Objective

Evaluate different configurations of a large language model (LLM) to determine the most effective setup for answering questions from a long context. Key considerations include memory optimization, latency, and response quality.

Experiment Details

1. **Variables**:
   - **PyTorch & Transformer Versions**:
     - PyTorch 2.4.0 with Transformers 4.43.3
     - PyTorch 2.5.1 with Transformers 4.46.1
   - **FlashAttention**: Enabled or disabled
   - **Quantization**: No quantization vs. 8-bit quantization.
   - **Environment Variable (PYTORCH_CUDA_ALLOC_CONF)**: Applied only for PyTorch 2.4.0 setup, with and without `expandable_segments:True`
2. **Experimental Setup**:
   - Conducted on a virtual machine with 2 A100 GPUs
   - Model: Meta-Llama-3.1-8B-Instruct
3. **Prompts & Context**:
   - Context size: 70,483 tokens
   - Recorded GPU memory usage, latency, response quality, and exceptions

Detailed Description of Results

Metric	PyTorch 2.4.0 & Transformers 4.43.3	PyTorch 2.5.1 & Transformers 4.46.1
Success Rate	Only 17% of the experiments completed. Failures due to "CUDA out of memory" exceptions.	100% completion rate of experiments
Memory Optimization	Higher memory usage. Memory failures reduced when PYTORCH_CUDA_ALLOC_CONF was set to `expandable_segments:True`.	Lower memory usage. 41% reduction in total memory (maximum allocated + reserved) across 2 GPUs
Latency	Latency was around 17s, but not meaningful due to the low experiment completion rate	Latency reduced by 44% with flashattention-2 (versus default setting). Latency increased by 80% with 8-bit quantization (versus default)
Response Quality	Lower quality; more incorrect responses.	Higher percentage of correct responses with flashattention (83% versus 50% for default). Quantization seems to reduce the quality of responses

General Observations Across Experiments

1. **Significant Performance Improvement with Newer Library Versions:** The newer PyTorch and Transformers libraries (pytorch 2.5.1 & transformers 4.46.1) demonstrate drastically improved memory management compared to the older versions (pytorch 2.4.0 & transformer 4.43.3). All experiments using the newer versions successfully completed, while the older versions frequently encountered memory errors and timeouts.
2. **FlashAttention Improves Response Quality and Reduces Latency:** FlashAttention consistently led to lower latency for successful prompts and also improved the average response quality, particularly noticeable with the newer library versions.
3. **8-bit Quantization Increases Latency, Mixed Impact on Response Quality:** While 8-bit quantization enabled completion of all prompts with the newer library versions, it significantly increased the latency compared to non-quantized setups1. It had a mixed impact on response quality, sometimes improving it and sometimes degrading it.

Highlight:

These results highlight the rapid and meaningful improvements in open-source libraries like PyTorch and Transformers, which have made substantial strides in efficiency and performance within a short time. These advancements not only enhance the feasibility of deploying AI systems for practical, long-context tasks but also drive broader adoption by reducing hardware constraints and making state-of-the-art AI more accessible to diverse users and industries.

Next Steps

1. **Evaluate other LLMs**:
   - Repeat the experiment with other high-quality llms such as Qwen2.5
2. **Other techniques**:
   - Evaluate other techniques such as torch.compile and additional quantizations

3. **LLM server**:
   - Repeat experiment with huggingface “text-generation-inference”

Data Tables

Completion of experiments

Total planned experiments = 24

Library version	Count of Experiments
pytorch2.4.0 & transformer4.43.3	4
pytorch2.5.1 & transformers4.46.1	24

Total memory

(Maximum allocated + reserved) across the 2 GPUs (GB):

Library version	Average of Pre_Gen_Total_Mem(gb)	Average of Post_Gen_Total_Mem(gb)
pytorch2.4.0 & transformer4.43.3	135.44	168.12
pytorch2.5.1 & transformers4.46.1	91.08	99.98

Results quality

pytorch2.5.1 & transformers4.46.1

-1 means the response was incorrect. 0 means the response was incomplete. 1 means the response was correct. 2 means the response was correct and detailed

Library version	-1	0	1	2
8_bit	33.33%	33.33%	0.00%	33.33%
8_bit_flash	50.00%	16.67%	16.67%	16.67%
default	50.00%	0.00%	16.67%	33.33%
flashattention	0.00%	16.67%	66.67%	16.67%
Grand Total	**33.33%**	**16.67%**	**25.00%**	**25.00%**

Latency

Across prompts (1-6) in seconds: pytorch2.5.1 & transformers4.46.1

Library version	1	2	3	4	5	6
8_bit	36.16	29.49	89.76	89.88	89.40	88.92
8_bit_flash	17.04	21.91	63.44	63.30	62.98	62.37
default	22.97	19.66	40.16	51.01	51.06	51.17

flashattention	15.02	17.55	22.05	25.87	25.89	25.96