HW5

Tguo67

Question(s):

Read the paper **CacheBlend: Fast Large Language Model Serving for RAG with Cached Knowledge Fusion**, summarize the paper, specifically, including the points below:

- What are the motivations/challenges of this work?
- How does the design of this paper address the challenges?
- How does the paper evaluate its design (experiment settings, workloads, metrics)?
- How does the evaluation prove its claims?

Note that it is essential to logically connect the motivation, design, and evaluation, rather than merely listing some points.

Related link:

Paper of CacheBlend: https://arxiv.org/pdf/2405.16444

Answers:

Motivation and challenges:

In RAG, prompts stitch together many retrieved chunks. Prefill dominates latency; simple prefix caching helps only when the reused text is the leading prefix, while naïve non-prefix reuse breaks cross-attention with the preceding text and can hurt answer quality. So the target goal is how to reuse cached chunks anywhere while not losing the accuracy of the full prefill.

Solutions to the challenges:

CacheBlend reuses K/V for every reused chunk but selectively recomputes a small subset of tokens per layer to reduce the attention deviation relative to full refill. It then pipelines this small recompute with loading caches from slower storage (CPU/SSD) so the I/O is hidden. A controller picks the minimal recompute ratio and feasible storage tier to keep TTFT low. These are implemented on vLLM.

Evaluation settings, workloads, metrics:

3 open LLMs (e.g., Mistral-7B, Yi-34B, Llama-70B), 4 benchmarks: multi-hop QA (2Wiki, MuSiQue) and summarization (SAMSum, MultiNews). Metrics: TTFT, throughput, and task quality (F1/ROUGE-L).

Results:

Across models/datasets, CacheBlend cuts TTFT by 2.2–3.3 times and boosts throughput 2.8–5 times without quality loss vs full recompute. It is direct evidence that selective recompute restores the needed cross-attention while reusing most K/V. The pipelining study shows similar TTFT even when caches sit on SSD, confirming the I/O-hiding claim.