Boosting AI Inferencing for LLM Models on Intel CPU-Powered Lenovo Servers, Part 1

Planning / Implementation

This paper is the first in a three-part series analyzing the performance of large language models (LLMs) on Intel CPUs. Through detailed benchmarking, we assess latency, throughput, and resource utilization, providing insights into the performance of different model architectures and their computational efficiency.

The main idea is to provide comparative insights into how different LLM architectures leverage Intel CPU resources rather than focusing on absolute performance values. If specific benchmark numbers do not meet business expectations, they should be interpreted as relative comparisons rather than definitive performance constraints.

This series of papers is specifically tailored to Intel's hardware and software ecosystem. This document, Part 1, showcases how LLM architectures impact the AI performance based on key inferencing benchmarks. In Part 2 and Part 3, we will introduce Intel-exclusive technologies such as:

- Intel Advanced Matrix Extensions (Intel AMX) for enhanced AI acceleration
- Intel Extension for PyTorch (IPEX) for optimized deep learning inference
- Intel Advanced Vector Extensions 512 (Intel AVX-512) for workloads and usages acceleration

These Intel-specific optimizations play a crucial role in improving inference performance and energy efficiency.

Intel CPUs and their use with AI workloads

The Lenovo Press paper Accelerating RAG Pipelines for Enterprise LLM Applications using OpenVINO on the Lenovo ThinkSystem SR650 V3 with 5th Gen Intel Xeon Scalable Processors highlights the SR650 V3 exceptional scalability and performance for Generative AI workloads. Designed to meet low-latency requirements (~100ms) for applications like real-time chatbots, this 2U server offers advanced features such as support for DDR5-5600 MT/s memory, Intel Advanced Matrix Extensions (AMX), and flexible storage configurations. Its energy-efficient design includes direct-water cooling (DWC) and high-efficiency power supplies, ensuring both performance and operational cost savings.

These attributes making it an ideal choice for demanding AI workloads, high-performance computing, and enterprise applications that require robust processing power and reliability.

In this series of papers, we conduct the benchmarking on varies Lenovo devices powered by Intel CPUs. The idea is to demonstrate that the impact of LLM architecture exist in both edge servers and rack servers. Since different LLMs serve distinct use cases, business should choose the right LLM to meet their requirements.

The servers in our lab environment that we are using for the series of paper are the following:

- Part 1 (this paper): ThinkEdge SE450 with the 3rd Gen Intel Xeon Gold 6338N processor
- Part 2: ThinkSystem SR650 V3 with the 4th Gen Intel Xeon Gold 6426Y processor
- Part 3: ThinkSystem SR650 V3 with the 5th Gen Intel Xeon Platinum 8570 processor

LLM Architectures

Large Language Models (LLMs) are designed using different architectures, each suited for specific tasks. Below is an explanation of these architectures and examples of popular LLMs:

- **Decoder-Only Models**:

  These models are designed for generative tasks, where text is generated token by token. They predict the next word in a sequence based on the input provided. Examples include GPT-series, Llama-series.

- **Encoder-Only Models**:

  These models focus on understanding and encoding the input data into a meaningful representation. They are optimized for tasks that require input analysis rather than generation. Examples include BERT, RoBERTa.

- **Encoder-Decoder Models**:

  These models combine both encoding and decoding mechanisms. The encoder processes the input to generate a meaningful representation, which the decoder uses to produce the output. Examples include T5, BART.

These architectures serve as the foundation for evaluating the performance of LLMs under varying workloads and hardware configurations in this study. Choosing an overly resource intensive LLM may lead to system crashes, while under-utilizing hardware can result in wasted computational potential and higher operational costs. The ability to pair the right LLM with the appropriate hardware ensures operational efficiency, cost-effectiveness, and consistent performance, making it a cornerstone of successful AI Inference.

The figure below illustrates the differences between encoder-only and decoder-only architectures, highlighting their distinct processing mechanisms.

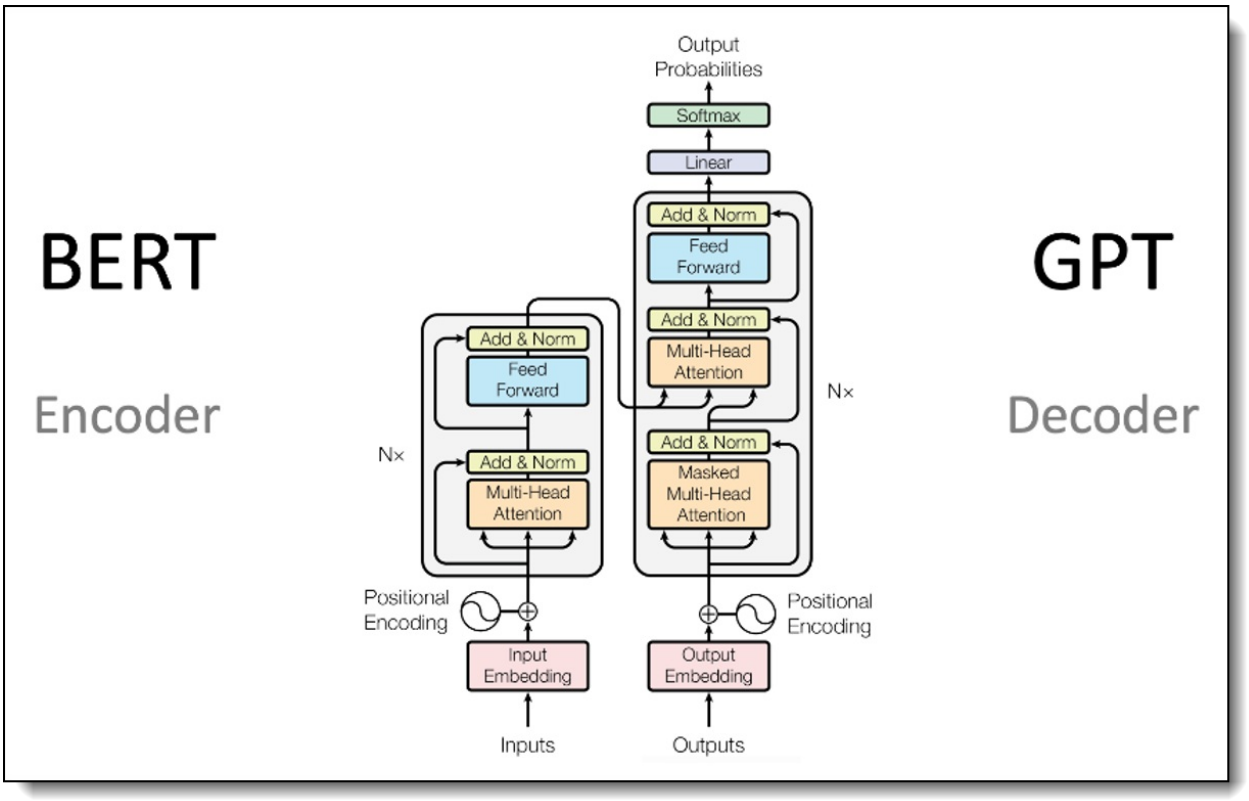

Figure 1. Transformer-based Architectures – BERT (Encoder-Only) vs. GPT (Decoder-Only)

Methodology

LLMs vary significantly in architecture, impacting their performance across different tasks and hardware configurations. To understand these differences, we evaluated three representative models—decoder-only, encoder-only, and encoder-decoder—while controlling for model size.

Models Evaluated

The following three models were evaluated:

- **Decoder-only**: OPT-350M, commonly used for Chatbots, code autocomplete, story generation, and summarization (prompt-based).
- **Encoder-only**: BERT-large-uncased, suited for tasks such as text classification (sentiment, topic), semantic search, information retrieval, and entity recognition.
- **Encoder-Decoder**: CodeGen-350M-mono, ideal for machine translation, text summarization, question answering, and paraphrasing.

The three selected LLMs—OPT-350M (decoder-only), BERT-large-uncased (encoder-only), and CodeGen-350M-mono (encoder-decoder)—were chosen due to their similar model sizes. This allows a direct comparison of how architecture influences performance, removing size as a confounding variable. The chosen models represent diverse use cases and architectures, providing comprehensive insights into how different LLMs utilize hardware resources.

Metrics Evaluated

To evaluate performance, the following metrics were assessed:

- **Time to First Token**: The time taken to wait before seeing the output.
- **Throughput**: The number of tokens or requests can be processed per second.
- **Resource Utilization**: CPU usage during inference tasks.

These metrics provide a holistic view of how each architecture performs under various workloads and hardware configurations.

Performance Evaluation

In this section, we go through three different key metrics in AI inferencing. The analysis provides insights into how architectural choices influence computational performance and practical deployment.

Evaluation 1: Time to First Token (TTFT)

The first evaluation tested the Time to First Token (TTFT) for input length range from 32 to 512 tokens, and batch sizes of range from 1 to 64 across all three model architectures. The findings are shown in the figure below.

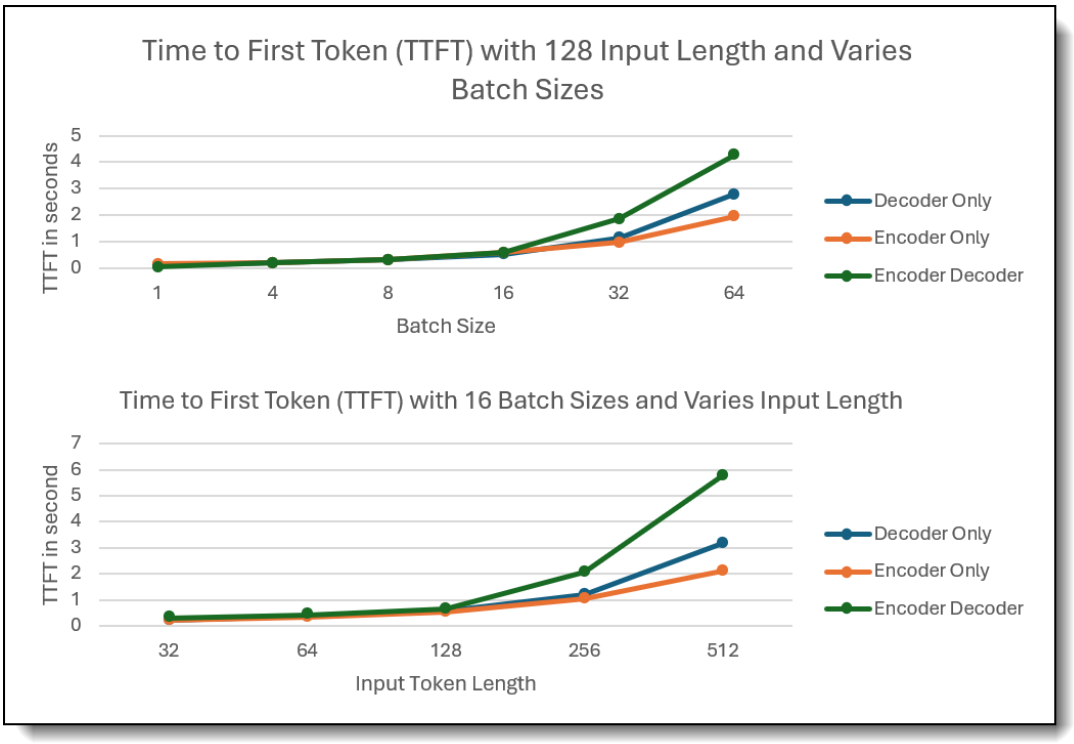

Figure 2. Evaluation 1 findings

Observation:

1. **Batch Size Comparison**: Encoder-Decoder model curve indicates that the additional multi-headed cross-attention layer presented in the architecture significantly impacts performance. We see Encoder-Decoder model exhibited the highest TTFT as batch size increased.
2. **Input Token Length Comparison**: Encoder-Decoder model is more sensitive to the input token length. Decoder-only model shows stable performance up to 256 length, then TTFT significantly increases after that.
3. **Model Comparison**: Encoder-only model demonstrated better scaling efficiency with increased batch sizes and input token length, maintaining relatively stable TTFT compared to the other architectures.

Evaluation 2: Throughput

The second evaluation focused on throughput performance for input length range from 32 to 512 tokens, and batch sizes of range from 1 to 64 across all three model architectures. The findings are shown in the figure below.

Figure 3. Evaluation 2 findings

Observation:

1. **Batch Size Comparison**: Encoder-only model remained relatively flat throughput across different batch sizes, showing consistent efficiency in handling varying loads. The other two models scaled the best in throughput, with optimal performance observed at batch size 16 for the 32-core configuration. This highlights the scalability of decoder involved models under appropriate batch size conditions.
2. **Input Token Length Comparison**: The graph demonstrated with increased input token length, the throughput decreases for all three architectures.
3. **Different Optimized Batch Size**: With fixed input length and CPU cores, LLMs with distinct architecture optimizes at different batch size. Encoder-only LLM optimized at 4 batches, Decoder-only LLM optimized at 32 batches while encoder decoder LLM can stretch to 64 batches.

Evaluation 3: CPU Utilization

The third evaluation analyzed average CPU utilization using a for input length range from 4 to 512 tokens, with batch sizes varying in the range of 1 to 48 in steps of 2. The findings are shown in the figure below.

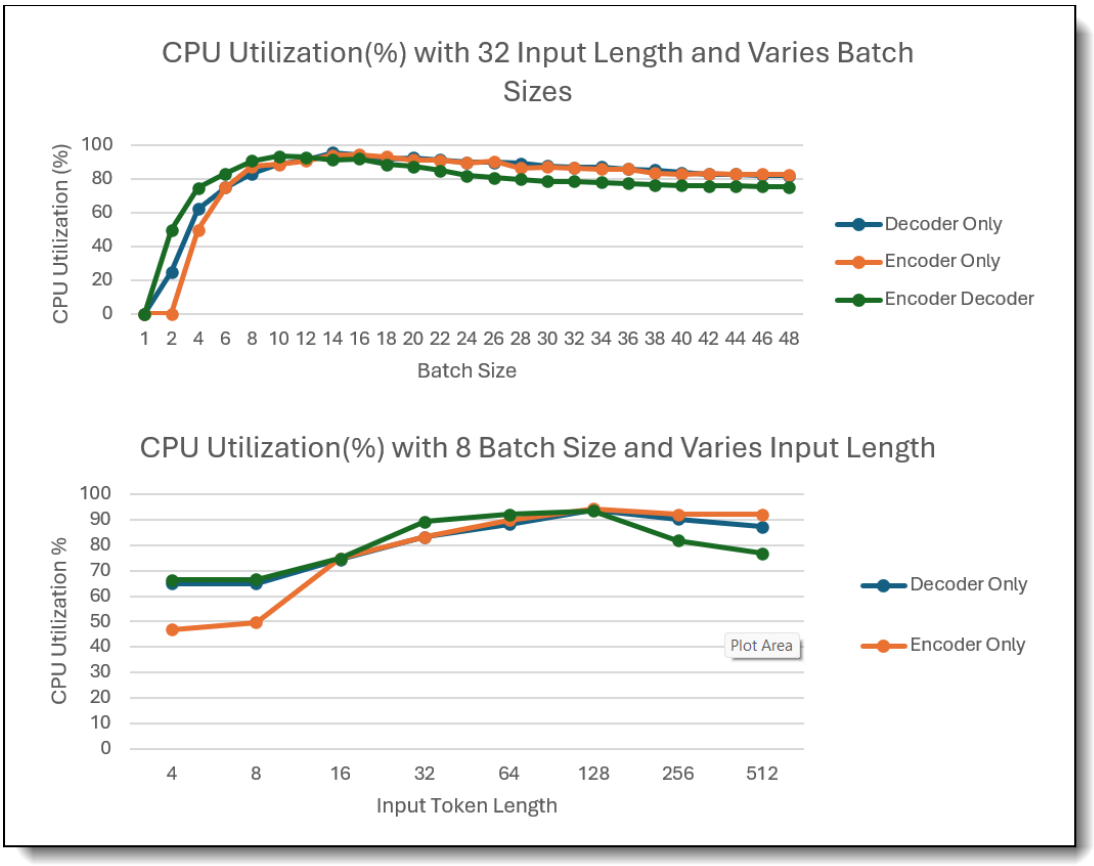

Figure 4. Evaluation 3 findings

Observations:

1. **Batch Size Comparison**: Across all batch sizes, CPU utilization followed a curve with an optimal batch size at which utilization peaked, after which it gradually decreased. This indicates diminishing returns as batch size grows beyond the CPU's processing capacity**.** More of the CPU's time is taken up by moving data around and the CPU cannot be fully utilized as it's bottlenecked by RAM.
2. **Input Token Length Comparison:** Decoder involved models require higher CPU utilization for lower input token length than the Encoder-only model, while Encoder-only model is more sensitive to input token length.
3. **Model Comparison**: Encoder-Decoder models consumed the most CPU resources, reaching the highest utilization (93%) at batch size 8 on the 32-core configuration. In contrast, Encoder-only and Decoder-only models reached their peak CPU utilization at batch size 14, showing more efficient scaling for larger batch sizes.

Conclusion

The evaluation of three LLM architectures—encoder-only, decoder-only, and encoder-decoder—on Intel CPUs highlights key performance trends and resource utilization patterns. Each architecture demonstrates unique scaling and efficiency characteristics under varying batch sizes and workloads. Selecting the appropriate architecture based on hardware constraints and workload demands is critical to achieving optimal performance.

Key findings are shown in the following table.

Table 1. Key findings

Metric of Interest	Key Insights	Recommendation
TTFT Performance	Encoder-Decoder models exhibit the highest TTFT as batch size increases.	Control batch sizes for encoder-decoder models to improve latency.
Throughput Scaling	Decoder-only models scale the best	Use decoder-only models provide more flexibility for throughput-intensive tasks.
CPU Utilization	Encoder-only models demonstrates better performance across batch sizes and input token length.	Optimize input token length and batch size for CPU-intensive tasks based on business requirements.

Future Work

While this study provides valuable insights into the performance of various LLM architectures on Intel CPUs, additional areas of exploration are needed to further refine the findings and address practical business challenges.

The following areas represent key directions for future work:

1. **Stress Testing on Lenovo ThinkEdge Devices**: Conduct comprehensive stress tests to identify the optimal LLM and working capacity aligned with business needs.
2. **On-Premises vs. Cloud Cost Estimation**: Estimate the cost of deploying LLMs on-premises versus in the cloud, tailored to different use cases and workloads.
3. **Refining Hardware Price Estimates**: Further refine cost estimates by incorporating Intel-accelerated techniques, such as Intel Neural Compressor and IPEX optimizations, to maximize price-to-performance efficiency.

Server configuration

The following table lists the configuration of the test environment.

Table 2. Server configuration

Component	Configuration
Server	ThinkEdge SE450 (300mm)
Processor	Intel Xeon Gold 6338N CPU @ 2.20GHz
Microarchitecture	Ice Lake
Sockets	1
Cores per Socket	32
CPU Cores	32
Turbo	Intel Turbo Boost Technology Enabled
Base Frequency	2.2GHz
Maximum Frequency	3.5GHz
NUMA Nodes	1
Installed Memory	256GB Total; 8x 32GB TruDDR4 3200 MHz (2Rx8 1.2V) RDIMM
Networking	2x Intel E810-XXV Ethernet Controller for SFP 4x Broadcom BCM57454 NetXtreme-E
Disk	2x M.2 5300 480GB SATA 6Gbps Non-Hot Swap SSD 2x 3.5" U.2 P5500 1.92TB Read Intensive NVMe PCIe 4.0 x4 HS SSD
BIOS	CME102Q-1.00
Microcode	0xd0003e7
OS	Ubuntu 22.04 Live Server
Kernel	5.15.0-126-generic

References

See the following documents for more information:

- Liu, B. Comparative analysis of encoder-only, decoder-only, and encoder-decoder language models. College of Liberal Arts & Sciences, University of Illinois Urbana-Champaign, Champaign, IL. https://www.scitepress.org/Papers/2024/128298/128298.pdf
- Accelerating RAG Pipelines for Enterprise LLM Applications using OpenVINO on the Lenovo ThinkSystem SR650 V3 with 5th Gen Intel Xeon Scalable Processors. By Rodrigo Escobar, Abirami Prabhakaran, David Ellison, Ajay Dholakia, Mishali Naik. https://lenovopress.lenovo.com/lp2025-accelerating-rag-pipelines-for-llms-using-openvino-on-sr650-v3

Authors

Kelvin He is a AI Data Scientist at Lenovo. He is a seasoned AI and data science professional specializing in building machine learning frameworks and AI-driven solutions. Kelvin is experienced in leading end-to-end model development, with a focus on turning business challenges into data-driven strategies. He is passionate about AI benchmarks, optimization techniques, and LLM applications, enabling businesses to make informed technology decisions.

David Ellison is the Chief Data Scientist for Lenovo ISG. Through Lenovo's US and European AI Discover Centers, he leads a team that uses cutting-edge AI techniques to deliver solutions for external customers while internally supporting the overall AI strategy for the World Wide Infrastructure Solutions Group. Before joining Lenovo, he ran an international scientific analysis and equipment company and worked as a Data Scientist for the US Postal Service. Previous to that, he received a PhD in Biomedical Engineering from Johns Hopkins University. He has numerous publications in top tier journals including two in the Proceedings of the National Academy of the Sciences.

Eric Page is an AI Engineer at Lenovo. He has 6 years of practical experience developing Machine Learning solutions for various applications ranging from weather-forecasting to pose-estimation. He enjoys solving practical problems using data and AI/ML.

Related product families

Product families related to this document are the following:

- Artificial Intelligence

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP2148, was created or updated on February 14, 2025.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP2148
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP2148.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
ThinkEdge®
ThinkSystem®

The following terms are trademarks of other companies:

Intel®, OpenVINO®, and Xeon® are trademarks of Intel Corporation or its subsidiaries.

Other company, product, or service names may be trademarks or service marks of others.