WHITE PAPER

Accelerating Large Language Model Use Cases with Intel Flex GPU and Intel Extension for PyTorch (IPEX)

OUTLINE

TABLES

FIGURES

Introduction

Large Language Models (LLMs) are powerful tools for generating text in a wide range of applications. By training on vast datasets, LLMs can produce coherent, contextually appropriate responses based on input prompts. Their ability to understand and generate human-like text makes them suitable for various tasks, such as content creation, question answering, and text summarization.

LLMs are highly versatile, enabling use cases across industries. They can assist in automating writing processes, enhancing customer interactions, improving language translation, or even generating creative content like stories and poems. Their applications span multiple domains, from everyday text-based tasks to more complex interactions in virtual assistants or conversational agents. In industry settings, LLMs have become essential for automating and improving text-based workflows. For example:

- Content Marketing: LLMs generate high-quality blog posts, product descriptions, and social media content, reducing the need for manual content creation. By producing engaging and tailored content efficiently, businesses can scale their marketing efforts with minimal human intervention.
- Customer Support: LLM-powered chatbots and virtual assistants handle routine customer queries in real-time, providing accurate and helpful responses. This not only enhances customer satisfaction but also reduces the workload on human agents, allowing them to focus on more complex issues.
- Healthcare: LLMs assist in medical documentation, including writing patient reports, summarizing clinical notes, and generating discharge summaries. They ensure that healthcare professionals save time on administrative tasks, allowing them to focus more on patient care.
- Legal and Financial: LLMs assist in drafting legal documents or generating financial reports, saving time on manual tasks while ensuring high accuracy in complex fields.

To ensure optimal performance and cost-effectiveness in LLM deployment, it's crucial to evaluate their inference capabilities. To this end, we benchmarked LLM models on the QCT DevCloud platform[1], a collaborative effort between Quanta Cloud Technology (QCT) and Intel. This platform is specifically designed to address the unique challenges of developing and testing high-performance computing (HPC) and AI applications.

QCT DevCloud provides a comprehensive remote development environment, granting access to cutting-edge hardware, storage, and networking resources. Leveraging the advanced features of Intel Xeon processors, such as enhanced security and accelerated HPC/AI capabilities, this platform streamlines development workflows. Precompiled workloads, a robust software stack, and tools like Intel oneAPI further simplify the process by eliminating the need for complex installations.

Moreover, QCT DevCloud enables testing across various CPU and GPU environments, including Intel Flex 170 GPUs. This flexibility empowers developers to efficiently evaluate system performance, reducing development time and resource constraints, ultimately leading to significant cost savings. By utilizing this platform, industries can optimize their LLM deployments, increasing productivity, reducing labor costs, and streamlining operations.

Experiment Setup

To evaluate the performance of LLM inference, we benchmarked the models on an Intel Flex 170 GPU, utilizing Intel Extension for PyTorch (IPEX) to optimize deep learning workloads. IPEX enhances PyTorch operations by maximizing the use of Intel hardware features, such as the Flex GPU, improving overall speed and efficiency.

In this use case, we evaluate three models to demonstrate AI performance across various tasks:

- **GPT-2**[2]: A generative language model from OpenAI, ideal for text generation tasks like completion, summarization, and creative writing.
- **BERT**[3]: A transformer-based model designed for natural language understanding, excelling in tasks like text classification and question answering.
- **ResNet**[4]: A convolutional neural network widely used for image classification, known for its effectiveness in computer vision tasks.

The ResNet model is included to compare the performance of LLMs with the traditional image classification model. The experiment setup of these models is summarized in the **Table 1** below.

	Batch Size	Configurations
GPT-2	64	Max Output Length: 128 tokens Iterations: 20
BERT	256	Max Output Length: 512
ResNet	512	Image Size: 224 × 224 pixels, 3 channels (RGB)

Table 1: Model Experiment Configurations

All models were optimized using **Intel Extension for PyTorch (IPEX)** on Intel Flex GPUs to enhance performance and efficiency compared to traditional CPU-based processing.

Intel Extension for PyTorch (IPEX) is a performance optimization library designed to enhance PyTorch's deep learning workloads on Intel hardware. IPEX integrates seamlessly with PyTorch, allowing users to take advantage of Intel-specific optimizations without modifying their existing code.

By leveraging IPEX, deep learning models like GPT-2, BERT, and ResNet can experience significant speedups on Intel Flex GPUs, delivering improved efficiency across various AI applications.

Evaluation Results & Analysis

The performance comparison results between Intel Flex GPU and Intel Xeon 8470 CPU are shown in Figure 1 ~ Figure 3 for the three tested models, respectively. The overall speedup factor of Intel GPU over Intel CPU is summarized in Table 2. The results show a significant speedup in text generation and inference when running models on the Intel Flex GPU with IPEX optimizations compared to the CPU. For **GPT-2,** the Intel Flex GPU was more than **3 times faster** than the CPU, demonstrating the efficiency gains in LLM text generation. For **BERT,** the GPU accelerated inference performance by almost **8 times**, highlighting the Flex GPU's ability to handle high-volume sequence-based tasks. Finally, although the speedup for ResNet image classification was less pronounced, the GPU still outperformed the CPU by **1.6 times**, showing how Intel Flex GPUs can enhance vision-related workloads as well. The result also indicates that the Intel Flex GPU can have better performance acceleration for the text-based models, like GPT-2 and BERT, than the vision models like ResNet.

	CPU (Intel Xeon 8470) Total Time (sec)	XPU (Intel Flex GPU) Total Time (sec)	Speedup Factor of Intel Flex GPU
GPT-2	85.12	27.37	3.10
BERT	5.557	0.697	7.97
ResNet	1.711	1.064	1.60

Table 2: Performance Comparison Between Intel CPU and Intel Flex GPU.

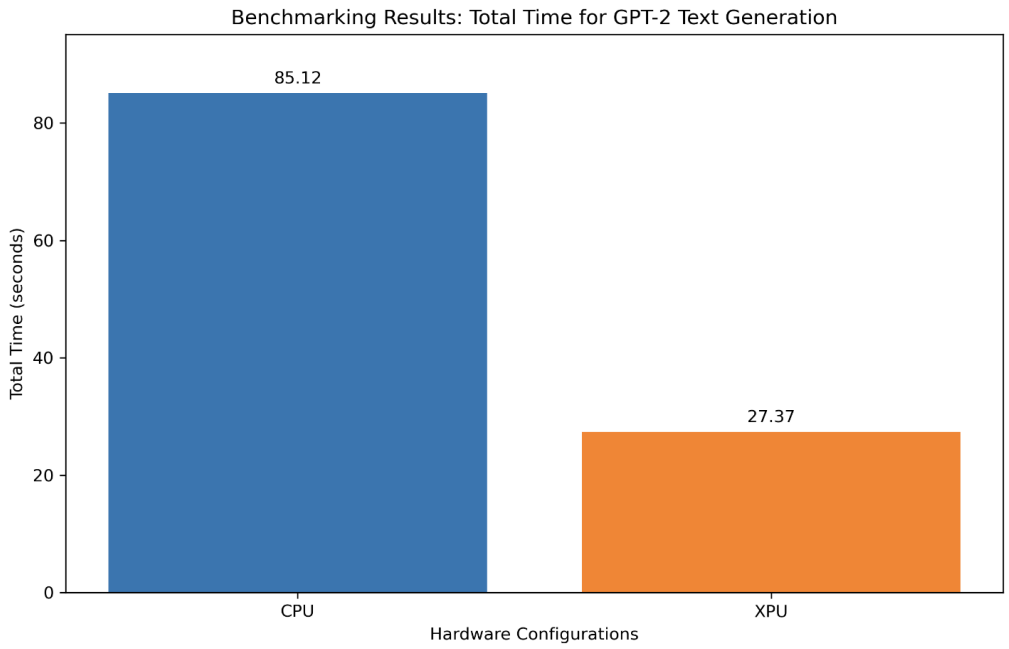

Figure 1: GPT-2 benchmark result.

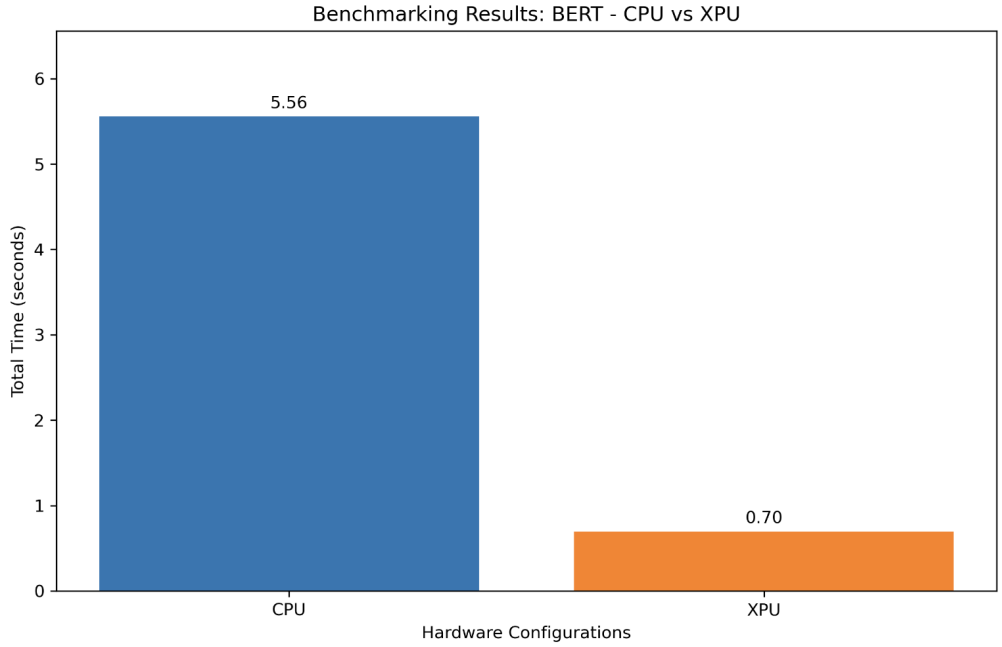

Figure 2: BERT benchmark result.

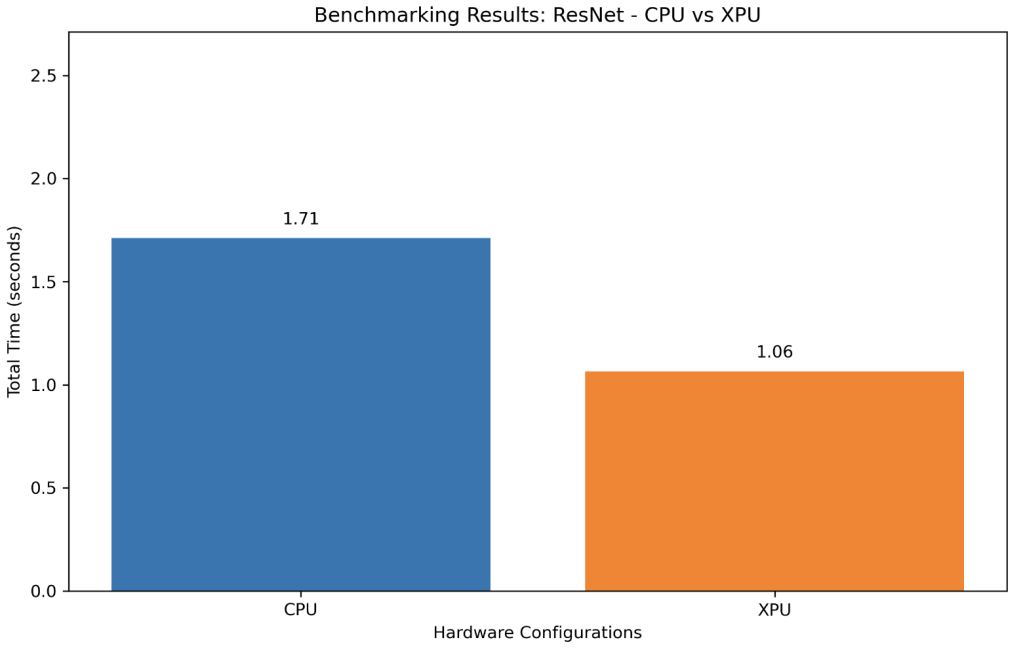

Figure 3 : ResNet benchmark result.

We also profiled the performance using **Intel VTune Profiler**, a tool designed to analyze and optimize code running on Intel hardware with the following command, “vtune -collect gpu-offload python {program name}.py”.

The profiling results for the three tested models are presented in **Figure 4** to **Figure 6**. These figures illustrate that the workload is compute-bound based on the following observations:

- **Execution Time Dominance**: **Figure 4** demonstrates that GPU 0 achieves an execution duration of 977.748 milliseconds, with a predominant allocation of time to the execution phase as opposed to host-to-device or device-to-host data transfer operations. This delineation underscores the computationally intensive nature of the workload, with minimal overhead attributed to data movement.
- **GPU Utilization**: The green bars in the timeline represent GPU computation tasks, which are active for most of the execution period. The dense and continuous nature of these bars suggests that the GPU is fully utilized and not idling, reinforcing the conclusion that the workload is compute-bound.
- **Call Stack Analysis**: The call stack reveals that most of the operations (e.g., Abstract Tensor and Torch library functions) are computation focused. This further emphasizes the focus on core computation tasks rather than memory-bound or other bottlenecks.

Similar patterns are also observed in **Figure 5** and **Figure 6**. According to the profiling data, the workload is primarily constrained by the computational capacity of the hardware. As a result, Intel GPUs demonstrate significant performance acceleration compared to CPUs in these scenarios.

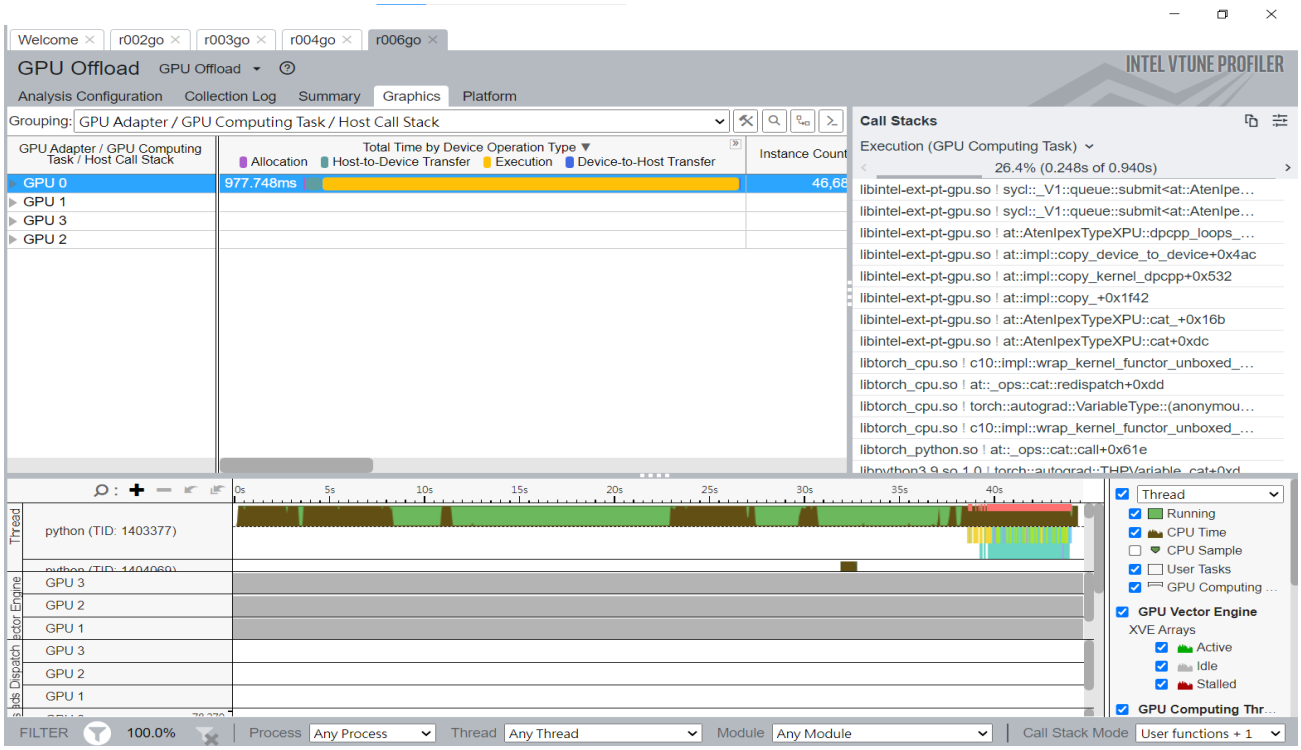

Figure 4: GPT-2 profiling result.

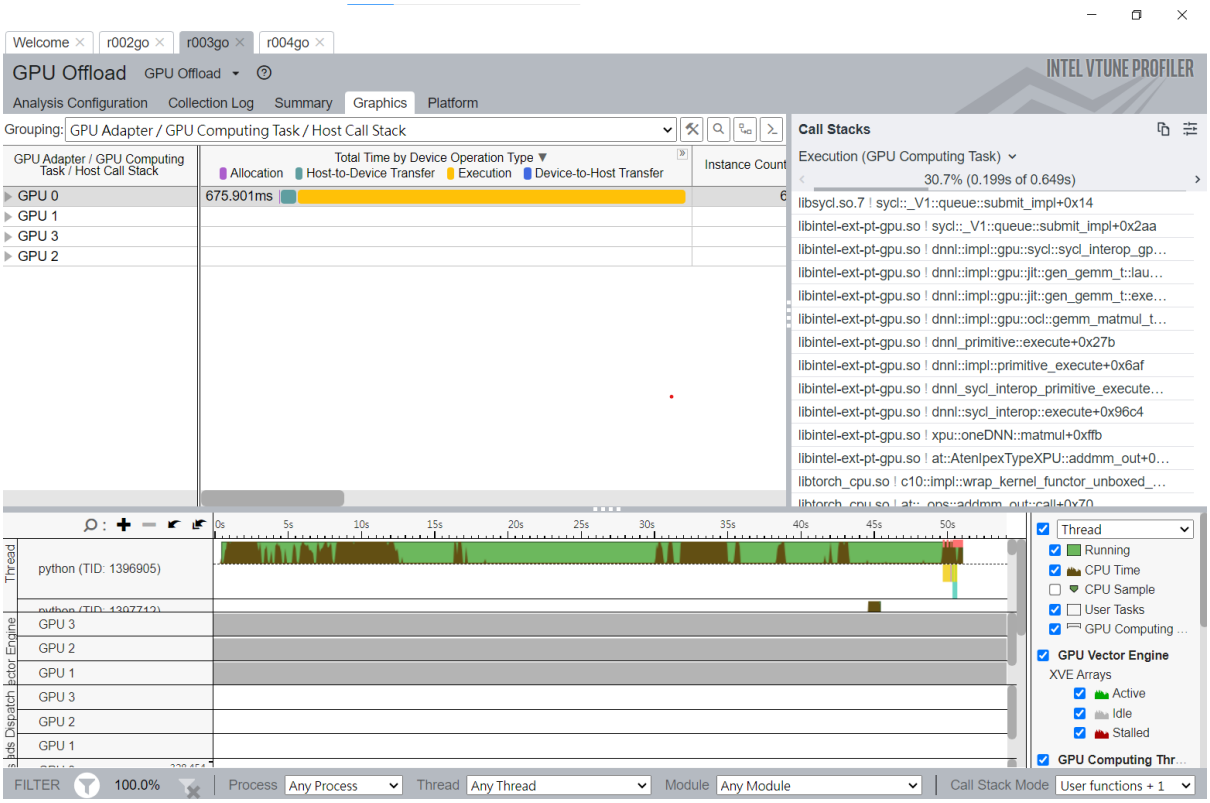

Figure 5: BERT profiling result.

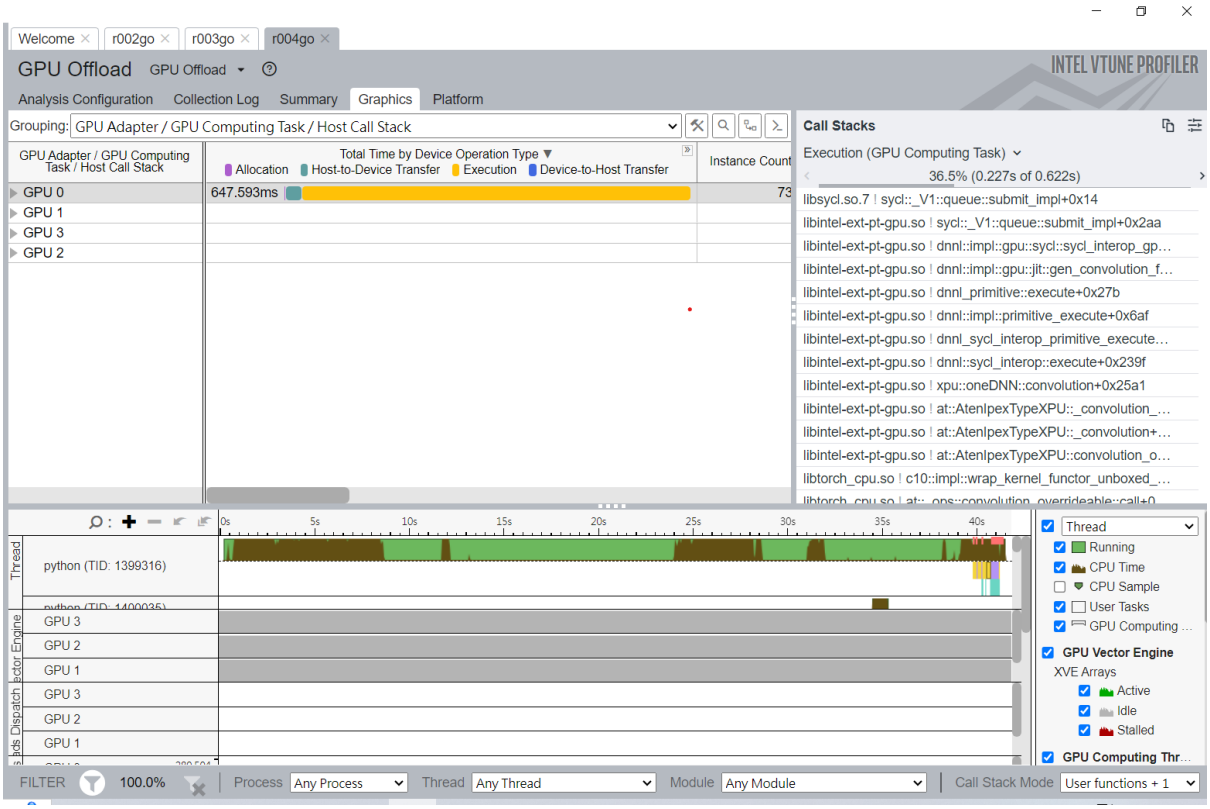

Figure 6: ResNet profiling result.

Conclusion

This experiment demonstrates the efficiency of using **Intel Flex 170 GPUs** in conjunction with **Intel Extension for PyTorch (IPEX)** for several tasks. The benchmarks show that GPU acceleration, combined with IPEX optimizations, drastically reduces the time required for generating outputs. With the Intel Flex GPU completing the task nearly three times faster than the CPU, this makes it an ideal solution for businesses and organizations looking to deploy AI applications.

By leveraging the power of **Intel Flex GPUs** and the efficiency enhancements provided by **IPEX**, industries can significantly optimize their AI workflows. The improved speed and resource efficiency can lead to reduced operational costs, faster response times, and enhanced user experience. The combination of Intel's hardware and software optimizations positions it as a highly scalable option for AI-driven applications. Additionally, the QCT DevCloud platform has streamlined the setup and preparation process for the system and development environment. The platform also enables efficient evaluation of resource requirements, helping us identify optimal configurations to support high-performance computing needs.

References

1. Ware A. QCT and Intel Codeveloped QCT DevCloud Program to Jumpstart HPC and AI Development. HPCwire. August 7, 2023. Accessed December 4, 2024. https://www.hpcwire.com/2023/08/07/qct-devcloud-program-provides-a-platform-to-jumpstart-hpc-and-ai-development/

2. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. *OpenAI Blog*. 2019;1(8):9.

3. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Published online May 24, 2019. doi:10.48550/arXiv.1810.04805

4. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Published online December 10, 2015. doi:10.48550/arXiv.1512.03385