac{1979\ TFlops/s}{3.35TB/s} = 591\ FLOPs/Byte$$



$$M \geq 295 \approx 300, B_{global} \geq 300 \times 32 = 9600, B_{local} \leq 30, \therefore S_{EP} \geq 320$$

Roofline Model for MoE Layer

上海创智学院 Shanghai Innovation Institute

For example: Gemm

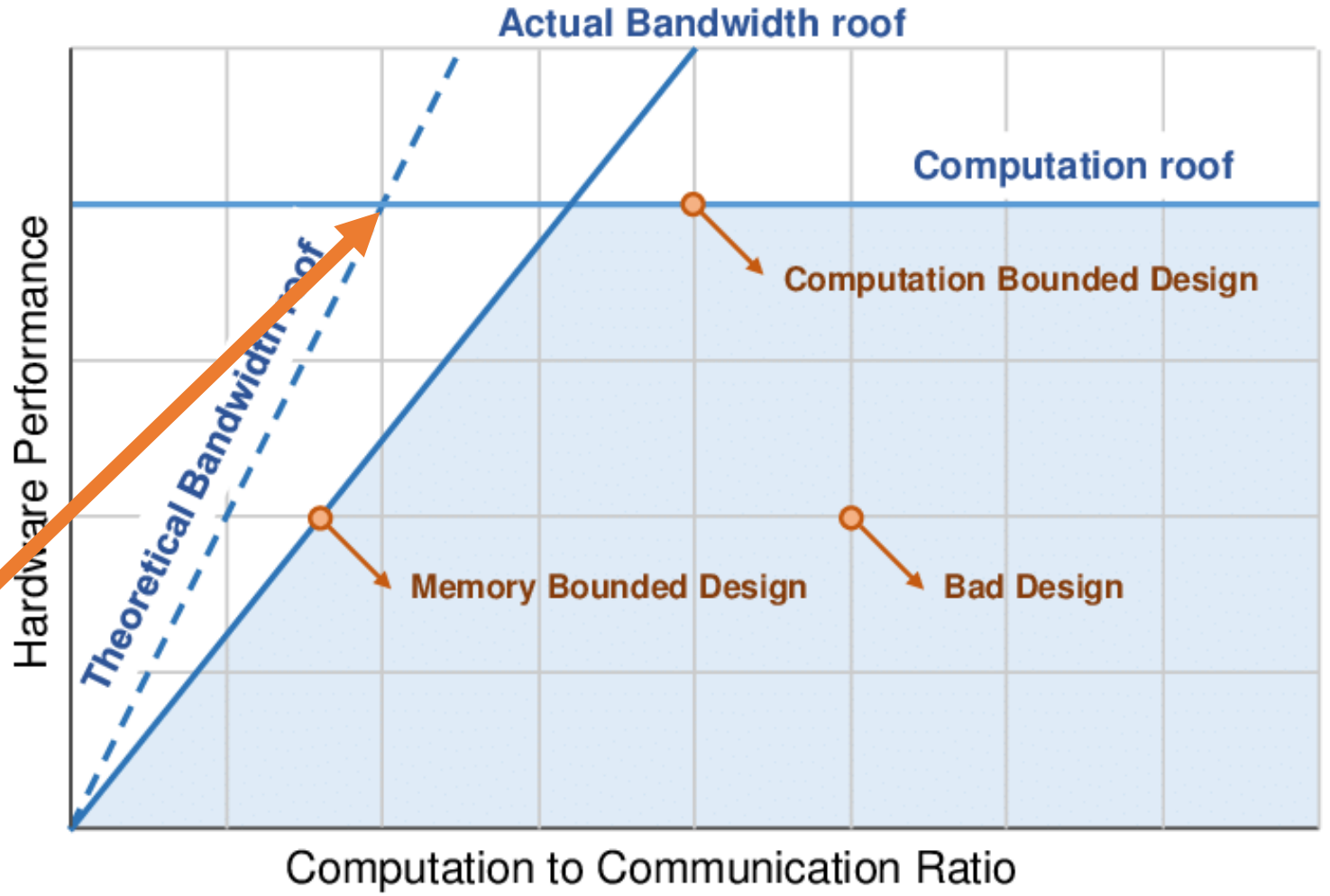
$$Ratio = \frac{2 \cdot M \cdot N \cdot K}{M \cdot K + N \cdot K + M \cdot N}$$

For MoE Gemm

M: Tokens to this expert
$N = 7168$: hidden size
$K = 2048$: intermediate size

H20 hardware roof @fp8

$$Ratio = \frac{296\ TFlops/s}{4\ TB/s} = 74\ FLOPs/Byte$$

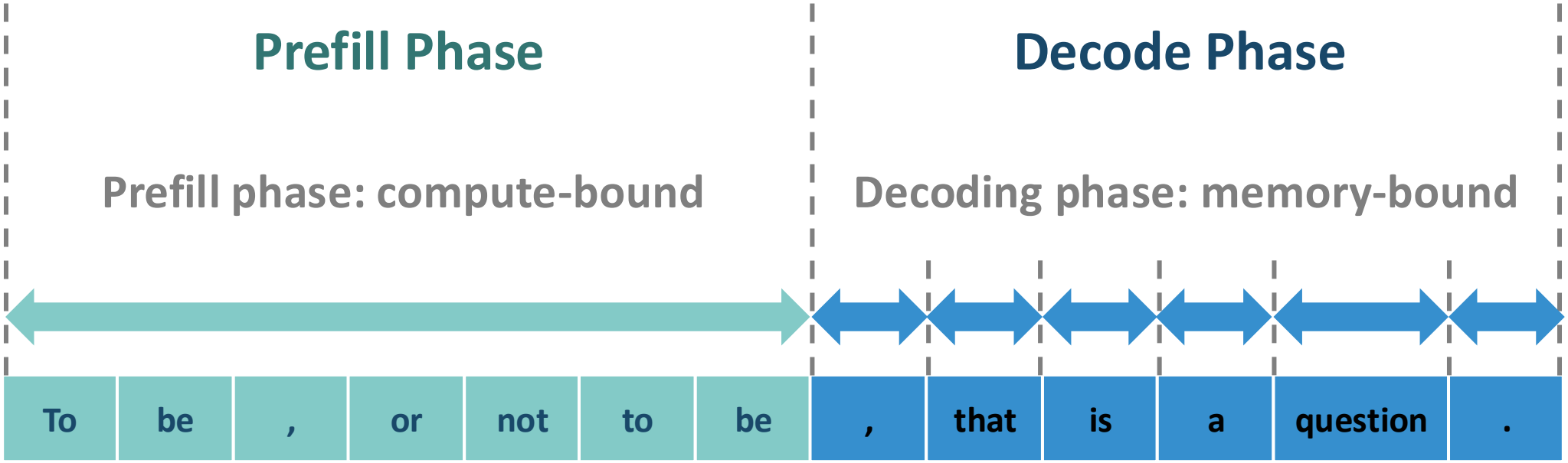


$M \geq 37 \approx 40, B_{global} \geq 40 \times 32 = 1280, B_{local} \leq 30, \therefore S_{EP} \geq 42$

03

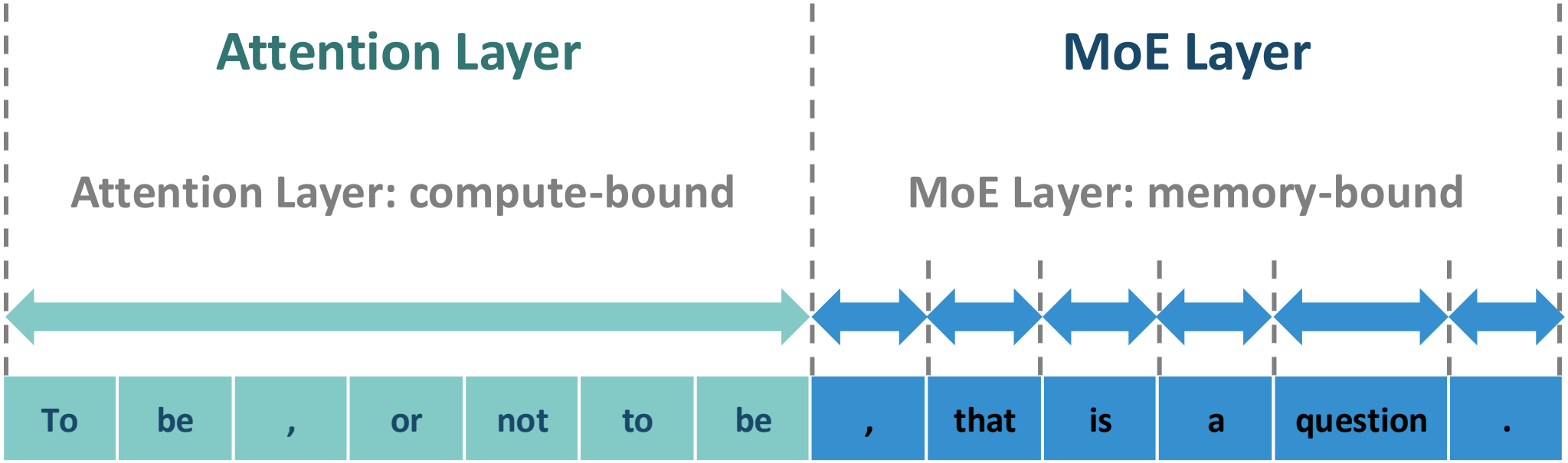
AF Disaggregation

Recap: Different Computation Paradigm

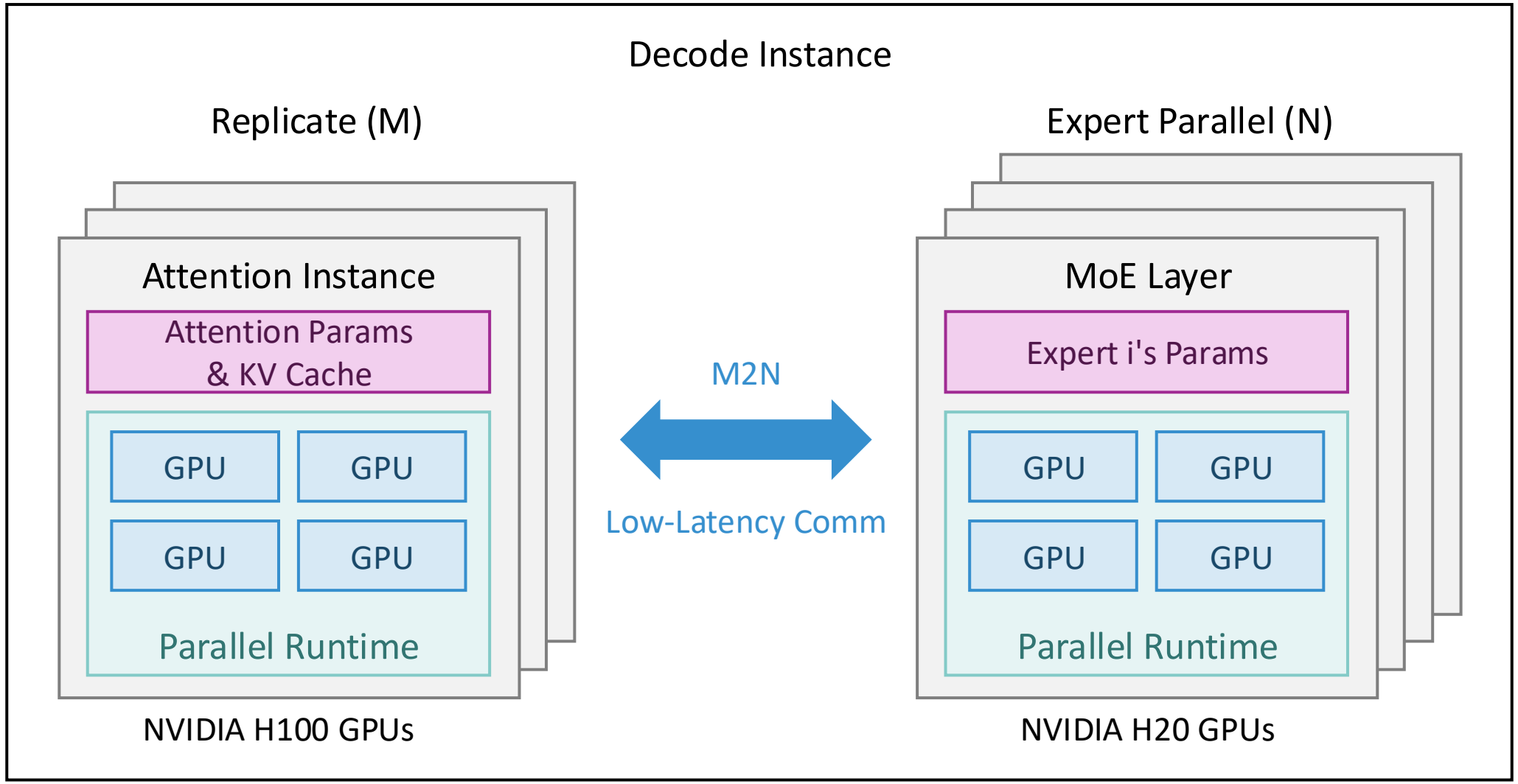


Different Computation Paradigm

上海创智学院
Shanghai Innovation Institute



Disaggregating Attention and MoE Layer



Pipeline Parallelism in AF-Disaggregation

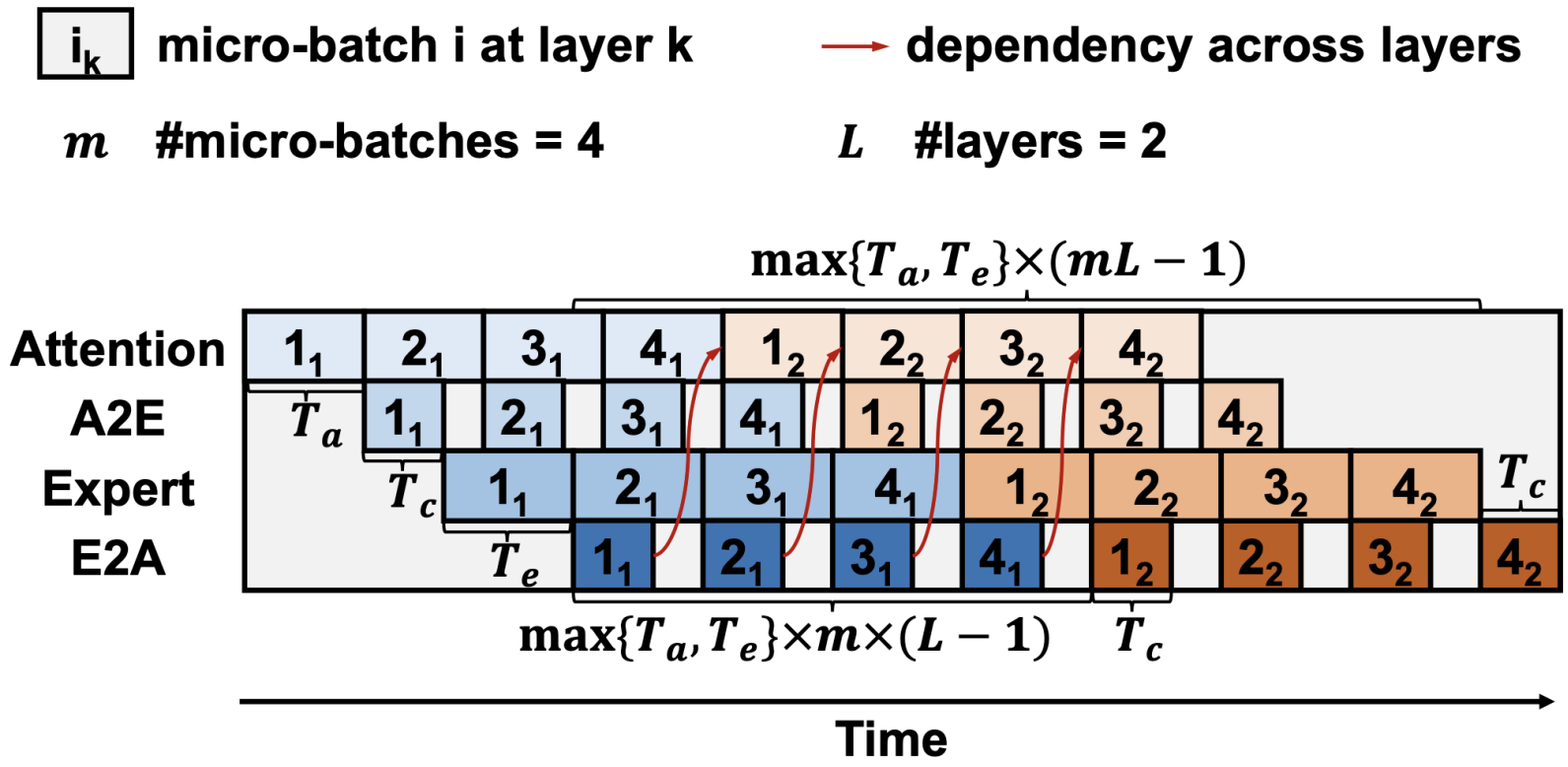
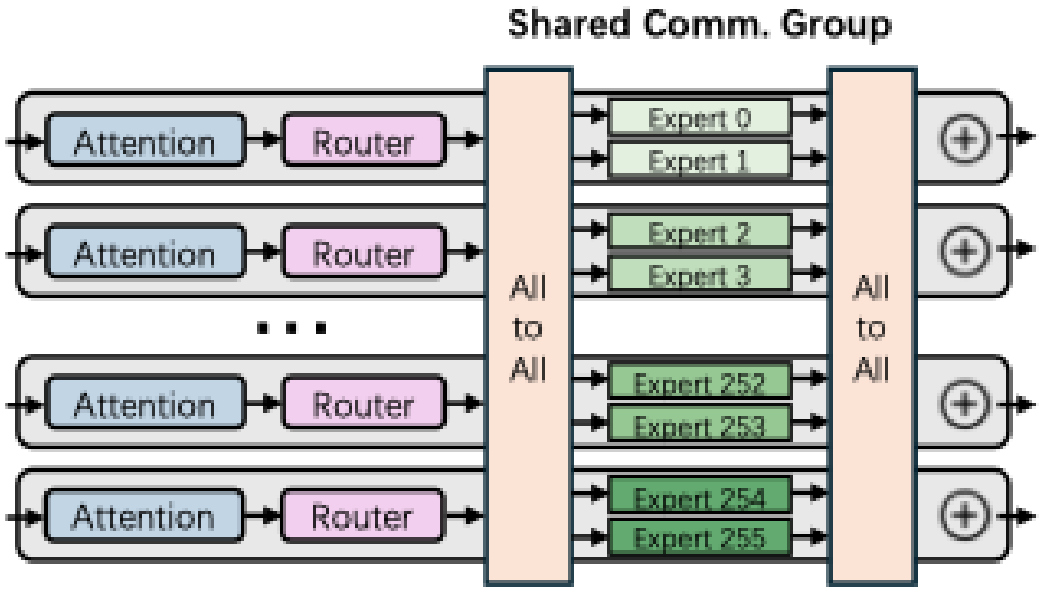
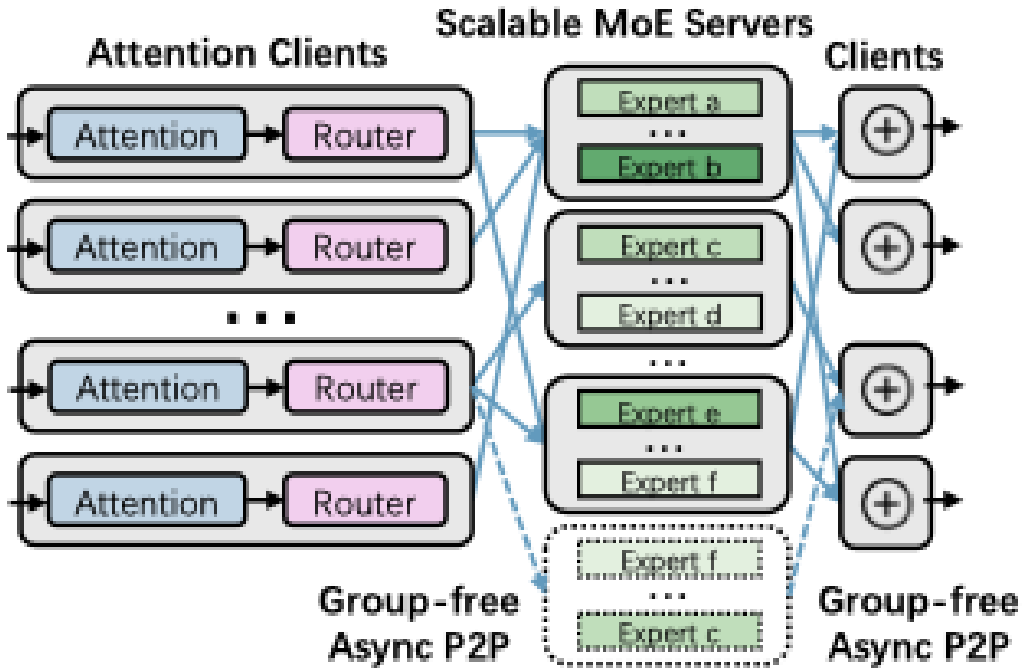


Figure 4 Illustration of ping-pong pipeline parallelism.

Expert-as-a-Service

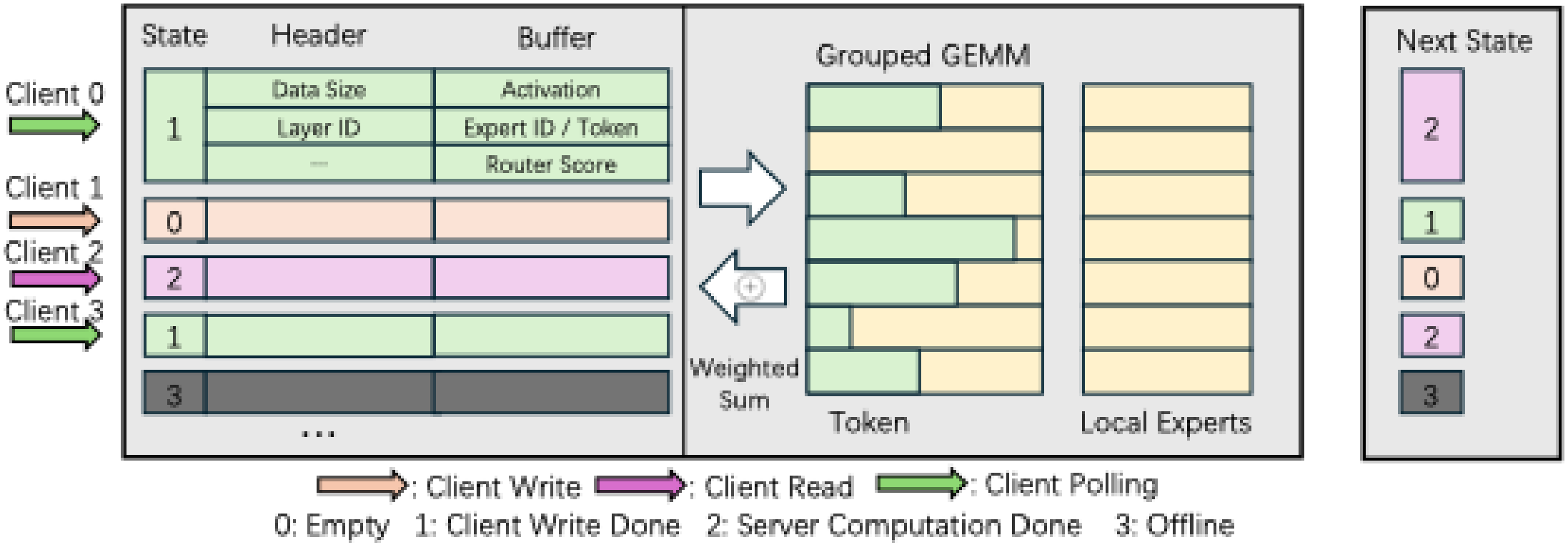


(a) Standard Expert Parallelism



(b) Decoupled EaaS Expert Service

Expert Server Design



Comparison between PDD and AFD

上海创智学院
Shanghai Innovation Institute

PD-Disaggregating

- Transfer **once** for single request
- Require **high throughput**
- Utilize GPU computation
- For better **SLO**

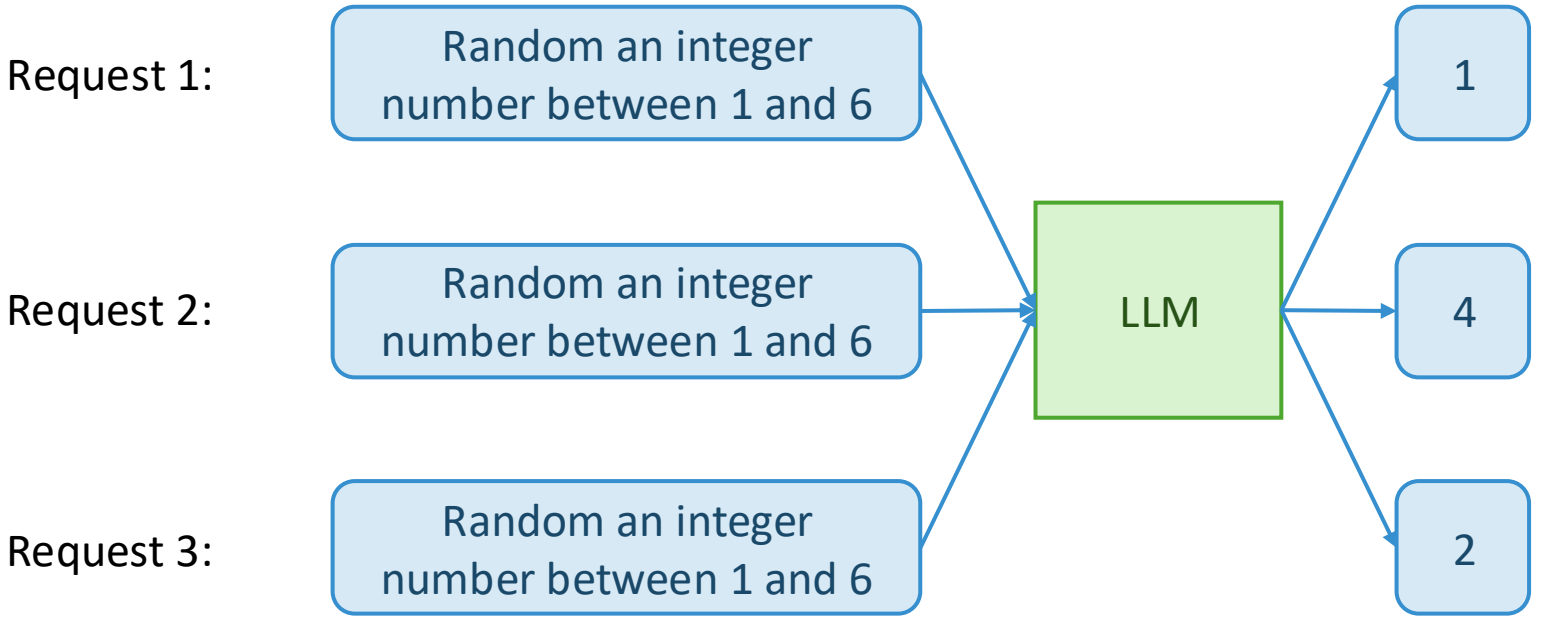
AF-Disaggregating

- Transfer **multi times** for single token
- Require **low latency**
- Utilize GPU computation
- For **fault tolerance**

04

Prefix Cache

Same Prompt Prefix



Recap: Paged Attention

Prompt: “Random an integer number between 1 and 6”

Request 1

Logical KV blocks

Random	an	integer	number
between	1	and	6

Request 2

Logical KV blocks

Random	an	integer	number
between	1	and	6

Physical KV blocks

				block 0
between	1	and	6	block 1
				block 2
				block 3
				block 4
				block 5
				block 6
Random	an	integer	number	block 7

Recap: Paged Attention

Prompt: “Random an integer number between 1 and 6”

Request 1

Logical KV blocks

Random	an	integer	number
between	1	and	6
1			

Request 2

Logical KV blocks

Random	an	integer	number
between	1	and	6
4			

Physical KV blocks

				block 0
between	1	and	6	block 1
1				block 2
				block 3
4				block 4
				block 5
				block 6
Random	an	integer	number	block 7

Paged Attention Natively Support Prefix Cache

上海创智学院
Shanghai Innovation Institute

However:

- The paged attention prefix cache can be only reused in an instance
- The GPU memory is limited, and can not store history prefix cache
- Real world contains many requests with the same prefix cache

MoonCake: Distributed KVCache Pool

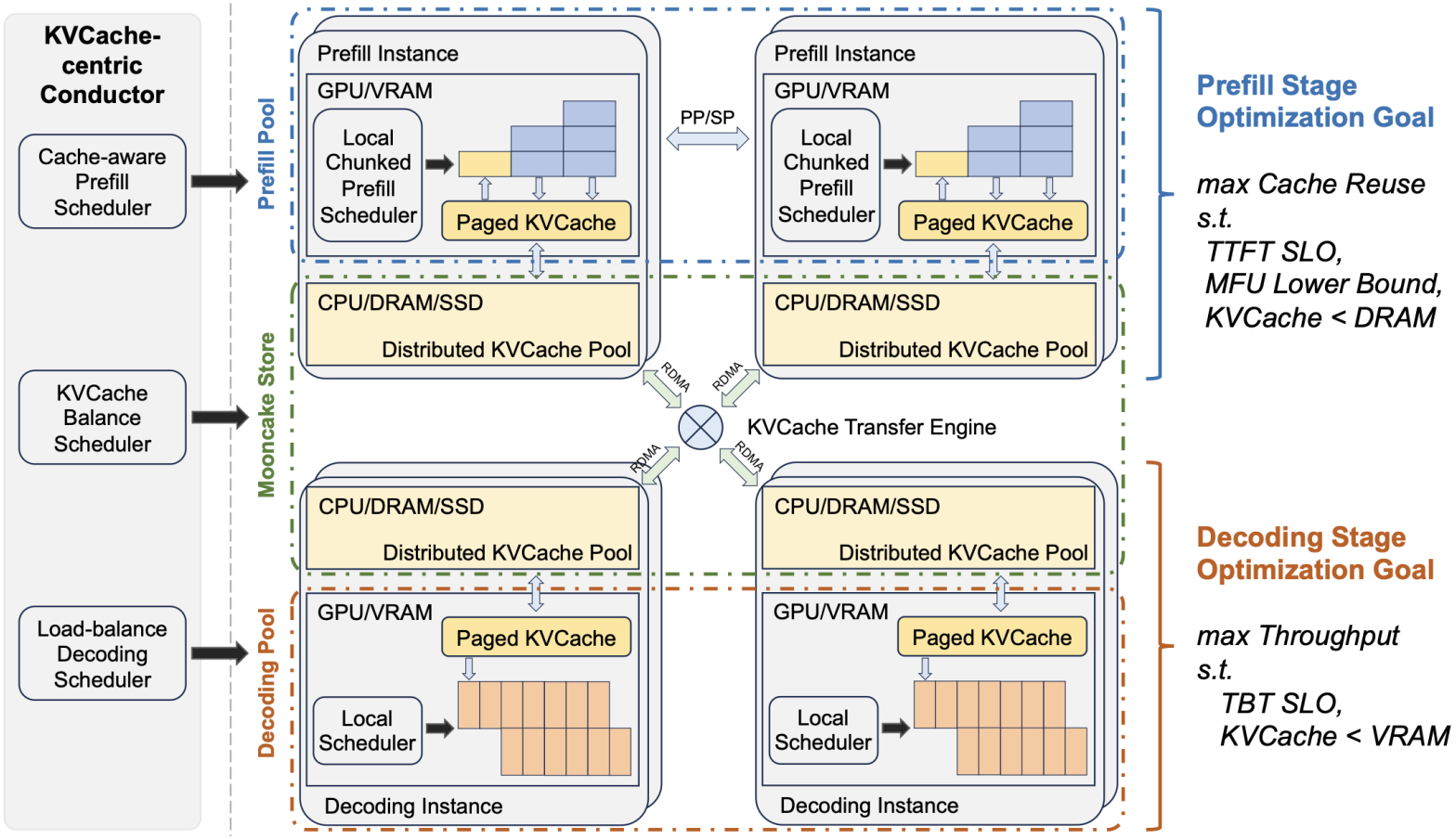


Figure 2: MOONCAKE Architecture.

Acknowledgement

The development of this course, including its structure, content, and accompanying presentation slides, has been significantly influenced and inspired by the excellent work of instructors and institutions who have shared their materials openly. We wish to extend our sincere acknowledgement and gratitude to the following courses, which served as invaluable references and a source of pedagogical inspiration:

- Machine Learning Systems[15-442/15-642], by **Tianqi Chen** and **Zhihao Jia** at **CMU**.
- Advanced Topics in Machine Learning (Systems)[CS6216], by **Yao Lu** at **NUS**

While these materials provided a foundational blueprint and a wealth of insightful examples, all content herein has been adapted, modified, and curated to meet the specific learning objectives of our curriculum. Any errors, omissions, or shortcomings found in these course materials are entirely our own responsibility. We are profoundly grateful for the contributions of the educators listed above, whose dedication to teaching and knowledge-sharing has made the creation of this course possible.

System for Artificial Intelligence

Thanks

Siyuan Feng
Shanghai Innovation Institute