Department of Computer Science and Engineering
School of Electrical and Computer Sciences
Indian Institute of Technology, Bhubaneswar

HIGH PERFORMANCE COMPUTER ARCHITECTURE

PROJECT PHASE 2

Analytical Modeling of LLM Computation and Communication on Real CPU/GPU Hardware

Team Members

Nali Bhavana - 24AI06013
Puvvula Nikhileswari - 24AI06017
Devesh Sharma - 24CS06002
Sapna Vishwakarma - 24CS06012

1 Introduction

Large Language Models (LLMs) have become essential for tasks like summarization, translation, question answering, and code generation. However, their high computational demands and varying hardware constraints pose challenges for real-world deployment. Efficient inference is crucial for reducing latency, optimizing resource utilization, and ensuring scalability across different platforms.

Understanding the computational behavior of LLMs across different hardware setups helps in identifying performance bottlenecks and improving execution efficiency. Profiling various models under diverse workloads provides insights into optimizing resource allocation. Such analysis is essential for designing scalable AI solutions while balancing speed, accuracy, and energy efficiency.

2 Methodology

2.1 Models Evaluated

We profiled four models:

- **Decoder-Only Models:** OPT-350M, BLOOM-560M
- **Encoder-Decoder Models:** T5-Small, BART-Base

2.2 Experiment Setup

- **Device:** Tesla GPU
- **Batch Sizes:** 1, 8, 32, 128
- **Metrics:** Latency, CUDA Execution Time, CPU Execution Time, Memory Bandwidth
- **Workloads:** Summarization, Question Answering, Code Generation, Translation

3 Initial Results and Observations

3.1 Latency vs. Batch Size

Trend: Latency increases with batch size but varies by model
Observation: Encoder-decoder models (T5, BART) have higher latency due to additional decoder computations.

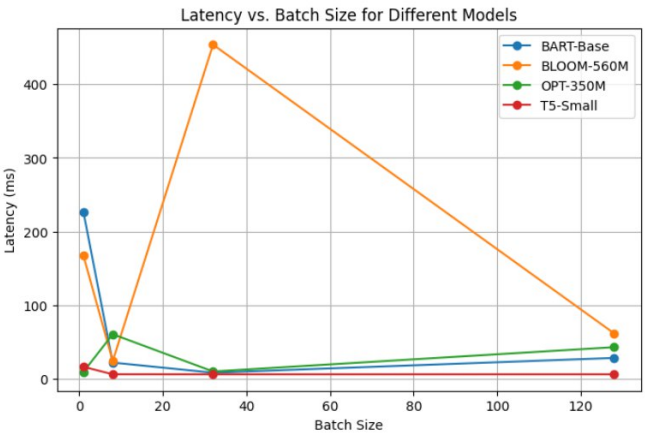


Figure 1: Latency vs Batch Size for different Models

3.2 CUDA vs. CPU Execution Time

Trend: Decoder-only models rely more on CUDA acceleration, while encoder-decoder models involve higher CPU computation overhead.
Observation: CPU bottlenecks impact performance at higher batch sizes.

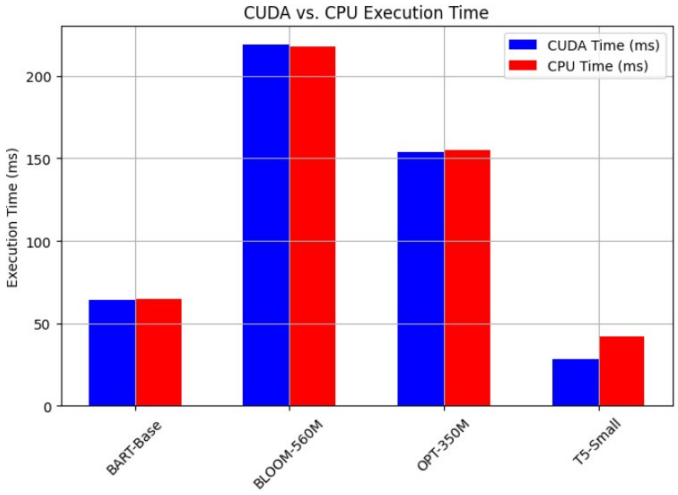


Figure 2: CUDA VS CPU Execution Time Across Models

3.3 Workload-Specific Performance

Summarization & Translation: Encoder-decoder models perform well but at higher latency.
Question Answering & Code Generation: Decoder-only models like OPT-350M are more efficient.

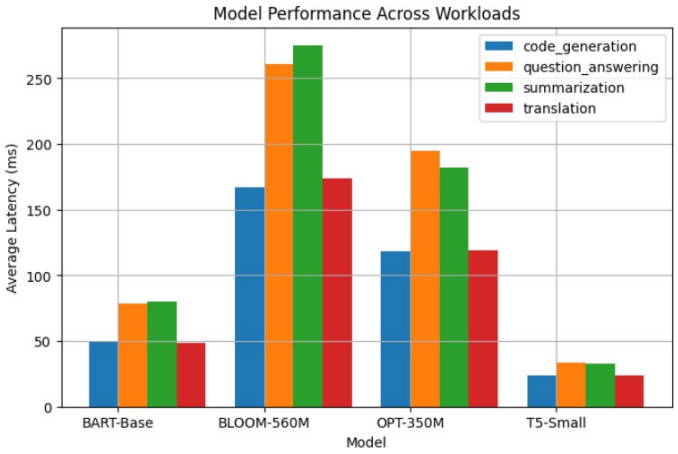


Figure 3: Latency Analysis Across Models

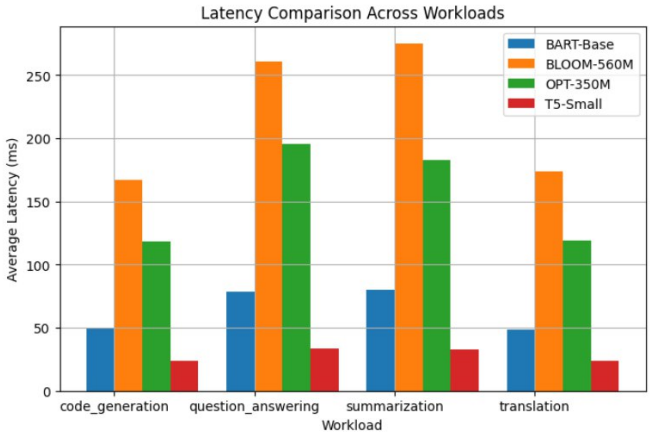
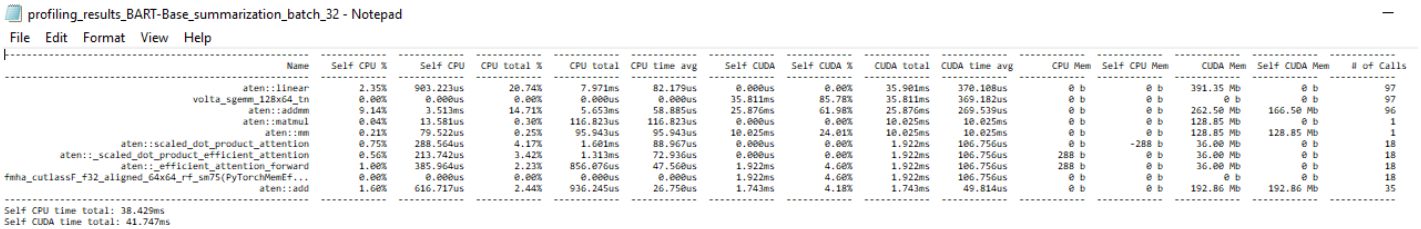


Figure 4: Latency Analysis Across Workloads



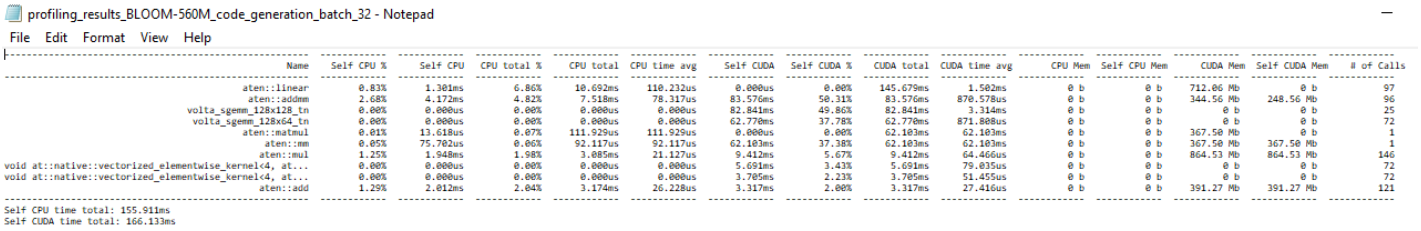
profiling_results_BART-Base_summarization_batch_32 - Notepad

File Edit Format View Help

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	Self CUDA	Self CUDA %	CUDA total	CUDA time avg	CPU Mem	Self CPU Mem	CUDA Mem	Self CUDA Mem	# of Calls
aten::linear	2.35%	903.223us	20.74%	7.971ms	82.179us	0.000us	0.00%	35.901ms	370.108us	0 b	0 b	391.35 Mb	0 b	97
volta_sgemm_128x64_tn	0.00%	0.000us	0.00%	0.000us	0.000us	35.811ms	85.78%	35.811ms	369.182us	0 b	0 b	0 b	0 b	97
aten::addmm	9.14%	3.513ms	14.71%	5.653ms	58.885us	25.876ms	61.98%	25.876ms	269.539us	0 b	0 b	262.50 Mb	166.50 Mb	96
aten::matmul	0.04%	13.581us	0.30%	116.823us	116.823us	0.000us	0.00%	10.025ms	10.025ms	0 b	0 b	128.85 Mb	0 b	1
aten::mm	0.21%	79.522us	0.25%	95.943us	95.943us	10.025ms	24.01%	10.025ms	10.025ms	0 b	0 b	128.85 Mb	128.85 Mb	1
aten::scaled_dot_product_attention	0.75%	288.564us	4.17%	1.601ms	88.967us	0.000us	0.00%	1.922ms	106.756us	0 b	-288 b	36.00 Mb	0 b	18
aten::_scaled_dot_product_efficient_attention	0.56%	213.742us	3.42%	1.313ms	72.936us	0.000us	0.00%	1.922ms	106.756us	288 b	0 b	36.00 Mb	0 b	18
aten::_efficient_attention_forward	1.00%	385.964us	2.23%	856.076us	47.560us	1.922ms	4.60%	1.922ms	106.756us	288 b	0 b	36.00 Mb	0 b	18
fmha_cutlassF_f32_aligned_64x64_rf_sm75(PyTorchMemEf...	0.00%	0.000us	0.00%	0.000us	0.000us	1.922ms	4.60%	1.922ms	106.756us	0 b	0 b	0 b	0 b	18
aten::add	1.60%	616.717us	2.44%	936.245us	26.750us	1.743ms	4.18%	1.743ms	49.814us	0 b	0 b	192.86 Mb	192.86 Mb	35

Self CPU time total: 38.429ms
Self CUDA time total: 41.747ms

Figure 5: Best Workload of BART Inference Analysis



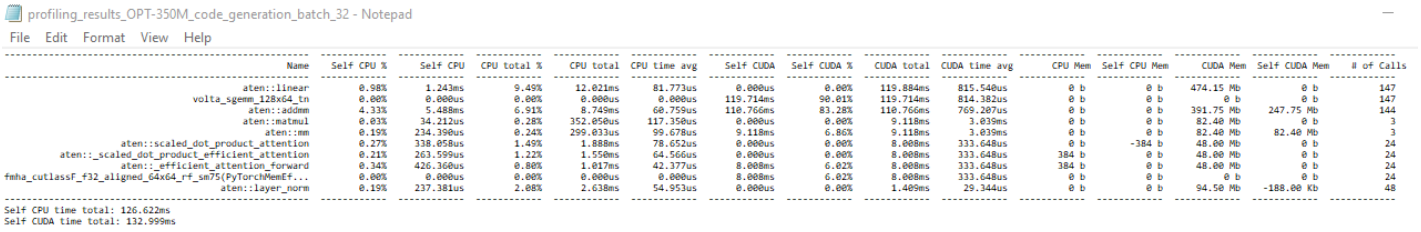
profiling_results_BLOOM-560M_code_generation_batch_32 - Notepad

File Edit Format View Help

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	Self CUDA	Self CUDA %	CUDA total	CUDA time avg	CPU Mem	Self CPU Mem	CUDA Mem	Self CUDA Mem	# of Calls
aten::linear	0.83%	1.301ms	6.86%	10.692ms	110.232us	0.000us	0.00%	145.679ms	1.502ms	0 b	0 b	712.06 Mb	0 b	97
aten::addmm	2.68%	4.172ms	4.82%	7.518ms	78.317us	83.576ms	50.31%	83.576ms	870.578us	0 b	0 b	344.56 Mb	248.56 Mb	96
volta_sgemm_128x128_tn	0.00%	0.000us	0.00%	0.000us	0.000us	82.841ms	49.86%	82.841ms	3.314ms	0 b	0 b	0 b	0 b	25
volta_sgemm_128x64_tn	0.00%	0.000us	0.00%	0.000us	0.000us	62.770ms	37.78%	62.770ms	871.808us	0 b	0 b	0 b	0 b	72
aten::matmul	0.01%	13.618us	0.07%	111.929us	111.929us	0.000us	0.00%	62.103ms	62.103ms	0 b	0 b	367.50 Mb	0 b	1
aten::mm	0.05%	75.702us	0.06%	92.117us	92.117us	62.103ms	37.38%	62.103ms	62.103ms	0 b	0 b	367.50 Mb	367.50 Mb	1
aten::mul	1.25%	1.948ms	1.98%	3.085ms	21.127us	9.412ms	5.67%	9.412ms	64.466us	0 b	0 b	864.53 Mb	864.53 Mb	146
void at::native::vectorized_elementwise_kernel<4, at...	0.00%	0.000us	0.00%	0.000us	0.000us	5.691ms	3.43%	5.691ms	79.035us	0 b	0 b	0 b	0 b	72
void at::native::vectorized_elementwise_kernel<4, at...	0.00%	0.000us	0.00%	0.000us	0.000us	3.705ms	2.23%	3.705ms	51.455us	0 b	0 b	0 b	0 b	72
aten::add	1.29%	2.012ms	2.04%	3.174ms	26.228us	3.317ms	2.00%	3.317ms	27.416us	0 b	0 b	391.27 Mb	391.27 Mb	121

Self CPU time total: 155.911ms
Self CUDA time total: 166.133ms

Figure 6: Best Workload of BLOOM Inference Analysis



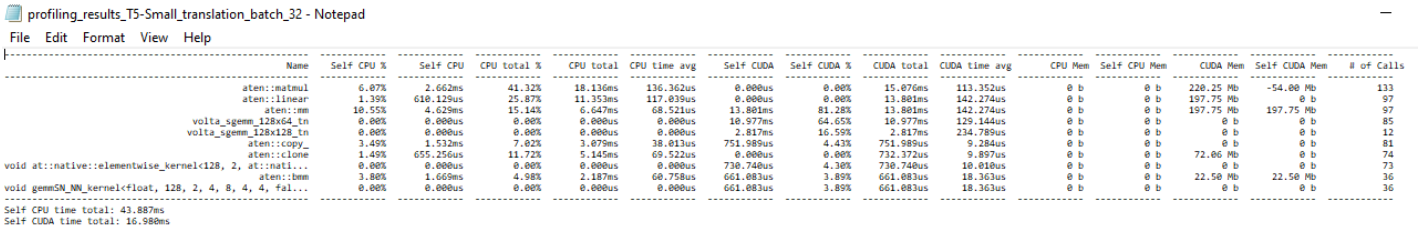
profiling_results_OPT-350M_code_generation_batch_32 - Notepad

File Edit Format View Help

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	Self CUDA	Self CUDA %	CUDA total	CUDA time avg	CPU Mem	Self CPU Mem	CUDA Mem	Self CUDA Mem	# of Calls
aten::linear	0.98%	1.243ms	9.49%	12.021ms	81.773us	0.000us	0.00%	119.884ms	815.540us	0 b	0 b	474.15 Mb	0 b	147
volta_sgemm_128x64_tn	0.00%	0.000us	0.00%	0.000us	0.000us	119.714ms	90.01%	119.714ms	814.382us	0 b	0 b	0 b	0 b	147
aten::addmm	4.33%	5.488ms	6.91%	8.749ms	60.759us	110.766ms	83.28%	110.766ms	769.207us	0 b	0 b	391.75 Mb	247.75 Mb	144
aten::matmul	0.03%	34.212us	0.28%	352.050us	117.350us	0.000us	0.00%	9.118ms	3.039ms	0 b	0 b	82.40 Mb	0 b	3
aten::mm	0.19%	234.390us	0.24%	299.033us	99.678us	9.118ms	6.86%	9.118ms	3.039ms	0 b	0 b	82.40 Mb	82.40 Mb	3
aten::scaled_dot_product_efficient_attention	0.27%	338.058us	1.49%	1.888ms	78.652us	0.000us	0.00%	8.008ms	333.648us	0 b	-384 b	48.00 Mb	0 b	24
aten::_scaled_dot_product_efficient_attention	0.21%	263.599us	1.22%	1.550ms	64.566us	0.000us	0.00%	8.008ms	333.648us	384 b	0 b	48.00 Mb	0 b	24
aten::_efficient_attention_forward	0.34%	426.360us	0.80%	1.017ms	42.377us	8.008ms	6.02%	8.008ms	333.648us	384 b	0 b	48.00 Mb	0 b	24
fmha_cutlassF_f32_aligned_64x64_rf_sm75(PyTorchMemEf...	0.00%	0.000us	0.00%	0.000us	0.000us	8.008ms	6.02%	8.008ms	333.648us	0 b	0 b	0 b	0 b	24
aten::layer_norm	0.19%	237.381us	2.08%	2.638ms	54.953us	0.000us	0.00%	1.409ms	29.344us	0 b	0 b	94.50 Mb	-188.00 Kb	48

Self CPU time total: 126.622ms
Self CUDA time total: 132.999ms

Figure 7: Best Workload of OPT Inference Analysis



profiling_results_T5-Small_translation_batch_32 - Notepad

File Edit Format View Help

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	Self CUDA	Self CUDA %	CUDA total	CUDA time avg	CPU Mem	Self CPU Mem	CUDA Mem	Self CUDA Mem	# of Calls
aten::matmul	6.07%	2.662ms	41.32%	18.136ms	136.362us	0.000us	0.00%	15.076ms	113.352us	0 b	0 b	220.25 Mb	-54.00 Mb	133
aten::linear	1.39%	610.129us	25.87%	11.353ms	117.039us	0.000us	0.00%	13.801ms	142.274us	0 b	0 b	197.75 Mb	0 b	97
aten::mm	10.55%	4.629ms	15.14%	6.647ms	68.521us	13.801ms	81.28%	13.801ms	142.274us	0 b	0 b	197.75 Mb	197.75 Mb	97
volta_sgemm_128x64_tn	0.00%	0.000us	0.00%	0.000us	0.000us	10.977ms	64.65%	10.977ms	129.144us	0 b	0 b	0 b	0 b	85
volta_sgemm_128x128_tn	0.00%	0.000us	0.00%	0.000us	0.000us	2.817ms	16.59%	2.817ms	234.789us	0 b	0 b	0 b	0 b	12
aten::copy_	3.49%	1.532ms	7.02%	3.079ms	38.013us	751.989us	4.43%	751.989us	9.284us	0 b	0 b	0 b	0 b	81
aten::clone	1.49%	655.256us	11.72%	5.145ms	69.522us	0.000us	0.00%	732.372us	9.897us	0 b	0 b	72.06 Mb	0 b	74
void at::native::elementwise_kernel<128, 2, at::nati...	0.00%	0.000us	0.00%	0.000us	0.000us	730.740us	4.30%	730.740us	10.010us	0 b	0 b	0 b	0 b	73
aten::bmm	3.86%	1.669ms	4.98%	2.187ms	60.758us	661.083us	3.89%	661.083us	18.363us	0 b	0 b	22.50 Mb	22.50 Mb	36
void gemmSN_NN_kernel<float, 128, 2, 4, 8, 4, 4, fal...	0.00%	0.000us	0.00%	0.000us	0.000us	661.083us	3.89%	661.083us	18.363us	0 b	0 b	0 b	0 b	36

Self CPU time total: 43.887ms
Self CUDA time total: 16.980ms

Figure 8: Best Workload of T5 Inference Analysis

4 Project Timeline

Table 1: Project Timeline

Task	Duration	Status
Model Selection	Week 1	Completed
Profiling Implementation of Basic Models	Week 2	Completed
Data Collection & GPU Setup	Week 3	Ongoing
Advanced Profiling (FLOPs)	Week 4	Planned
Optimization Strategies	Week 5	Planned
Report & Documentation	Week 6	Planned

5 Next Steps

- **Memory Profiling:** Use nvidia-smi to track memory consumption across models.
- **Compute FLOPs Estimation:** Integrate fvcore.nn.FlopCountAnalysis to estimate model computational complexity.
- **Optimization Strategies:** Investigate quantization and pruning techniques to improve efficiency.
- **Validation on Real Hardware:** Test models on different GPU architectures to analyze scalability.

6 Conclusion

This project analyzed the computational and communication characteristics of LLMs on real CPU/GPU hardware. Our profiling indicates that decoder-only models are more efficient for tasks like code generation and QA, while encoder-decoder models offer better summarization and translation accuracy but with higher latency. CPU execution overhead highlights the need for CUDA acceleration.

References

1 Amey Agrawal, Nitin Kedia, Jayashree Mohan, Ashish Panwar, Nipun Kwatra, Bhargav Gulavani, Ramachandran Ramjee, Alexey Tumanov. "Vidur: A Large-Scale Simulation Framework For LLM Inference."

2 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. "Attention Is All You Need."