High-Performance Computing Pipeline Architecture for Nexa LLM Training and Inference

Allan

May 2025

Abstract

This paper presents a high-performance computing (HPC) pipeline designed for efficient training and deployment of domain-specialized large language models (LLMs) under the Nexa Mixture-of-Experts (MOE) architecture, with model sizes ranging from 110M to 2.2B parameters. The pipeline maximizes system utilization, minimizes latency, and leverages hardware heterogeneity across multi-GPU and multi-core CPU systems. Key features include dynamic batching, asynchronous I/O, mixed precision training, and gradient checkpointing, enabling scalable and elastic throughput. We detail the architecture, engineering strategies, current performance metrics, and planned improvements, positioning the pipeline as a modular, fault-tolerant HPC environment for scientific machine learning applications.

1 Introduction

The Nexa LLM suite, comprising Mini (110M parameters), CoT (750M parameters), and UltraMax (2.2B parameters), is designed for domain-specialized tasks in scientific machine learning (SciML). This paper introduces a high-performance computing (HPC) pipeline tailored for the pretraining and inference of these models under the Mixture-of-Experts (MOE) architecture. The pipeline addresses key challenges in LLM training: maximizing hardware utilization, minimizing latency, and ensuring scalability across heterogeneous hardware. It incorporates dynamic batching, asynchronous I/O, mixed precision training, and fault-tolerant checkpointing, making it suitable for models ranging from lightweight to large-scale.

The pipelines primary objectives are:

- Pretraining LLMs from scratch on domain-labeled corpora.

- Efficient MOE routing and gradient updates.

- Maximizing GPU utilization through dynamic batching and asynchronous I/O.

- Supporting mixed precision and gradient checkpointing for scalability.

- Enabling elastic throughput for models from 110M to 2.2B parameters.

2 System Architecture

The pipeline is structured as a modular, scalable system, as illustrated in Figure 1. It comprises five core components: dataset preprocessing, CPU dispatcher, dynamic batch generator, GPU compute engine, and checkpointing/optimizer modules.

Figure 1: Schematic of the Nexa LLM HPC pipeline, showing data flow from preprocessing to evaluation.

2.1 Data Pipeline and Asynchronous I/O

Tokenized data is preprocessed into semantically coherent chunks and dispatched across CPU cores using an asynchronous I/O system. The system includes:

- I/O monitoring to prevent bottlenecks.

- Dynamic feed rate adjustment based on GPU occupancy.

2.2 CPU Load-Aware Chunking

Dynamic chunk sizing adjusts batch size and sequence length based on real-time CPU load monitoring and a scheduling queue for per-batch shaping, ensuring efficient resource allocation.

2.3 Multi-GPU Execution

The pipeline leverages distributed computing for:

- Optimal GPU saturation.

- Auto-balancing using workload telemetry.

- Peer-to-peer GPU communication to minimize PCI bottlenecks.

2.4 GPU Compute Core

Heavy tensor operations are routed to GPUs, utilizing:

- Flash Attention for UltraMax models.

- Mixed precision (FP16/BF16) for memory optimization.

- Gradient checkpointing to enable 2.2B parameter scaling.

2.5 Checkpointing and Recovery

The training loop supports periodic stateful gradient saves, resume-safe interrupts, and full-stack validation restore for model, optimizer, and scheduler states.

2.6 Custom Optimizer Integration

The pipeline supports modular optimizers, including:

- Adam: Fast convergence.

- AdamW: Weight-decay aware.

- Azure Sky: A hybrid stochastic approximation and Adam optimizer for global minima discovery.

3 Performance Metrics

The pipelines performance is summarized in Table **??**.

Component	Utilization	Status
CPU Preprocessing	90–95%	Healthy
GPU Weight Compute	95–99%	Optimal
IO Wait Latency	3–7 ms	Acceptable
Multi-GPU Load Split	Balanced	Stable
Memory Management	Tuned	Stable

Table 1: Performance metrics for the Nexa LLM HPC pipeline

4 Nexa LLM Targets

The Nexa MOE suite includes three models, as shown in Table **??**. Each is trained using semantic routing and domain-specialized sub-experts, supporting tasks from general-purpose inference to chain-of-thought reasoning and long-form memory applications.

Model	Parameters	Purpose
Nexa MOE Mini	110M	Base expert model for all domains
Nexa MOE CoT	750M	Chain-of-thought reasoning, STEM
Nexa UltraMax	2.2B	Long-form memory + domain specialists

Table 2: Nexa LLM model specifications

5 Planned Improvements

Future enhancements are divided into three phases:

5.1 Phase 1: Execution Layer

- Dynamic multi-GPU load tuner for real-time tensor allocation.

- I/O autotuning monitor to eliminate latency spikes.

- Inference mode preloading for memory-stable weight compaction.

5.2 Phase 2: Intelligence and Adaptivity

- Self-tuning Azure Sky V2 optimizer using hyperparameter optimization.

- Profiling layer and logging dashboard for live metrics.

- Long context window support (20K+ tokens) with parallel attention.

5.3 Phase 3: Scientific Model Integration

- Support for SciML tasks, including protein folding, physics simulations, and symbolic expression generation.

6 Future Work: VORTEX HPC Stack

The next evolution, VORTEX, will be implemented in Rust for memory safety and performance. It will feature a custom LLVM/CUDA compiler backend, zero-copy tensor execution, and optimized throughput for both inference and pretraining.

7 Conclusion

The Nexa LLM HPC pipeline is a modular, fault-tolerant environment designed for maximal utilization, domain-aware design, and scalable simplicity. By integrating dynamic batching, asynchronous I/O, and mixed precision training, it achieves high efficiency across multi-GPU and multi-core CPU systems. Ongoing improvements and the VORTEX stack will further enhance its capabilities for scientific machine learning applications.