vExpert: Virtualizing Expert Storage for Adaptive Load Balancing in Distributed MoE Inference

Wenxun Wang, Xiuhong Li, Yida Wang, Chen Tang, Zongle Huang, Ke Hong, Yu Wang and Yongpan Liu

Department of Electronic Engineering, Tsinghua University
Beijing National Research Center for Information Science and Technology

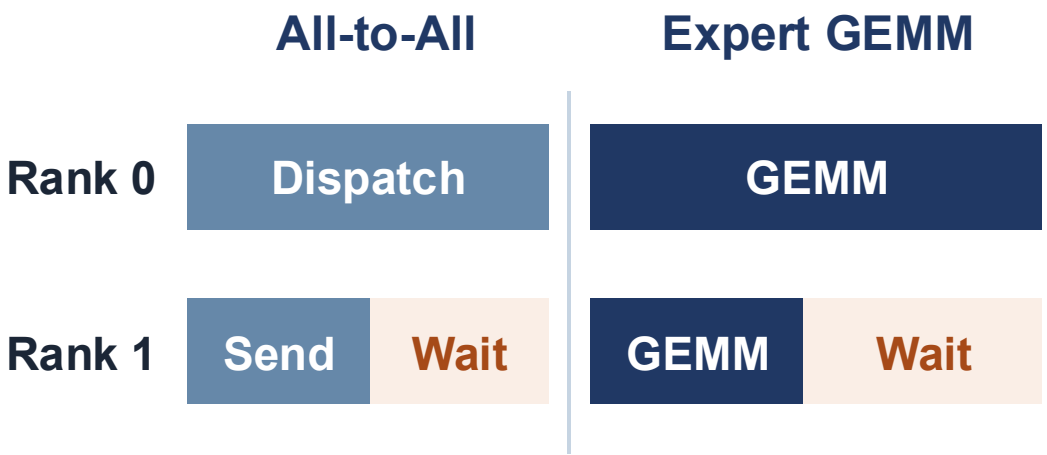
NeurIPS 2026



Routing Skew Delays All-to-All and Expert GEMM

The busiest rank sets the pace of each decode layer

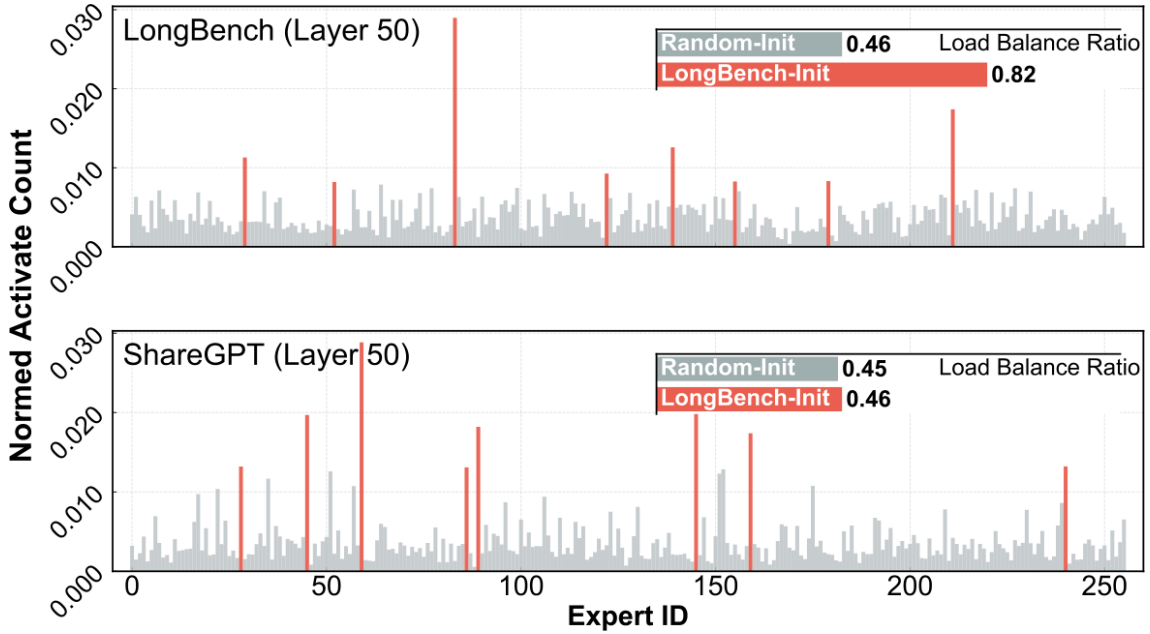
Where ranks wait



Illustrative timing for two ranks

The hot rank exchanges more tokens and runs a larger expert GEMM.

How the same placement transfers



Load balance ratio **0.82 → 0.46**

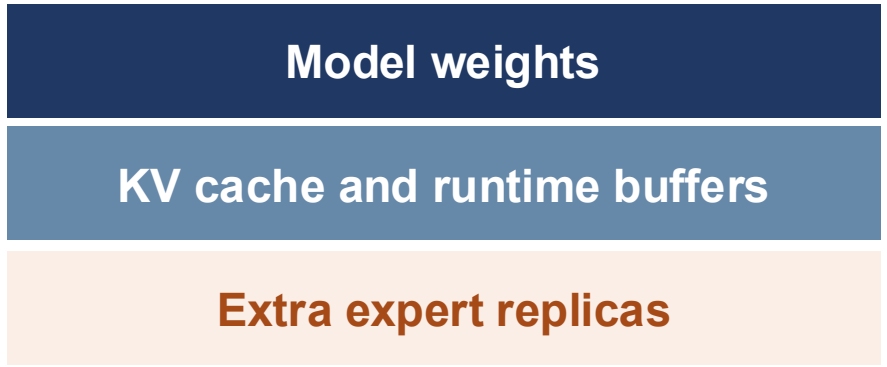
Adapting to Dynamic Load Balance for Efficient MoE decoding

HBM Limits the Number of Resident Expert Replicas

Expert Parallel Load Balancing (EPLB) splits a hot expert's tokens across its replicas.

Per-GPU HBM budget

HBM allocation



1–2 **extra experts per GPU**

in the typical deployments

When the hot expert changes

	Iteration A	Iteration B
Hot expert	**E2**	**E6**
Resident replica	**E2**	**E2**
Can share load?	**Yes**	**No**

E2 still occupies the replica slot when E6 becomes hot.

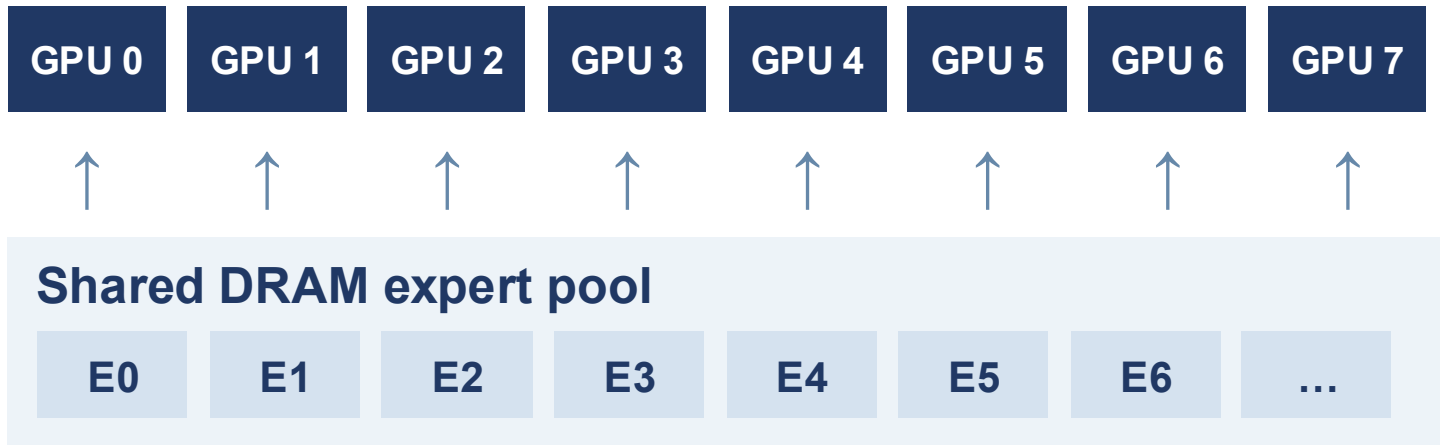
GPU-resident EPLB incurs Memory Overhead and Frequent Rebalancing Stall

A Node Can Hold the Full Expert Pool in DRAM

All eight GPUs can fetch expert weights from one shared host pool.

Node-local Redundant Expert Storage

GPU HBM



Any local GPU can request a replica from the pool.

Eight-GPU node

1.8 TB

Host DRAM per node

8 × 64 GB/s

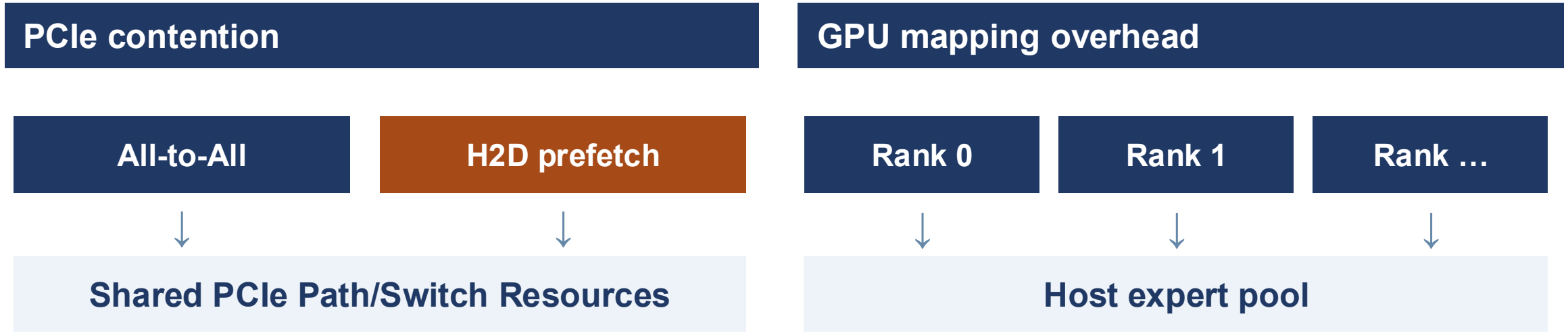
Eight PCIe Gen5 ×16 links

Nominal one-way H2D bandwidth

A shared DRAM Expert Pool Enables Dynamic Replica Placement

The Cost of On-Demand Expert Replication

Prefetch traffic competes with All-to-All, and each GPU process needs access to the host pool.



Decode throughput with naive offloading

542 → 497 tok/s

Each process maps the expert weights.

Page tables consume HBM on each GPU.

Shared-memory quotas also limit pool size.

H2D contend with A2A for PCIe, Per-Process Mappings incur Memory Overhead

vExpert Virtualizes Expert Storage

A GPU can serve different logical experts through the same physical HBM slot.

Shared expert pool in host DRAM

E0	E1	E2	E3
E4	E5	E6	E7

All eight weights stay available

for on-demand H2D prefetch.

One reusable slot in GPU 0 HBM

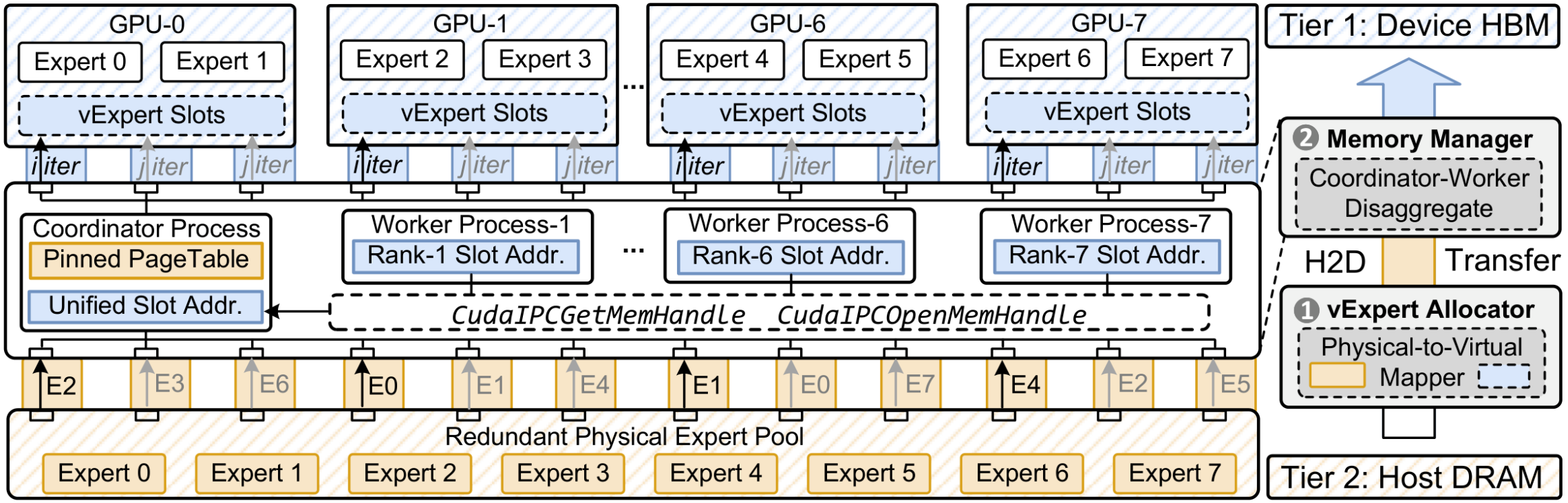
Placement	Resident experts	Replica slot
Batch A	E0 E1	E2
Batch B	E0 E1	E6

The slot address stays fixed as its weights change.

Decouple Logical Visibility from Physical Residency for Expert Weights

Two-Tier Expert Storage in HBM and DRAM

The allocator chooses where a replica runs. The memory manager delivers its weights.



Cluster-wide allocator
Selects the expert, target rank and HBM slot.

Node-local memory manager
Prefetches weights from DRAM into that slot.

Allocator for Placement Mapping, Manager for Storage and Transfers

vExpert ·

Allocator: The Latency Trade-off in Expert Prefetching

Timeline · layer ℓ

No prefetch



With prefetch



Extra A2A time from PCIe contention

Allocator performance model

Benefit in layer ℓ + 1

Busiest rank handles fewer tokens:
less A2A traffic and less GEMM work.

Cost in layer ℓ

H2D lowers effective A2A bandwidth.

Choose replicas to minimize latency

Include zero replicas as a candidate.

H2D transfers balances workloads, PCIe Contention limits A2A Bandwidth

vExpert

Allocator: Expert Placement Updates Reuse the CUDA Graph

Recent routing counts drive placement updates at batch boundaries.

Cluster-wide allocation

Recent expert loads → **GPU allocator · ≤10 ms** → New expert-to-slot map

Memcpy node

Update the source pointer

Host weights: E2 → E6

Destination: same GPU slot address

Routing metadata

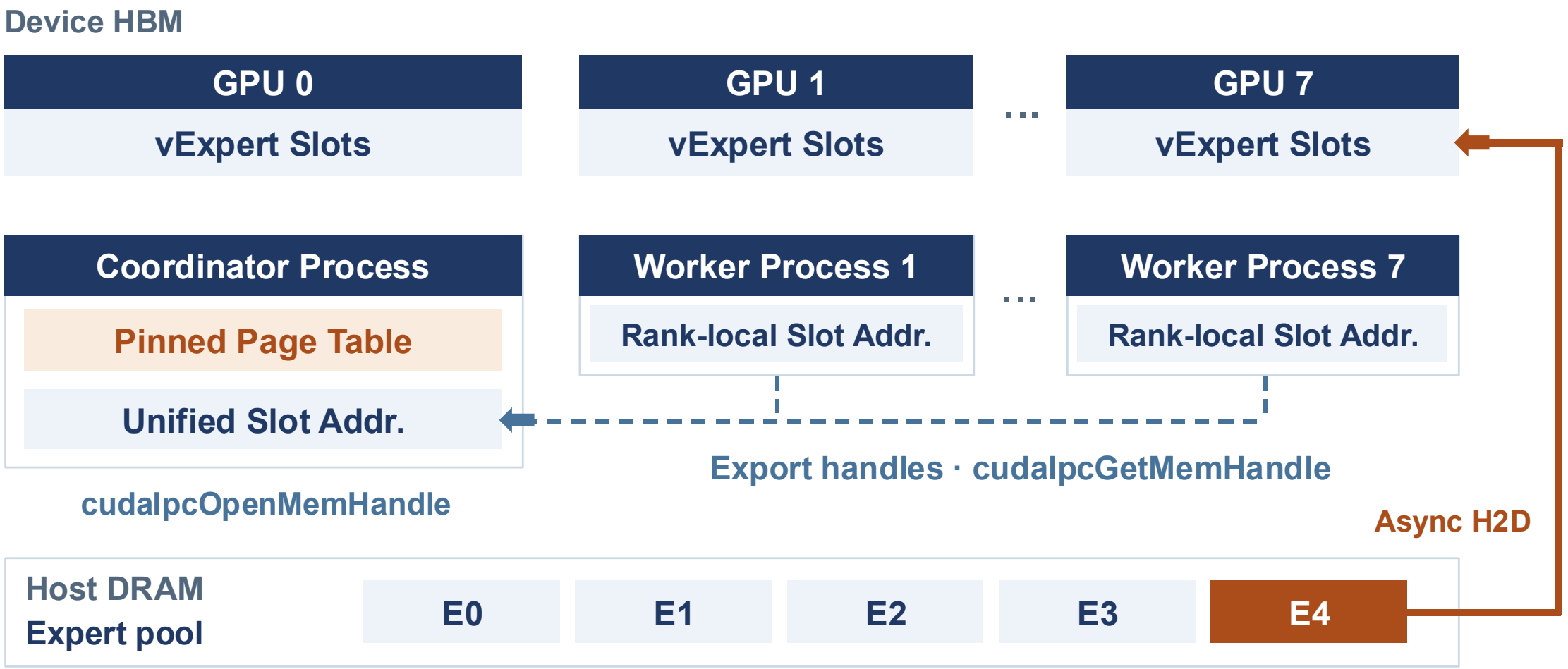
Update the map in place

Write the new expert-to-rank mapping.

Replay the graph using the updated map.

Update every 10 iters while Reusing Graphs, only 1.5% end-to-end Overhead

The Coordinator Uses CUDA IPC to Prefetch Expert Weights



Workers export local Handles, Coordinator manages Mappings and Transfers

vExpert

Decode Throughput Gains at EP32 and EP64

DeepSeek-V3.1-671B · H100 cluster · batch size 256

EP32

+11%

End-to-end decode throughput

Load balance ratio ↑



EP64

+20%

End-to-end decode throughput

Load balance ratio ↑



Baseline: SGLang EPLB with one resident expert replica per GPU.

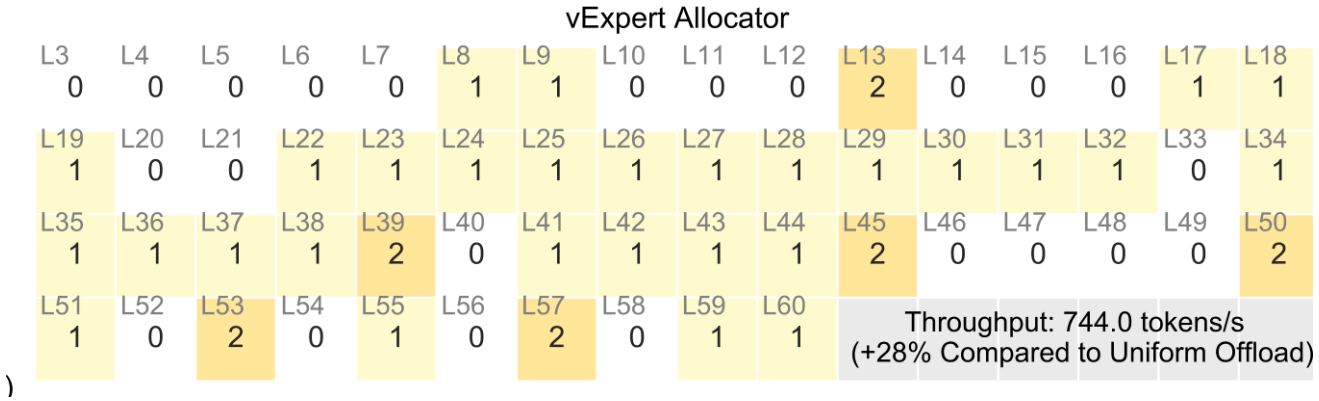
vExpert improves throughput over EPLB by 11% at EP32 and 20% at EP64

vExpert ·

Layer-Wise Prefetch Beats Uniform Prefetch

DeepSeek-V3.1 · EP64 · batch size 256 · 58 MoE layers in one decode iteration

Slot allocation across layers



Layers 3–60 · Each cell shows its slot count.

Load balance by slot count

Slots	Layers	Before	After
0	23	0.697	**0.697**
1	29	0.438	**0.763**
2	6	0.347	**0.768**

744 tok/s with layer-wise prefetch

Uniform baseline: one replica per layer.

41 slots to 35 skewed layers, balancing load-balancing gains against contention

vExpert ·

vExpert Runs a 1.3 TB Pool with a 900 GB Shared-Memory Quota

DeepSeek-V3.1 · 256 experts · two NUMA domains per node

Expert pool per node	Host DRAM	Without manager	With manager
One shared copy	646 GB	Runs	Runs
Two NUMA-local copies	**1.3 TB**	**OOM**	**Runs**

The coordinator registers the host pool. Workers share GPU slot handles via CUDA IPC.

GPU Memory Consumption per node

One copy **11.9 GB → 1.5 GB**

Two copies **18.4 GB → 2.3 GB**

Manager avoids Shared-memory Limits and reduces GPU Memory Overhead