Scaling FlashAttention: Multi-GPU Implementation and Comparative Analysis on Llama Models

Venu Gopal Kadamba (vk2636)
Abhisek Upadhyaya (au2216)

Why Distributed FlashAttention?

Summary

Problems:

1. Quadratic memory and computational cost of the self-attention mechanism in Transformers
2. Infeasible to train models efficiently on long contexts
3. Slow and expensive training

Flash Attention Benefits:

1. Linear memory usage relative to sequence length
2. Longer context lengths
3. Faster training times

Goals:

1. Implement Distributed FlashAttention for LLaMA2 7B across 4 GPUs
2. Profile performance metrics (arithmetic intensity, latency, bandwidth)
3. Port implementation to LLaMA3
4. Compare performance between LLaMA2 and LLaMA3 implementations

Distributed FlashAttention Benefits:

1. Distributed training
2. Scaling by overcoming single GPU memory limitations
3. Reduce communication overhead
4. Eliminates redundant recomputation during backpropagation

NYU

Technical Challenges

Challenges

Allocation of GPUs in HPC

LLaMA 2 7B:

1. Memory Issues due to limited cumulative memory of 4 available GPU
2. Determining the metric computation
3. Dependency version issue between the “Triton”, “Torch” and “Flash-Attention”

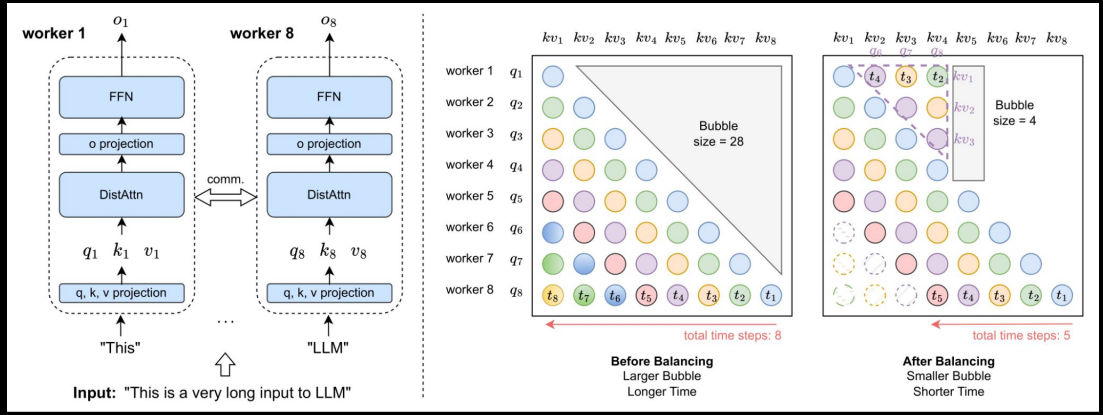
LLaMA 3:

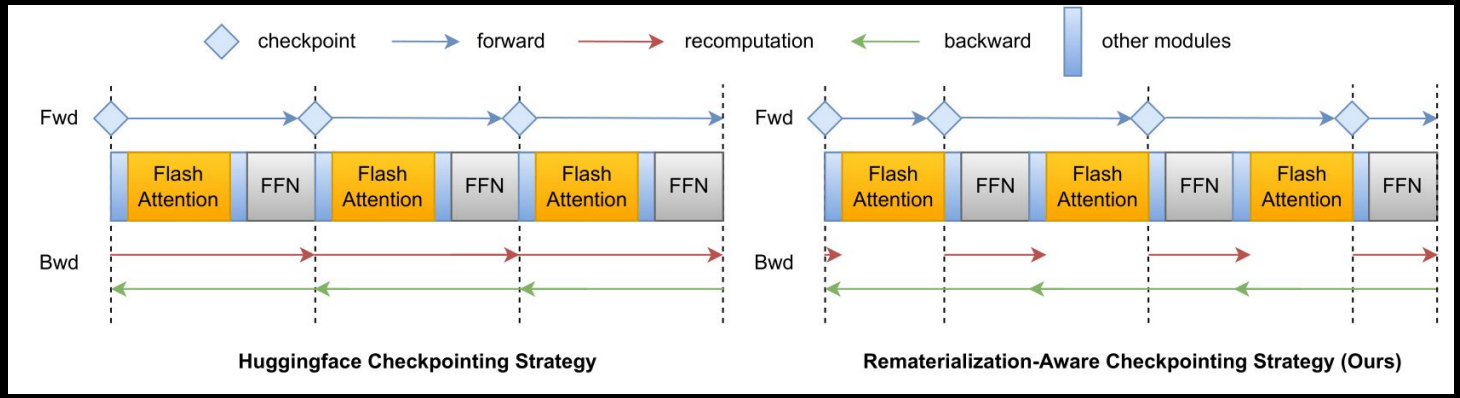
1. Transferring the existing code base to new architecture
2. Resolving Dependency issues for the new system
3. “rope_scaling” by upgrading “hugging_face” and “cache” issue by downgrading it
4. Memory Issues due to limited cumulative memory of 4 available GPU

NYU

Technical Approach

LLaMA2 and LLaMa3





NYU

Metrics for Distributed Flash Attention

To evaluate performance metrics such as Latency and Bandwidth for LLama models using torch.profiler data.
Metrics and Formulas

1. Latency (Total Execution Time)
Represents the total time taken for CUDA and CPU operations.

Latency (seconds) = Total CUDA Time + Total CPU Time

2. Bandwidth (Memory Transfer Rate)
Measures the rate of memory transfer between GPU and memory, computed as the memory transferred divided by the CUDA execution time.

Bandwidth (GiB/s) = (Total Memory Transferred (bytes) / Total CUDA Time (seconds)) * 10^{-9}

Implementation Workflow
- Profiler Data Extraction: Extracted metrics include:
  - CUDA Memory Usage: Total memory transferred during computation.
  - CUDA Time: Execution time on GPU.
  - CPU Time: Execution time on CPU.

- Metric Aggregation:
  - Summed across all profiler events for accurate measurement.

Results

Summary

Latency Trends:

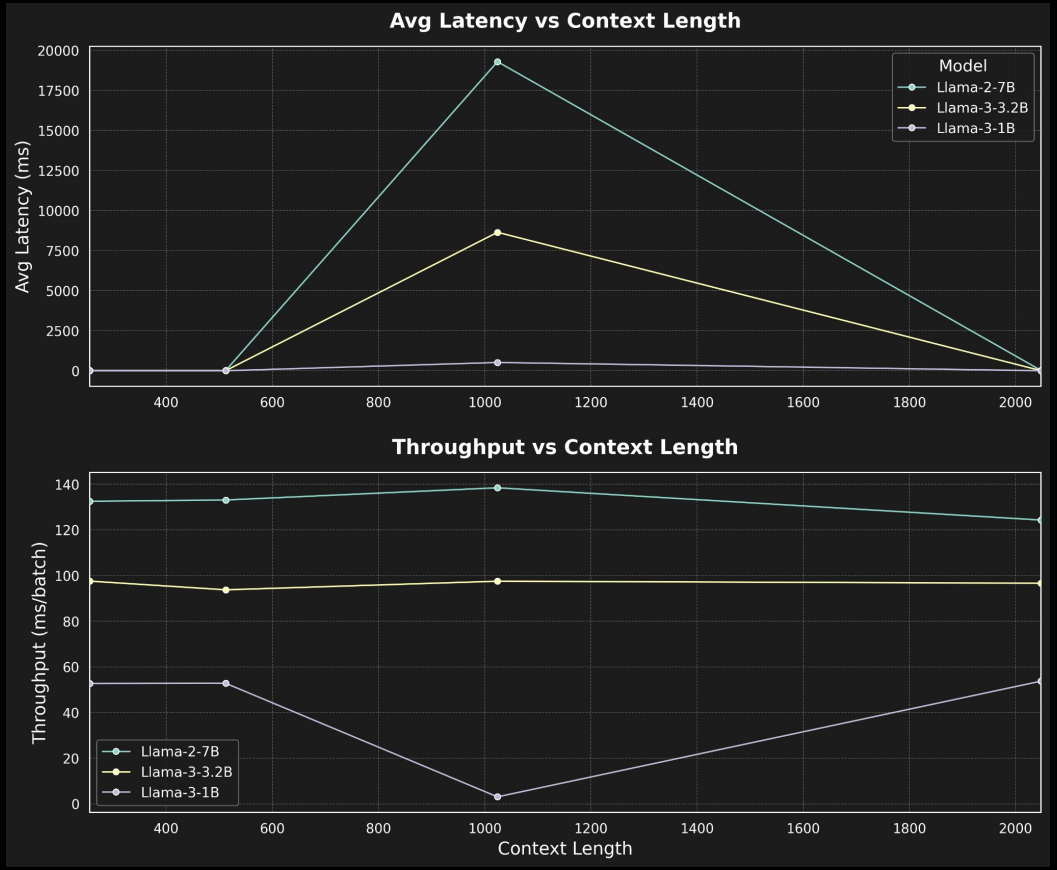
- LLaMA-3-1B consistently exhibits the lowest latency across all context lengths, indicating its computational efficiency.

- **Bandwidth:** Bandwidth increases with context length; LLaMA-3-1B achieves the highest at 3.07 GiB/s.
- **Memory Usage:** LLaMA-2-7B has the highest memory requirements; memory demand grows with context length.
- **Training Efficiency:** LLaMA-3-1B has faster forward and backward pass times; communication overhead is consistent.

Experimental Evaluation

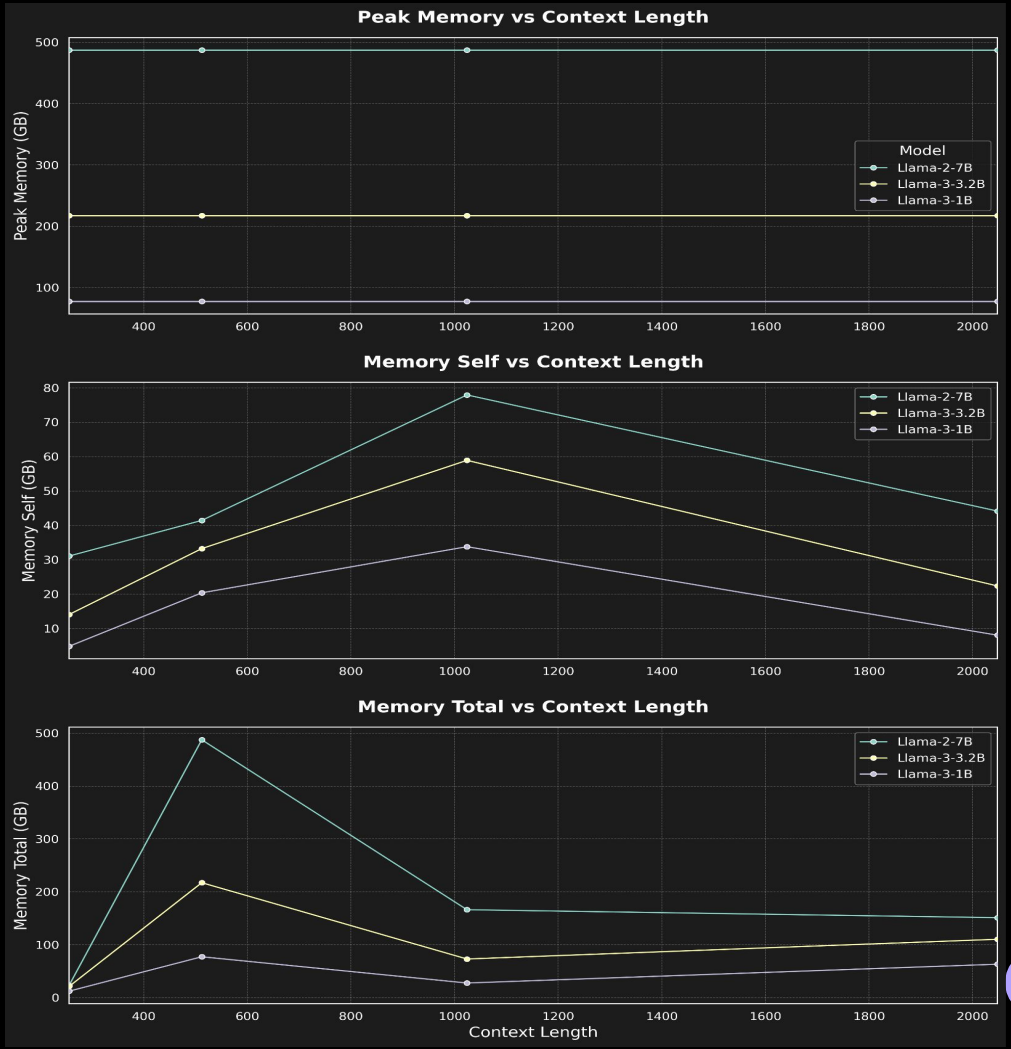
Training Performance Metrics

Training Performance Metrics (Per GPU Performance)



NYU

Memory Usage Analysis



Training Efficiency

Model	Metric	Forward Pass	Backward Pass	Communication Time
Llama-2-7B	256	18.77s	132.55s	18.78s
Llama-3-3.2B	256	8.52s	97.57s	8.51s
Llama-3-1B	256	3.08s	52.73s	3.09s
Llama-2-7B	512	132.62ms	538.26ms	68.42ms
Llama-3-3.2B	512	93.55ms	261.17ms	35.01ms
Llama-3-1B	512	53.34ms	157.83ms	21.22ms
Llama-2-7B	1024	58.97s	182.11s	133.28s
Llama-3-3.2B	1024	26.67s	77.11s	58.15s
Llama-3-1B	1024	10.79s	26.62s	21.00s

Observations and Conclusion

- **Performance Insights:** Distributed FlashAttention demonstrates effective scaling for LLaMA models, with improvements in bandwidth and latency for smaller parameterized models (LLaMA-3-1B).
- **Challenges:** Larger models require optimization to address memory and latency bottlenecks.
- **Future Work:** Create a single package for the distributed flash attention, so that it can be used for other family of model’s.

GitHub Repository

https://github.com/venugopalkadamba/hpml-final-project

NYU

Thank You

NYU