DISTRIBUTED INFERENCE PERFORMANCE OPTIMIZATION FOR LLMS ON CPUS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) hold tremendous potential for addressing numerous real-world challenges, yet they typically demand significant computational resources and memory. Deploying LLMs onto a resource-limited hardware device with restricted memory capacity presents considerable challenges. Distributed computing emerges as a prevalent strategy to mitigate single-node memory constraints and expedite LLM inference performance. To reduce the hardware limitation burden, we proposed an efficient distributed inference optimization solution for LLMs on CPUs. We conduct experiments with the proposed solution on 5th Gen Intel® Xeon® Scalable Processors, and the result shows the time per output token for the LLM with 72B parameter is 140 ms/token, much faster than the average human reading speed about 200ms per token.

1 INTRODUCTION

With the unprecedented success of Large Language Models (LLMs) across diverse domains Cui et al. (2023), Thirunavukarasu et al. (2023), the performance of LLM inference is paramount for extensive LLM applications Zhao et al. (2023). In the deployment of LLMs, we encounter numerous challenges, including substantial memory consumption, stringent latency targets, and long sequence lengths. Moreover, these challenges hindered the practical applications in low-resource environments.

As we known, LLMs primarily utilize the Transformers architecture Vaswani et al. (2017), which exhibits high parallelism. However, efficiently deploying these models in practical applications presents challenges. This is because inference generation occurs token by token, with each token's computation relying on previously generated tokens. Multiple deployment optimization solution for LLM has been proposed, such as Miao et al. (2023), Agrawal et al. (2023) and Zheng et al. (2023). While these solutions are primarily designed for GPUs, when GPU hardware resources are limited, we can explore alternative options on CPUs. Therefore, an efficient distributed solution for LLM on CPUs is proposed. A better distributed solution for LLM inference performance optimization on CPU is crucial for cost-savings, efficient hardware usage, and optimal inference strategies. It could help achieve the required scalability and efficient low-latency inference.

In this paper, we propose three approaches which helps optimize the distributed inference performance for LLMs on CPUs. We conduct experiments with the proposed solution on 5th Gen Intel® Xeon® Scalable Processors, and the results indicate that the LLM with 72B parameters achieves a time per output token of 140 ms, significantly surpassing the average human reading speed of approximately 200 ms per token.

2 APPROACH

To optimize the distributed inference performance, minimizing communication cost wherever possible is important Bajović et al. (2016). We utilize the oneAPI Collective CommunicationsLibrary (oneCCL) designed with the aim of creating a unified standard API compatible with various types of hardware accelerators. Furthermore, we proposed multiple optimization approaches to enhance the LLM inference performance on CPUs as follows.

2.1 IMPROVE SCALABILITY BY MINIMIZING SYNCHRONIZATION

During the initial phase of each inference round, the proposed solution broadcasts token IDs rather than broadcasting the values of Embedding part obtained based on token IDs. Similarly, we adopt an effective approach which is for each worker to compute top-k before performing the reduction at the end of each inference round. The implementation is shown in Figure 1 .

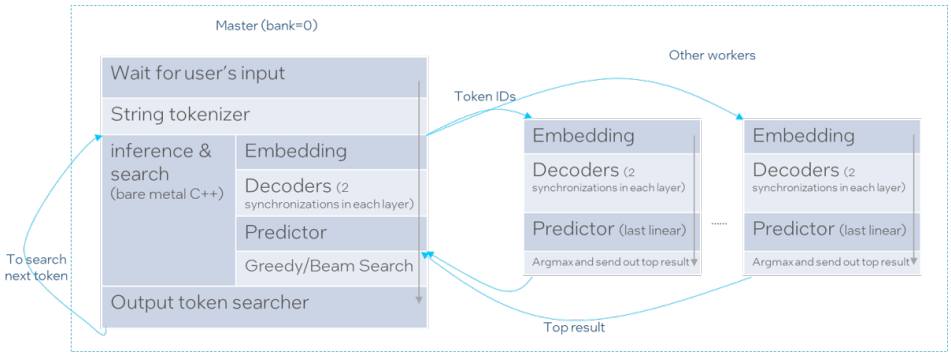

Figure 1: Distributed inference based on oneCCL.

2.2 ONE-TIME SYNCHRONIZATION

Optimizing communication cost based on each model's structure is essential. For models like GPT-J and Falcon, where attention and feed-forward network sections run in parallel, it's possible to achieve communication efficiency by ensuring that each decoder layer performs only one time synchronization which shows in Figure 2 .

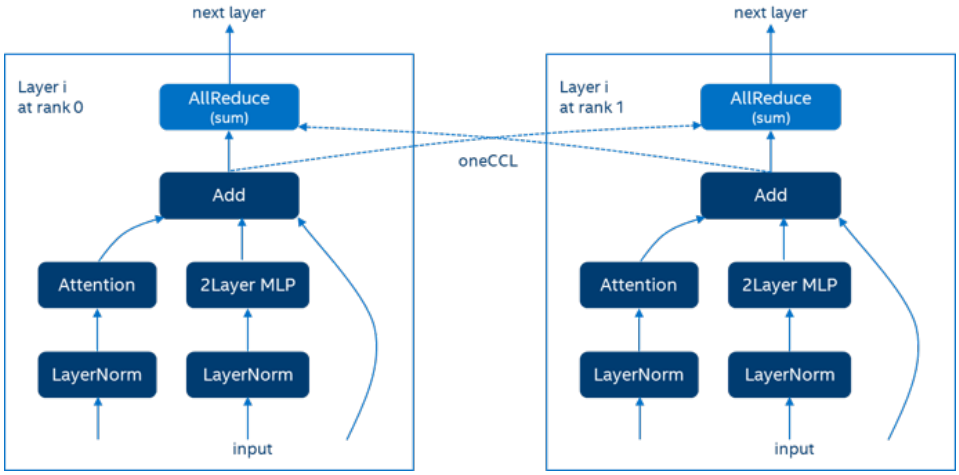

Figure 2: One time synchronization.

2.3 MINIMIZE MEMORY COPY

As we are aware, when the computation module and communication module interact, data copying is often involved in practice. Therefore, a more aggressive optimization approach can be pursued to eliminate these copies. This involves the computation module, during its last operation before communication, directly writing the results to the location of the communication module, achieving a zero-copy implementation.

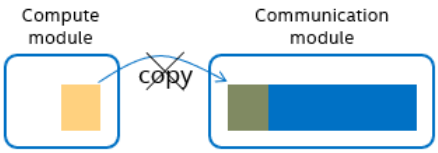

Figure 3: Minimize memory copy.

3 EXPERIMENT RESULTS

We conducted experiments using Qwen which is a large-scale pre-trained model developed by Alibaba Group Bai et al. (2023) with model parameter sizes of 72B. Qwen-72B is a Transformer-based large language model. To illustrate performance results, we measured the per token time of the next token generation on the 4 * Intel® Xeon® Scalable Processors 8575C, where each device has 1 socket, and each socket has 48 cores. With input size = 512 tokens and batch size = 1, the results shows time per output token is 140 ms/token, much faster than human reading speed.

4 CONCLUSION

We propose an efficient distributed inference performance optimization solution for LLMs on CPUs by leveraging oneCCL. The experiment results shows the promising per-token generation latency is 140 ms. In our future endeavors, we aim to enhance distributed LLM inference as a contribution to the open-source community. Furthermore, we intend to expand our approach to encompass a wider variety of CPUs, thereby empowering generative AI on CPUs in resource-limited environments.

REFERENCES

Amey Agrawal, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S Gulavani, and Ramachandran Ramjee. Sarathi: Efficient llm inference by piggybacking decodes with chunked prefills. *arXiv preprint arXiv:2308.16369*, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Dragana Bajović, José MF Moura, Joao Xavier, and Bruno Sinopoli. Distributed inference over directed networks: Performance limits and optimal design. *IEEE Transactions on Signal Processing*, 64(13):3308–3323, 2016.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. Receive, reason, and react: Drive as you say with large language models in autonomous vehicles. *arXiv preprint arXiv:2310.08034*, 2023.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Hongyi Jin, Tianqi Chen, and Zhihao Jia. Towards efficient generative large language model serving: A survey from algorithms to systems. *arXiv preprint arXiv:2312.15234*, 2023.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Zangwei Zheng, Xiaozhe Ren, Fuzhao Xue, Yang Luo, Xin Jiang, and Yang You. Response length perception and sequence scheduling: An llm-empowered llm inference pipeline. *arXiv preprint arXiv:2305.13144*, 2023.