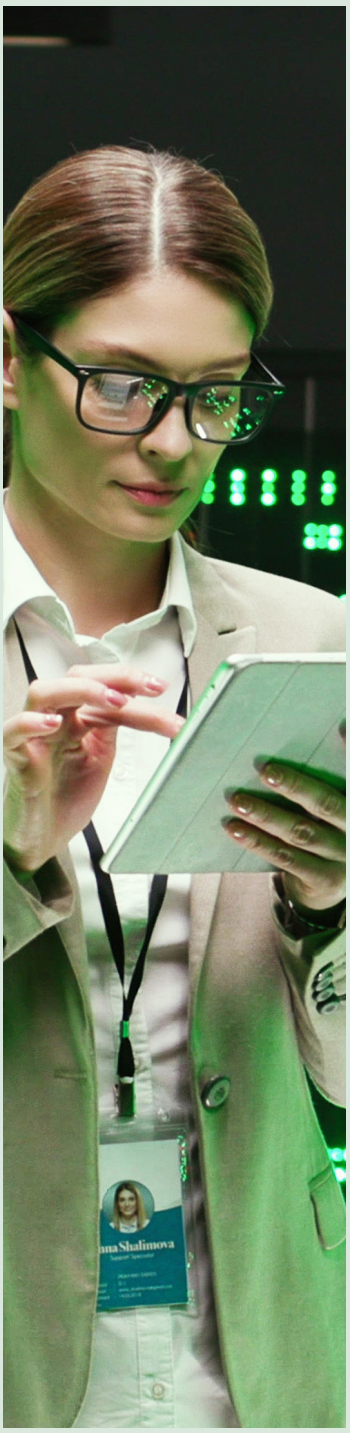
WHITE PAPER



PROCESSOR BENCHMARK REPORT FOR HOSTING OPENSOURCE LLMS ON INTEL CPU INFRASTRUCTURE

intel.

Infosys
Navigate your next

Contents

Benchmark Report for Hosting OpenSource LLMs

1. Problem Statement 3
2. Solution Details 4
3. High-Level Architecture 5
4. Key Performance Highlights 6
5. Conclusion & Recommendation 7

Executive Summary

Infosys cloud and infrastructure operations workloads increasingly rely on **Large Language Models (LLMs)** to automate IT operations and deliver operational intelligence. GPU-first deployments can be cost-intensive and capacity-constrained. This evaluation assesses **Intel Xeon 6 processors with Intel AMX (Advanced Matrix Extensions)** as a **CPU-native GenAI inference platform** capable of serving open-source LLMs at predictable performance and improved cost efficiency for enterprise AIOps use cases.

Problem Statement

Infosys manages the IT landscape of one of largest Automobile manufacturers in the world,. As part of this project, Infosys manages the customer data center across the globe, with Software Defined Infrastructure architecture and zero touch IT Ops enabling the next generation services for the end users.

As AIOps adoption expands, multiple use cases depend on LLM inference, for example:

Day-0 and Day-2 operations including health checks, incident summarization, problem management, root cause analysis, service request fulfilment, automated change management and assistance, knowledge retrieval & automation copilots

The current GPU-centric approach faces recurring challenges:

Escalating CapEx/OpEx Cost for GPU procurement and operations

Limited availability of large-scale GPU resources

Goal: Identify a CPUas an alternative that can host open-source LLMs with predictable performance for enterprise AIOps workloads.

Evaluation Objectives

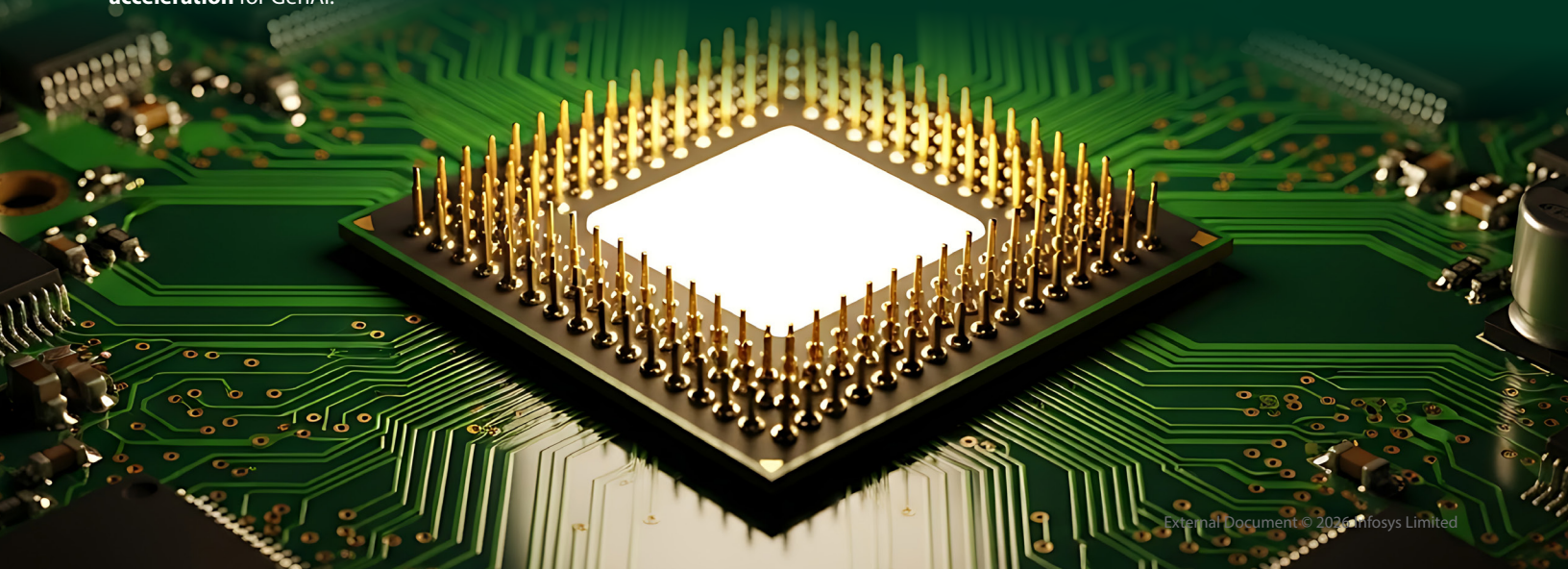
The assessment focused on whether Intel Xeon Processors can support GenAI inference with:

Predictable performance at lower cost for non–latency-critical workloads

Handling the LLM (current case QWEN 14B) with stable throughput under moderate concurrency

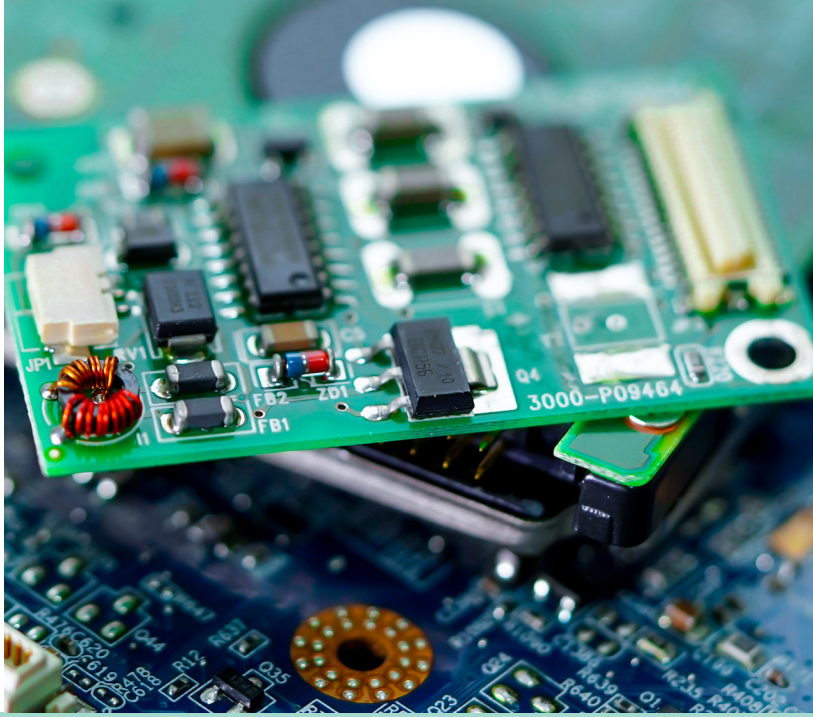
Simplified software stack simplicity by leveraging Intel enterprise optimized software stack, time to adapt and time to market cycle should be easy and effective while keeping orchestration to compute efficiency

Note: Container orchestration may be used for deployment consistency, but the primary focus of this document is **Xeon + AMX compute acceleration** for GenAI.



Solution Details and Architecture

Intel Xeon 6 processors are designed for scalable, general-purpose data center workloads and can be leveraged for GenAI inference where the dominant constraints often include memory bandwidth, cache behavior, and token-by-token generation patterns.



Key Solution artefacts

2.1 Intel AMX (Advanced Matrix Extensions)

Intel AMX is an instruction set extension that accelerates matrix operations commonly found in deep learning workloads. It enables CPUs to execute key tensor computations more efficiently, improving performance for inference tasks without introducing discrete accelerators.

AMX in the Context of LLM Inference

LLM inference heavily relies on repeated linear algebra operations across transformer layers (e.g., projections and feed-forward network computations). AMX helps by accelerating the dense matrix multiplications that dominate these compute paths, particularly when using reduced precision formats.

2.2 Precision Modes That Matter (BF16 / INT8)

In production inference, many deployments use reduced precision (e.g., **BF16 or INT8)** to improve throughput and reduce memory footprint. AMX acceleration is most impactful when the inference stack effectively exploits these precisions through optimized kernels and libraries.

2.3 Software Stack and Serving Approach (Compute-Centric)

In this evaluation, vLLM is used as a high-throughput inference and serving engine built specifically to improve utilization and memory efficiency during LLM serving. The capabilities most relevant to this evaluation are:

- **Paged Attention for KV-cache efficiency:** vLLM manages attention key/value (KV) cache more efficiently than naïve contiguous allocation, reducing fragmentation and improving concurrency under load.
- **Continuous batching (in-flight batching):** Instead of fixed batches that wait for the longest request, vLLM continuously admits new requests as others complete, improving utilization and stabilizing latency/throughput under mixed prompt and output lengths.
- **Streaming outputs:** Supports token streaming to improve perceived latency and user experience for interactive AIOps copilots.
- **OpenAI-compatible server mode:** Enables integration with existing clients and gateways without rewriting application logic.
- **Distributed inference support (tensor/pipeline parallelism):** vLLM provides tensor parallelism and pipeline parallelism options for distributing model execution across devices/workers.
- **Broad hardware support including Intel CPUs:** vLLM documentation explicitly lists Intel CPUs as supported serving targets.

Intel Xeon 6 Performance Optimization: NUMA Awareness, Core Pinning, and Tensor Parallelism

To achieve best performance on multi-socket Xeon 6 servers, **deep CPU-level tuning** was applied in addition to model/runtime configuration. The objective was to maximize **local memory access**, reduce cross-socket traffic, and ensure CPU threads stay close to their data.

NUMA-aligned "tensor parallelism" for best throughput (multi-worker partitioning)

- For larger models and higher concurrency, the serving configuration can use **vLLM distributed inference** (tensor parallelism) to partition execution across multiple workers.

Outcome of these optimizations

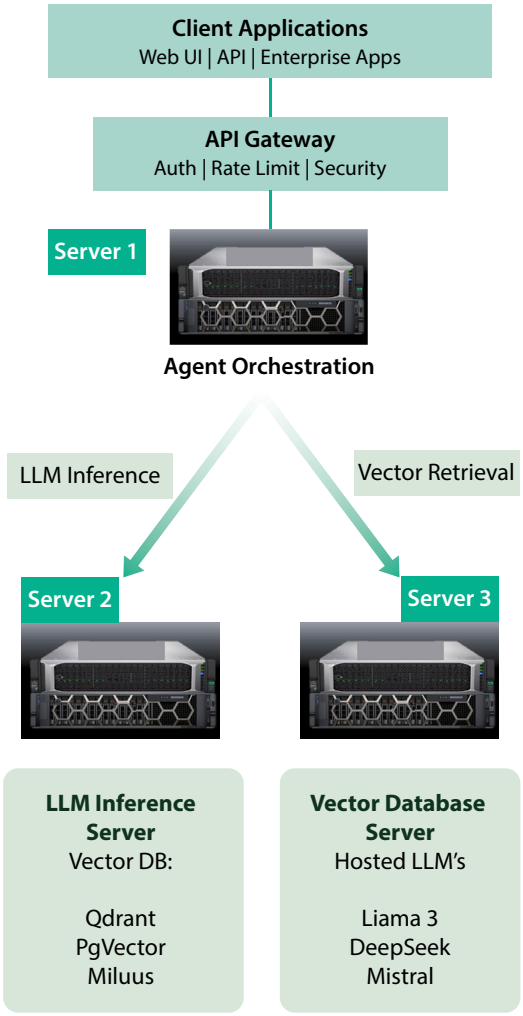
By combining vLLM's scheduling/KV-cache efficiencies with Xeon-specific NUMA and affinity tuning, the deployment achieves more predictable TTFT and stable token streaming under moderate concurrency, improving the performance-per-cost profile on Intel Xeon 6.

2.4 Infra Details

Platform: Intel Xeon 6 Processors	**Accelerator:** Intel AMX (Advanced Matrix Extensions) is a specialized instruction set architecture extension designed to accelerate AI and machine learning workloads	**Architecture:** Kubernetes-based CPU clusters with autoscaling
Inference Runtime: vLLM with token streaming	**Model Scale Tested:** Up to **14B parameters**	**Use Cases:** Moderate concurrency pipelines, Test and Dev Environment, batch processing, etc.

High-Level Architecture

Schematic Diagram



Below is a clean enterprise schematic architecture for an Agentic AI platform using three Intel Xeon 6 servers, where each server is optimized for a specific role:

Server 1 → Agent orchestration + tools
Server 2 → Vector database / knowledge
Server 3 → LLM inference

This separation is considered a best practice in production systems because agentic architectures typically separate model reasoning, memory retrieval, and orchestration logic into independent scalable components.

How Each Xeon 6 Server Is Optimized

Agent Server (Orchestration)

Role
- Decision making
- Tool selection
- Workflow orchestration

Typical stack
- LangGraph / CrewAI / AutoGen
- Python microservices
- Redis (session memory)
- API gateway

CPU usage pattern
- high concurrency
- lightweight inference
- many API calls

Why Xeon works well
- high core count
- networking
- microservice workloads

LLM Inference Server

Role
- Run the actual large language models

Typical stack
- vLLM
- Intel IPEX/OpenVINO Optimization

Typical models
- Llama-3
- DeepSeek
- Qwen
- Code LLM

Xeon features useful
- AMX (matrix acceleration)
- Tensor Parallelism
- Numa Pinning
- AVX 512

Vector Database Server

Role
- semantic search
- long-term agent memory
- Vector databases store embeddings and allow semantic retrieval which enables RAG and persistent memory for agents.

Typical stack
- Milvus
- Qdrant
- PgVector
- FIASS
- Redis

Components
- embedding generation
- indexing
- similarity search

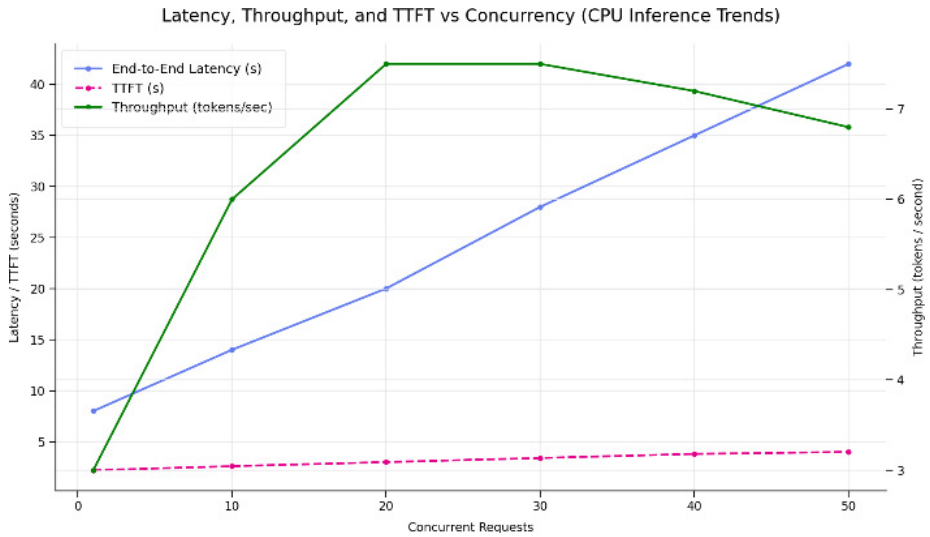
The AI pipeline can leverage a simple cloud-native architecture designed to efficiently serve LLMs at scale. Traffic enters the system through an API Gateway or Ingress controller, which intelligently routes inference requests into the underlying Kubernetes cluster infrastructure.

The Kubernetes layer forms the backbone of the deployment, featuring strategically configured node pools powered by Intel Xeon 6 processors that provide the computational power necessary for demanding AI workloads.

To ensure optimal resource utilization and cost efficiency, the cluster implements autoscaling capabilities that dynamically adjust capacity based on real-time demand patterns.

Each large language model is deployed as dedicated serving pods within this orchestrated environment, allowing for isolated and scalable model management. The inference runtime utilizes vLLM, a high-performance serving framework optimized for LLM workloads, with token streaming capabilities enabled to provide real-time response generation and improved user experience through progressive content delivery rather than waiting for complete response generation.

Key Performance Highlights



Average Metrics Observed

- Time to First Token (TTFT): ~3.4 seconds
- End-to-End Latency: ~22 seconds (dependent on token count)
- Throughput: ~7.5 tokens/ second
- Streaming Stability: Consistent inter-token latency (~0.10s)

Insights

- TTFT consistently remained within 2-4 seconds
- Latency scaled with output size (8s–42s)
- CPU inference demonstrated stable and predictable behavior across workloads

Concurrency & Scalability

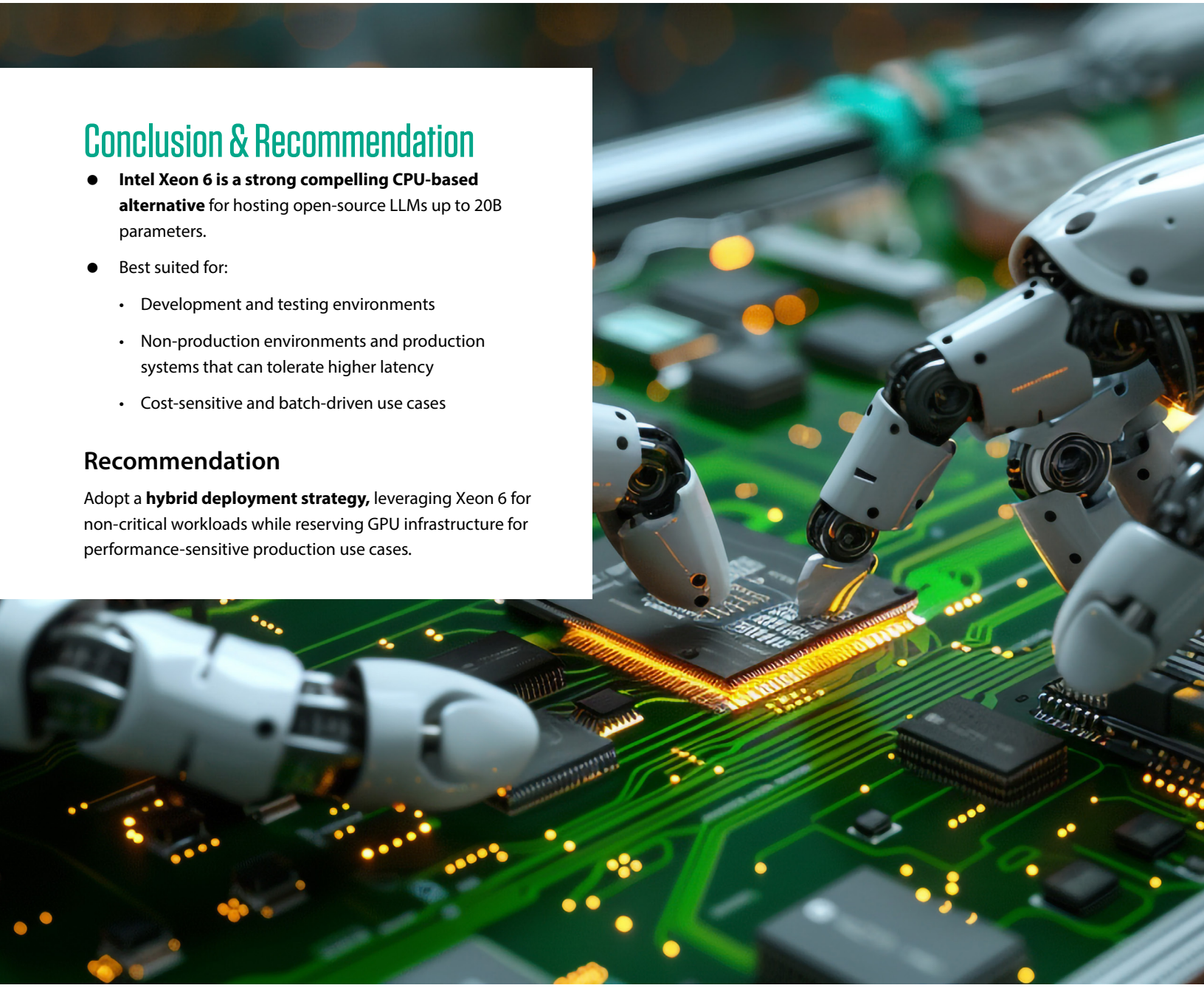
- Concurrency tested up to 50 parallel requests
- Success rate: 100% across all levels
- Latency increased linearly with load, as expected in CPU-only inference
- Throughput saturation observed beyond ~30 concurrent requests

Implication

Xeon 6 provides **reliable parallel processing** with predictable degradation, suitable for controlled environments rather than latency-critical production systems.

Comparative Assessment

Aspect	Intel Xeon 6	GPUs
Performance	Adheres to required SLAs	Industry-leading
Cost Efficiency	High	Low (expensive)
Availability	High	Constrained
Best Fit	Dev/Test, Batch, Production that can tolerate higher latency.	Latency-critical production



Conclusion & Recommendation

- **Intel Xeon 6 is a strong compelling CPU-based alternative** for hosting open-source LLMs up to 20B parameters.
- Best suited for:
  - Development and testing environments
  - Non-production environments and production systems that can tolerate higher latency
  - Cost-sensitive and batch-driven use cases

Recommendation

Adopt a **hybrid deployment strategy,** leveraging Xeon 6 for non-critical workloads while reserving GPU infrastructure for performance-sensitive production use cases.

Infosys & Intel: A Strategic Partnership Powering Enterprise.

Infosys and Intel share a strategic partnership focused on co innovation, enterprise infrastructure modernization, and scalable AI adoption. By combining Infosys' strengths in AI led IT operations (AIOps), platform engineering, and global enterprise delivery with Intel's leadership in CPU architecture, silicon innovation, and optimized software stacks, the partnership enables enterprises to deploy practical and cost effective GenAI solutions on industry standard CPU infrastructure.

Through joint innovation programs, proofs of concept, and go to market initiatives, Infosys and Intel help customers confidently adopt GenAI for real world AIOps workloads balancing performance, cost efficiency, sustainability, and enterprise governance—while accelerating the journey toward more autonomous, resilient, and intelligent IT operations.

Authors

Prabhas Harlapur
Solutions Architect, Intel

Prasanth M S
Global Alliance Manager, Intel

Omkar Arvind Darvekar
Principal Technology Architect, Infosys

Abhinandan Chakraborty
Principal Consultant - Technical Alliance, Infosys

For more information, contact askus@infosys.com

Infosys®
Navigate your next