DRAM 기반 스토리지를 활용한 RAG 기반 LLM 추론 가속화 연구

KiHyun Kim[1], Jongman Kim[2], Youngjae Kim[1]
[1]*Sogang University,* [2]*Soteria Inc.*

Large Language Model(LLM)은 질의응답, 번역, 요약 등 다양한 자연어 처리 작업에 널리 활용되고 있다. LLM은 auto-regressive 방식으로 작동하며, 이전에 생성된 출력 토큰을 기반으로 순차적으로 다음 토큰을 예측하고 생성한다. 이 과정에서 Key-Value(KV) 캐시는 중요한 역할을 한다. KV 캐시는 이전 토큰에 대한 중간 계산 결과를 GPU 메모리에 저장하여 모든 토큰을 매번 다시 계산하지 않고 빠르게 참조할 수 있도록 지원한다. 이를 통해 모델은 연산량을 줄이고 토큰 생성 속도를 향상시킬 수 있다. 한편, Retrieval-Augmented Generation(RAG) [1] 기술은 LLM의 환각(hallucination)을 개선하는 데 중요한 역할을 한다. RAG는 외부 대형 벡터 데이터베이스에서 사용자 입력과 관련된 문서를 검색하여 LLM 입력에 추가적인 맥락을 제공한다. 이를 통해 모델은 더 정확하고 일관된 답변을 생성할 수 있다. 그러나 이 과정에서 LLM의 입력 및 출력 길이가 길어지고, 이에 따라 KV 캐시의 크기도 증가하게 된다. 이는 제한된 GPU 메모리 자원에 과부하를 초래할 수 있다.

이 문제를 해결하기 위해, 기존에는 KV 캐시를 GPU 메모리 대신 CPU 메모리로 오프로딩하고 필요 시 GPU 메모리로 다시 불러오는 접근 방식이 제안되었다 [2]. 이 방식은 GPU 메모리의 부담을 줄이는 데 도움을 주지만, CPU 메모리 역시 용량이 제한적이기 때문에 대규모 입력 데이터를 처리할 때 한계가 있다. 기존 방식에서는 CPU 메모리가 부족한 경우 KV 캐시를 파일 시스템 인터페이스를 사용하여 디스크에 저장하는 방식을 채택한다. 하지만 이러한 방식은 OS 파일 시스템의 소프트웨어 스택 호출로 인한 오버헤드가 발생한다.

본 연구에서는 CPU 메모리의 확장 솔루션으로 PCIe 기반 메모리 스토리지의 효용성을 탐구한다. 메모리 스토리지는 CPU 메모리의 추가적인 메모리 계층으로 활용되어 호스트 시스템과 직접 통신한다. 이를 통해 파일 시스템 호출 스택을 우회하여 데이터 전송 오버헤드를 최소화할 수 있다. 또한, 호스트와 메모리 스토리지 간 효율적인 KV 캐시 전송을 위한 맞춤형 프로토콜을 설계하여 고속 KV 캐시 오프로딩을 가능하게 한다. 이를 바탕으로 기존 CPU 메모리 오프로딩 방식과 메모리 스토리지를 도입한 방식 간의 성능 차이를 분석하고자 한다. 특히, 위 두 방식을 사용한 LLM 추론에서의 지연시간(latency)을 평가함으로써 메모리 스토리지가 기존 CPU 메모리 및 디스크 기반 오프로딩 방식을 대체할 수 있는 효율적인 메모리 확장 솔루션으로서의 적합성을 탐구한다.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. K¨uttler, et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," in Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474, 2020. [2] Y. Sheng, L. Zheng, B. Yuan, Z. Li, M. Ryabinin, D. Y. Fu, Z. Xie, B. Chen, C. Barrett, J. E. Gonzalez, P. Liang, C. Ré, I. Stoica, and C. Zhang, "FlexGen: High-Throughput Generative Inference of Large Language Models with a Single GPU," *arXiv preprint*, arXiv:2303.06865, 2023.