Research Paper Reading (CENT)

Research Problem

The paper begins with mentioning the high cost of LLM inference using GPUs which they attribute to LLMs requirement for large memory capacities (due to unique key-value caches per prompt) and low operational intensity (spend more time waiting for data than doing work on it). The main issue of current deployment/inference methods on GPUs simply means that it ends up being expensive for the consumer.

The solution that the paper proposes is not only GPU-free but in fact faster, more energy and cost efficient. In simpler terms, using their solution in place of current GPU-based deployment means reduced costs and improved efficiency, making LLM inference cheaper for consumers, more profitable for providers, better for the environment and a better user experience overall.

Prior Research

Prior research that the paper mentions includes Processing In-Memory (PIM) architectures, Processing-Near-Memory (PNM) architectures, Pipeline and Tensor Parallel mappings and Compute eXpress Link (CXL) memory expansion. The paper builds upon these prior works by combining these components, that have been studied individually in the past, to allow for the distribution and parallel execution of LLM inference tasks more efficiently with the need for GPUs. The novelty of the paper lies in the integration of these components and even a custom ISA to achieve this. The paper does not seem to explicitly make mention of the problems that these prior works sought to solve, but they generally could be inferred. For example, it is faster to move data within memory (PIM/PNM) than to move it between memory and a processor (GPU/CPU), so the problem there would have been memory bandwidth bottlenecks.

High-level Ideas

- Distribution of Workload among CXL devices:
  CENT makes use of multiple CXL devices to expand memory capacity and allow for data communication across them. This effectively enabled the breakdown and distribution of LLM inference tasks across multiple devices, allowing for parallel execution and reduced latency.
- Hierarchical PIM-PNM Architecture within CXL devices: After workload distribution, the hierarchical architecture allows for further specialized processing/execution of tasks by components of the CXL device. For example, the PIM components are primarily used for MAC (multiply and accumulate) operations responsible for matrix-vector multplications that make up 99% of arithmetic operations for a transformer block, while the PNM components are designed to handle less frequent operations such as the square root or inversion. These hardware components effectively replicate the functions of GPUs but in a more efficient manner by being closer to memory and reducing data movement.
- Custom ISA to achieve the above:
  A custom ISA was developed to achieve data movement and arithmetic operations across the CXL devices and their PIM/PNM components.
- Hybrid Tensor-Pipeline Parallel Mapping: The paper builds upon prior works on pipeline and tensor mapping to develop a hybrid mapping strategy where each decoder transformer block is distributed across multiple consecutive CXL devices to form pipeline stages and within each stage, tensor parallelism is used to further split workloads (e.g., fully connected layers) across the multiple CXL

devices in that stage. While implementation details are sparse, it is obvious that implementing and coordinating this hybrid mapping strategy would not have been easy. Such a strategy exploits parallelism at the level of multiple prompts processing (pipeline) as well as the distribution of operations/specific layers across devices (tensor) for improved efficiency.

Evaluation

Methodology

Primarily, the paper compares the performance between their CENT architecture and a GPU-based baseline. The CENT architecture is modeled using the Ramulator2 simulator, amongst others, for CXL devices, using 32 CXL devices to match the average power consumption of the GPU baseline. The experiments/comparisions were ran using the Llama line of models of varying sizes as well as vaying GPU and CXL device counts to see how performance scaled. Besides CENT vs GPU, the paper also evaluates CENT against other prior works in heterogeneous architectures such as GPU-PIM and CXL-PNM to see how CENT compares against them.

Results

The results of the experiments show that CENT outperforms the GPU baseline in terms of latency, energy consumption and cost across all Llama model sizes tested. Comparing CENT against prior works, CENT outperforms GPU-PIM archictectures in terms of cost while outperforming CXL-PNM architectures in terms of throughput. Besides performance and costs, the other selling point of CENT is its configurability and scalability in terms of CXL device counts and supported operations.

Artifacts

The evaluation platform that can be reproduce their results using the AiM simulator publicly available on their GitHub repository. Through the evaluation platform, one can simulate and reproduce results comparing the CENT architecture and GPU baselines as Figures 12 to 15 in the paper, which are namely the latency, throughput, energy consumption and cost comparisons.

Strengths and Weaknesses

- Costs were modeled and estimated rather than measured, which could lead to slight deviations from real-world costs.
- However, with that being said, virtually every single cost estimation/assumption made in the paper was explicitly mentioned, which allows the reader to verify and validate the assumptions made.
- Results analysis was comprehensive, highlighting not only the end-to-end latency, but also latency and throughput breakdowns at different stages, context lengths and more. It was particularly interesting to see that a low pipeline but high tensor parallelism setup was optimal for CENT, indicating that optimizing for per prompt performance was more effective than optimizing for multiple prompts in parallel.
- Performance improvements in CENT were justified and explained, while areas in which CENT underperformed were also acknowledged and explained.
- Comparisons against prior works lent credibility to the claims made by the paper as it compared against multiple baselines rather than just a single one, showing that the authors had done their due diligence in evaluating their work and did not cherry-pick baselines that they could easily beat. It was

particularly enlightening to see that while CENT did not see a landslide win in terms of throughput, its main selling-point after comparisions was cost-efficiency, which it excelled at.

- The section on Related Works came at the end of the paper rather than the beginning, which made it slightly difficult to understand how the proposed work built upon prior works, the problem that CENT was trying to solve that prior works could not solve and the novelty of the proposed work.
- Artifact provided does not allow the reader to fully reproduce all results in the paper, particularly those comparing against prior works. However, the artifact does allow for reproduction of the main CENT vs GPU results, which is still commendable.

Knowledge Learned

Frankly, prior to reading this paper, I had very little knowledge about CXL devices, PIM/PNM architectures and how LLM inference tasks were executed on hardware, i.e. much of the vocabulary and concepts were new to me. My primary takeaway from this paper was the conceptual understanding of pipeline and tensor parallelism, where tasks can be scheduled and pipelined across multiple devices (familiar) as well as the distribution of operations/layers across multiple devices (new to me). The inner workings of CXL devices, PIM and PNM architectures were also new to me and currently still opaque, but I now have a basic understanding of their purpose and functions, particularly in their ability to reduce data movement and memory bottlenecks. The concept of a CXL network was also new to me, so I am curious to learn more about this and how it differs from conventional networks.