Report On The Chinese LineShine System

Tech Report No. ICL-UT-26-01

Jack Dongarra

University of Tennessee, Knoxville
University of Manchester

June 23, 2026

Prepared by
UNIVERSITY OF TENNESSEE,
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE,
INNOVATIVE COMPUTING LABORATORY

Overview

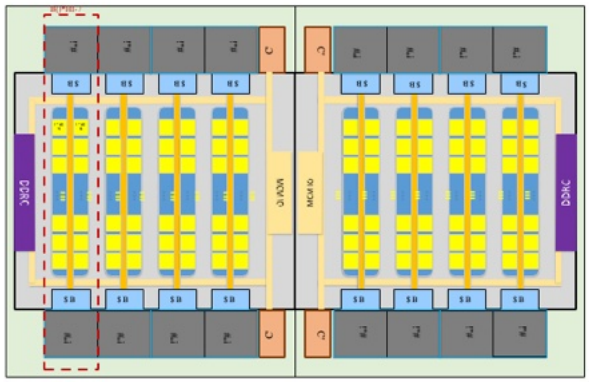
The LineShine compute system was designed and built by National Supercomputing Center in Shenzhen and Shenzhen Cloud Computing Center Co. Ltd with partners. LineShine, officially known in Chinese as **灵晟** (Ling Sheng), translates literally to "Intelligent Splendor" or "Intelligent Brightness. The system is installed at the National Supercomputing Center (NSCC) in Shenzhen. Development for the LineShine hardware started in 2023.

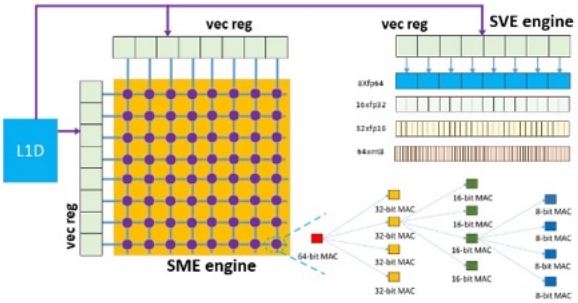
Figure 1. National Supercomputer Center (NSCC) in Shenzhen

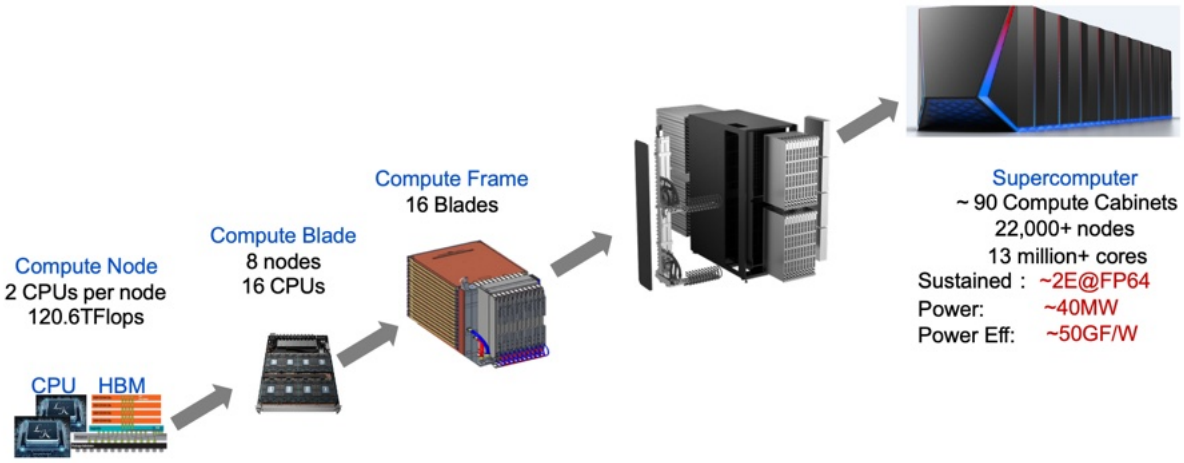
Figure 2. LineShine System as installed in NSCC Shenzhen

LineShine is a CPU-only system that uses a custom Chinese processor. The exact processor model, fabrication process, and manufacturer are not stated. The LineShine system is built on the ARMv9-based LX2 processors. Each LX2 processor contains two compute dies, for a total of 304 Armv9 cores. The package includes 32 GB of on-package HBM, delivering about 4 TB/s of aggregate bandwidth. In addition, each compute die is attached to 128 GB of off-package DDR memory, organized into four NUMA domains, giving 256 GB of DDR per LX2 processor. Thus, an LX2 combines a small, very high-bandwidth HBM tier with a larger DDR memory tier, with data movement between them handled by a dedicated SDMA engine. The LX2 supports FP64/FP32/FP16/INT8 via SME and SVE units, delivering up to 60.3/120.6/241.2 TFLOPS in FP64/FP32/FP16. Nodes are interconnected via the LingQi high-speed network with a dual-plane multi-rail fat-tree topology, offering 1.6 Tb/s bandwidth per node. Below is a further breakdown of the hardware.

Each LX2 processor uses one compute chiplets and has a total of 304 CPU cores organized into eight CPU clusters containing 38 cores each. Every core includes Arm SVE (Scalable Vector Extension) and SME (Scalable Matrix Extension) units that accelerate vector and matrix operations used in AI training and scientific computing that support FP64, FP32, BF16, FP16, and INT8 data formats. Each core is equipped with 32 KB L1 instruction cache and 32 KB L1 data cache, while every cluster shares a 28.5 MB L2 cache. The peak FP64 performance is 60.3 TFLOP/s at 690W.







Each CPU/package includes 64 GB of HBM, with an HBM bandwidth of about 4 TB/s, plus roughly 500 GB of DDR5 memory.

The system packaging is described as:

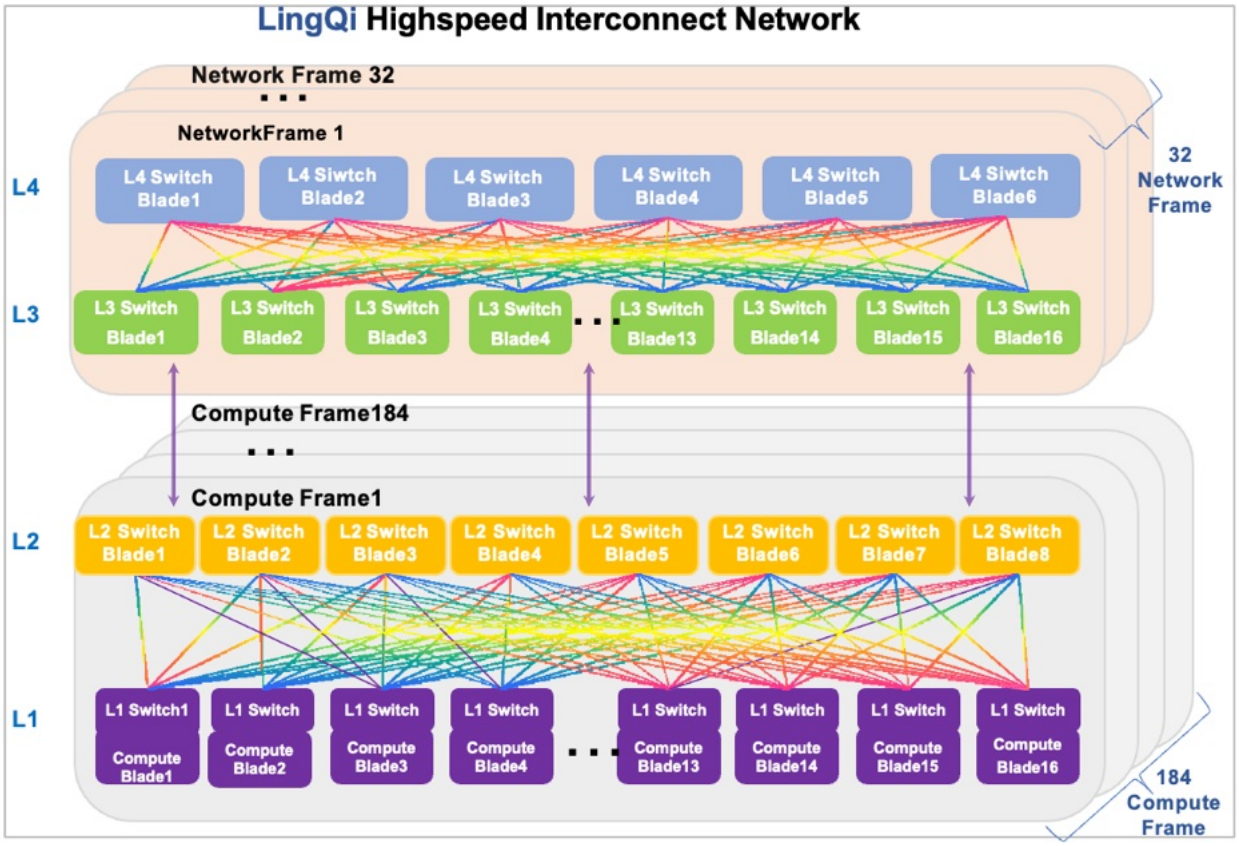
- 8 nodes per blade
- 16 blades per frame
- 2 frames per cabinet
- 90+ cabinets total
- 22,000+ nodes

This gives a theoretical peak of about 30.87 PFLOP/s per cabinet, leading to roughly 2.7 EFLOP/s peak FP64 performance across about 90 cabinets.

Theoretical Peak Performance of Various Precisions Across 90 Cabinets

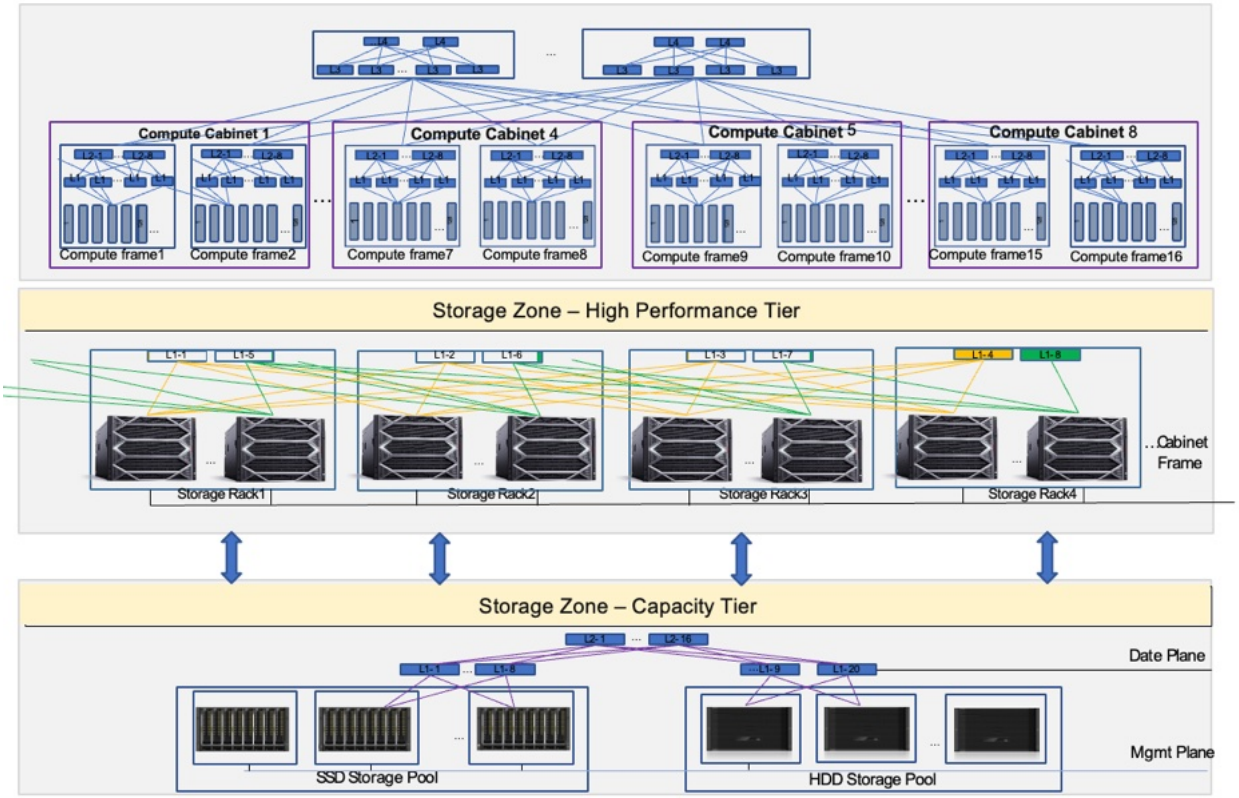
Precision	Performance per node	Total Performance
FP64	120.6 TFLOP/s	2.7 EFLOP/s
FP32	241.2 TFLOP/s	5.4 EFLOP/s
FP16	482.4 TFLOP/s	11 EFLOP/s
INT8	1920 TOPS	40.2 EOPS

	System Memory
Per CPU/socket	32 GB HBM + 256 GB DDR5
Per node	64 GB HBM + 512 GB DDR5
Per cabinet	256 nodes × per-node memory = 16 TB HBM + 128 TB DDR5
Full system, ~90 cabinets	about 1.5 PB HBM + 11.5 PB DDR5
Combined full-system memory	roughly 13 PB total memory

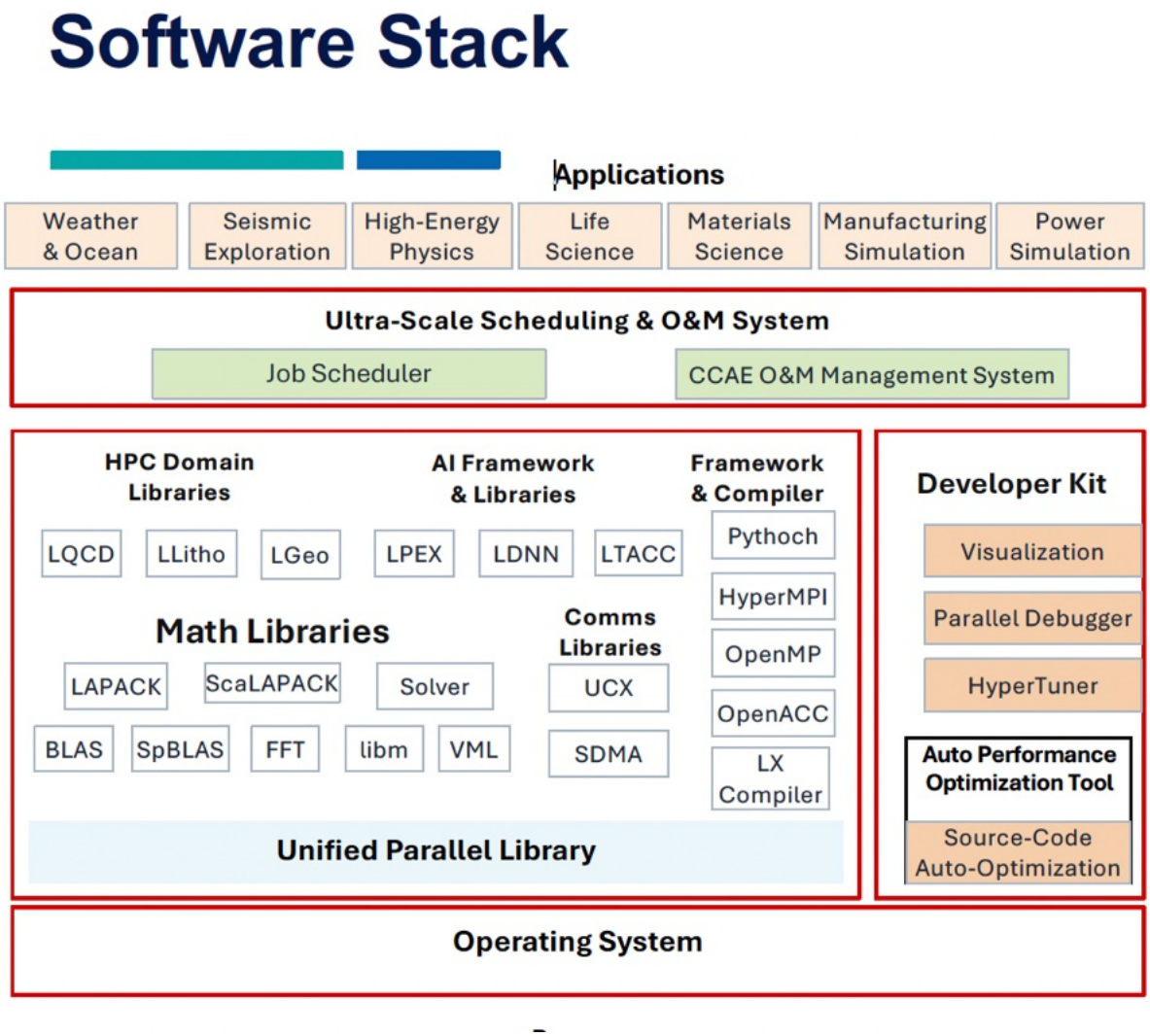


- Large-scale network: 22K+ nodes, 200K ports
- 4-layer fat-tree topology
- Full-system bisection bandwidth: ≥ 3.5 PbpsCE
- Single-hop latency: 1.07 μs
- Four-layer network with only one optical layer
- Credit-based flow control for lossless communication

Tiered Storage & Distributed File System



- Tiered architecture balances bandwidth, capacity, and cost
- Unified data path for job I/O, staging, and lifecycle migration
- Distributed file system with scalable metadata and shared namespace
- Performance tier for active jobs; capacity tier for retention and cold data
- Capacity; 200 PB – 1000 PB (planed)



The system software stack is designed to expose the full capability of the underlying hardware while reducing overhead and improving programmability. At the operating-system level, deep kernel isolation reduces operating-system noise and service jitter, which is especially important for tightly coupled parallel applications. The communication layer supports multi-rail networking, allowing traffic to be balanced across eight network ports and improving aggregate communication bandwidth.

Above this layer, the system provides a unified parallel acceleration library that is compatible with widely used ecosystems such as OpenBLAS, FFTW, OpenMP, and PyTorch. This helps users move existing applications and development workflows onto the system more easily. The library also provides the foundational mathematical and acceleration routines needed to exploit the hardware capabilities of the SME units and high-bandwidth memory.

The compiler and math libraries include optimizations that are aware of both SME and HBM. These include the automatic transformation of suitable computations into SME matrix operations, the insertion of prefetching instructions, and the placement of frequently used data in

high-bandwidth memory. Optimized dense-compute primitives improve the efficiency of SME execution, while sparse-compute primitives are tuned to make better use of HBM bandwidth.

An automatic performance optimization tool further reduces the burden on application developers. Without requiring source code changes, it can identify algorithms suitable for acceleration and transform them into SME-based matrix multiplication. It also automates more than thirty domain-specific optimization techniques to improve hardware affinity and overall application performance.

Finally, the scheduler and management platform provide customized support for different job types, scheduling policies, resource allocation, and job lifecycle management. Together, these software layers make the system easier to use, better balanced, and more capable of exploiting the hardware's specialized acceleration and memory features.

SME-Enabled, HBM-Aware Matrix Acceleration

SME-enabled, HBM-aware approach to matrix acceleration that is intended to serve a wide range of computation patterns, from traditional HPC workloads to AI workloads. These patterns include stencil computations, FFTs, GEMM, graph analytics, and attention mechanisms. The main idea is to map diverse computational kernels onto efficient matrix operations to make better use of the hardware.

A central feature is SME matrix function. The SME unit is used to handle unified matrix multiplication across both HPC and AI applications. For stencil-like computations, this includes multi-row update matrix multiplication and outer-product stencil formulations. Dense linear algebra includes square-tile matrix multiplication in GEMM. For Transformer models, it includes tall-and-skinny matrix multiplications such as those used in attention operations. This matrix function approach can achieve more than 40% SME utilization, showing that nontraditional workloads can still benefit from matrix-oriented acceleration.

While SME is well-suited for matrix-oriented computation, SVE is used for row-wise and irregular operations. Examples include single-row updates in stencil computations, boundary handling in GEMM, and online softmax in Transformer models. By interleaving SME and SVE instructions scheduling, the system can improve instruction-level performance, with reported IPC improvements of up to 1.59×.

A second major theme is memory-aware data placement. The system uses proactive buffer management between HBM and DDR memory. Pre-allocating an HBM buffer pool reduces the memory footprint by about 3.9 GB. It also improves cache behavior by blocking computations to keep working tiles cache-resident, packing tiles into SME-friendly layouts, and prefetching non-contiguous data needed in later iterations. These optimizations improve cache hit rates by up to 28%.

The system provides a suite of SME-HBM mapping methods. The HStencil, aimed at stencil computations; KirbyMM, a matrix multiplication optimization effort; SMEAtten, aimed at Transformer attention; The performance results show significant gains: stencil workloads

improve by up to 4.1× over auto-vectorization, GEMM improves by 1.11× to 1.75× over the vendor library, and attention workloads show an average improvement of 13.62× over the state of the art.

Overall, the message is that efficient execution with SME and HBM relies on the software stack being aware of both the computation pattern and the memory hierarchy. The suite of SME-HBM mapping methods make application optimization process practical by transforming computations into matrix-friendly forms, combining SME with SVE for irregular work, and carefully managing data placement in HBM and cache. This system can accelerate both conventional HPC kernels and emerging AI workloads.

Benchmark performance

An important result is an HPL sustained performance of about 2.198 EFLOP/s, for submission to the TOP500 list. The run is achieving roughly 80% of peak efficiency, with power consumption around 40–42 MW during the HPL run. The matrix size mentioned for the HPL run is n = 37,647,359.

The system also reportedly ran HPL-MxP, achieving about 7.92 EFLOP/s using FP16-based mixed precision. Giving about a three times improvement over the HPL algorithm.

HPCG

The system has a strong HPCG result of 22.0049 PFLOP/s, achieving 1% of HPL efficiency.

AI and application use

The system is described as an HPC-plus-AI convergence machine. It is not just intended for the Linpack benchmark. For the SC26 conference, there were 14 Gordon Bell-related submissions that used the LineShine computer, with several using AI and others using low-precision computation. Of the 14 submissions, 3 are finalists in the Gordon Bell Prize and 3 are finalists for the Gorgon Bell Climate Prize. The system is said to support both AI inference and training, as well as traditional scientific applications using mixed precision.

A major point is that the machine represents a new strategic future direction for AI-era supercomputing: instead of attaching GPUs as external accelerators, it integrates matrix acceleration into the CPU. This is useful for “AI for science,” where simulation, mixed precision, matrix operations, and AI workflows are increasingly intertwined.

The new LineShine system is a large-scale, CPU-based supercomputer designed for converged HPC and AI workloads. Its processors integrate matrix engines directly into the CPU, supporting FP64, FP32, FP16, and INT8 computation, and combine high-bandwidth memory with DDR5 capacity memory. At full scale, the system reaches a theoretical FP64 peak performance of roughly 2.7 EFLOP/s, with an HPL result of about 2.19 EFLOP/s at around 40 MW. Beyond

HPL, the system has demonstrated strong mixed-precision performance on HPL-MxP and is being used for scientific applications that combine simulation, AI, and low-precision numerical methods.

Key takeaway

This machine is significant because it is not simply another GPU-based exascale system. Its distinguishing feature is a CPU-centric architecture with built-in matrix engines and high-bandwidth memory, paired with a co-designed software stack and aimed at the convergence of traditional HPC, mixed-precision numerical computing, and AI-for-science workflows. As Prof. Yutong Lu, chief designer of LineShine supercomputer, stated, “We have proposed a new online-acceleration supercomputing architecture and further demonstrated its feasibility and efficiency.” LineShine points toward a new direction for HPC-AI convergence in the post-exascale era.

Applications

The LineShine system is being positioned not simply as a machine for peak benchmark performance, but as a platform for large-scale scientific and engineering applications. Its value is demonstrated across a broad set of domains, including climate and ocean prediction, materials and chemistry, molecular dynamics, drug discovery, engineering simulation, and scientific AI. Across these application areas, the reported average scaling efficiency is 84.4%**,** with several applications operating in a full-system regime of more than 12 million cores**.** This breadth is important: it shows that the system is intended to support real scientific workloads, not only carefully tuned benchmark kernels.

A major example is an AI-accelerated 1-km global Earth-system simulation. This application integrates atmosphere, ocean, land, and ice components into a coupled prediction system. The workflow combines large-ensemble data assimilation with 30-day global kilometer-scale forecasts. AI-based physical parameterizations are used for atmospheric prediction and for ocean near-inertial waves, delivering a 62% reduction in computational cost. Additionally, techniques such as mixed-precision, exascale I/O preprocessing, load balancing, and auto-tuning help reduce development, porting, and optimization costs. The model achieves a new 1-km throughput record of 1.73 simulated years per day for the atmosphere**,** 3.08 simulated years per day for the ocean**,** and approximately 1 simulated year per day for the coupled system**.** Jointly on model-generated and ECMWF high-resolution data, a 1km AI atmospheric model performs a 10,240-member ensemble data assimilation and 30-day forecast cycle in two hours, scaling to 20,000 nodes and 12.16 million cores at 65.8% efficiency**.**

The system also supports extreme-scale direct numerical simulation of wall turbulence, aimed at resolving long-standing questions about near-wall scaling at very high friction Reynolds numbers. Such simulations are exceptionally expensive because higher Reynolds numbers require larger grids and longer runs for statistical convergence. The LineShine implementation uses co-optimization across compute, memory, and communication: SME/SVE co-acceleration

for high-order finite-difference stencil kernels, FP32 storage with FP64 critical-path computation, and topology-aware communication strategies to reduce and overlap transposes. The application reports a completed production simulations at $Re\tau \approx 2.1 \times 10^4$ over 1.5 million steps, with ongoing simulations at $Re\tau \approx 4.3 \times 10^4$. These runs provide high-fidelity numerical evidence for the near-wall scaling debate in a regime that is more discriminating than previous simulations.

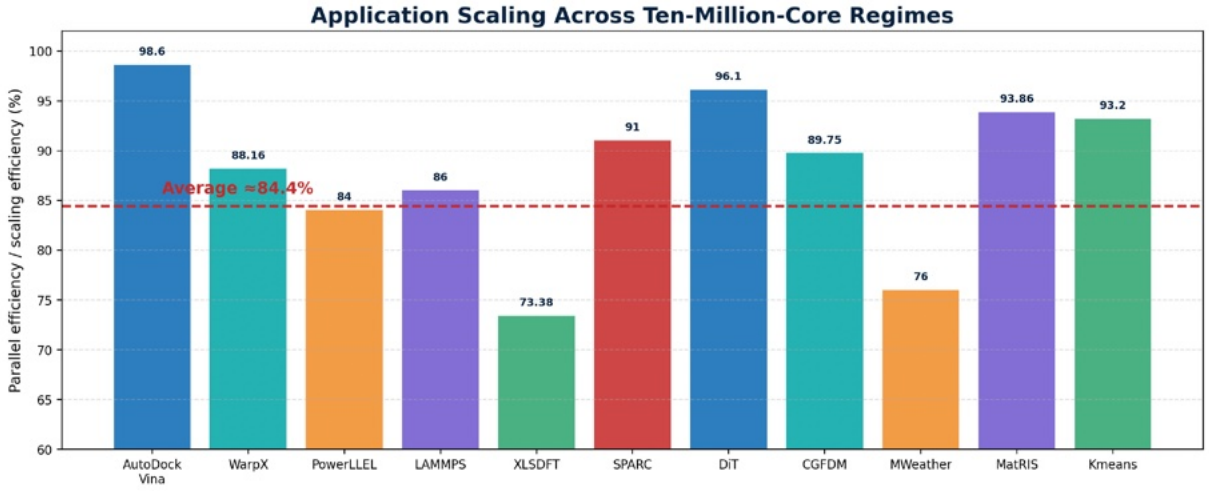
In drug discovery, LineShine is used for a 10-trillion-compound synthesizable screening workflow. The key challenge is that enforcing chemical synthesis rules expands the search space from roughly 10^{10} to 10^{13} candidates, while coupling fast AI inference with slower high-fidelity physical simulations creates severe load imbalance. The proposed approach confines AI generation to established reaction templates, ensuring synthesizability, and uses a self-evolving HPC-AI framework in which reinforcement learning navigates chemical space while high-fidelity simulations periodically anchor the model. A hierarchical, asynchronous scheduling mechanism decouples fast AI tasks from expensive HPC simulations. The reported result compresses a 10-trillion-candidate high-fidelity screening campaign into 22.6 hours, using 20,480 LineShine nodes and 12.45 million cores, with 80.9% weak-scaling efficiency and 65.8% strong-scaling efficiency.

For materials science, the system supports spin–lattice simulation of skyrmion formation in FeGe at device-relevant scales. Skyrmions are nanoscale topological spin structures of interest for low-power information devices, but resolving their finite-temperature helix-to-skyrmion transition requires explicitly coupling lattice motion with spin dynamics. The NEPSPIN methodology uses local spin-aware descriptors to reduce complexity from $O(N^3)$ to $O(N)$, while many-core optimization includes OpenMP fusion, SVE2 vectorization, row-major floating-point layout, and SME pipelining. The reported simulation reaches 1.34 trillion atoms and spins, using 12.45 million CPU cores with 90% weak scaling efficiency, and covers a physical domain of $3.02 \times 2.41 \times 2.41\ \mu m^3$.

Another striking application is whole-brain simulation at the 100-trillion-synapse scale. CerebroSim transforms medical and physiology into executable spiking neural networks, enabling biologically constrained studies of human-scale brain dynamics. The implementation includes delay-aware spike broadcast, race-free synaptic dynamics computation, and sparse synapse storage compression. The reported simulation includes 86 billion neurons and 100 trillion synapses on 18,432 LineShine nodes, achieving 24.44 PFLOP/s, 91% weak scaling, and 94% strong scaling. This demonstrates the machine's relevance not only to traditional physics-based simulation but also to computational neuroscience and large-scale biologically inspired modeling.

Additional application cases further broaden the picture. Large-scale earthquake simulation converts strong-motion physics into actionable geographic risk information for emergency decision-making, including a reported trillion-grid real-earthquake simulation using mixed precision while retaining absorbing-boundary and free-surface physics. Full-aircraft wall-resolved large-eddy simulation supports high-precision aerodynamic design with roughly 1 trillion degrees of freedom and 10 billion cells, using high-order methods and AI convergence boosters. Billion-parameter universal machine-learned interatomic potentials are trained at

exascale, aiming to deliver quantum-accurate molecular dynamics across molecules, crystals, catalysis, molecular crystals, and metal-organic frameworks. Generative compression for Earth observation data uses geographic, temporal, and spectral priors to reconstruct remote-sensing data at ultra-low bit rates.



Taken together, these applications illustrate a shift from traditional scientific computing toward scientific intelligence computing. The common pattern is no longer simulation alone, or AI alone, but a coupled workflow: simulation, data assimilation, AI inference, model training, optimization, visualization, and uncertainty-aware evaluation. The application ecosystem, therefore, requires more than hardware. It depends on a layered software stack: domain frameworks, intelligent orchestration, data and model infrastructure, numerical libraries, AI operators, compilers, runtimes, communication systems, scheduling, monitoring, energy management, and trusted evaluation.

The broader message is that post-exascale computing will be judged by useful scientific throughput, not simply peak FLOP/s. Real performance depends on energy efficiency, data movement, software maturity, numerical trust, reproducibility, and the ability to integrate data and models as long-term strategic assets. In this sense, LineShine's application portfolio argues for system–algorithm–software–data–model co-design as the organizing principle for future supercomputing.

Appendix A.

Table A-1. Comparison with top machines on the TOP500

	LLNL El Capitan	**ORNL Frontier**	**RIKEN Fugaku**	**Sunway TaihuLight**	**LineShine**
Theoretical Peak	2.74 EFLOP/s	2.0 EFLOP/s	514 PFLOP/s	125.4 PFLOP/s = CPEs + MPEs Cores per Node = 256 CPEs + 4 MPEs	2.7 EFLOP/s

				Supernode = 256 Nodes System = 160 Supernodes Cores = 260 * 256 * 160 = 10.6M	
HPL Benchmark FLOP/s	1.809 EFLOP/s	1.353 EFLOP/s	415 PFLOP/s	93 PFLOP/s	2.198 EFLOP/s
HPL % peak	66%	65%	81%	74.16%	81%
HPCG benchmark	17.4 PFLOP/s	14.1 PFLOP/s	13 PFLOP/s	.371 PFLOP/s	21.019 PFLOP/s
HPCG % peak	0.60%	0.70%	2.8%	0.30%	1 %
Compute nodes	11,520	9,408	152,064 (This is on 96% of the full system)	40,960	22,680
Node	4 AMD Instinct MI300A APUs 96 AMD EPYC CPU cores per node 4 GPUs per node	Optimized 3rd Gen AMD EPYC (2 TFLOP/s, 64 cores) Plus 4 AMD Instinct MI250X GPUs / node (53 TFLOP/s, 220 cores each)	48 cores	256 CPEs + 4 MPEs	2 LX2 608 cores per node
Node peak performance	250.8 TFLOP/s	214 TFLOP/s (2 CPU + 4*53 GPU) TFLOP/s	3.4 TFLOP/s	12	120.6 TFLOP/s
Node memory	512 GB	1024 GB	32 GB HMB2	32 GB per node	64 GB HBM + 512 GB DDR5
System memory	5,760 GB	9,408 nodes*1,024 GB/node = 9.6 PB	4.85 PB	1.31 PB (32 GB × 40,960 nodes)	13 PB
Configuration	90 cabinets	9,408 nodes in 74 cabinets	158,976 nodes, 432 cabinets	Node peak performance is 3.06 TFLOP/s, or 11.7 GFLOP/s per core. 260 cores/node CPE: 8 FLOPs/core/cycle (1.45 GHz × 8 × 256 = 2.969 TFLOP/s) MPE (2 pipelines) 2 × 4 × 8 FLOPs/core/cycle (1.45 GHz × 1= 0.0928 TFLOP/s)	8 Nodes per blade, 16 blades per frame and 2 frames per cabinet and 90+ cabinets in the system.

				Node peak performance: 3.06 TFLOP/s 1 thread/core Nodes connected using PCI-E The topology is Sunway network. 256 nodes = a supernode (256 × 3.06 TFLOP/s = . 783 PFLOP/s) 160 supernodes make up the whole system (125.4 PFLOP/s) The network system consists of three different levels, with the central switching network at the top, the super node network in the middle, and the resource-sharing network at the bottom. 4 SNs = cabinet Each cabinet ~3.164 PFLOP/s 256 nodes per SN 1,024 nodes (3 TFLOP/s each) per cabinet 40 cabinets ~125 PFLOP/s	
Total system	CPU cores: 11,424 nodes × 96 CPU cores/node = 1,096,704 CPU cores GPU compute units: 11,424 nodes × 4 MI300A/node × 228 CUs/MI300A = 10,418,688 GPU compute units	(64 CPU cores + 4*220 GPU cores) * 9408 nodes = 8,881,152 cores	7,630,848 = 158,976 * 48 cores	10,649,600 cores = Node (260) × supernodes(256 nodes) × 160 supernodes	13,698,720 cores
Power (processors,	36 MW	29 MW	28.33 MW (7.33*3.863)	15.3 MW	42 MW for full system

memory, interconnect)					
Footprint	557.4 m^2	370 m^2 Each Cabinet weigh is 8,000 pounds without water.	1,920 m^2	605 m^2	332.64 m^2