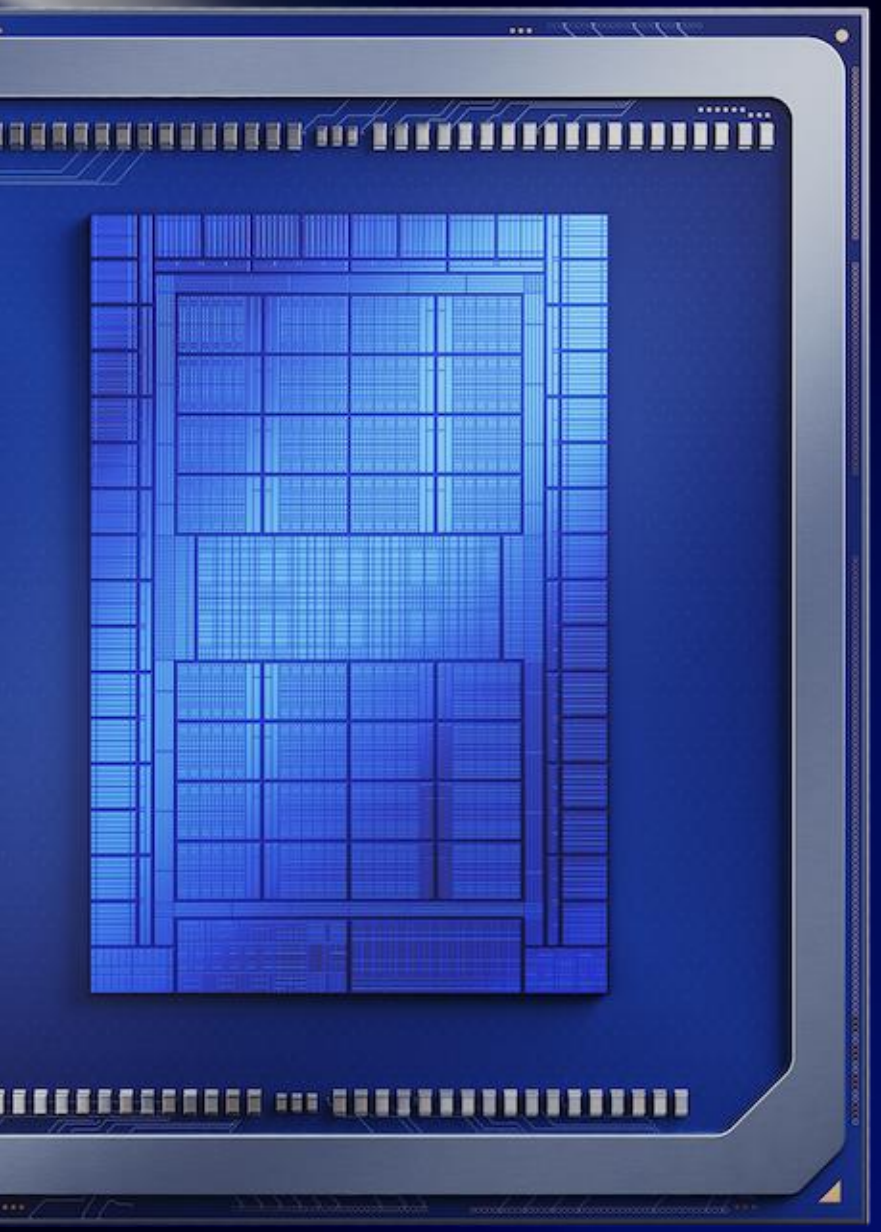


Crescent Island: GPU Designed for Agentic AI Inference

Sumit Mohan, Chief Architect, Enterprise AI Systems
Dr. Hong Jiang, Intel Fellow, Chief GPU Compute Architect

Hot Chips – Aug 2026

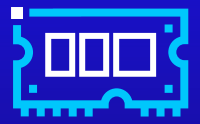
Agentic AI Workload Compute Demands

Continuous Iteration

More Steps Drive
Latency Pressure

- Single-turn inference time
- Tool call turnaround time
- Scheduling

Expanding Context



More Context Drives
Memory Pressure

- Memory capacity
- Long context handling
- Memory Bandwidth

Concurrent Orchestration

More Agents Drive
System Throughput Pressure

- CPU-GPU coordination
- Heterogenous orchestration
- Power density management

Accelerator Portfolio for Agentic AI

Scaling from local agents to large scale intelligence

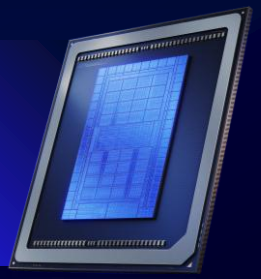
Local Agent Compute Boxes	Edge AI	AI Builder Workstations	Physical AI	Enterprise AI	Agentic AI Data Centers

Local Agent Computers → **Intelligence Centers**

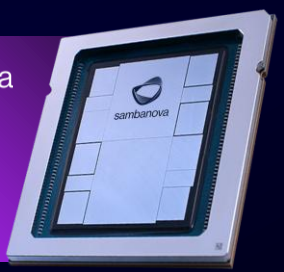
Integrated
Intel® Arc™ Pro GPUs

Workstation
Intel® Arc™ Pro GPUs

Data Center
Intel® GPUs

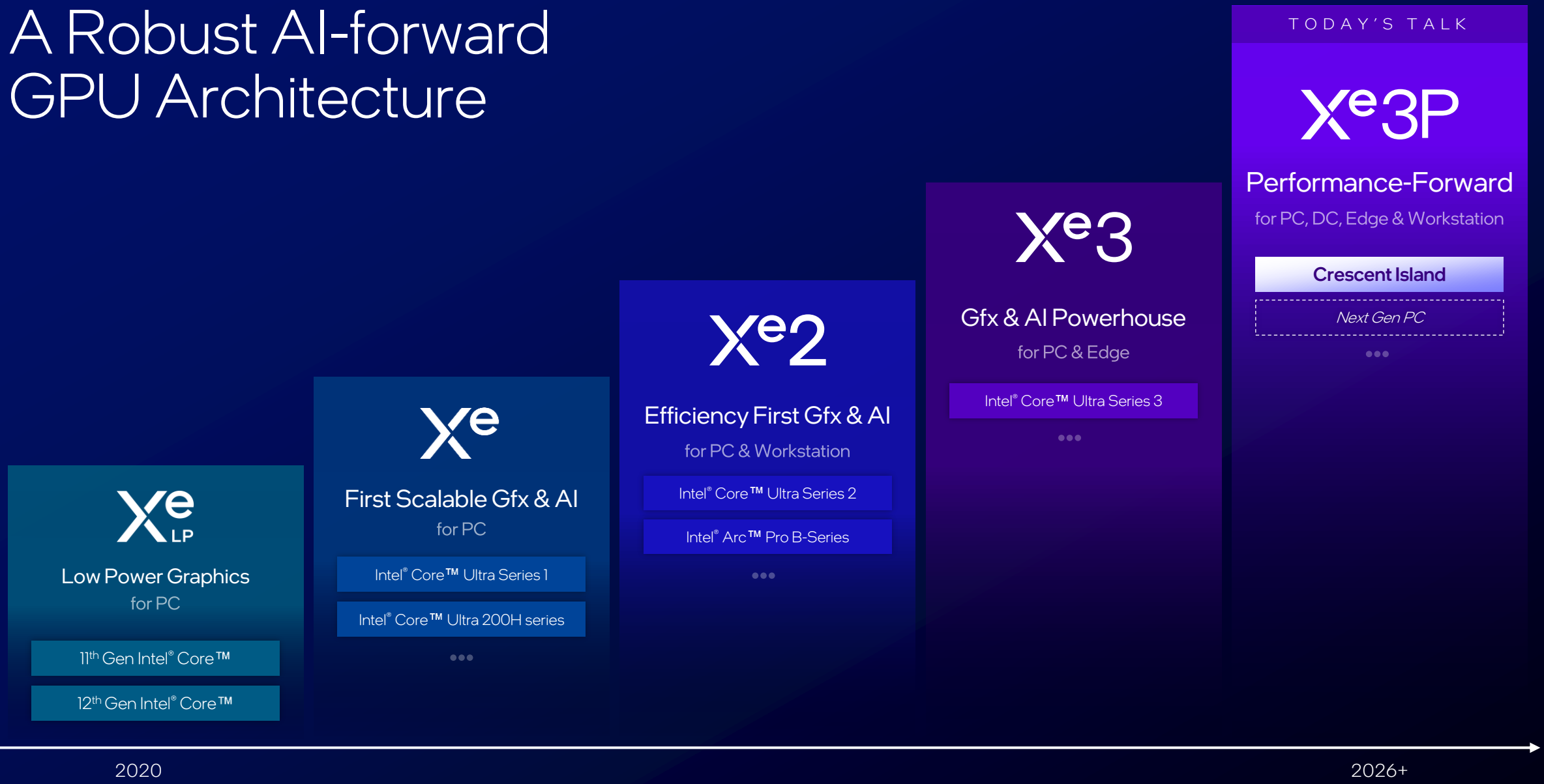


sambanova
SN50 RDUs

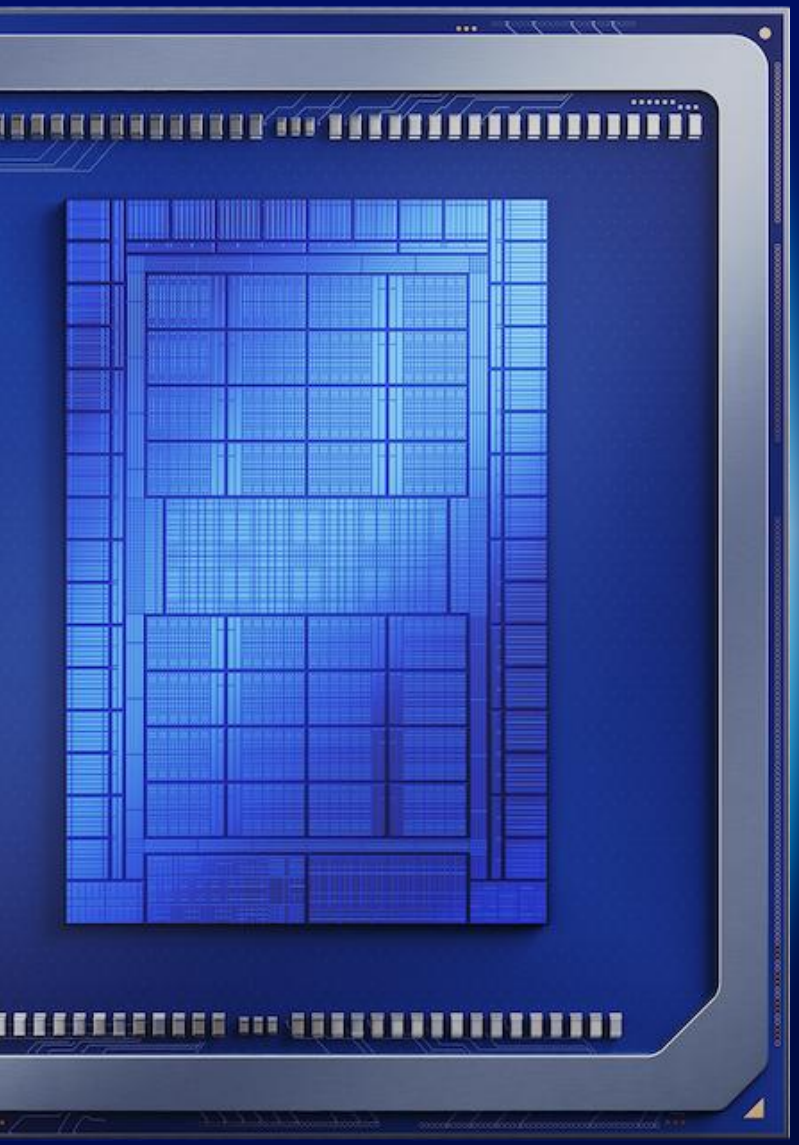


intel

A Robust AI-forward GPU Architecture



intel



Next Gen

Intel® GPU

Codenamed
Crescent Island

Designed for tokens/watt, built on a reliable open software stack

Built for Agentic AI	Large Context Capability	Power Optimized
Xe3P AI optimized GPU IP	up to 480GB Capacity	350W Air-cooled PCIe
FP4/MXFP4 - FP64 Widest range datatypes	LPDDR5x Low power memory	Open & Robust software stack

*Intel branded PCIe card has 160GB LPDDR5x; design enables partners to build ODM branded cards with flexible options up to 480GB memory.

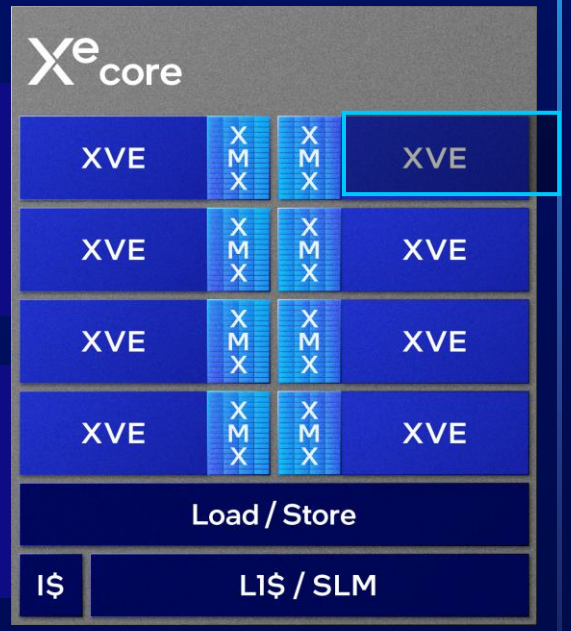
intel

3rd Generation

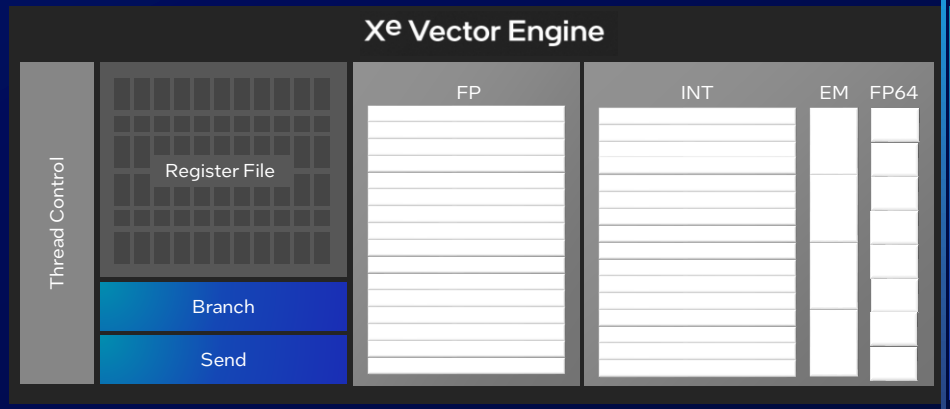
Xe core

8 Vector Engines

8 XMX Engines



Xe Vector Engine



GPU IP with FP4 Precision

3-way co-issue

Extended Math & FP64

Xe Matrix Extensions

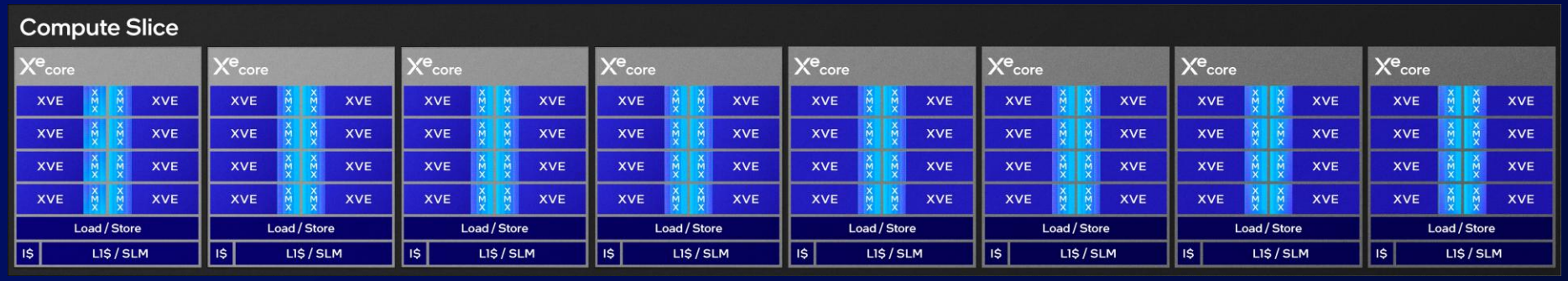
Xe Matrix Extensions

AI Acceleration Engines with **16-deep systolic array**

Optimized from engine to core for a more efficient performance

intel

Crescent Island Architecture



32 Xe-cores

256 XMX engines

intel

Xe-Architecture: Key Compute/AI Features over generations

	Xe2 Codename: Battlemage Client dGPU (Arc B-series)	Xe3 Codename: Panther Lake Client iGPU design	Xe3p Codename: Crescent Island AI GPU
GPGPU (FE, XeCore/EU)	20 Xe Cores XMX: 4-deep systolic FP16/BF16/TF32, INT8/INT4	Up to 12 Xe Cores XMX: 4-deep systolic FP16/BF16/TF32, INT8/INT4	**32** XeCores XMX: **16-deep systolic** FP16/BF16/TF32, INT8/INT4 **FP8 + FP4** format support **Microscaling (MX)** format support
	FP64 (8 FMA/XeCore)	FP64 (8 FMA/XeCore) Variable Registers per Thread (VRT) Improved thread dispatch & preemption	Full rate FP64 (**64 FMA**/XeCore) **Same Address Multi-Queue** Support for **Sigmoid and Tanh** Transcendental/EM improvements
	GRF per XeCore: 512KB	GRF per XeCore: 512KB	GRF per XeCore: **1MB**
Memory Sub-system	256KB L1$/SLM per XeCore 18MB L2 cache 24GB GDDR6	Larger L1$/SLM per XeCore Shared system memory	**512KB** L1$/SLM per XeCore **32MB** unified L2 **Up to 480GB** LPDDR5x

Crescent Island SoC Chip

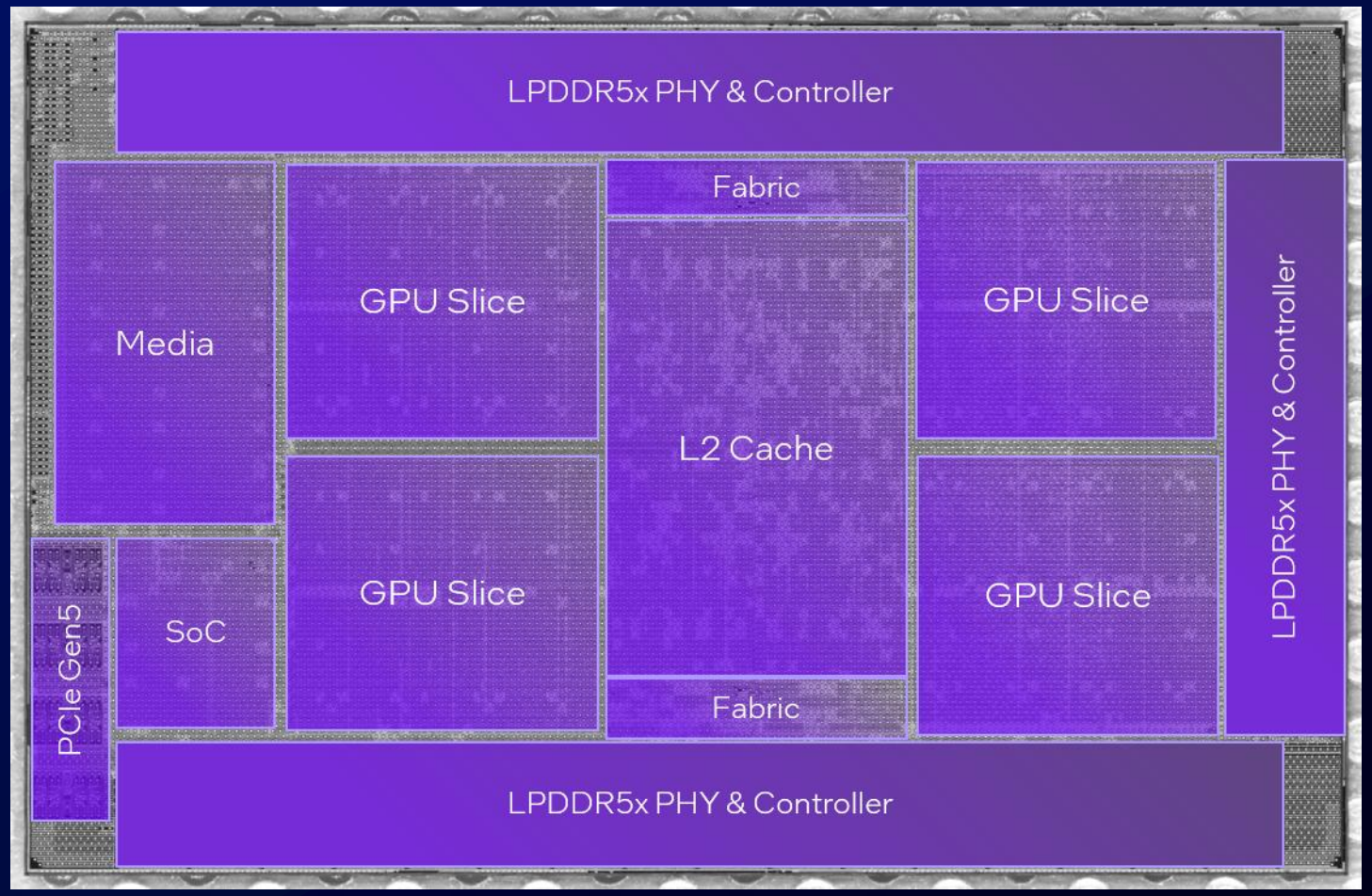
32 Xe Cores
High-FLOP Compute Behind Every Reasoning Step

32MB Unified L2
Bandwidth Filter that Keeps the Matrix Engines Fed

LPDDR5x
KV-Cache Capacity for Long Context and Concurrent Sessions

Media Engine
(4 Decoder + 4 Encoder)
Compressed-Domain Visual I/O for Multimodal Agents

PCIe Gen5 x16
Scale-Up on an Open Standard PCIe switch fabric

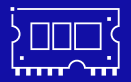


intel

RAS Features

GPU IP and Media Reliability

- ECC and Parity protection across key memory and compute blocks
- Fine grained error management such as masking and thresholding
- Local error containment and fault isolation at Xᵉcore and memory subsystem level recovery
- Dynamic Error Injection and Control
- End-to-End Cmd/Address/Data Poison Propagation



LPDDR Memory Reliability

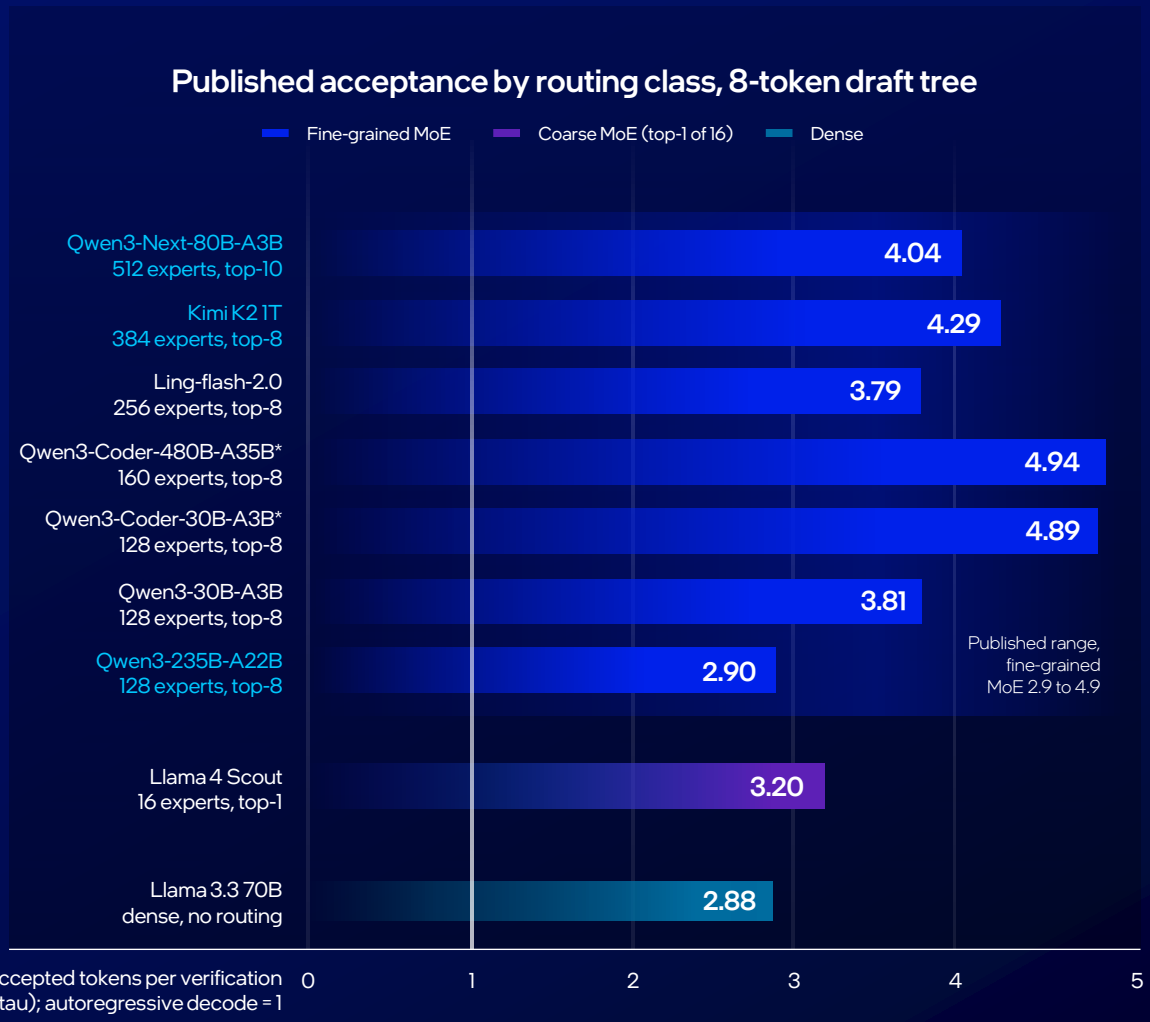
- ECC over DMI enables drastically reduced SDC rates without bandwidth / capacity impact
- Dynamic Page Offlining
- Dynamic Error Injection and Control
- hPPR (hard Post Package Repair)
- Data Poison Propagation
- Patrol Scrubbing

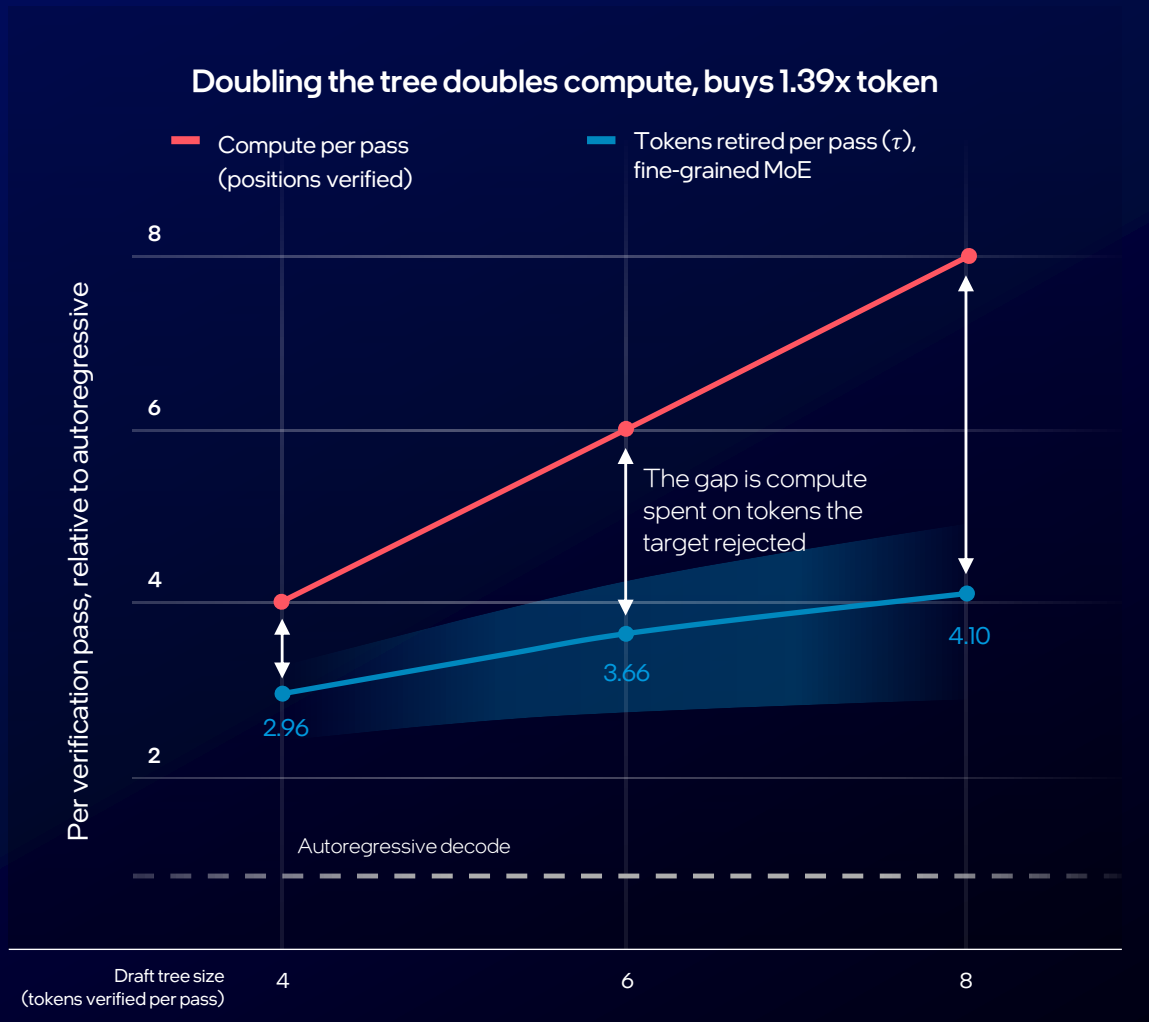
PCIe Advanced Error Reporting (AER) support

Speculative Decoding

Decode Becomes a Compute Problem

Measured on frontier MoE: 2.9 to 4.9 tokens retired per verification pass, paid for in compute that autoregressive decode leaves idle





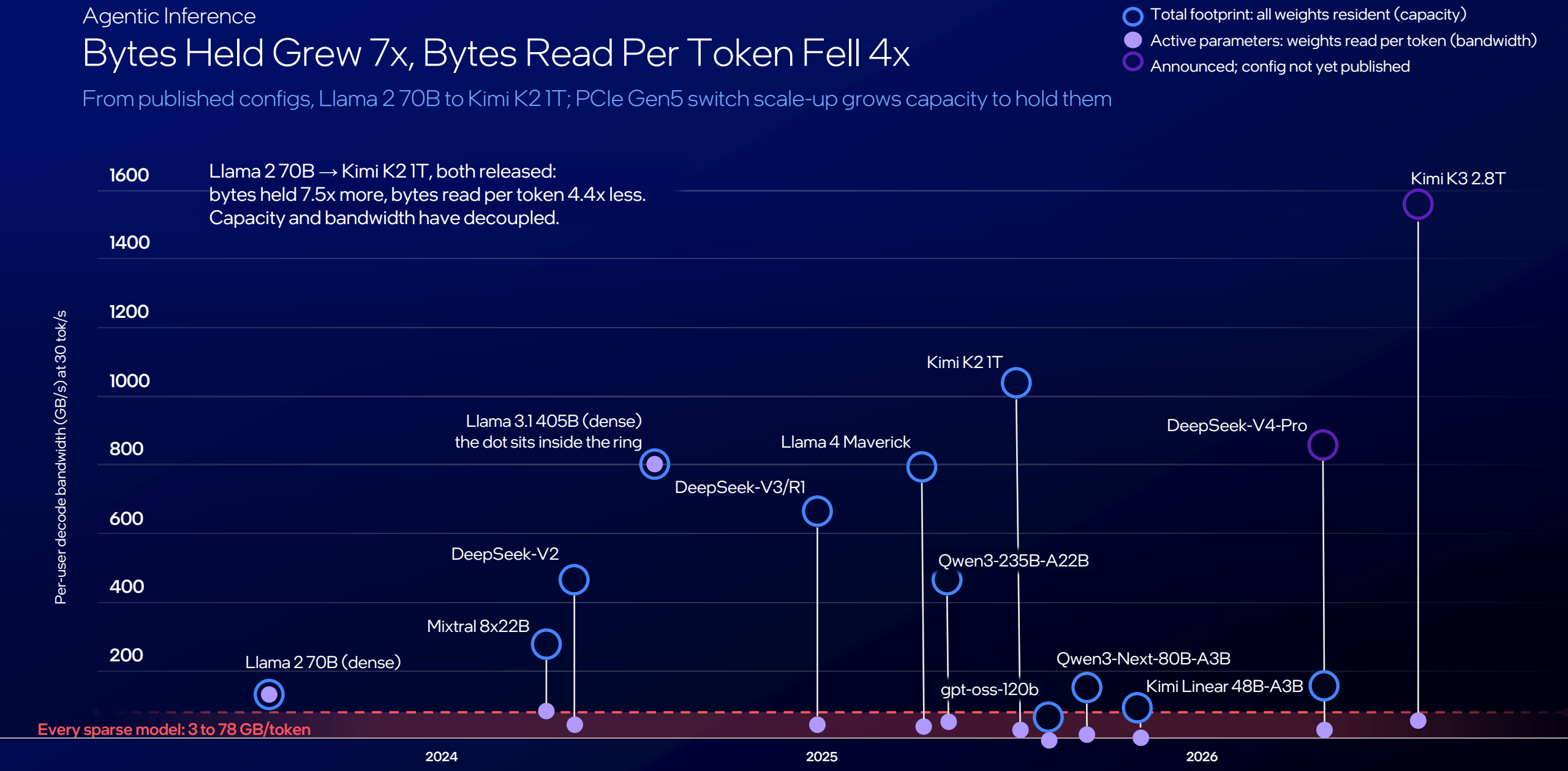
Source: LMSYS SpecBundle (SpecForge/SGLang) datasets, technical papers, and project announcements; see appendix/reference slide for complete citations. Accepted lengths shown; throughput metrics excluded.

intel

Agentic Inference

Bytes Held Grew 7x, Bytes Read Per Token Fell 4x

From published configs, Llama 2 70B to Kimi K2 1T; PCIe Gen5 switch scale-up grows capacity to hold them



Source: Public model repositories, technical papers, and vendor announcements; see appendix/reference slide for complete citations. Weight footprint only (KV cache excluded).

intel

Benefits of higher memory capacity on Crescent Island

Run more concurrent sessions, long context workloads, and larger models for better TCO

More KV sessions

On same model size results in higher aggregate throughput

Run long context RAG/agentic sessions

On same model size

Run larger models with fewer GPUs

Leading to better performance

CRI can easily fit model weights and KV cache stored in FP8 while a GPU with lower memory cannot

Crescent Island: designed for tokens/watt & agentic AI

Silicon Features

Agentic AI ready
High memory capacity and sustained compute

LPDDR5x memory
Higher density, lower power. Enhanced reliability with no BW/capacity loss

Performance optimized for AI
Featuring larger context lengths, higher memory, and performance/TCO

Prefill optimized
High FLOP/W, compute-bound workloads

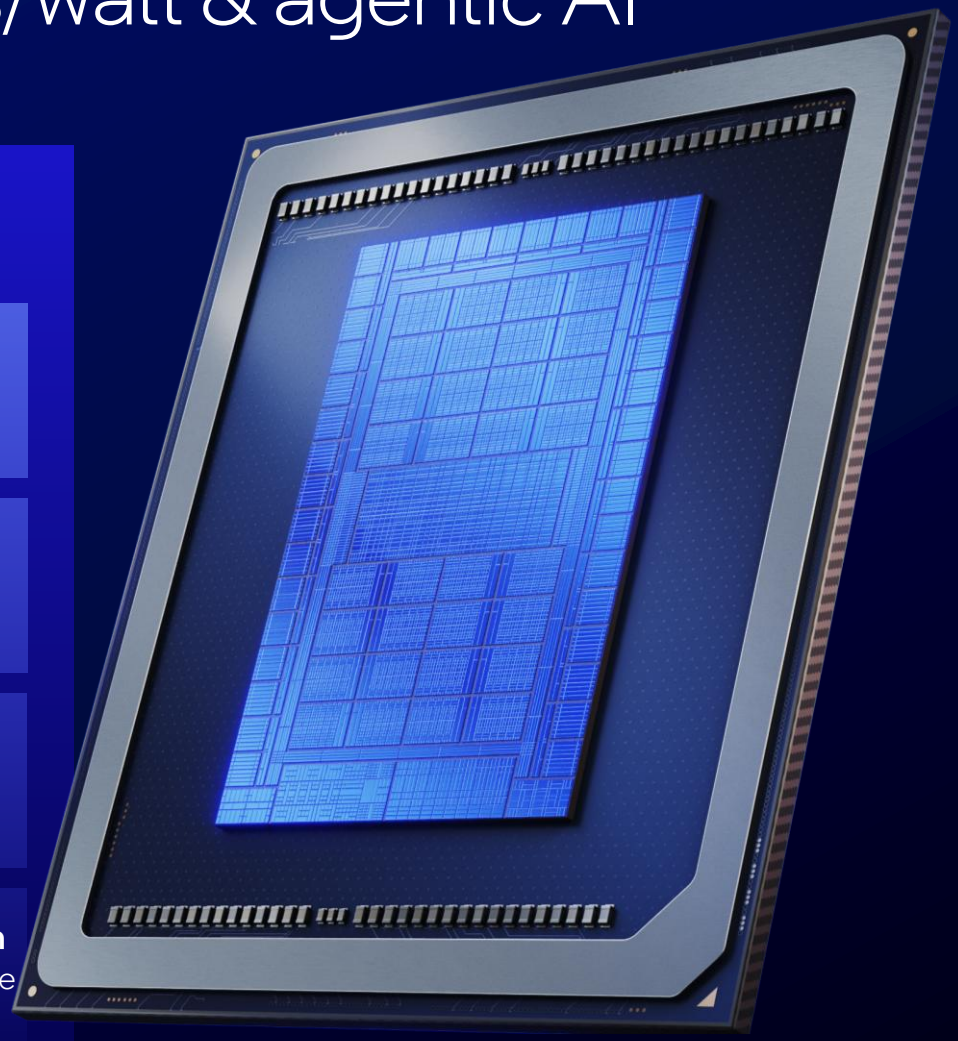
Software Features

Broad model support
Supports language, reasoning, multimodal, and diffusion models

KV cache efficiency
Enables KV-cache-aware routing and offloading for scalable inference

Open framework ecosystem
Built to work with leading AI serving frameworks

Heterogeneous agent orchestration
Runs across heterogeneous infrastructure without changing agent code



intel

An Industry Standard

AI Development Stack

Open, Upstreamed and Day 0
Ready for Intel® GPU

Trusted by an Ecosystem

Used by 100s of ISVs across a wide variety of applications

Built to Feel Familiar

Open ecosystems, and developer-trusted libraries & frameworks

Support You Can Rely on

Day 0 functionality and performance with fully validated firmware, drivers and reference systems

AI Runtimes

Frameworks

Compilers

Libraries & Tools

Orchestration & OS

Crescent Island

intel

An Industry Standard

AI Software Stack

Open, Upstreamed and Day 0 Ready for Intel® GPU

Compilers
- Rapid kernel generation
- Triton
- SYCL-TLA

Libraries & Tools
- Multi-node & scale-out libraries
  - oneCCL
  - NIXL w UCX
- Performance libraries
  - oneDNN
- Heterogenous programming
  - SYCL (+OpenMP)
- Developer tools
  - Intel® vTune
  - Intel® GDB

Orchestration & OS
- OS
- Hypervisor
- Orchestration

Level Zero
- Hardware runtimes
  - Intel compute runtime
- Kernel launch & data control
  - OpenCL

AI Runtimes

Frameworks

Compilers

Libraries & Tools

Orchestration & OS

Crescent Island

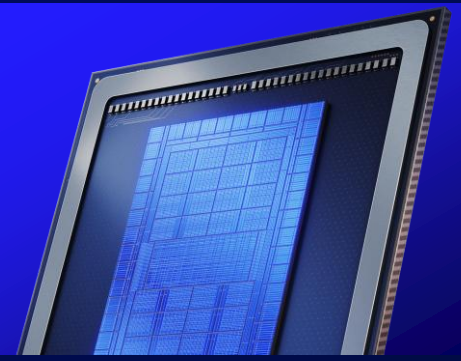
intel

Intel® GPU

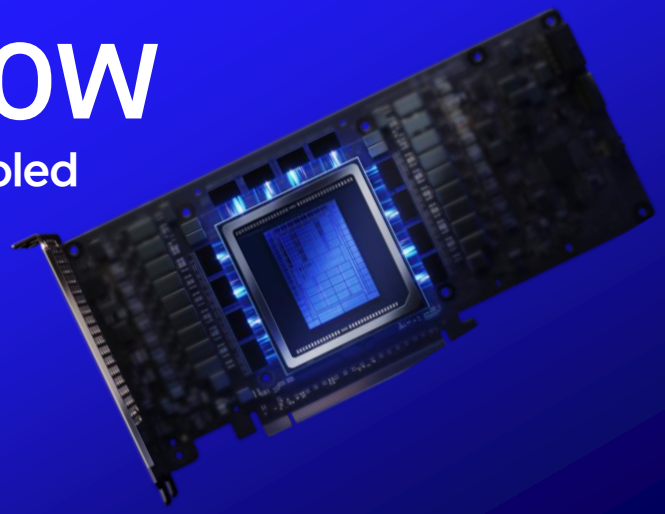
Code-named Crescent Island

Performant. Efficient.
Inference optimized GPU for agentic AI.

Up to 480 GB LPDDR 5x Memory



350W
Air cooled PCIe



Agentic AI Ready
more concurrent sessions, long context workloads, and larger models

Widest range of Data types
from FP4-FP64

Prefill optimized
High FLOP/W, optimized for compute-bound workloads

Open software stack

Industry standard frameworks Day 0 functionality

32 Xe Cores
Xe3P GPU IP

intel

intel

Notice & Disclaimers

Performance varies by use, configuration and other factors. Learn more at intel.com/performanceindex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Methodology & Sources: Bytes Held Grew 7x, Bytes Read Per Token Fell 4x

Methodology

- Weight footprint only, on a linear axis. Candidate replacement, not an addition.
- Model configurations were read from config.json in each vendor's public Hugging Face repository, huggingface.co/<repo id>/blob/main/config.json
- arXiv references correspond to the model publications listed. arXiv ids resolve at arxiv.org/abs/<id>.
- DeepSeek-V4-Flash, DeepSeek-V4-Pro and Kimi K3 2.8T are announced only, configurations not yet published; drawn with dashed rings and carrying no headline claim.
- Weights charged at each checkpoint's native format. KV cache excluded: this is the weight term only.

Models & Data Sources

- Llama 2 70B: meta-llama/Llama-2-70b-hf, arXiv:2307.09288.
- Mixtral 8x22B: mistralai/Mixtral-8x22B-v0.1, arXiv:2401.04088.
- DeepSeek-V2: deepseek-ai/DeepSeek-V2, arXiv:2405.04434.
- Llama 3.1 405B: meta-llama/Llama-3.1-405B, arXiv:2407.21783.
- DeepSeek-V3 / R1: deepseek-ai/DeepSeek-V3, arXiv:2412.19437 and arXiv:2501.12948.
- Llama 4 Maverick: meta-llama/Llama-4-Maverick-17B-128E-Instruct (gated; config read from mirror unsloth/Llama-4-Maverick-17B-128E-Instruct), Meta AI blog 5 April 2025.
- Qwen3-235B-A22B: Qwen/Qwen3-235B-A22B, arXiv:2505.09388.
- gpt-oss-120b: openai/gpt-oss-120b, arXiv:2508.10925.
- Kimi K2 1T: moonshotai/Kimi-K2-Instruct, arXiv:2507.20534.
- Qwen3-Next-80B-A3B: Qwen/Qwen3-Next-80B-A3B-Instruct, Qwen Team blog 10 September 2025.
- Kimi Linear 48B-A3B: moonshotai/Kimi-Linear-48B-A3B-Instruct, arXiv:2510.26692.

Methodology & Sources: Decode Becomes a Compute Problem

Methodology

- Accepted lengths from the LMSYS SpecBundle performance dashboard.
- Drafting method EAGLE-3, Li, Wei, Zhang and Zhang, arXiv:2503.01840, served by SGLangat batch 8. Means over gsm8k, math500, mtbench, humaneval, livecodebench, financeqa, gpqa.
- Expert counts from each model's config.json.
- The source also publishes wall-clock throughput; it is deliberately not shown.
- Full record: Speculative_Decoding_Slide_References.md.

Models & Data Sources

- SpecForge project (LMSYS Org / SGLang)
  - docs.sglang.io/SpecForge/SpecBundle.
  - Data file: sgl-project.github.io/SpecForge/SpecBundle/raw_data/data.json, retrieved 13 August 2026,
  - archived verbatim as specbundle_raw_data.json.
  - Repository: github.com/sgl-project/SpecForge, Apache License 2.0.
- Announcements:
  - lmsys.org/blog/2025-07-25-spec-forge/
  - lmsys.org/blog/2025-12-23-spec-bundle-phase-1/.
- Targets as named in the dashboard:
  - Qwen3-30B-A3B-Instruct-2507, Qwen3-235B-A22B-Instruct-2507, Qwen3-Next-80B-A3B-Instruct-FP8, Qwen3-Coder-30B-A3B-Instruct,Qwen3-Coder-480B-A35B-Instruct, Kimi-K2-Instruct, Ling-flash-2.0, Llama-4-Scout-17B-16E-Instruct, Llama-3.3-70B-Instruct.


intel