IMPRESS: An Importance-Informed Multi-Tier Prefix KV Storage System for Large Language Model Inference

Weijian Chen, Shuibing He, Haoyang Qu, Ruidong Zhang,
Siling Yang, Ping Chen, Yi Zheng [$], Baoxing Huai [$], Gang Chen

浙江大学 Zhejiang University

$ HUAWEI CLOUD

USENIX FAST 2025

Large Language Model (LLM) Inference

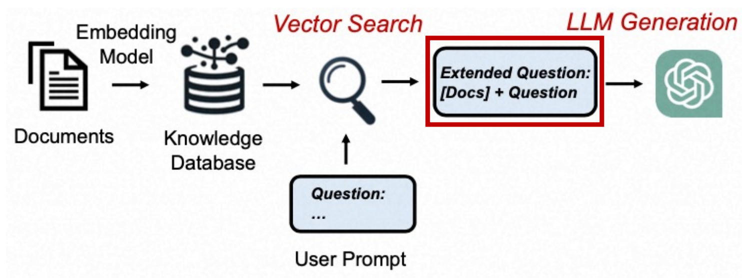
- LLM has been applied in a range of fields

Chat	Search	Code	Q&A
ChatGPT	perplexity	CURSOR	CHATPDF

- Context-rich prefixes + user queries = LLM requests

- Many requests share identical prefixes

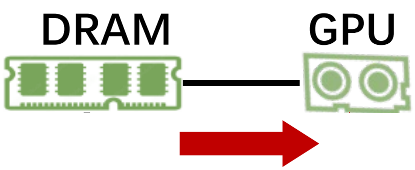
Prompt:
[Instructions]
You are an AI chatbot. You are having a conversation with a human by following rules:
- You do not have a name.
- You are helpful, creative, clever, and friendly
...
[Examples]
Human: Hello, who are you?
AI: I am an AI chatbot. How can I help you?
...
[Question]
Human: Tell me about the second world war.

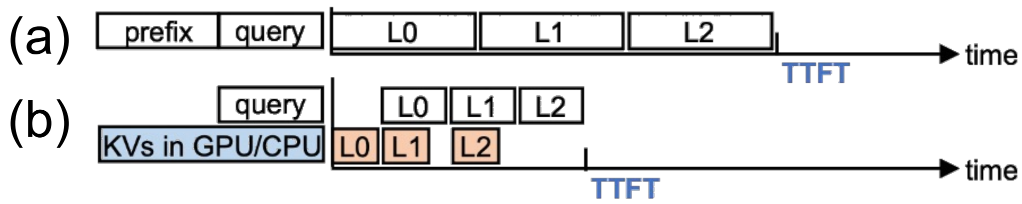


* Image Source: Internet

Prefix KV Storage System

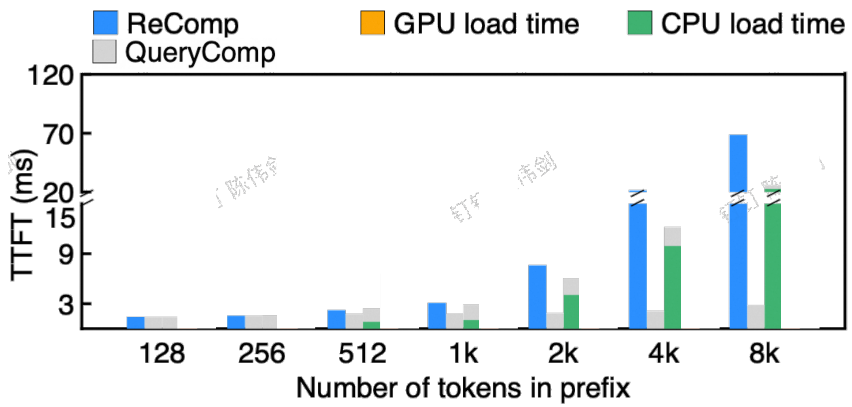
➢ Shared prefix KVs can be **restored and reused**





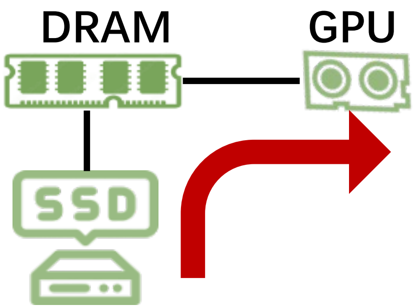
* Assume a three-layer simple LLM

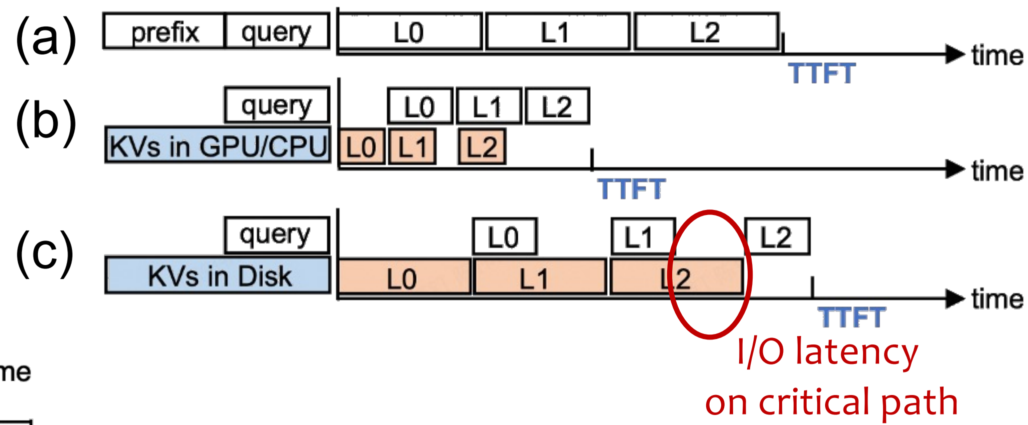
Time-to-First-Token (TTFT) can be reduced.

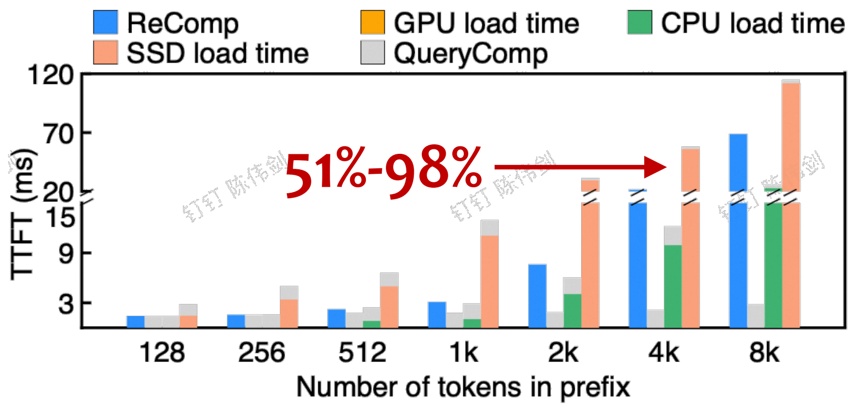


Prefix KV Storage System

- When shared prefix KVs needs to be stored into **SSD**







Prefix KV loading from SSD to GPU has become a new bottleneck

Related Work

- Most existing systems store prefix KVs only in GPU and/or CPU memory

  PromptCache-MLSys24, RAGCache-arxiv24, ChunkAttention-arxiv24, SGLang-arxiv23

Limited space in GPU and CPU memory quickly becomes exhausted

- Pre-loads them into CPU memory based on the scheduler's predictions

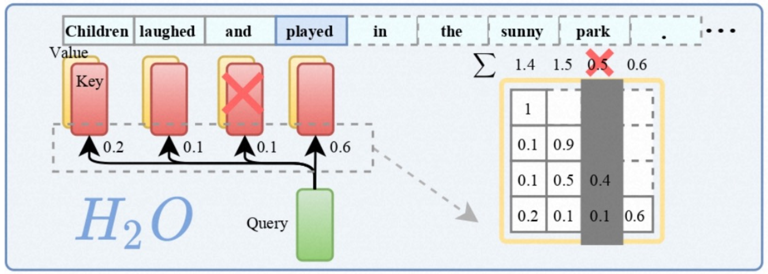
  AttentionStore-ATC24

Limitations exists under high request volumes or in preemptive scheduling

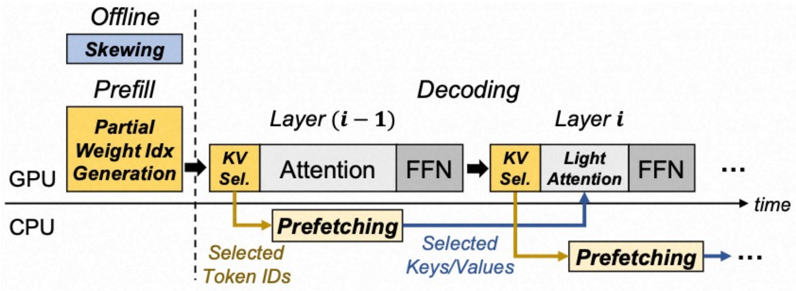
Is it possible to reduce KV data that needs to be loaded?

Opportunity from KV Importance

➢ Only preserve important KVs during decoding phase achieves the same level accuracy



H2O-NeurIPS23

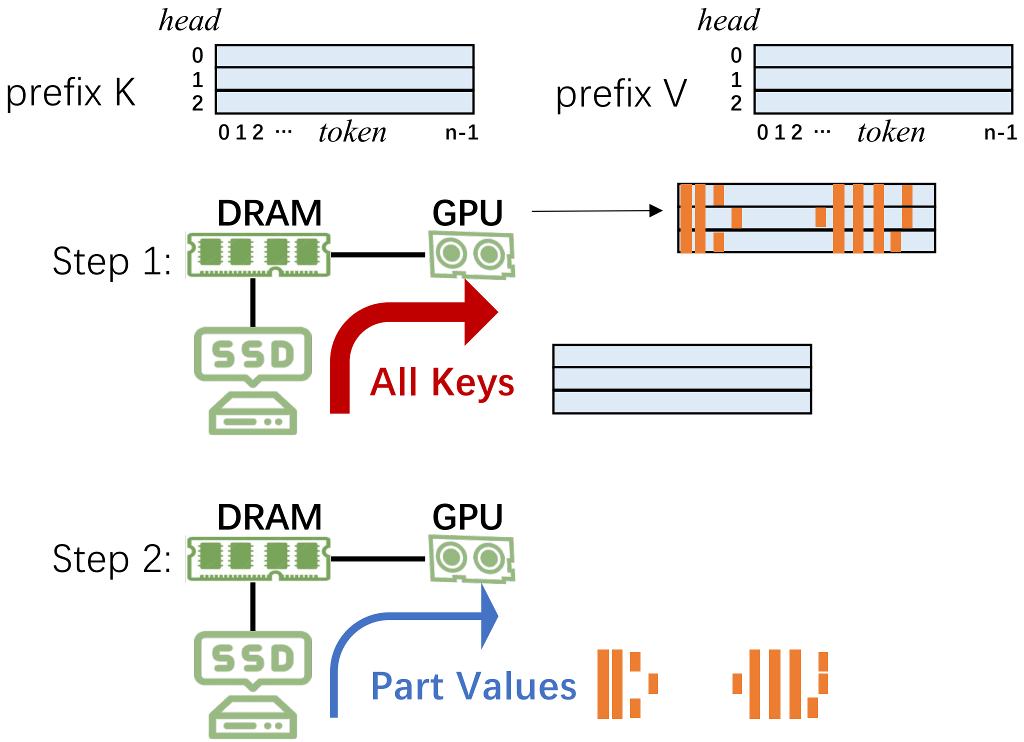


InfiniGen-OSDI24

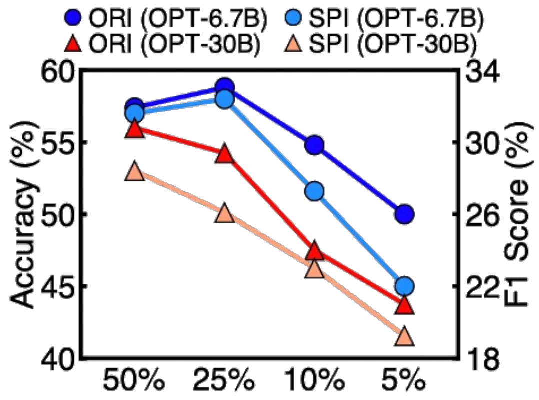
How about only load important KVs during prefill to reduce I/O bottleneck and TTFT?

Challenge 1

➢ **A large amount of I/O is introduced to identify important KVs.**



- Pre-determine important KVs?
  Accuracy Drop.

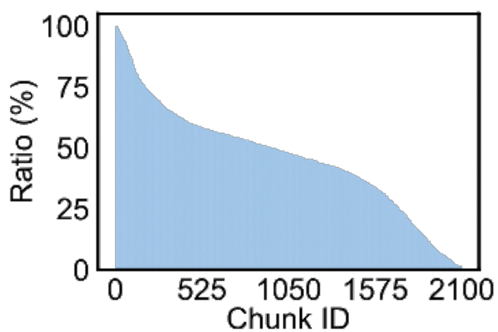


- SPI: statically pre-determine importance
- ORI: original dynamically determine importance

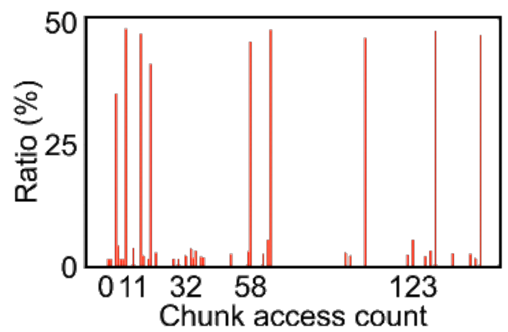
Challenge 2

➢ **The existing prefix KV storage and caching systems are suboptimal considering the importance of tokens' KVs.**

1. Storage： read amplification (Each chunk contains a mix of important and unimportant KVs.)
2. Caching： based solely on recency or frequency (ignore the importance of KVs)



(a) The ratio of important KVs within each chunk.

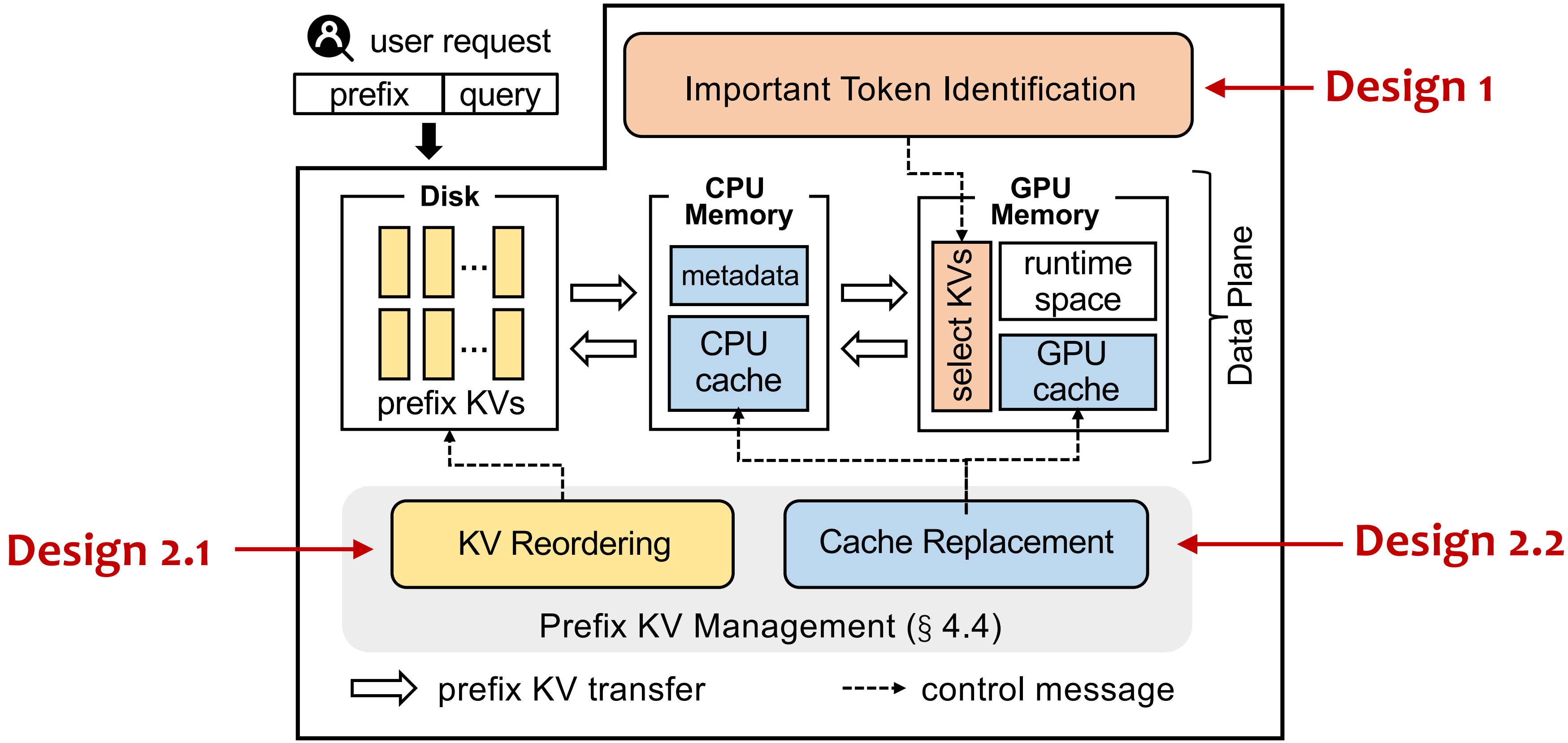


(b) Average ratio of important tokens in all chunks for a given chunk access frequency.

Outline

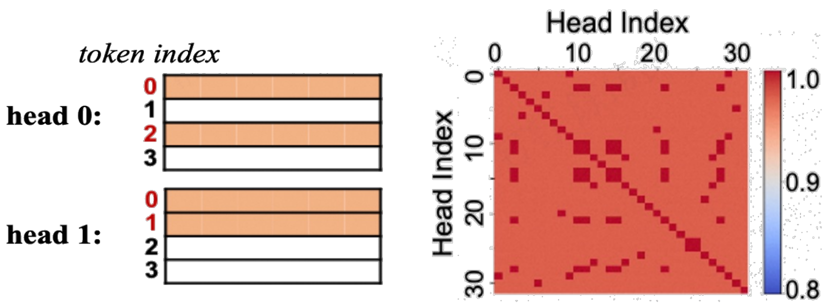
- Background & Motivation
- Observation & Design of IMPRESS
- Evaluation
- Summary & Conclusion

IMPRESS Architecture



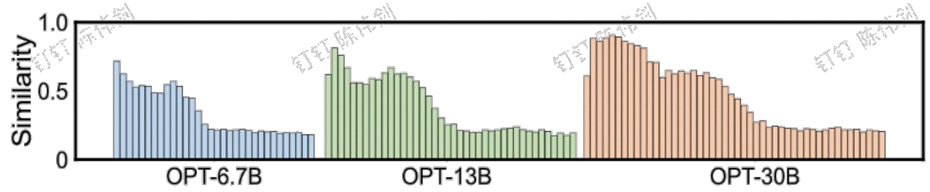
Observation

- There is a high similarity in the set of important token indices across different heads within the same layer of an LLM.

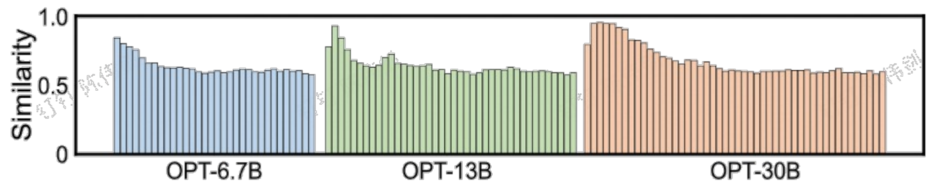


Similarity measurement:

h0={0, 2} h1={0, 1} $J(h_0, h_1) = \frac{|h_0 \cap h_1|}{|h_0 \cup h_1|} = \frac{1}{3}$



(a) select the top 10% most important tokens

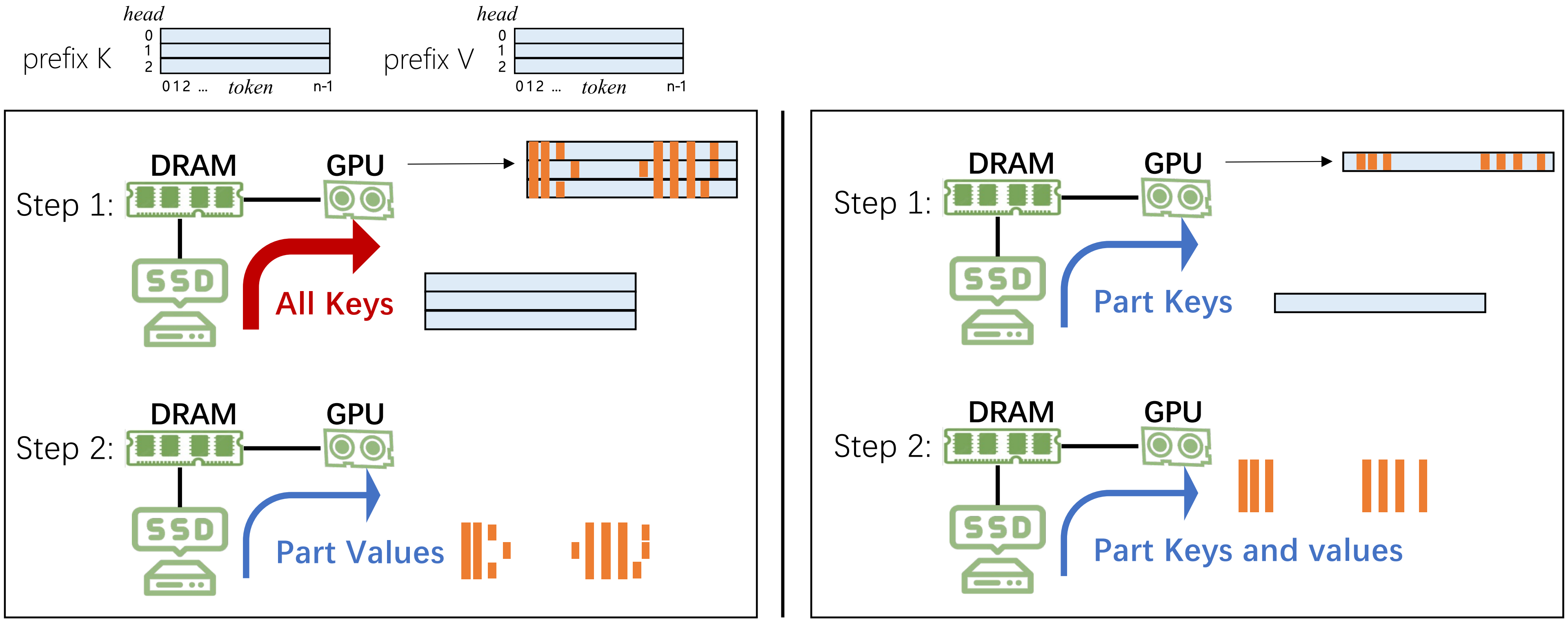


(b) select the top 40% most important tokens

The similarity of important tokens indices exists across different LLM scales and important KV ratios.

1 Similarity-Guided Important Token Identification

Key idea: Use the important token index set from a few selected heads to **approximate** the important token index sets for the remaining heads

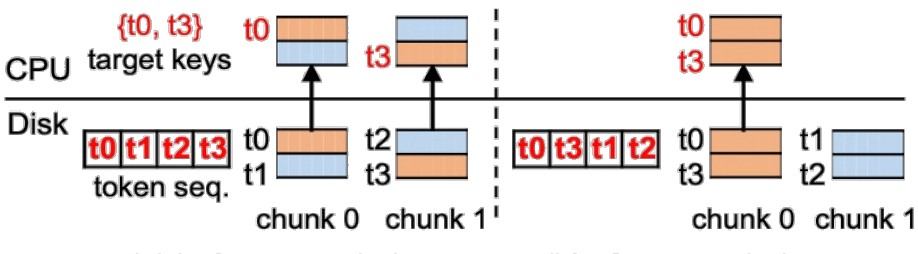


(a) Without our method

(b) With our method

2.1 KV Reordering

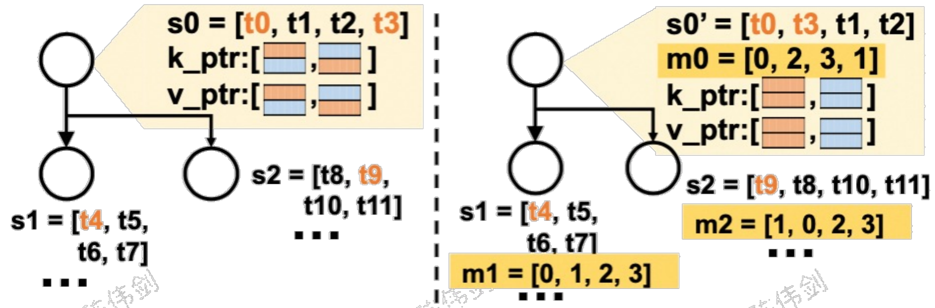
- Target: Reduce read amplification
- Key idea: reorder and repack important KVs into denser chunks



(a) before recordering (b) after recordering

- Problem: KV reordering may destroy the radix tree structure by altering the token order

1. avoid cross-node reordering
2. Add mapping list to recovery



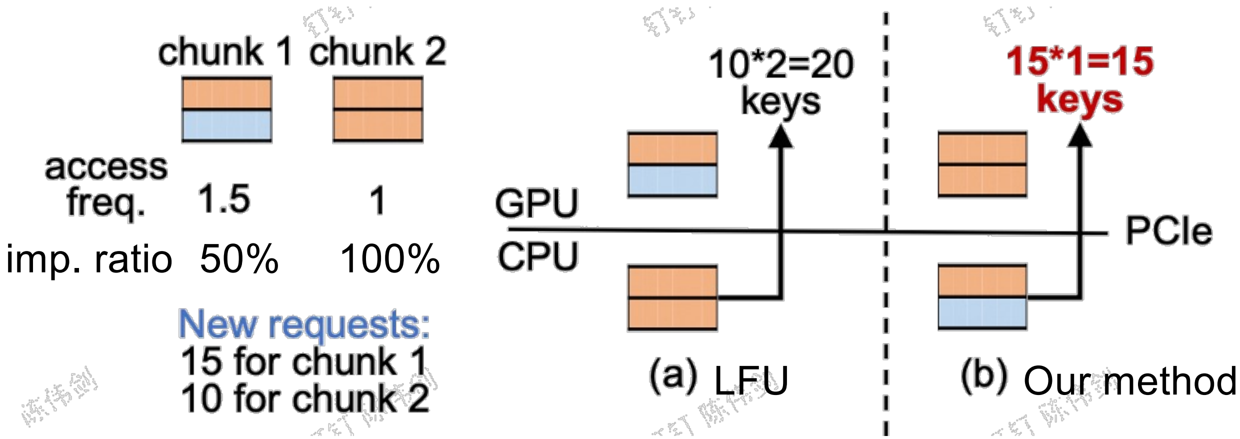
(a) before recordering (b) after recordering

2.2 Score-Based Cache Management

➢ Key idea： Data admission and cache replacement based on scoring.

➢ The score = the chunk access frequency * proportion of important KVs.

The higher the score, the higher the priority for admission into the faster medium cache.



score for chunk 1: 1.5 * 50% = 0.75
score for chunk 2: 1 * 100% = 1

Experimental Setup

- System configuration

CPU	2 × AMD EPYC 7763
GPU	1× NVIDIA A100 (80GB)
Memory & SSD	128 GB DRAM, 2TB SSD (5GB/s)

- Workloads and datasets

Datasets	PIQA, RTE, COPA, and OpenBookQA Prefix sizes: 55GB, 57GB, 64GB, 65 GB
Models	OPT-6.7B, OPT-13B, OPT-30B

- Compared systems

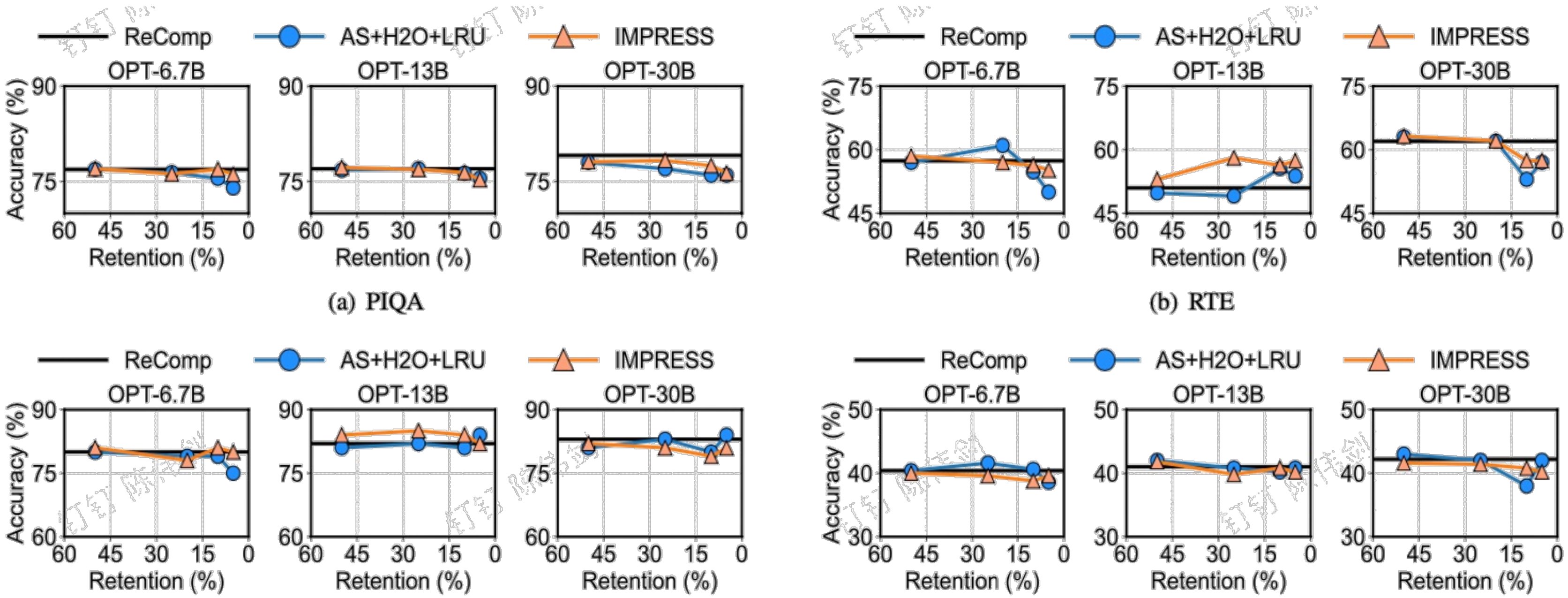
ReComp	Recomputation without reusing prefix KVs
AS-like	AttentionStore with async KV loading, without scheduler
AS+H2O+LRU	Add H2O on top of AttentionStore with LRU
AS+H2O+LFU	Add H2O on top of AttentionStore with LFU
IMPRESS	Our three optimizations on top of H2O

- Default settings.

  (1) cache size: 10GB GPU HBM, 32GB CPU DRAM

  (2) Chunk size: keys or values of 64 tokens

Model Inference Accuracy



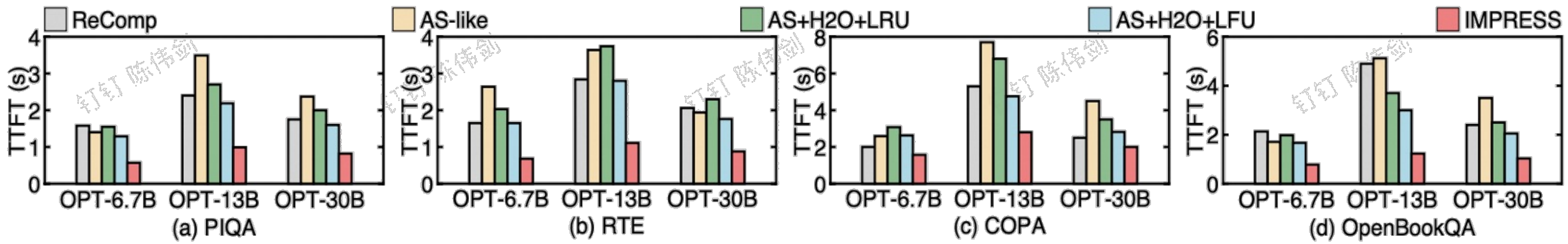
(a) PIQA (b) RTE

(c) COPA (d) OpenBookQA

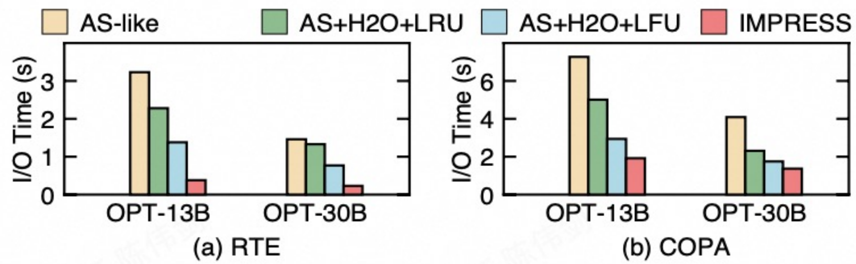
Compared to ReComp, the average inference accuracy drop is less than 0.2%

Time-to-first-token (TTFT)

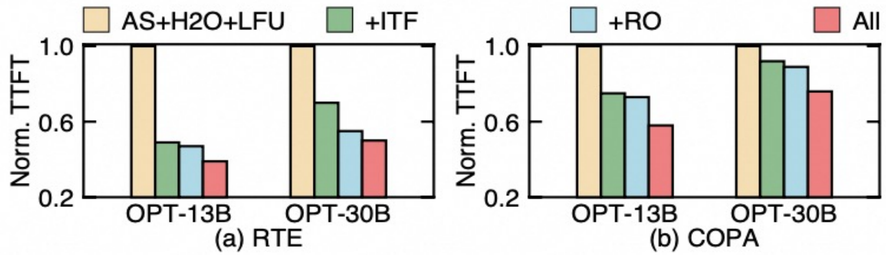
✓ TTFT



✓ I/O time

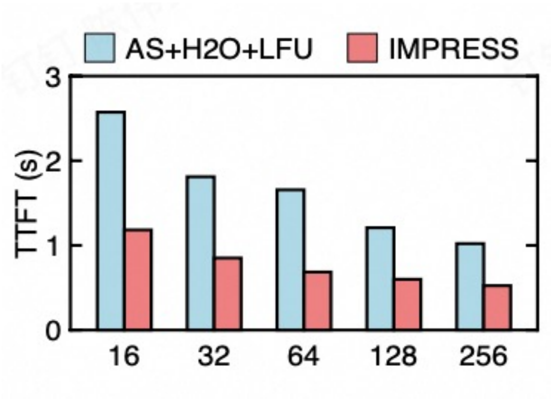


✓ Individual technique

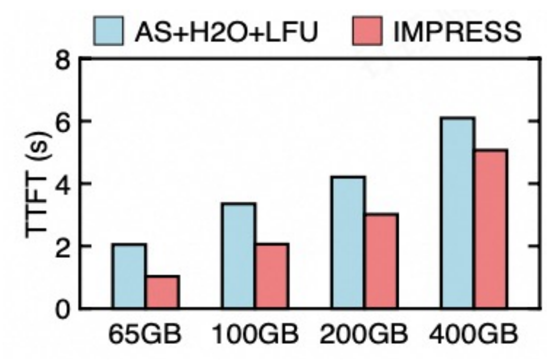


IMPRESS outperforms alternatives, with a 1.2×~2.8× improvement over SOTA solutions, due to a 1.5×~3.8× reduction in I/O time.

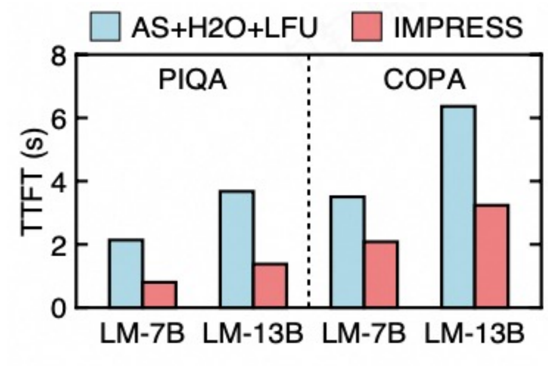
Sensitivity Analysis



Various chunk sizes



Various dataset scales



Results on Llama models

IMPRESS outperforms the leading alternative on various cases.

More evaluations: checkout our paper

Summary & Conclusion

- **Problem**
  - I/O becomes the bottleneck when shared prefix KVs are loaded from SSD for LLM
- **Key idea**
  - Only load important KVs during prefill phase
- **Challenges**
  - A large amount of I/O is introduced to identify important KVs
  - Storage and caching systems are suboptimal
- **Techniques in iCache**
  - Similarity-Guided Important Token Identification
  - KV Reordering & Score-Based Cache Management
- **Results**
  - IMPRESS outperforms the alternatives with the same level of inference accuracy

Thanks & QA

IMPRESS: An Importance-Informed Multi-Tier Prefix KV Storage System for Large Language Model Inference

浙江大学
Zhejiang University

HUAWEI
HUAWEI CLOUD

Contact Email: weijianchen@zju.edu.cn

ISCS Lab: https://shuibing9420.github.io