Standard Retrieval Augmented Generation on Intel: From Search to Answers

Planning / Implementation

Retrieval Augmented Generation (RAG) pairs an LLM with a search/retrieval layer so answers are grounded in external knowledge, reducing hallucinations and keeping responses current for knowledge intensive tasks.

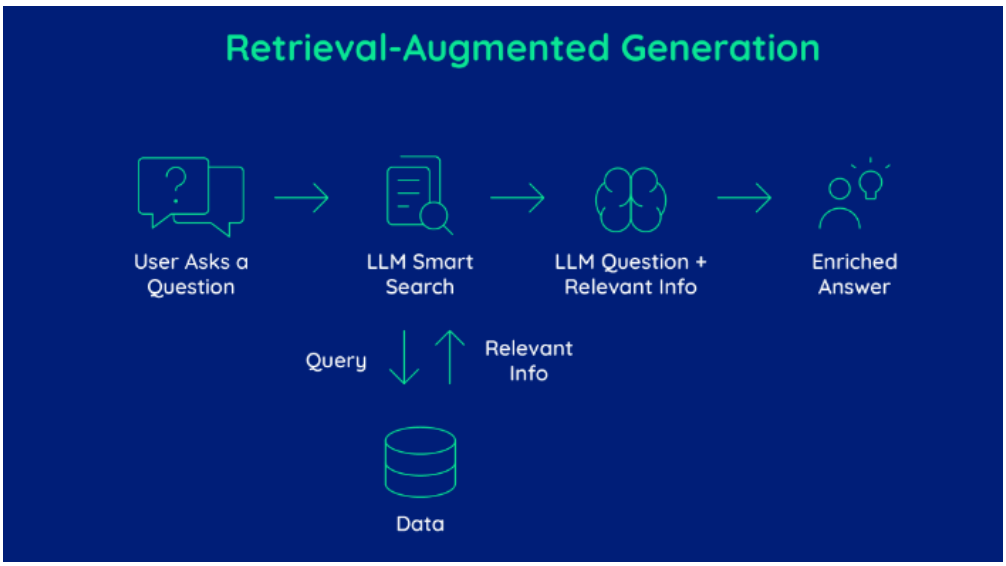

Figure 1. Retrieval-Augmented Generation framework

RAG now underpins common enterprise use cases such as internal search, customer support, and analytics on major platforms (e.g., Azure AI Search, Google Cloud Vertex AI RAG), reflecting broad industry adoption.

With this paper, you'll have a reproducible CPU-first benchmarking baseline for standard RAG, To ensure fairness, we present two comparisons:

- A gen-over-gen Intel comparison, Xeon 6th-Gen vs 5th-Gen
- A cross-vendor baseline, Intel 5th-Gen 8592+ vs AMD EPYC 9554

Evaluation Metrics & Model Selection

This section defines the performance metrics used to assess a standard RAG pipeline (embedding throughput, retrieval latency, TTFT, and per-token generation) and explains why each matter for production. It then lays out our 2×2 model matrix (embedding × generator) and the selection criteria, followed by the dataset, chunking, retrieval-k, and inference settings to ensure fair, reproducible comparisons across platforms.

We implemented the RAG pipeline in Python 3.10, using LangChain for orchestration and Chroma as the

vector database.

The pipeline comprises of three components:

1. Document preprocessing and chunking
2. Dense embedding + ANN search in Chroma
3. Instruction-tuned LLM generation

Measurements (TTFT/TPOT/retrieval) are runtime-agnostic and reproducible with any CPU inference backend.

In this section:

- System performance metrics
- Models
- Dataset

System performance metrics

We focus on metrics that matter for production RAG:

- Embedding Portion:
  - **Embedding Throughput** (documents/sec or samples/sec): higher is better.
  - **Retrieval Latency** (ms): time for content search; lower is better.
- Generation Portion
  - **TTFT (Time to First Token)** (s): interactive latency; lower is better.
  - **TPOT(Time Per Output Token)** (ms): average time to generate each individual output token after first token, also known as inter-token latency; lower is better.

Notes on interpretation: TTFT governs perceived responsiveness; per-token latency dominates long responses; embedding throughput and retrieval latency dominate indexing and high-QPS workloads.

Models

We evaluate a **2×2** matrix combining two embedding encoder**s** with two instruction-tuned LLMs, selected for wide adoption and CPU-friendliness; full model cards are listed in Resources.

- **Embedding Models**
  - Alibaba-NLP/gte-base-en-v1.5 (768-d dense encoder).
  - sentence-transformers/all-MiniLM-L6-v2 (384-d; fast, widely adopted).
- **Generative Models**
  - Mistral-7B-Instruct-v0.2 (7B, strong generalist).
  - Qwen3-4B-Instruct-2507 (compact 4B, latency-friendly).

For more information about these models, see the links in the Resources section.

Dataset

To exercise realistic enterprise lookups, we use a focused documentation corpus with controlled crawl depth and deterministic chunking, enabling repeatable retrieval and generation profiles.

Hugging Face documentation: Five well-known areas of Hugging Face usage. A key parameter here is max_depth, which controls how deeply we crawl from the starting page

```
# Define the documentation sections to scrape, with varying crawl depths.
urls_to_load = [
    {"url": "https://huggingface.co/docs/transformers/index", "max_depth": 2},
    {"url": "https://huggingface.co/docs/datasets/index", "max_depth": 2},
    {"url": "https://huggingface.co/docs/tokenizers/index", "max_depth": 1},
    {"url": "https://huggingface.co/docs/peft/index", "max_depth": 1},
    {"url": "https://huggingface.co/docs/accelerate/index", "max_depth": 1}
]
```

Performance Results & Analysis

In this section, we present side-by-side results for the standard RAG pipeline: embedding throughput, retrieval latency, time-to-first-token, and per-token generation. Results are broken down by workload and platform, emphasizing gen-over-gen uplift on 6th Gen vs 5th Gen Intel Xeon, and the comparison of Xeon 5th Gen to AMD EPYC with practical takeaways for SLAs and capacity planning.

- Embedding Throughput
- Embedding Retrieval
- Generation Latency – TPOT
- Generation Latency – TTFT

Embedding Throughput

The figures below show the embedding throughput across models and CPUs. Higher values indicate faster embedding speed.

Intel Xeon 8592+ VS AMD 9554

Figure 2 shows embedding throughput (samples/sec) for two embedding models on AMD EPYC 9554, Intel Xeon 8592+, and Xeon 8592+ with IPEX..

For **gte-base-en-v1.5**, Xeon 8592+ already beats EPYC 9554 (23 vs 8 samples/s), and Xeon 8592+ with IPEX jumps to 38 samples/s, roughly **4.7×** AMD and **1.6×** over vanilla Xeon. For **MiniLM-L6-v2**, EPYC 9554 is slightly ahead of stock Xeon (58 vs 50 samples/s), but Xeon 8592+ with IPEX reaches 86 samples/s, about **1.5×** AMD and **1.7×** standard Xeon—showing that IPEX consistently lifts Xeon's embedding throughput.

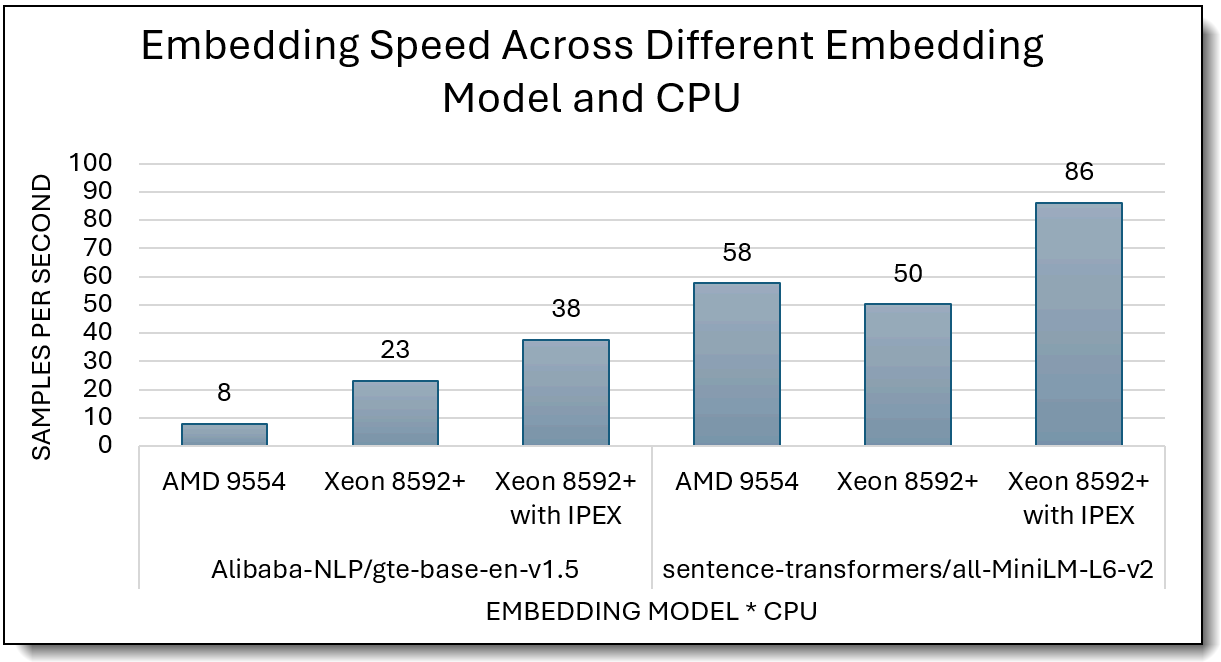

Figure 2. Embedding Throughput (higher is better)

Intel Xeon 6787P VS Intel Xeon 8592+

Figure 3 compares embedding throughput (samples/sec) for Xeon 8592+ with IPEX vs Xeon 6787P with IPEX across two embedding models.

On both embedding models, crucially the 6th-Gen Xeon delivers a measurable uplift versus the 5th-Gen Xeon, generation over generation, adding about **23%** more throughput on GTE (47 vs 38) and about **54%** on MiniLM (133 vs 86), confirming a consistent uplift across encoders.

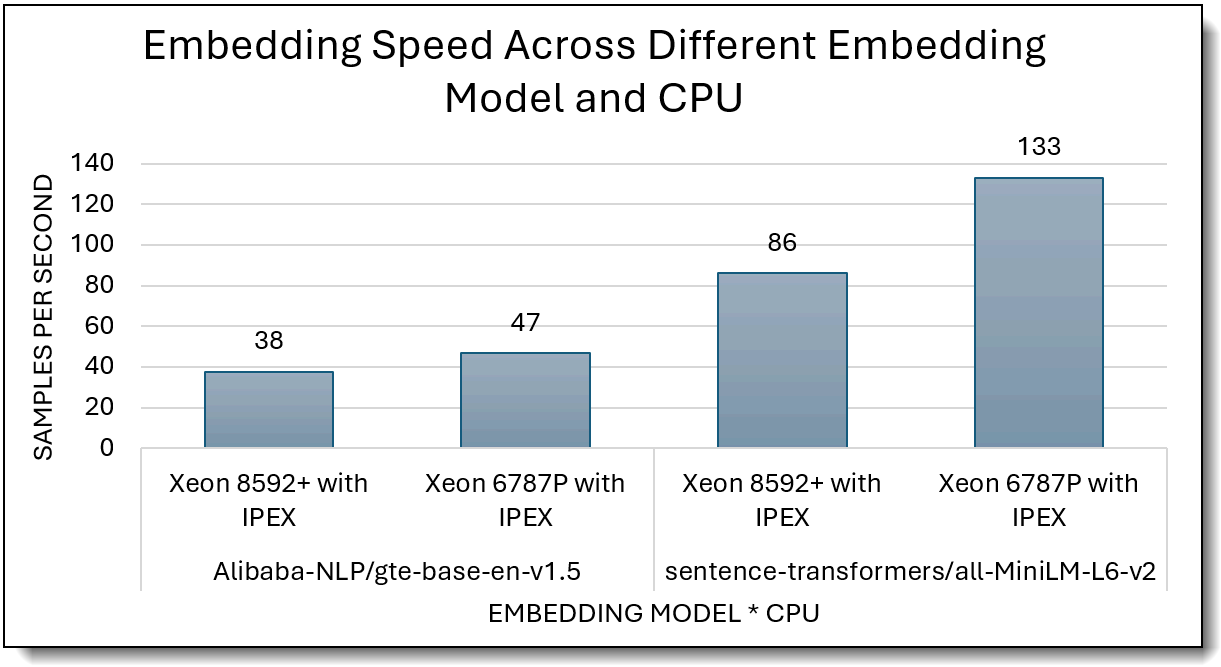

Figure 3. Embedding Throughput (higher is better)

Embedding Retrieval

The figures below show the speed of documents retrieval across models and CPUs. Lower values indicate faster retrieval.

Intel Xeon 8592+ VS AMD 9554

Figure 4 shows **retrieval latency (ms, lower is better)** for AMD 9554 vs Intel Xeon 8592+ with and without IPEX, across two embedding models.

For **gte-base-en-v1.5**, Xeon 8592+ cuts latency from 52.9 ms on AMD to 19.9 ms, and Xeon 8592+ with IPEX further reduces it to 15.2 ms. That's about a **3.5×** improvement over AMD. For **MiniLM-L6-v2**, AMD sits at 20.6 ms, stock Xeon is slightly slower at 25.5 ms, but Xeon 8592+ with IPEX again becomes the fastest at **14.2** ms, clearly outperforming both.

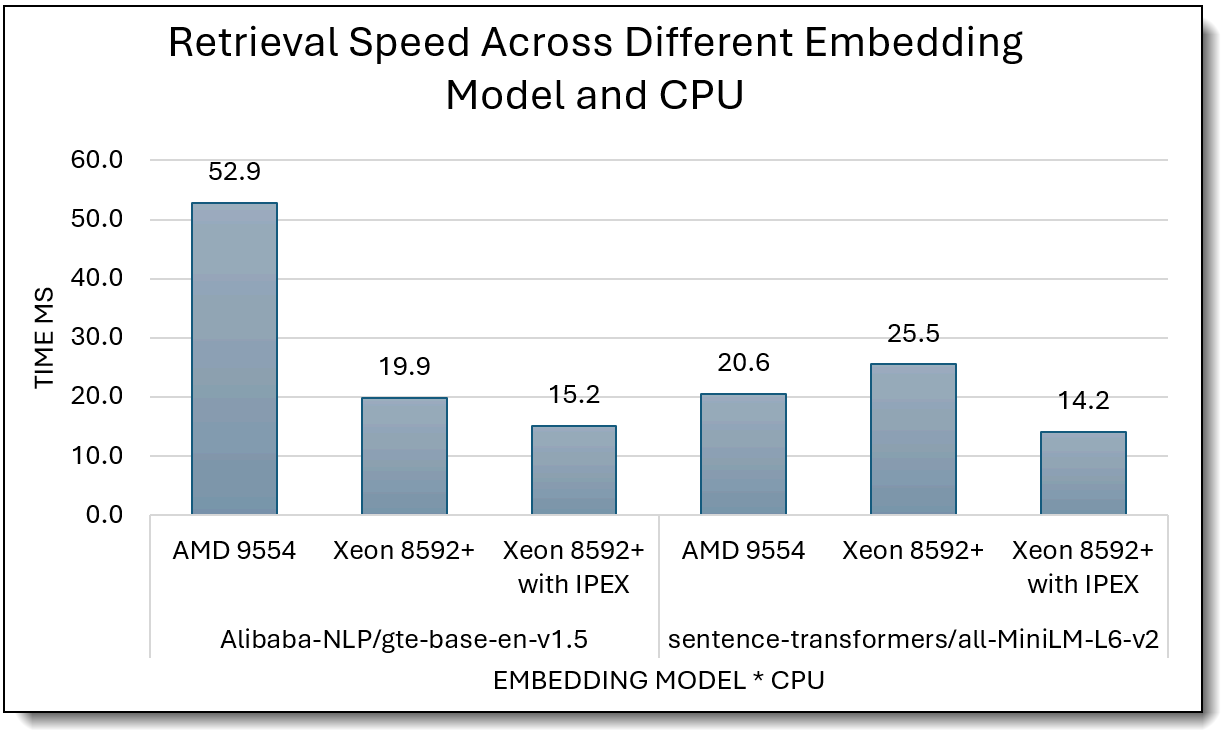

Figure 4. Retrieval Latency (lower is better)

Intel Xeon 6787P VS Intel Xeon 8592+

Figure 5 compares retrieval latency (ms, lower is better) for Xeon 8592+ with IPEX vs Xeon 6787P with IPEX across two embedding models.

For **gte-base-en-v1.5**, retrieval drops slightly from 15.2 ms on 8592+ to 14.8 ms on 6787P about 3% improvement. For MiniLM-L6-v2, latency improves from 14.2 ms to 13.8 ms. Overall, Xeon 6787P with IPEX is consistently but modestly faster than Xeon 8592+ with IPEX for retrieval in both models.

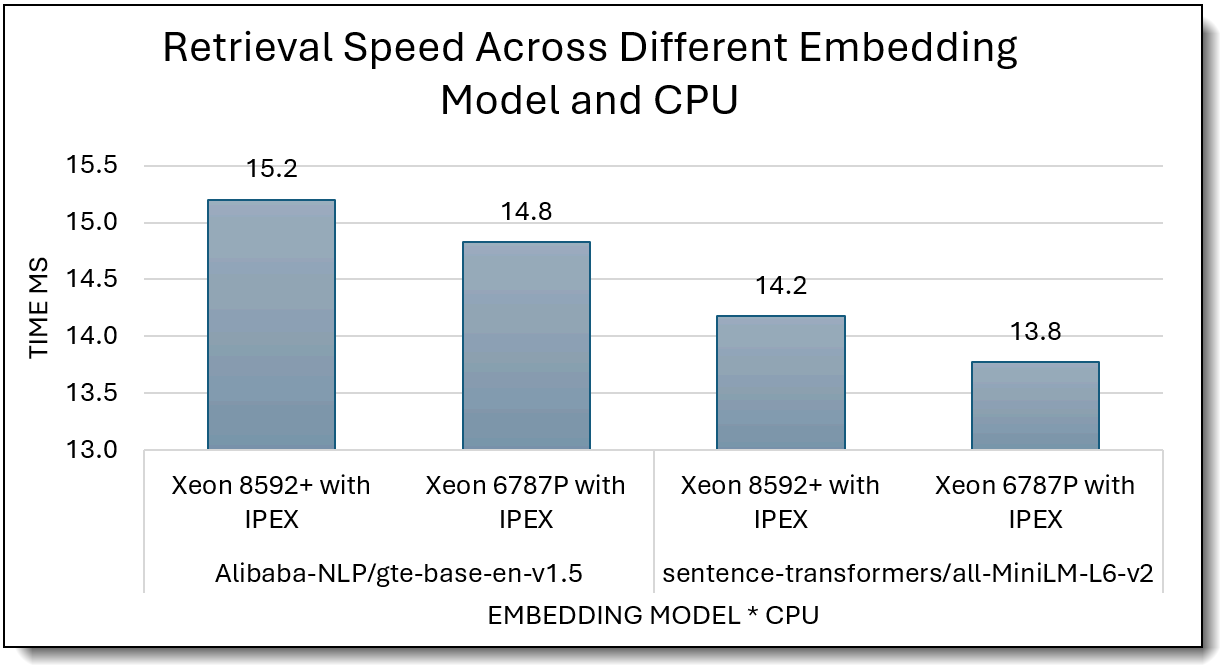

Figure 5. Retrieval Latency (lower is better)

Generation Latency – TPOT

The figures below show Time per Output Token (TPOT) across generative model and CPUs. Lower values indicate faster streaming once the first token appears.

Intel Xeon 8592+ VS AMD 9554

Figure 6 reports **TPOT (time per output token, ms; lower is better)** for Mistral-7B and Qwen3-4B across AMD 9554, Xeon 8592+, and Xeon 8592+ with IPEX.

For **Mistral-7B**, AMD (313 ms) is better than stock Xeon (398 ms), but Xeon 8592+ with IPEX again leads with 104 ms, about **3×** faster than AMD. For **Qwen3-4B**, AMD and Xeon are similar (283 ms vs 317 ms), while Xeon 8592+ with IPEX drops TPOT to 86 ms, roughly **3.3×** faster than AMD, which makes the Xeon+IPEX setup clearly best for streaming speed.

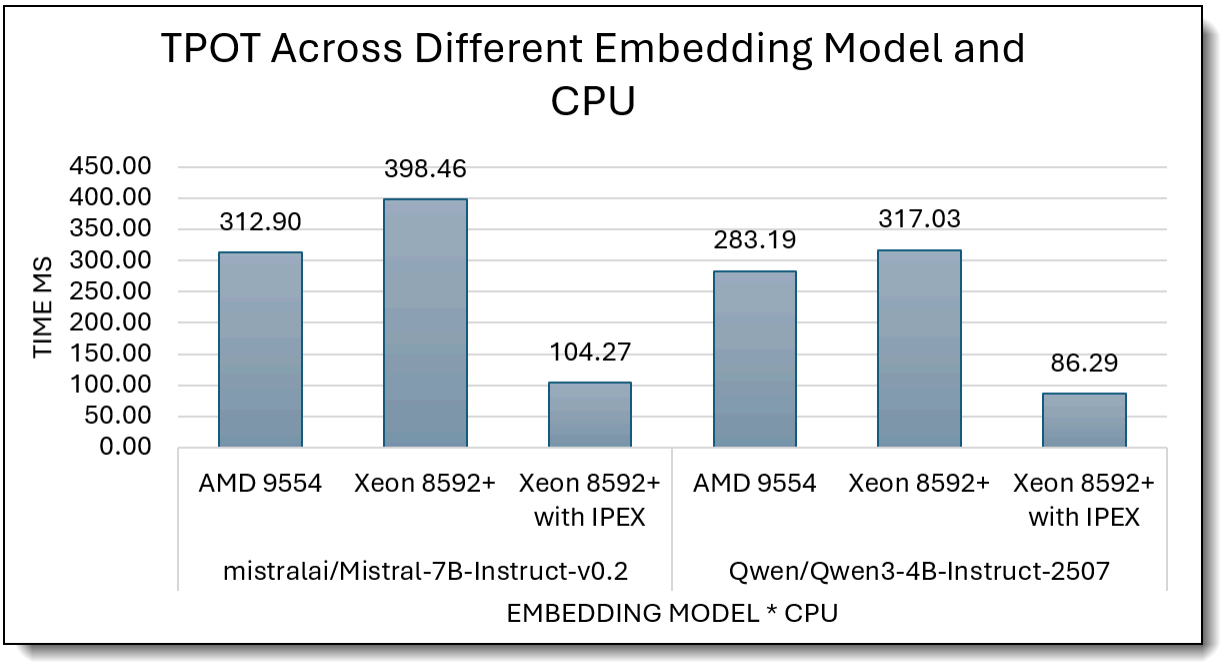

Figure 6. Time per Output Token (Lower is better)

Intel Xeon 6787P VS Intel Xeon 8592+

Figure 7 reports TPOT (time per output token, ms; lower is better) for Xeon 8592+ with IPEX and Xeon 6787P with IPEX on two LLMs.

For **Mistral-7B-Instruct-v0.2**, TPOT drops from 104.27 ms on 8592+ to 96.78 ms on 6787P; for **Qwen3-4B-Instruct-2507**, it improves from 86.29 ms to 81.71 ms. Overall, Xeon 6787P with IPEX consistently streams tokens a bit faster (roughly 5–7% lower TPOT) than Xeon 8592+ with IPEX for both models.

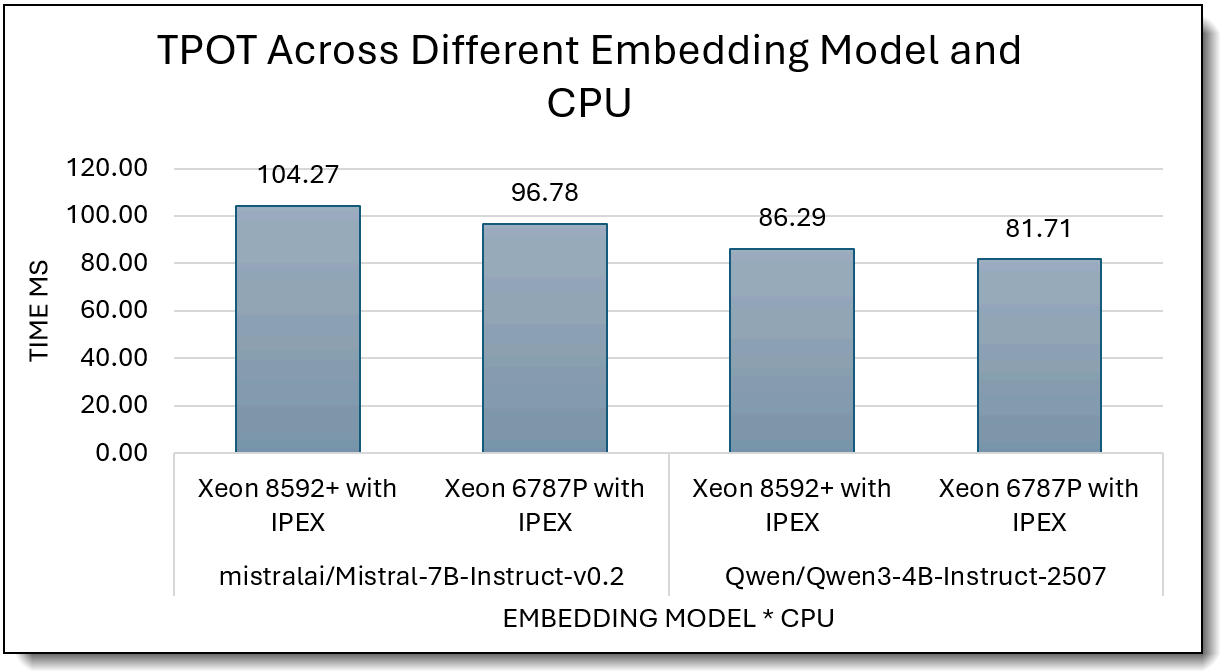

Figure 7. Time per Output Token (Lower is better)

Generation Latency – TTFT

The figures below show Time to First Token (TTFT) across generative models and CPUs. Lower values improve perceived responsiveness, aligning with the near 1-second interaction threshold for fluid UX.

Intel Xeon 8592+ VS AMD 9554

Figure 8 shows TTFT (time to first token, seconds; lower is better) for Mistral-7B and Qwen3-4B across AMD 9554, Xeon 8592+, and Xeon 8592+ with IPEX.

For **Mistral-7B**, AMD and stock Xeon are similar (3.58 s vs 4.14 s), but Xeon 8592+ with IPEX drops TTFT to 2.54 s, drops **30%** in TPOT, clearly the fastest. For Qwen3-4B, AMD is slightly better than stock Xeon (2.95 s vs 3.26 s), while Xeon 8592+ with IPEX again leads with a much lower TTFT of 1.46 s, roughly 50% drop.

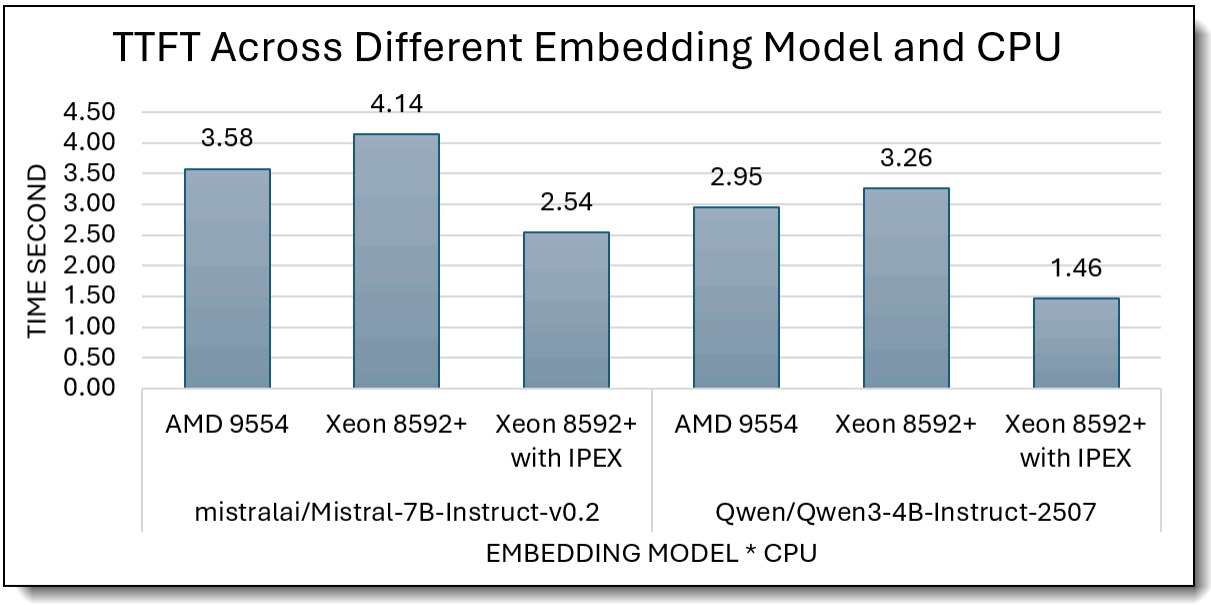

Figure 8. Time to First Token (Lower is better)

Intel Xeon 6787P VS Intel Xeon 8592+

Figure 9 compares TTFT (time to first token, seconds; lower is better) for Xeon 8592+ with IPEX vs Xeon 6787P with IPEX on two LLMs.

For **Mistral-7B-Instruct-v0.2**, TTFT drops from 2.54 s on Xeon 8592+ to 1.25 s on Xeon 6787P, about a **2×** improvement. For **Qwen3-4B-Instruct-2507**, TTFT improves from 1.46 s to 0.95 s, roughly a **35%** reduction. Overall, Xeon 6787P with IPEX consistently delivers much faster first-token times and translating to earlier response onset and snappier UX in interactive RAG.

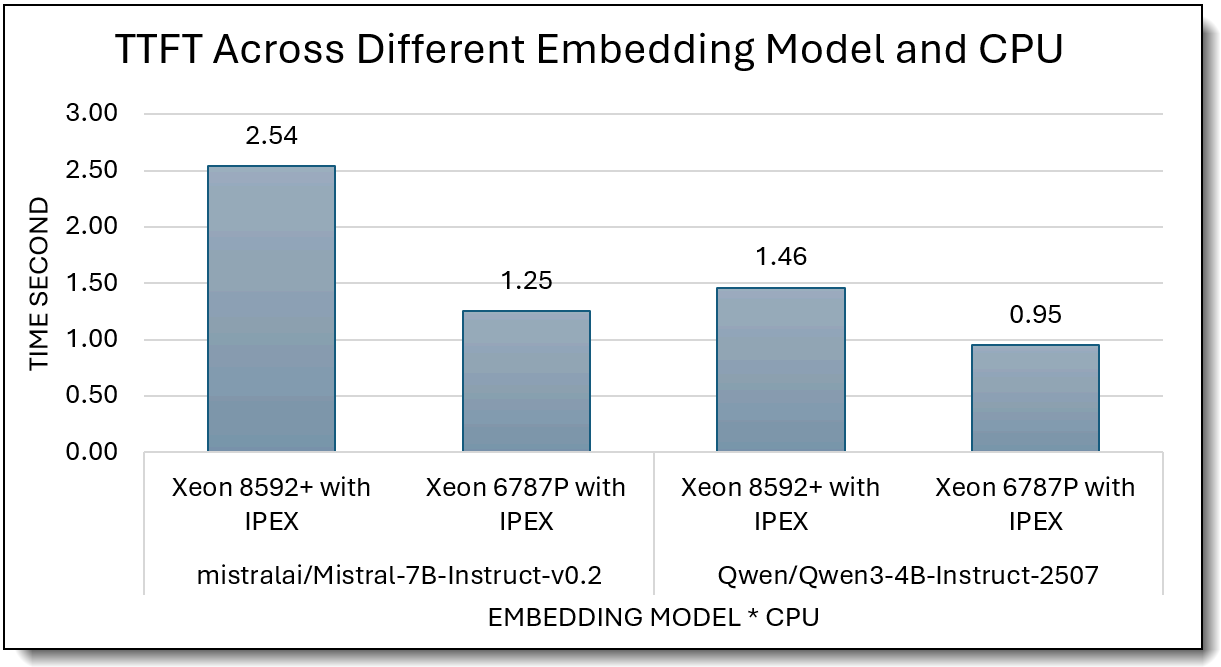

Figure 9. Time to First Token (Lower is better)

Conclusion and Business Impact

The results described in the previous section show that Intel Xeon (6787P) delivers meaningful, measurable gains over its predecessor and EPYC 9554 for the *entire* RAG pipeline:

- **05-1.5×** improvement in embedding throughput on common encoders.
- **Up to 3×** faster retrieval.
- **Up to 3x** faster per-token generation and **1.3 - 2×** better TTFT.

These gains compound across solid RAG services to:

Interactive Chat / Support (streaming UI)

Table 1 converts TTFT/TPOT + retrieval into end-to-end totals time for generating 100- and 200-token replies, useful for SLA mapping in chat/support.

Table 1. Interactive chat totals (lower is better): retrieval + TTFT + tokens×TPOT.

Embedding × LLM	TTFT (s)	Retrieval (ms)	Total @100 tok (s)	Total @200 tok (s)
GTE × Qwen-4B	0.97	1.89	9.744	18.516
MiniLM × Qwen-4B	0.97	1.32	9.743	18.515
GTE × Mistral-7B	1.25	1.89	9.885	18.518
MiniLM × Mistral-7B	1.25	1.32	9.884	18.517

Doc QA / Analyst Copilots (heavier answers)

Table 2 shows totals time for generating 300/400 tokens, matching analyst/copilot usage where answers are longer.

Table 2. Doc-QA totals (lower is better): retrieval + TTFT + tokens×TPOT

Embedding × LLM	TTFT (s)	Retrieval (ms)	Total @300 tok (s)	Total @400 tok (s)
GTE × Qwen-4B	0.97	1.89	27.288	36.066
MiniLM × Qwen-4B	0.97	1.32	27.287	36.064
GTE × Mistral-7B	1.25	1.89	27.151	35.784
MiniLM × Mistral-7B	1.25	1.32	27.150	35.783

Based on our measurements, Intel Xeon 6th-Gen (6787P) shows consistent generation-over-generation gains over 5th-Gen (8592+). These results indicate that CPU-first RAG is viable on Xeon 6 for the tested model sizes and workloads.

- TTFT (time to first token):
  - On Xeon 6787P, Qwen-4B reaches 0.97 s and Mistral-7B 1.25 s.
  - Per Nielsen Norman Group's interaction thresholds, within 1 second keeps users "in flow" and within 10 seconds risks loss of attention. Your Qwen-4B TTFT meets the 1 s ideal, and Mistral-7B is just over but still well within an acceptable range for an interactive chat UX.
- TPOT (per-token generation speed):
  - On 6787P you measured 86–88 ms/token, i.e.,11–12 tokens/second. For English, a token is roughly ¾ of a word (4 characters), so 11 tokens/s corresponds to about 8–12 words/s in practice. That is above typical on screen reading speeds (200–300 wpm = 3–5 words/s), so the stream should feel comfortably fast to users.

Based on our end-to-end benchmarks, 6th Gen Intel Xeon 6787P offers a clear gen-over-gen uplift versus

the 5th Gen Intel Xeon CPU across the RAG pipeline; it delivers acceptable TTFT (0.97–1.25 s) and streaming speeds (86–88 ms/token ≈ 11 tokens/s) that keep interactive chat and doc-QA responsive. Combined with its strong retrieval/embedding performance, Xeon 6 enables CPU first RAG that meets mainstream UX SLOs without accelerators for the tested model sizes.

Additional papers on RAG

This paper is Part 1 of 4 of a series of papers on Retrieval Augmented Generation.

The papers in the series are as follows:

- **Part 1**: This paper
- **Part 2**: Multimodal RAG: extend the standard pipeline to support cross-modal retrieval and grounding (e.g., text with images/tables/code), compare hybrid indexes and multimodal encoders/LLMs, and quantify modality-aware latency and throughput alongside retrieval quality. Read the paper here: Accelerating Multimodal RAG with Intel Xeon and OpenVINO: When Vision Meets Language
- **Part 3**: Multimodal RAG Architecture Trade-offs: systematically compare three multimodal RAG designs, text-mediated, hybrid, and multimodal embedding–based, analyzing indexing cost vs. query-time latency, retrieval accuracy vs. grounding fidelity, and scalability under SLA constraints to establish practical design guidelines for enterprise deployment. Read the paper here: End-to-End Latency Benchmarking for Three Multimodal RAG Architectures
- **Part 4**: Agentic Long-Context RAG: introduce planning and tool use with validation loops (self-check, re-query, re-rank), add long-context memory, and evaluate end-to-end task success, stability across multi-step trajectories, and cost–latency trade-offs. Coming soon.

System Configurations

The following table lists the configuations of the systems under test.

Table 3. System Configurations

Platform	Lenovo ThinkSystem SR650 V4	Lenovo ThinkSystem SR650 V3	Lenovo ThinkSystem SR675 V3
CPU	Intel Xeon 6787P processor, 86 cores / 172 threads @ 3.8 GHz	Intel Xeon 8592+ processor, 64 cores / 128 threads @ 3.9 GHz	AMD EPYC 9554 processor 64 core / 128 threads @ 3.75 GHz
Memory	16x 64 GB DDR5 @6400GH	16x 32 GB DDR5 @6400GH	24x 64Gb DDR5 @6400GH
OS	Ubuntu 22.04.5 LTS (Linux kernel 6.8.0-59-generic)	Ubuntu 22.04.5 LTS (Linux kernel 6.8.0-59-generic)	Ubuntu 22.04.5 LTS (Linux 5.15.0-153-generic)

Resources

For more information, see these resources:

- Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP.” arXiv:2005.11401 https://arxiv.org/abs/2005.11401
- Intel Extension for PyTorch: https://intel.github.io/intel-extension-for-pytorch/
- Intel Advanced Matrix Extensions (Intel AMX): https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/what-is-intel-amx.html
- Intel Xeon 6787P (Intel Xeon 6 / Granite Rapids) product page: https://www.intel.com/content/www/us/en/products/sku/241844/intel-xeon-6787p-processor-336m-cache-2-00-ghz/specifications.html
- UX response-time thresholds (≈0.1 s instant / ≈1 s flow / ≈10 s attention): Nielsen Norman Group https://www.nngroup.com/articles/response-times-3-important-limits/
- Average adult silent reading speed (200–300 wpm): https://en.wikipedia.org/wiki/Words_per_minute

Model cards:

- sentence-transformers/all-MiniLM-L6-v2 https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
- Alibaba-NLP/gte-base-en-v1.5 https://huggingface.co/Alibaba-NLP/gte-base-en-v1.5
- Mistral-7B-Instruct-v0.2 https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
- Qwen3-4B-Instruct-2507 https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507

Author

Kelvin He is an 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.

Related product families

Product families related to this document are the following:

- Artificial Intelligence
- Processors

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.

© Copyright Lenovo 2026. All rights reserved.

This document, LP2322, was created or updated on November 6, 2025.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP2322
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP2322.

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®
ThinkSystem®

The following terms are trademarks of other companies:

AMD®, AMD EPYC®, and EPYC® are trademarks of Advanced Micro Devices, Inc.

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

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Azure® is a trademark of Microsoft Corporation in the United States, other countries, or both.

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