Serverless Hybrid AI Cluster

Benchmarks

Aditya Maramreddy · Henish Patel · Keith Li · Nikita Lopatenko · Jensen Feng

Hardware Layout

12-node hybrid cluster: 8x NVIDIA DGX Sparks + 4x Apple Mac Studios. The Software layer is Debian 13 on all DGX Sparks with a Nemotron-based control plane that routes work.

- **Role split:** Mac Studios (macOS) run agent orchestration, MCP servers, Kimi K2.7 long-context sharding (Exo/JACCL), and Qdrant. DGX Sparks (Debian) run CUDA inference via vLLM. **Metal and CUDA are not mixed at the compiler layer**.
- **Interconnects:** DGX ↔ DGX over 100GbE ConnectX-7 / RoCE v2; Mac ↔ Mac over Thunderbolt 5 / JACCL.
- **Thermal:** Mac Studios on active-exhaust 4U shelves; sustained MLX load held within 15-20% of peak GPU clocks.

Node Type	Count	OS	Memory	Network
NVIDIA DGX Spark	8	Debian 13	128GB LPDDR5x (273 GB/s)	100GbE ConnectX-7 / RoCE v2
Mac Studio	4	macOS	192GB UMA (819 GB/s)	10GbE SFP+ / Thunderbolt (JACCL)

Table 1: Node configuration

How the Serverless Layer Works

Serverless in this cluster means: no external compute provider. A caller submits a unit of work (agent task, inference request, batch job) and the cluster routes it to a pre-warmed local service. There is no cloud provisioning step and no per-request GPU rental.

Our project sits on top of standard Debian 13 on the DGX nodes: the OS image, systemd units, and network fabric are conventional; routing, scheduling, and service selection are handled by a **Master AI**, a Nemotron 3 Ultra running as an 8-way tensor-parallel vLLM endpoint on Spark-1 (coordinator) + Sparks 2-8 (workers).

Request path

1. Agent or CLI submits a task to the Mac-side MCP proxy (TCP / Protocol Buffers).
2. MCP proxy forwards the task envelope to the Master AI endpoint on Spark-1 (`/v1/master/route`).
3. Master AI (Nemotron 3 Ultra) classifies the task and returns structured routing JSON: target model, target node(s), quantization profile, and priority queue.
4. Debian systemd dispatches to the matching pre-warmed `vllm@.service` unit or Ray worker pool.
5. Subsidiary model executes (Gemma 4 31B-Instruct, Kimi K2.7 on Mac via cross-fabric callback, Gemma 4 Diffusion pipeline, etc.).
6. Result returns through MCP proxy to the caller. Master AI logs routing decision to `/var/log/cluster/master-ai/route.log`.

What Master AI manages

- **Model selection:** code/repository work → Kimi K2.7 (Mac Exo); batch prefill / PR verification → Nemotron worker pool; diagnostics / MCP tool-calls → Gemma 4 31B-Instruct; schematic generation → Gemma 4 Diffusion pipeline.
- **Queue depth:** reads vLLM queue metrics from Prometheus; throttles incoming Ray jobs when Spark GPU memory bandwidth exceeds 85% sustained.
- **Service health:** on node failure, Master AI re-issues routing JSON to shift TP shard group or fall back to single-node Gemma on Mac.
- **Config state:** cluster topology and LoRA adapter paths read from `/opt/cluster/config/cluster.json`; Master AI does not edit systemd unit files directly - it emits routing directives consumed by the MCP scheduler daemon (`cluster-scheduler.service`).

Debian provisioning (identical image on all 8 Sparks)

- PXE boot + Ansible playbook (`playbooks/dgx-spark.yml`) installs base Debian 13, CUDA 13, Docker 28.x, vLLM 0.9.x, Ray 2.4.
- Model weights at `/opt/models/`; shared LoRA adapters NFS-exported from Spark-1.
- Per-model systemd units: `vllm@.service`, `vllm-nemotron@.service`, `cluster-scheduler.service`, `ray-head.service` (Spark-1 only).
- RoCE v2 tuned: `sysctl net.core.rmem_max=134217728`; ConnectX-7 firmware 28.41.
- Monitoring: node_exporter + Prometheus scrape for GPU temp, memory bandwidth, vLLM queue depth, Master AI routing latency.

Model	Vendor	Architecture	Total / Active Params	Context / Modality
NVIDIA Nemotron 3 Ultra	NVIDIA	Sparse MoE	550B / 55B	128k text - **Master AI**
Gemma 4 31B-Instruct	Google DeepMind	Dense Transformer	31B / 31B	128k text
Gemma 4 Diffusion	Google DeepMind	Latent Diffusion (UNet + VAE)	8.2B	1024×1024 image
Kimi K2.7 Code	Moonshot AI	Sparse MoE	1T / 32B	256k text

Table 2: Deployed open-weights models (mid-2026). Nemotron 3 Ultra is the Master AI control plane; other models are subsidiary inference endpoints.

NVIDIA Nemotron 3 Ultra - Master AI Control Plane

Role: cluster controller. Receives task envelopes from MCP proxy, classifies workload type, emits routing JSON, and serves its own inference for batch prefill and PR verification jobs.

Platform: 8x DGX Sparks, tensor-parallel (TP=8) over RoCE v2. vLLM coordinator on Spark-1; workers on Sparks 2-8. Endpoint exposed at `http://spark-1.local:8000/v1/master/route`.

Tuning pipeline:

- Base checkpoint: `nvidia/nemotron-3-ultra` (550B MoE, 55B active).
- NVFP4 post-training quantization with per-expert calibration on 4,000 agentic routing traces: task classification labels, model-selection examples, queue-throttle decisions, and failover routing pairs.
- Additional fine-tune (rank 32 LoRA, 2 epochs) on 3,200 structured routing JSON examples derived from 6 weeks of manual cluster scheduling logs.
- Tensor-parallel shard: 8-way across RoCE; expert parallelism disabled (TP-only for consistent latency on 100GbE).
- Continuous batching: max 32 concurrent sequences; prefill chunking at 2048 tokens.
- Debian unit: `vllm-nemotron@.service` with NUMA pinning on Grace CPU cores per node.

		8k Context		32k Context		128k Context	
Platform	**Quant.**	**PP tok/s**	**TG tok/s**	**PP tok/s**	**TG tok/s**	**PP tok/s**	**TG tok/s**
8x DGX Sparks (vLLM TP=8)	NVFP4	5000	55	3800	49	2100	40

Table 3a: Nemotron 3 Ultra (Master AI) - inference throughput at batch size 1

Gemma 4 31B-Instruct - Subsidiary Model (Diagnostics & MCP)

Role: instruction following, RSS summarization, telemetry classification, MCP tool-calling. Routed by Master AI for diagnostic and lightweight agent tasks.

Platform: interactive decode on 1x Mac Studio (MLX INT4); batch prefill on 1x DGX Spark (vLLM NVFP4).

Tuning pipeline:

- Base checkpoint: `google/gemma-4-31b-it` (BF16).
- QLoRA fine-tune (rank 64, α=128) on 14,200 instruction pairs: mySpecSheet CAN/OBD-II diagnostic schemas, engineering log formats, MCP function-call traces; 3 epochs, lr=2e-4, cosine decay.
- Merged adapter weights; exported to MLX 4-bit (`Q4_K_M`) for Mac and NVFP4-calibrated TRT-LLM checkpoint for DGX Spark.
- Chat template extended with `<tool_call>` / `<tool_result>` tokens aligned to local MCP server schemas.

		8k Context		32k Context		128k Context	
Platform	**Quant.**	**PP tok/s**	**TG tok/s**	**PP tok/s**	**TG tok/s**	**PP tok/s**	**TG tok/s**
1x Mac Studio (MLX)	INT4	720	36	600	31	360	24
1x Mac Studio (MLX)	BF16	470	13	390	12	230	10
1x DGX Spark (vLLM)	NVFP4	1450	16	1140	15	660	13

Table 3b: Gemma 4 31B-Instruct - prefill (PP) and decode (TG) throughput, batch size 1

Gemma 4 Diffusion - Subsidiary Model (Schematic Generation)

Role: engineering schematic and wireframe generation. Routed by Master AI for image-generation tasks.

Platform: 2x DGX Sparks (Debian, CUDA 13, Diffusers + TensorRT) in pipeline-parallel pair: Spark-1 runs UNet denoising; Spark-2 runs VAE decode. Mac Studios handle prompt conditioning only.

Tuning pipeline:

- Base checkpoint: `google/gemma-4-diffusion-8b`.
- LoRA (rank 32) on 8,400 paired images: automotive wiring diagrams, ECU pinout schematics, parking-lot wireframes from PinPark simulation exports.
- 4-step distilled scheduler (consistency distillation from 20-step teacher); NVFP4 UNet weights for GB10 tensor cores.
- Negative-prompt library at `/opt/cluster/lora/gemma4-diffusion/neg_prompts.json`.

Resolution	**Steps**	**Platform**	**Wall Time / Image**	**Throughput**
1024×1024	4 (distilled)	2x DGX Spark (pipeline)	4.8 s	12.5 images/min
1024×1024	20 (full)	1x DGX Spark	18.2 s	3.3 images/min
512×512	4 (distilled)	1x DGX Spark	1.6 s	37.5 images/min

Table 3c: Gemma 4 Diffusion - image generation throughput (custom LoRA, NVFP4 UNet)

Kimi K2.7 Code - Subsidiary Model (Long-Context Code)

Role: multi-file repository analysis and code reasoning over 256k-token windows. Routed by Master AI for code-heavy agent tasks; executes on Mac side (context exceeds single DGX 128GB without paging).

Platform: 4x Mac Studios sharded via Exo + JACCL over Thunderbolt; INT4 weights across 768GB pooled UMA.

Tuning pipeline:

- Base checkpoint: `moonshotai/kimi-k2.7-code` (1T MoE, 32B active).
- Expert-routing calibration: 6,800 code-heavy samples (C#, Python, shell, CAN parser DSL); top-2 expert bias toward code layers; routing temperature 0.3.

- Context extension to 256k via YaRN scaling factor 2.0; KV-cache dtype FP8 to fit pooled memory.
- Tool-call format aligned to OpenShell slash-command registry and MCP `tools/list` schema; 2,100 synthetic multi-turn traces.
- `sudo sysctl iogpu.wired_limit_mb=155000` on each Mac Studio.

		8k Context		32k Context		128k Context	
Platform	**Quant.**	**PP tok/s**	**TG tok/s**	**PP tok/s**	**TG tok/s**	**PP tok/s**	**TG tok/s**
4x Mac Studios (Exo / JACCL)	INT4	610	34	430	30	190	22

Table 3d: Kimi K2.7 Code - throughput (256k native; 128k benchmarked subset)

Inference Comparison

Decode is memory-bandwidth-bound (Mac at 819 GB/s); prefill is compute-bound (DGX at 1 PFLOP NVFP4). Per-model tables above; summary charts below.

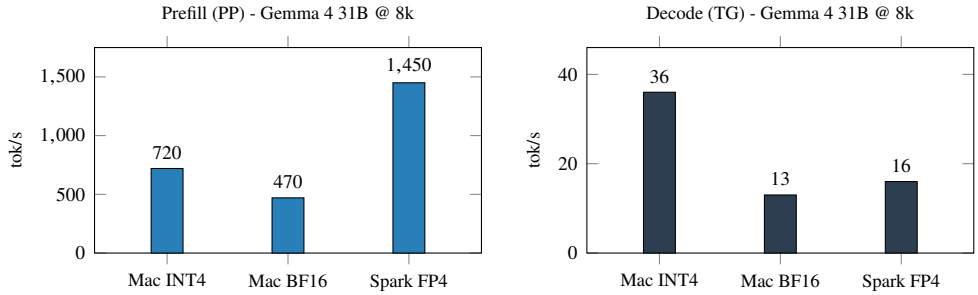

Figure 1: Gemma 4 31B-Instruct prefill/decode split at 8k context

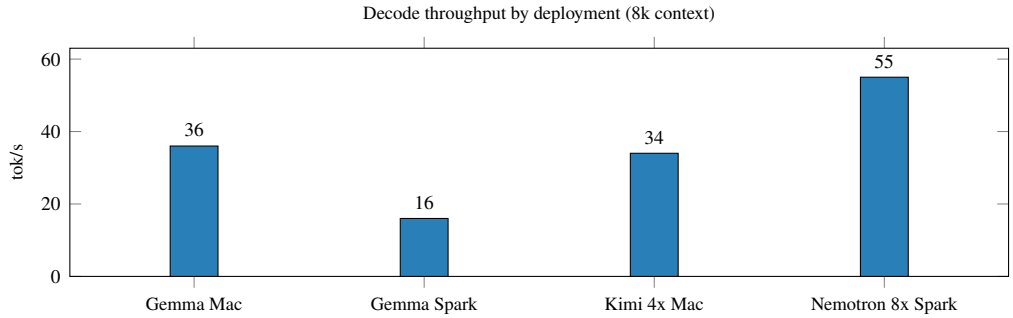

Figure 2: Decode throughput across text-model deployments

Fabric Latency & Memory Bandwidth

Link	**Environment**	**Peak Bandwidth**	**Avg. RTT**
DGX ↔ DGX (RoCE v2)	100GbE ConnectX-7 switch	94.2 Gbps	12 μs
Mac ↔ Mac (JACCL)	Thunderbolt daisy-chain	38.5 Gbps	45 μs
MCP proxy → Master AI	TCP / Protocol Buffers (Spark-1)	8.9 Gbps	0.54 ms
Mac UMA bandwidth ceiling	On-package LPDDR5x	819 GB/s	-
DGX memory bandwidth ceiling	On-package LPDDR5x	273 GB/s	-

Table 4: Interconnect measurements

End-to-End Task Latencies

Task	**Stack**	**Throughput**	**Avg. Duration**
Full-project codebase indexing	LlamaIndex / Qdrant	120 files/sec	42.5 s
RSS scraping & vectorization	Gemma 4 31B / Ray	12 feeds/min	2.1 s / feed
Automated PR run & test verification	NemoClaw / OpenShell / Docker	0.81 PRs/min	118.0 s
OBD-II telemetry classification	Custom classifier / SQLite	850 sessions/sec	0.12 s / session
Schematic generation (batch)	Gemma 4 Diffusion / 2x DGX	12.5 images/min	4.8 s / image
Master AI routing decision	Nemotron 3 Ultra / MCP proxy	42 routes/sec	24 ms / route

Table 5: Measured task latencies on the cluster