HYPHA

Distributed Transformer Inference Across Commodity CPU Systems

A Two-Node and Three-Node Xeon Proof

Hypha Distributed Inference v0 | August 27, 2026

THREE_WAY_CPU_DISTRIBUTED_INFERENCE_PROVEN

HP840 layers 0-9 -> R740 layers 10-23 -> T7610 layers 24-27 + final norm + LM head produced token 12095 (' Paris'), exactly matching an unsplit HP840 CPU reference.

Core finding

A single transformer forward path was physically partitioned across separate networked CPU hosts, with intermediate hidden state transmitted between machines and the final next-token decision matching the unsplit reference.

Abstract

This paper documents Hypha distributed transformer inference v0 on commodity x86 CPU systems. The experiment partitions one Qwen2.5-7B-Instruct transformer across physical machines rather than sending independent prompts to separate models. Each stage executes a contiguous portion of the same model, passes the intermediate hidden state over the local network, and continues the same forward computation on the next host. The work progressed from a two-machine CPU split to a three-machine Xeon pipeline and preserved the next-token decision against an unsplit CPU reference.

The two-machine proof established HP840 CPU -> Dell R740 CPU execution with an exact first-token match. A subsequent 34-minute validation run produced 835 of 835 matching teacher-forced token decisions, with periodic 16-token free-running checks maintaining a full 16-token exact prefix. The three-machine proof then extended the same execution path across HP840, R740, and T7610. The final three-way result again produced the exact same next token as the unsplit HP840 reference. Measured network transit was small relative to compute time, while stage timings exposed an obvious load-imbalance target for optimization.

1. System Architecture

Hypha treats the participating machines as stages in one transformer execution graph. The model is not replicated for independent inference requests; instead, the transformer layer stack is divided and the hidden representation is transferred between stage boundaries.

Stage	Host	CPU	Transformer responsibility	Role
A	HP840	2 x Xeon E5-2699 v3, 36C/72T	Layers 0-9	Embedding + early transformer
B	Dell R740	2 x Xeon Gold 6230, 40C/80T	Layers 10-23	Middle transformer
C	Dell T7610	2 x Xeon E5-2637 v2, 8C/16T	Layers 24-27 + norm + LM head	Final transformer + token decision

Execution flow

HP840 0-9	hidden state over LAN	R740 10-23	hidden state over LAN	T7610 24-27 + head

2. Method

The CPU experiments used Qwen/Qwen2.5-7B-Instruct, a 28-layer transformer. Stages loaded the same model weights in float32 and replaced non-owned transformer layers with pass-through modules. Stage A generated the hidden representation through its assigned early layers, serialized that tensor, and sent it to the next host. Stage B continued the transformer computation over its assigned layer range and forwarded the resulting hidden state. Stage C completed the remaining transformer layers, final normalization, and language-model head to produce next-token logits.

- Two-node proof: HP840 executed layers 0-10; R740 executed layers 11-27 plus final norm and LM head.
- Three-node proof: HP840 executed layers 0-9; R740 executed layers 10-23; T7610 executed layers 24-27 plus final norm and LM head.
- Correctness reference: a full unsplit Qwen2.5-7B-Instruct forward pass on HP840 CPU using the same prompt.
- Primary prompt used for the milestone proof: "The capital of France is".
- Correctness criterion: distributed next-token ID must equal the unsplit reference next-token ID.

3. Two-Node CPU Proof

The first CPU milestone split the model across HP840 and R740. HP840 computed the initial transformer section and transmitted the intermediate hidden state to R740, which completed the model and produced the next token.

Metric	Observed result
Model	Qwen2.5-7B-Instruct, 28 transformer layers
Partition	HP840 0-10 -> R740 11-27 + norm + LM head
Prompt	The capital of France is
Prompt tokens	5
Hidden-state payload	73,761 bytes
HP840 Stage A	833.213 ms
R740 Stage B	948.477 ms
Network / HTTP	20.469 ms
Distributed total	1,802.158 ms
Distributed next token	12095 (' Paris')
Unsplit HP840 reference	1,137.109 ms
Reference next token	12095 (' Paris')
Exact match	YES

3.1 Extended CPU Validation

The two-node implementation was then exercised in a longer validation run. The run ended after approximately 34 minutes because the R740 network path became unreachable; the inference comparisons recorded before that network interruption remained exact.

Metric	Observed result
Observed duration	2,050.460 seconds / 34.174 minutes
Teacher-forced token decisions	835
Matches	835
Mismatches	0
Agreement	100.000000%
Free-running checks	2
Free-running exact prefix	16 / 16 tokens on both checks
Median Stage A	259.728 ms
Median Stage B	1,297.251 ms
Median network	6.457 ms
Median split total	1,524.446 ms
Median unsplit reference	739.357 ms
Median payload	202,913 bytes

4. Three-Node Xeon Proof

The next step added a third physical CPU host to the same model execution path. This changed the R740 from the terminal stage into a middle transformer stage and moved the final transformer layers, normalization, and LM head to the T7610.

Metric	Observed result
Partition	HP840 0-9 -> R740 10-23 -> T7610 24-27 + norm + LM head

Metric	Observed result
Prompt	The capital of France is
Prompt tokens	5
HP840 -> R740 payload	73,509 bytes
R740 -> T7610 payload	73,509 bytes
HP840 Stage A	562.317 ms
HP840 -> R740 network	16.971 ms
R740 Stage B	712.295 ms
R740 -> T7610 network	12.041 ms
T7610 Stage C	2,589.604 ms
Distributed total	3,893.229 ms
Distributed next token	12095 (' Paris')
Unsplit HP840 reference	970.222 ms
Reference next token	12095 (' Paris')
Exact match	YES

5. Findings

The CPU v0 experiments establish four practical results.

- One transformer can be physically partitioned across multiple independent CPU machines while preserving a single forward inference path.
- Intermediate hidden state can cross ordinary LAN links between transformer stage boundaries and continue computation on the next machine.
- The tested distributed executions preserved the next-token decision of the unsplit reference, including 835/835 teacher-forced decisions in the extended two-node run and the exact three-node milestone token.
- For the three-node run, compute time dominated transport time. The two measured LAN transitions totaled 29.012 ms, while the three compute stages totaled 3,864.216 ms.

5.1 The Current Bottleneck Is Placement

The three-node result immediately exposes the next engineering target. Stage C on the T7610 consumed 2,589.604 ms, far more than Stage A on HP840 or Stage B on R740. The network links were comparatively inexpensive. The current partition therefore proves the architecture while also showing that equal or intuitive layer counts are not the correct scheduling policy. Hypha can improve performance by assigning layers according to measured stage service time and host capability rather than by machine count alone.

6. Why This Matters

Conventional local inference is constrained by the memory, compute, and accelerator capacity of one machine. Hypha v0 demonstrates another execution model: treat a collection of existing systems as parts of a single inference path. A transformer stage can execute where capacity exists, transfer its hidden representation, and continue elsewhere. The machines do not need to be identical, and the three-node proof used three different Xeon generations and core counts.

This creates a foundation for a larger reasoning and inference fabric in which CPUs, GPUs, and specialized accelerators can be scheduled as stage resources rather than isolated boxes. The CPU work is especially useful because it establishes the mechanics independently of GPU-specific interconnects or vendor-specific multi-GPU runtimes.

7. Optimization Phase

The proof phase is complete enough to move directly into optimization. The next work is about making the demonstrated execution path faster, broader, and more dynamic.

- Rebalance transformer layer boundaries using measured per-stage latency.
- Spread the same model across additional Hypha CPU nodes and measure the point where network and serialization costs begin to dominate.
- Replace full-tensor JSON-style transport with more efficient binary tensor transport and persistent connections.
- Introduce stage-aware KV-cache handling so autoregressive generation does not recompute the complete prefix at every step.
- Allow Hypha to select stage placement dynamically from observed host capability and measured service time.
- Mix CPU, GPU, and specialized accelerator stages where each resource provides the best stage economics.
- Add stage failure and reassignment behavior so the inference path can recover through the fabric.

8. Conclusion

Hypha distributed CPU inference v0 moved from a two-machine split to a three-machine Xeon transformer pipeline. In the three-node proof, HP840 executed layers 0-9, R740 executed layers 10-23, and T7610 executed layers 24-27 plus final normalization and the LM head. The distributed path produced token 12095, " Paris", exactly matching the unsplit HP840 reference.

The extended two-node validation also recorded 835 matching teacher-forced token decisions with no mismatches before a network interruption ended the run. Together, these results demonstrate that Hypha can carry one transformer computation across separate commodity CPU hosts while preserving the tested token

decisions. The next phase is optimization: better placement, faster transport, wider distribution, cache-aware generation, and mixed-resource scheduling.

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.

Appendix A. Reproducible Milestone Output

```
=== HYPHA THREE-MACHINE CPU DISTRIBUTED INFERENCE ===
HP840  CPU layers 0-9
R740   CPU layers 10-23
T7610  CPU layers 24-27 + norm + lm_head

=== THREE-MACHINE SPLIT RESULT ===
prompt='The capital of France is'
prompt_tokens=5
AB_payload_bytes=73509
BC_payload_bytes=73509
stage_a_hp840_ms=562.317
network_hp840_to_r740_ms=16.971
stage_b_r740_ms=712.295
network_r740_to_t7610_ms=12.041
stage_c_t7610_ms=2589.604
split_total_ms=3893.229
split_next_token_id=12095
split_next_token=' Paris'

=== CORRECTNESS ===
reference_ms=970.222
reference_token_id=12095
reference_token=' Paris'
EXACT_MATCH=True

THREE_WAY_CPU_DISTRIBUTED_INFERENCE_PROVEN
```