CS598AIE Paper Review: ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning

Jiaqi Lou (Netid: jiaqil6)

September 20, 2024

1 The problem the paper is trying to tackle.

This paper addresses the challenges posed by the GPU memory wall, which limits the size of deep learning models and restricts user access due to the limited availability of large GPU clusters. Thus, this paper tries to tackle this challenge by efficiently leveraging the node heterogeneity to employ CPU memory and NVMe storage to overcome the GPU memory limitation.

2 What's the impact of the work, e.g., why is it an important problem to solve?

The paper is important from two perspectives. Firstly, it enables larger models to multi-trillions of parameters by expanding GPU memory with heterogeneous memory and storage on the node and smartly scheduling/overlapping/hiding the communication overhead. Secondly, it considers the user experience by simplifying the training process with many automation designs and allowing more researchers to be able to train large models without large GPU clusters.

3 The main proposed idea(s). + A summary of your understanding of different components of the proposed technique, e.g., the purpose of critical design choices.

- Memory-centric tiling tries to capture the data fetch and release pattern such that it can break large operators into smaller tiles to be executed sequentially and reduce working memory requirements.
- Bandwidth-centric partitioning. ZeRO-Infinity uses an all-gather instead of a broadcast when a parameter needs to be accessed so that all PCIe links are utilized.
- Overlap centric design. This work not only overlaps the GPU-GPU communication with GPU computation but also overlaps the NVMe to CPU to GPU communication. It keeps track of the operator sequence of each iteration and prefetches any parameters that are required in the future 3 steps' operator during the forward pass. The backward pass follows the same overlapping strategy to transfer data back to NVMe storage. In this way, the data movement overhead can be hidden by overlapping with computation.
- Offload model states to NVMe storage and activation to CPU memory. Infinity offload engine consists of DeepNVMe and pinned memory management layer. The engine optimizes a low-level NVMe library to achieve high parallelism and near-peak I/O performance. It also manages the pinned memory smartly by reusing buffers.
- Ease-inspired implementation makes the framework easy to use without any model refactoring.

4 Your perceived strengths and weaknesses of the work, e.g., novelty, significance of improvements, quality of the evaluation, easy-to-use.

4.1 Strengths

- The memory requirement analysis is very helpful for understanding the memory bottleneck during the training process.
- It is easy to use because ZeRO-infinity does not need model refactoring and allows easy accessibility for researchers to train large models.

4.2 Weaknesses

- I think the design may depend on hardware to achieve the best performance For example, the number and position of NVMe storage on the node, CPU memory capacity, etc. In particular, the analysis part may not be generalized to other hardware setups.

5 Is there room for improvement? If so, which directions you may want to explore or idea you have for improving the techniques?

New technologies such as GPUDirect RDMA, NVMe over Fabrics, CXL, and GPUDirect Storage allow direct data movement between storage and GPU memory. This paper's overlapping strategy assumes a two-hop NVMe storage to CPU to GPU data movement path. What factors should be considered if we want to leverage these new techniques to avoid buffering at CPU memory? Also, with these new techniques, storage and memory modules do not need to sit inside the same node. I think tiered memory and storage system may be useful in this case to improve bandwidth but sacrifice extra latency.