IMPERIAL
Department of Electrical
and Electronic Engineering

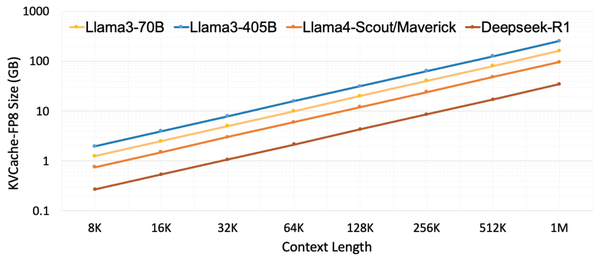
Speculative Sparse KV Cache Prefetching - Accelerating Agentic AI on Large Scale GPU systems

Can Xiao,
Aaron Zhao

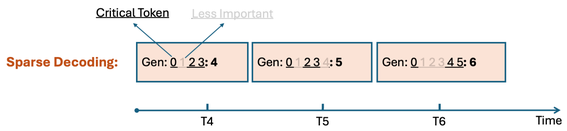
Introduction

Long context large language models
LLM models are becoming context heavy.
- 100x more tokens in agent workloads

KV cache – runtime storage challenge
KV Cache is the runtime storage for the previous context information. Large KV cache limits the batch size.
- With 80 GB of GPU memory and a 10 GB KV cache per request, the maximum batch size is 8.

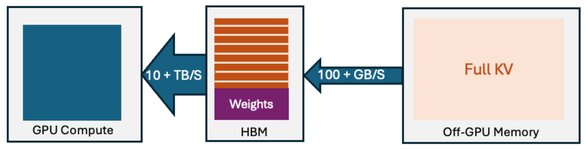


How to reduce the memory requirement on KV cache?



Sparse KV Cache

Not all tokens are important in the KV cache. When doing computation, we can evict some unimportant tokens.
But token importance varies across decoding steps, the tokens that can be skipped vary across decoding steps.



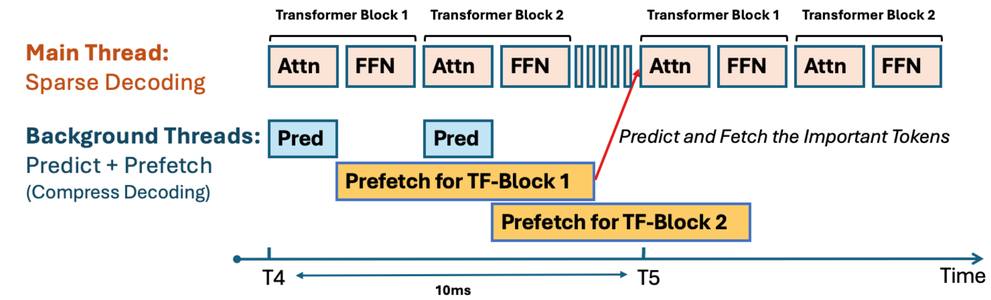
KV Retrieval System

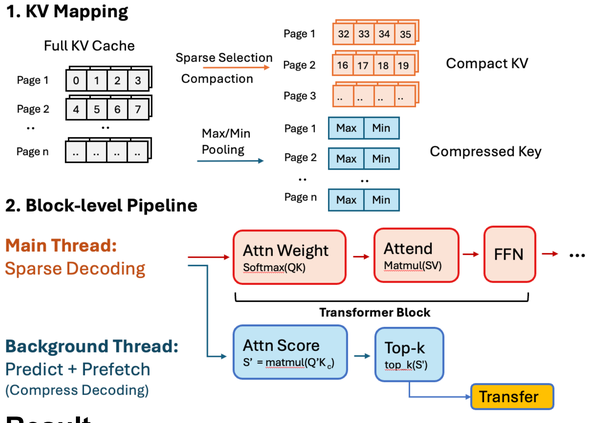
KV cache retrieval system keeps the advantage from sparse KV system and avoid accuracy drop from token importance variant.
- The important tokens are stored on GPU Memory.
- The new important KV is fetched from off-GPU memory.

Key bottleneck comes from KV cache retrieval system is the bandwidth gap. Ideally, the fetching process needs to overlap with decoding.

Speculative Sparse KV Cache Prefetching System

High level view
- Main Thread for token generation, background threads are for token retrieval
- The prefetching time is at least within time between tokens (TBT), e.g. 10ms
- Multi-thread design can guarantee that sparsity selection and memory prefetching can be overlapped with token generation.





Detailed pipeline
- Input of the main/background thread will be the normal/speculative token (Q/Q').
- Main thread compute the attention with compact KV cache. Each page inside compact KV will only store the important token. The important token will be updated during the prefetching stage.
- Background thread compute the attention score with pooled Key values. It will get the top-k based on the attn score only and trigger a transfer event.

Result
With limited token budge, we can achieve near lossless performance loss.

Long Reasoning Benchmark: AIME24

Qwen3-8B @ 20K output token budget

TopK Tokens	Sparsity Ratio	Accuracy
Full KV cache	0	0.7
4K	0.6 ~ 0.8	0.7
2K	0.8 ~ 0.9	0.7

Long Context Benchmark: longbench-gov_report

LLAMA-3.1-8B Instruct @ 10K input token

TopK Tokens	Sparsity Ratio	score
Full KV cache	0	0.3453
4K	0.6 ~ 0.8	0.3355
2K	0.8 ~ 0.9	0.3195