Hypha Gemma 3 27B BF16 Distributed Capacity Demonstration

Verified two node pipeline parallel inference across an RTX A6000 and RTX 3060

Hypha executed the original 51.10 GiB Gemma 3 27B checkpoint across two GPUs even though the checkpoint exceeds the VRAM of either card independently. The final run preserved BF16 weights, disabled CPU offload, used a verified 53 / 9 layer partition, generated a natural stop, and completed clean Hypha lifecycle recovery.

Proof condition	Verified result
Checkpoint	google/gemma-3-27b-it at pinned revision
Precision	BF16 with no quantization
CPU offload	0 GiB
Partition	53 layers on A6000 and 9 layers on RTX 3060
Generation	241 completion tokens with finish reason stop
Throughput	5.05 completion tokens per second

Hypha Compute

September 14 2026
Version 0.1

Executive Summary

The demonstration proved a specific capacity claim: Hypha can execute the original Google Gemma 3 27B instruction checkpoint in BF16 across an RTX A6000 and an RTX 3060 without CPU offload. The checkpoint contains 54,864,813,280 tensor bytes, or 51.096839 GiB. This exceeds the usable VRAM capacity of the 48 GB class A6000 and is more than four times the 12 GB RTX 3060 capacity.

The final operation used two-node pipeline parallelism with 53 transformer layers on the A6000 and 9 layers on the RTX 3060. It generated 241 completion tokens in 47.722 seconds and returned finish reason stop. End-to-end throughput was 5.05 completion tokens per second. Estimated mean board power during inference was 169.96 W on the A6000 and 55.49 W on the RTX 3060, for 225.45 W combined.

The immediately preceding run generated 1,862 completion tokens at 4.97 tokens per second, which provides a longer sustained measurement. It ended at the configured context boundary rather than with a natural stop. The final run therefore supplies the clean completion proof, while the longer run supplies the more sustained throughput and power sample.

Demonstration Objective

The objective was to run the unquantized original checkpoint across two dissimilar consumer and workstation GPUs while preserving BF16 tensor precision and setting CPU offload to zero. The demonstration also had to record the layer partition, inference throughput, GPU memory use, sampled board power, Hypha job outcomes, cleanup results, and lease audit state.

- The model checkpoint had to exceed either GPU's individual VRAM.
- Both GPU stages had to participate in one pipeline-parallel vLLM runtime.
- The final response had to complete with finish reason stop without an explicit generation-token cap.
- The remote worker had to be released through a Hypha cleanup operation, with both heavy-inference locks free afterward.

System Under Test

Component	A6000 stage	RTX 3060 stage
Host	ubuntu at 10.11.211.153	hp840 at 10.11.241.52
Hypha node	node-4678f275-77e3-44d2-b761-98c22d7e047a	node-7c4db209-4bf6-48d4-b4aa-0157f7b9d303
GPU	NVIDIA RTX A6000	NVIDIA GeForce RTX 3060
GPU UUID	GPU-fc102ba5-6b3a-376b-783e-f95f6f680f48	GPU-5443c72e-1f22-a088-979e-5ade65c318ad
Reported VRAM	49,140 MiB	12,288 MiB
Model path	/home/ubuntu/hypha-models/gemma-3-27b-it	/home/maysjack/hypha-models/gemma-3-27b-it

The Quadro M6000 installed in the A6000 host was deliberately excluded. Post-run telemetry showed it at 9 MiB used, confirming that it did not participate.

Checkpoint Integrity

The model was downloaded from google/gemma-3-27b-it at revision 005ad3404e59d6023443cb575daa05336842228a. The staged model contained 24 verified files. The safetensors index referenced 12 weight shards, and every tensor header was read before execution.

Checkpoint property	Measured value
Tensor data type	BF16 for all indexed tensors
Transformer layers	62
Tensor bytes	54,864,813,280 bytes
Tensor size	51.096839 GiB
Total staged file bytes	54,904,366,994 bytes
Weight shards	12 safetensors files

Memory Placement

The final partition assigned layers 0 through 52 to the A6000 stage and the remaining 9 transformer layers to the RTX 3060 stage. Embedding, normalization, multimodal, runtime, cache, and communication allocations add overhead beyond the raw layer weights. The table compares calculated BF16 weight estimates with vLLM's observed model-loading values.

Stage	Layers	Calculated BF16 weights	Observed vLLM model load
RTX A6000	53	44.175 GiB	44.6 GiB
RTX 3060	9	10.335 GiB	10.62 GiB

The final proof reached 48,321 MiB on the A6000 and 11,740 MiB on the RTX 3060. These peaks include model weights, cache, activations, CUDA context, and distributed runtime allocations. They are not pure weight measurements.

Runtime Configuration

Setting	Final value
vLLM version	0.29.0
Distributed executor	mp
Pipeline parallel size	2
Tensor parallel size	1
Layer partition	53,9

Setting	Final value
Data type	bfloat16
Quantization	none
CPU offload	0 GiB
Maximum model length	2,048 tokens
Explicit generation cap	none
Maximum sequences	1
Maximum batched tokens	512
Execution mode	eager
A6000 memory utilization target	0.97
RTX 3060 memory utilization target	0.95
Allocator	expandable segments enabled

Hypha Execution Method

The final operation was derived from the already proven llm.distributed.qwen3_coder_30b.pp40_8.v1 source rather than introducing a new transport or wrapper. The operation retained the established two-node vLLM multiprocess pattern, explicit GPU UUID selection, offline model paths, pipeline partition environment variable, local head API, and remote headless worker.

Hypha installed the checksum-verified operation independently on both pinned nodes. It submitted the RTX 3060 worker first and then the A6000 head. The head served the request on localhost, measured the generation interval, and returned the response and runtime measurements through the Hypha job result.

After the head completed, the driver submitted the verified Gemma cleanup operation to the HP840 before waiting for the worker job. This ordering reflects the observed distributed lifecycle: the head stops its server after the response, while the remote worker can remain alive until explicitly released.

Compatibility Adaptation

vLLM 0.29.0 initially allocated a full vocabulary language-model head before tying that head to the embedding tensor. On this checkpoint, the transient duplicate allocation exceeded the A6000's loading headroom. The Hypha operation applied a checksum-guarded compatibility adaptation that constructed the temporary tied head on the meta device and then tied it to the existing embedding weight.

The adaptation did not modify the checkpoint, tensor values, tensor data type, layer count, or CPU-offload setting. It changed only the transient construction path. The operation registered restoration logic and returned the installed vLLM source to its original checksum when the inference process exited.

The final operation also set PYTORCH_CUDA_ALLOC_CONF to expandable_segments:True. This addressed a measured 20 MiB fragmented allocation failure on the RTX 3060. It is a CUDA allocator behavior setting and does not move weights to CPU memory.

Final Proof Results

Metric	Final proof	Sustained run
Finish reason	stop	length at context boundary
Prompt tokens	72	186
Completion tokens	241	1,862
Inference time	47.722 s	374.451 s
End-to-end throughput	5.05 tok/s	4.97 tok/s
A6000 peak VRAM	48,321 MiB	48,326 MiB
RTX 3060 peak VRAM	11,740 MiB	11,755 MiB
A6000 mean board power	169.96 W	184.82 W
RTX 3060 mean board power	55.49 W	57.09 W
Combined mean board power	225.45 W	241.91 W
Combined inference energy	10,758.95 J	90,584.19 J
Energy per completion token	44.643 J	48.6489 J

Board-power estimates use telemetry arrival timestamps recorded on the Ubuntu head. Values between samples were linearly interpolated, with the nearest endpoint held at the measured inference boundaries. The final proof included 24 A6000 samples and 23 RTX 3060 samples during inference.

Lifecycle Verification

The final head job returned ok=true, finish reason stop, a terminal record, no active lease, and a clean lease audit. The worker returned ok=true with exit status 0 after the requested cleanup. Both cleanup jobs returned ok=true and confirmed that the heavy-inference lock was free.

Lifecycle check	Result
Head result	Completed with ok=true
Head lease audit	Clean with no active lease
HP840 worker release	TERM requested through Hypha cleanup
Worker result	ok=true and exit_status=0
Worker lease audit	Clean with no active lease

Lifecycle check	Result
A6000 lock	Free
HP840 lock	Free
Forced KILL	Not used

What This Demonstration Proves

The result proves that this exact 51.10 GiB BF16 checkpoint can execute on this two-node Hypha configuration even though neither GPU can hold the checkpoint independently. It also proves that Hypha can stage the model, install the derived operation, coordinate both pipeline stages, collect throughput and board-power telemetry, return a natural completion, release the remote worker, and close both leases cleanly.

What This Demonstration Does Not Prove

This is a capacity demonstration on one model revision, one vLLM version, one layer partition, and one hardware pair. It does not establish linear scaling, lower cost, lower energy than a single larger GPU, production service-level availability, or the best possible partition. The short proof run is sufficient for the stop condition but is not a statistical performance study. The longer run offers a more sustained sample, yet both results should be repeated before reporting variance or comparative efficiency.

Conclusion

Hypha completed the requested distributed-capacity proof. The original Gemma 3 27B checkpoint ran in BF16 across the RTX A6000 and RTX 3060 with CPU offload set to zero. The verified 53 / 9 split produced a natural stop at 5.05 completion tokens per second and an estimated combined mean board power of 225.45 W. The checkpoint exceeds either card's individual VRAM, both cards participated, and the post-run cleanup and lease audits were clean.

Reproducibility Record

Record	Identifier
Checkpoint revision	005ad3404e59d6023443cb575daa05336842228a
Final Hypha operation	llm.distributed.gemma3_27b.bf16.pp53_9.proof.v1
Final operation SHA256	3e1cf5e1c88373817a8a740b466015d8a5d7c4134b2346af7d9fb5356791b4d7
Cleanup operation	hypha.recover.gemma3_27b_worker.v1
Cleanup SHA256	be5b2f44c25626fc253b589405f556cca3ca7ef4d8797a9da5b0826ec0f817fe
Receipt directory	/home/ubuntu/hypha-benchmarks/gemma27b-53-9-3842a43e54ab42be8f34ca00bf0f461a
Registry	http://10.11.228.211:8080 namespace default

HYPHA | Patent Pending

© 2026 TKT, Inc. All Rights Reserved.
HYPHA technology is the subject of pending patent applications. The contents of this white paper, including its text, illustrations, diagrams, and technical descriptions, are protected by applicable copyright laws. No portion may be reproduced, distributed, or transmitted without prior written permission from TKT, Inc.