



# MoE Expert Parallelism Sizing: Traffic, Capacity, Fabric

Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-10-27 · moe expert parallelism sizing · expert parallel size · moe capacity factor · moe all-to-all bandwidth · deeppep · hybridgep · megatron core moe · moe token dropping · nvlake infiniband moe · moe inference expert parallelism

Original article: <https://gpusmith.com/articles/en/moe-expert-parallelism-sizing-traffic-capacity-fabric>



## In this article

1. Executive Summary
  2. Introduction and Background
  3. Key Changes in How MoE Parallelism Is Sized
  4. Traffic and Capacity Worksheets
  5. Fabric Placement: Scale-Up Versus Scale-Out
  6. Dispatcher Candidates and Support Matrix
  7. Implementation Considerations and Process Changes
  8. Data Analysis and Evidence
  9. Implications and Future Directions
  10. Frequently Asked Questions (FAQs)
  11. Conclusion
- 

## Executive Summary

Sizing expert parallelism (EP) for a mixture-of-experts (MoE) model is a placement and traffic problem before it is a throughput problem. EP shards the experts across GPUs, so every MoE layer performs a dispatch all-to-all that sends token vectors to the ranks holding their selected experts and a combine all-to-all that returns the results. Megatron Core recommends the alltoall dispatcher when EP is applied ([docs.nvidia.com](https://docs.nvidia.com)). The number of experts must divide evenly by the EP size ([docs.nvidia.com](https://docs.nvidia.com)), and total parallelism must divide the world size.

The first-order planning arithmetic is short. Per-expert capacity in current Megatron documentation is tokens per rank times top-k times capacity factor divided by the number of experts ([docs.nvidia.com](https://docs.nvidia.com)). Dispatch bytes per layer and per direction are assigned token vectors times hidden size times bytes per element. For a DeepSeek-V3-shaped layer (hidden size 7168, top-8 routing, 256 routed experts), a rank holding 4,096 tokens produces 32,768 assignments and about 470 MB per direction in BF16, before any deduplication, padding or metadata. These are planning hypotheses, and runtime kernels such as DeepEP and HybridEP change what actually crosses each link ([github.com](https://github.com)).

Placement is where the decision gets made. Megatron guidance keeps EP and TP inside the NVLink domain because NVLink offers much higher bandwidth than cross-node interconnects ([github.com](https://github.com)), yet the same guidance warns that very large MoE models can exceed NVLink bandwidth with EP traffic alone ([github.com](https://github.com)). Published all-to-all shares of step time range from 16% to 36% in an early GShard study ([arxiv.org](https://arxiv.org)), to 30-40% in Megatron guidance ([github.com](https://github.com)), to "up to 60%" in a 2026 Megatron-Core report ([arxiv.org](https://arxiv.org)). That spread is the reason no vendor figure should stand in for a measurement on the target topology.

Dispatcher choice is version and hardware specific. The README lists allgather, alltoall and flex dispatchers, and HybridEP is documented as targeting GB200 NVL72 and multi-node NVLink ([github.com](https://github.com)), while DeepEP is listed for Ampere, Hopper, B200 and B300 GPUs ([docs.nvidia.com](https://docs.nvidia.com)). Capacity-factor token dropping needs the alltoall dispatcher ([docs.nvidia.com](https://docs.nvidia.com)) and is claimed to lift throughput by 10-30% in imbalanced models ([docs.nvidia.com](https://docs.nvidia.com)), which makes quality and convergence checks mandatory. The July 2026 IETF document on

EP communication is an Internet-Draft that expires on 5 January 2027 and is not a standard ([www.ietf.org](http://www.ietf.org)). The recommended gate set: per-expert token histograms, dispatch and combine bytes from profiler or NCCL traces, tail latency, dropped and padded token counts, and a recorded software and hardware bundle.

## Introduction and Background

This report is written for distributed-training and inference platform architects, performance engineers and private-AI buyers who must decide whether a mixture-of-experts model should keep expert parallelism inside a scale-up domain (an NVLink-connected set of GPUs) or let it cross nodes over a scale-out fabric such as InfiniBand or Ethernet. GPU Smith, a private AI infrastructure engineering consultancy, publishes this worksheet. Every figure below is either a dated primary specification or an explicitly labeled planning calculation. No benchmark results from the publisher are reported.

An MoE layer replaces one dense feed-forward block with many expert blocks and a router that selects a few of them per token. Mixtral 8x7B illustrates the ratio: 47B total parameters but 13B active per token ([arxiv.org](https://arxiv.org)). DeepSeek-V3 reports 671B total and 37B active ([arxiv.org](https://arxiv.org)). Memory residency follows total parameters and compute follows active parameters, while network traffic follows the number of token copies sent to experts, the hidden size, the element width and where the experts physically sit.

The evidence base is current as of 27 October 2026. The Megatron Core MoE documentation page for v0.15.0 defines the capacity relation used below ([docs.nvidia.com](https://docs.nvidia.com)), the Megatron-LM MoE README on the main branch carries the dispatcher guidance ([github.com](https://github.com)), and Megatron Bridge 0.6.2 documents the EP and TP recommendations used in the compatibility matrix ([docs.nvidia.com](https://docs.nvidia.com)). DeepSpeed provides a second framework reference ([www.deepspeed.ai](https://www.deepspeed.ai)). Model shapes come from published configuration files, not model-family summaries.

Three limits apply throughout:

- **Line rate is not realized bandwidth.** NCCL's test documentation introduces "bus bandwidth" so that measured results can be compared with hardware peak ([github.com](https://github.com)).
- **Phases differ.** Training, prefill and decode have different token populations and are separated below.
- **Examples are placeholders.** Every worked example is labeled "(Hypothetical Example)" and should be replaced with profiler data before any purchase or production decision.

## Key Changes in How MoE Parallelism Is Sized

### Four budgets replace one

A dense model is usually sized by parameter memory and arithmetic. An MoE model needs four budgets:

- **Full model residency:** every expert must live somewhere, so DeepSeek-V3-class models need roughly the memory of 671B parameters.
- **Active parameters per token:** the compute-side number.
- **Activation and token-dispatch traffic:** scales with tokens, top-k and hidden size.

- **Optimizer and gradient communication:** expert gradients are reduced across expert-data-parallel replicas, while non-expert gradients are reduced across the wider data-parallel group.

Megatron Bridge encodes the first and second budgets in its TP guidance: because only a fraction of parameters are active per token, tensor parallelism (TP) can often stay at 1 or 2 for MoE models. DeepSeek-V3 went further and trained without TP at all, using memory optimizations instead ([arxiv.org](https://arxiv.org)). Moving TP down frees GPUs for EP, which raises the third budget.

## Expert parallelism becomes its own process group

In traditional layouts the EP group is a sub-group of the data-parallel group used by attention. Megatron's folding design decouples them: the documentation states that the EP group in MoE is a sub-group of DP in attention in the unfolded case ([docs.nvidia.com](https://docs.nvidia.com)), and the README says both context-parallel (CP) and EP traffic can remain within an NVLink domain after folding ([github.com](https://github.com)). The MoE Parallel Folding paper reports that without folding, EP all-to-all is forced across the lower-bandwidth inter-node fabric ([arxiv.org](https://arxiv.org)) and that MoE layer latency then rises significantly ([arxiv.org](https://arxiv.org)). For fine-grained MoE models the README recommends expert tensor parallelism (ETP) of 1 ([github.com](https://github.com)), and its own measurement-free rule of thumb is EP8TP1 over EP4TP2 for Mixtral 8x7B ([docs.nvidia.com](https://docs.nvidia.com)).

## Dispatcher choice is now a runtime decision

The same logical EP layout can run on different communication back ends. Megatron Core recommends alltoall for EP, notes that allgather is better with TP only or very large top-k ([docs.nvidia.com](https://docs.nvidia.com)), and exposes a flex dispatcher whose back end flag is documented as experimental. Inference engines expose equivalent switches: vLLM defaults to an allgather and reducescatter implementation ([docs.vllm.ai](https://docs.vllm.ai)), and offers separate DeepEP modes for prefill ([docs.vllm.ai](https://docs.vllm.ai)) and decode ([docs.vllm.ai](https://docs.vllm.ai)). SGLang documents the same split ([docs.sglang.io](https://docs.sglang.io)).

## Capacity semantics decide what crosses the wire

Under Megatron's default dropless strategy no token is dropped or padded ([docs.nvidia.com](https://docs.nvidia.com)). Setting a capacity factor makes tokens beyond an expert's capacity droppable by selected probability ([docs.nvidia.com](https://docs.nvidia.com)), and the documentation states the dropping happens before the token exchange between EP ranks ([docs.nvidia.com](https://docs.nvidia.com)). Dropped tokens therefore do not consume all-to-all bandwidth, which is why the capacity factor appears in both the quality budget and the traffic budget. Switch Transformer defined the same idea: overflowing tokens skip the expert and pass through the residual path ([arxiv.org](https://arxiv.org)), and GShard likewise passes overflowed tokens along residual connections ([arxiv.org](https://arxiv.org)).

# Traffic and Capacity Worksheets

## Configuration inputs to pin

The worksheet starts from a configuration table that must be filled from the real model files and launch scripts. The pinned values below use the published DeepSeek-V3 and Mixtral configuration files as illustrations.

*Table 1* below lists the inputs that every later calculation uses, with the example values and where each is read from.

| Input                          | DeepSeek-V3 shape (Hypothetical Example)                                                    | Mixtral 8x7B shape (Hypothetical Example)                    | Where to pin it                 |
|--------------------------------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------|---------------------------------|
| Routed experts per MoE layer   | 256 ( <a href="https://huggingface.co">huggingface.co</a> )                                 | 8 ( <a href="https://huggingface.co">huggingface.co</a> )    | Model config file               |
| Shared experts                 | 1 ( <a href="https://huggingface.co">huggingface.co</a> )                                   | 0 (not in config)                                            | Model config file               |
| Top-k routed experts per token | 8 ( <a href="https://huggingface.co">huggingface.co</a> )                                   | 2 ( <a href="https://huggingface.co">huggingface.co</a> )    | Model config file               |
| Hidden size H                  | 7168 ( <a href="https://huggingface.co">huggingface.co</a> )                                | 4096 ( <a href="https://huggingface.co">huggingface.co</a> ) | Model config file               |
| MoE layers                     | 61 layers, first 3 dense, so 58 MoE ( <a href="https://huggingface.co">huggingface.co</a> ) | 32 layers, all MoE (assumed)                                 | Config plus layer pattern       |
| Element width b                | 2 bytes (BF16), 1 byte if FP8 dispatch ( <a href="https://github.com">github.com</a> )      | 2 bytes (BF16)                                               | Launch flags, precision recipe  |
| Tokens per rank T              | 4,096 (one sequence of 4K, assumed)                                                         | 4,096 (assumed)                                              | Micro-batch and sequence length |
| EP, TP, PP, CP, DP             | EP64, TP1, PP16 (paper's training layout) ( <a href="https://arxiv.org">arxiv.org</a> )     | EP8, TP1 (assumed)                                           | Launch scripts                  |
| Dispatcher                     | alltoall or flex (DeepEP or HybridEP)                                                       | alltoall                                                     | Launch scripts, version pin     |
| Topology boundary              | 8 GPUs per node, InfiniBand between nodes                                                   | one 8-GPU NVLink node                                        | Cluster inventory               |

The shared expert is replicated across EP ranks rather than distributed ([docs.nvidia.com](https://docs.nvidia.com)), so it adds compute and memory but no dispatch traffic in Megatron, whereas DeepSeek's decode deployment treats the shared expert as a routed one so each token selects 9 experts ([arxiv.org](https://arxiv.org)). A calculation that ignores which convention the pinned runtime uses is off by one expert in nine.

## Token assignments and the capacity worksheet

Each token produces  $k$  assignments, so a rank with  $T$  tokens creates  $T$  times  $k$  assignment vectors per MoE layer. With  $T = 4,096$  and  $k = 8$  that is 32,768 assignments, or a mean of 128 per expert across 256 experts. Megatron's relation gives per-expert capacity as tokens per rank times top-k times capacity factor divided by the number of experts. The MoE Parallel Folding paper describes capacity in terms of tokens exceeding "the capacity of a given expert" ([arxiv.org](https://arxiv.org)), but its summary formula omits top-k, so the denominator and the meaning of "tokens per rank" must be confirmed in the pinned code rather than assumed. The Megatron page does not state a rounding rule in the passages reviewed, so the table uses exact values.

*Table 2* below applies the Megatron relation to the DeepSeek-V3 shape under one balanced and one skewed case, where the skewed case is defined as a single hot expert receiving twice the mean.

| Capacity factor | Per-expert capacity | Balanced case (128 each)         | Skewed case (hot expert receives 256)                               | Padded slots if all experts padded |
|-----------------|---------------------|----------------------------------|---------------------------------------------------------------------|------------------------------------|
| 1.0             | 128                 | 0 dropped, 0 padded              | 128 dropped (50% of that expert's tokens, 0.39% of all assignments) | 0                                  |
| 1.25            | 160                 | 0 dropped, 32 padded per expert  | 96 dropped (37.5% of that expert's tokens)                          | 8,192 of 40,960 slots (20%)        |
| 1.5             | 192                 | 0 dropped, 64 padded per expert  | 64 dropped (25%)                                                    | 16,384 of 49,152 slots (33%)       |
| 2.0             | 256                 | 0 dropped, 128 padded per expert | 0 dropped                                                           | 32,768 of 65,536 slots (50%)       |
| Dropless        | none                | exact assignments                | no drops, hot expert receives 256                                   | none                               |

These values are the author's arithmetic from the documented relation, not measurements. Table 2 shows two coupled effects. A capacity factor of 1.0 is the cheapest in bytes and compute but drops tokens as soon as any expert exceeds its mean, and real routers do exceed it: the Mixtral authors observe locality in routing that makes oversubscription of certain experts more likely under EP ([arxiv.org](https://arxiv.org)). Raising the factor removes drops but, if padding is transmitted, raises dispatch bytes in proportion to the factor. Megatron documents padding as filling under-filled experts up to capacity ([docs.nvidia.com](https://docs.nvidia.com)), but the passages reviewed do not say whether padded slots cross the network, so that is a profiler question. Switch Transformer reports that transformer variants performed better at lower capacity factors of 1.0 and 1.25 in its comparison ([arxiv.org](https://arxiv.org)) and that larger factors add computation and communication cost, and a later Megatron paper benchmarked with a capacity factor of 1 to avoid dropless jitter ([arxiv.org](https://arxiv.org)).

## Rank-level skew and the straggler effect

All-to-all completes only when the slowest rank has received its data. NVIDIA measured DeepSeek R1 routing in TensorRT-LLM and reported that, without load balancing, the hottest rank receives 1.56 times more routed tokens than the mean ([nvidia.github.io](https://nvidia.github.io)). NVIDIA defines this imbalance ratio as  $(\max - \text{mean}) / \text{mean}$ , so 1.56 is an excess over the mean and the hottest rank carries roughly 2.56 times the mean load; the figure is an average over layers and iterations for DeepSeek-R1 at EP32 on GB200, not a per-step peak. Applied to Table 3's balanced 462 MB bound, that gives about 1.18 GB (462 MB x 2.56) on the straggler. MoETuner reports that equal expert counts per GPU still leave activation skewed ([arxiv.org](https://arxiv.org)). In inference, SGLang measured expert-parallel load balancing (EPLB) speedups of 1.49x for prefill and 2.54x for decode ([lmsys.org](https://lmsys.org)), which suggests skew penalizes decode more.

## First-order dispatch and combine bytes

The per-layer, per-direction estimate is assigned token vectors times hidden size times bytes per element, multiplied by the remote fraction (EP minus 1) divided by EP when experts are spread uniformly. Combine returns the expert outputs over the reverse path, so a forward pass has two all-to-alls per MoE layer and the backward pass has two more. The Megatron-Core report likewise states that expert-parallel all-to-all volume

scales with token count, not expert count ([arxiv.org](https://arxiv.org)). NCCL expresses all-to-all as each rank providing an input buffer divided into one chunk per destination ([docs.nvidia.com](https://docs.nvidia.com)), and it can be composed from grouped send and receive calls ([docs.nvidia.com](https://docs.nvidia.com)).

Table 3 below gives the byte bounds for the DeepSeek-V3 and Mixtral shapes with the profiler columns left open on purpose.

| Case (all Hypothetical Examples)                                          | Assignment vectors       | Bound per layer, one direction | Forward round trip, all MoE layers                                       | Observed profiler bytes | Variance       |
|---------------------------------------------------------------------------|--------------------------|--------------------------------|--------------------------------------------------------------------------|-------------------------|----------------|
| DeepSeek-V3 shape, EP64, BF16, no dedup, remote share 63/64               | 32,768                   | 462 MB                         | 53.6 GB over 58 layers (about 107 GB with backward, excluding recompute) | to be measured          | to be measured |
| Same, only copies leaving the node (7 of 8 remote)                        | 28,672                   | 411 MB                         | 47.7 GB                                                                  | to be measured          | to be measured |
| Same, one copy per destination node, expected 4.6 remote nodes            | about 18,800 node copies | 270 MB                         | 31.3 GB                                                                  | to be measured          | to be measured |
| Same, node-limited to at most 3 remote nodes (assumption-dependent bound) | at most 12,288           | 176 MB                         | 20.4 GB                                                                  | to be measured          | to be measured |
| Same, capacity factor 1.25 padded and sent                                | 40,960 slots             | 587 MB                         | 68.1 GB                                                                  | to be measured          | to be measured |
| Mixtral shape, EP8 inside one node, BF16                                  | 8,192                    | 58.7 MB                        | 3.8 GB over 32 layers                                                    | to be measured          | to be measured |

The rows follow from simple multiplication. The node-limited rows reflect DeepSeek-V3's design choice to dispatch each token to at most 4 nodes ([arxiv.org](https://arxiv.org)), which exists because NVLink offers 160 GB/s against 50 GB/s for InfiniBand in that cluster ([arxiv.org](https://arxiv.org)); Megatron exposes the same constraint as group-limited routing. The deduplicated rows track what DeepEP documents: it removes redundant tokens during cross-node communication. For the expected node count, eight uniformly random experts across eight nodes touch  $8 \times (1 - (7/8)^8) \approx 5.25$  nodes on average, but the home node is touched only with probability about 0.66, so the expected number of remote nodes is  $7 \times (1 - (7/8)^8) \approx 4.6$ . The at-most-3-remote-nodes row is an assumption-dependent bound: it holds only if the home node is one of the 4 nodes DeepSeek-V3 allows, which the paper does not guarantee; a strict 4-remote-node bound would be 16,384 copies, about 235 MB per layer per direction.

What the bounds omit matters as much. They exclude routing probabilities returned alongside tokens, index and metadata tensors, alignment padding such as Megatron's router padding of the routing map to multiples of 16 or 32 ([github.com](https://github.com)), quantization scale tensors, duplicate-token handling choices inside a given kernel, and

retransmissions. They also assume the dispatcher moves each vector once. Quantization changes the element width: Megatron lists FP8 dispatch as sending tokens in FP8 instead of BF16, which halves the dispatch row while the combine path may stay wider, so the two directions should be sized separately.

## Memory residency for the same shape

Using the DeepSeek-V3 configuration, the routed experts use an intermediate size of 2,048 ([huggingface.co](https://huggingface.co)). Assuming a gated three-matrix expert (an assumption, since the config file does not state it), one expert holds 3 times 7,168 times 2,048, or 44.0 million parameters, and 58 layers of 256 routed experts hold about 654B of the 671B total. At EP64 each rank holds 4 experts in each of 58 layers, about 10.2B parameters before PP splits the layers, or roughly 10 GB in FP8 or 20 GB in BF16 for weights alone. As a cross-check, TensorRT-LLM reports 24 MiB per DeepSeek R1 expert in its FP4 deployment ([nvidia.github.io](https://nvidia.github.io)), which is consistent with 44.0 million parameters at half a byte plus scale tensors.

## Fabric Placement: Scale-Up Versus Scale-Out

### Which collective runs on which link

The placement question is a map from parallel groups to links. *Table 4* below assigns each group to the fabric it uses in a common eight-GPU-node layout and flags where the group crosses a boundary.

| Group          | Collective                                                 | Typical link when sized inside a node           | Crosses node or rack when                  | Contention note                                      |
|----------------|------------------------------------------------------------|-------------------------------------------------|--------------------------------------------|------------------------------------------------------|
| TP             | all-reduce or reduce-scatter and all-gather on activations | NVLink                                          | TP greater than GPUs per scale-up domain   | Shares NVLink with EP when folded groups overlap     |
| EP             | dispatch and combine all-to-all                            | NVLink inside the domain, NIC/InfiniBand beyond | EP times TP exceeds 8 on an eight-GPU node | Dominant MoE traffic, scales with tokens times top-k |
| DP / expert-DP | gradient all-reduce or reduce-scatter                      | InfiniBand across nodes                         | Almost always                              | Overlaps with backward all-to-all                    |
| PP             | point-to-point activations                                 | InfiniBand across stages                        | PP stages on different nodes               | Latency sensitive, small messages                    |
| CP             | ring or all-gather on KV                                   | NVLink if folded                                | CP times EP exceeds the domain             | Folding keeps CP and EP in NVLink                    |

Megatron's guidance is explicit that EP and TP are communication-intensive and should stay inside the NVLink domain ([github.com](https://github.com)), and when scaling beyond one node prefer PP over expanding TP/EP across nodes ([github.com](https://github.com)). The IETF draft raises the contention point that all-to-all and all-reduce compete for bandwidth when run concurrently ([www.ietf.org](https://www.ietf.org)). That draft is a work-in-progress Internet-Draft whose boilerplate states that it is inappropriate to use such drafts as reference material ([www.ietf.org](https://www.ietf.org)), so it is used here only as a source of framing: its mode-selection rule compares moving experts to tokens by volume ([www.ietf.org](https://www.ietf.org)), which for fine-grained experts is a measured question rather than a formula to adopt.

## Reference link speeds and the scale-up boundary

The scale-up boundary has moved with each GPU generation. The NVIDIA H100 page lists fourth-generation NVLink at 900 GB/s of GPU-to-GPU interconnect ([www.nvidia.com](http://www.nvidia.com)), and a DGX H100 or H200 has eight GPUs per node ([docs.nvidia.com](https://docs.nvidia.com)) with InfiniBand at up to 400 Gbps per card ([docs.nvidia.com](https://docs.nvidia.com)). A DGX B200 has two fifth-generation NVLink switches with 14.4 TB/s aggregate bandwidth ([docs.nvidia.com](https://docs.nvidia.com)) and eight single-port ConnectX-7 NICs ([www.nvidia.com](http://www.nvidia.com)) at up to 400 Gb/s ([www.nvidia.com](http://www.nvidia.com)). GB200 NVL72 provides 1.8 TB/s of GPU-to-GPU NVLink ([www.nvidia.com](http://www.nvidia.com)) and 130 TB/s across the NVLink Switch System ([www.nvidia.com](http://www.nvidia.com)), in a rack of 18 compute trays holding four Blackwell GPUs each ([docs.nvidia.com](https://docs.nvidia.com)), with four ConnectX-7 400G NICs per tray in the DGX GB200 configuration ([docs.nvidia.com](https://docs.nvidia.com)). For scale-out, ConnectX-8 offers up to 800 Gb/s ([www.nvidia.com](http://www.nvidia.com)) and Quantum-X800 provides 144 ports at 800 Gb/s ([www.nvidia.com](http://www.nvidia.com)).

Line rate converts as 400 Gb/s equals 50 GB/s and 800 Gb/s equals 100 GB/s, in a single direction. Table 3's 270 MB expected-case deduplicated figure at 50 GB/s would take about 5.4 ms per layer per direction at nominal line rate, and its 176 MB node-limited bound about 3.5 ms; the 411 MB undeduplicated bound would take about 8 ms. These are lower bounds under an idealized assumption that a rank's whole share moves at full NIC rate, and the measured outcome depends on message sizes, congestion, routing and GPU copy engines. The Parallel Folding paper's test cluster used 400 Gbps inter-node InfiniBand, and DeepEP's published H800 kernel tests were run on hardware with about 160 GB/s of NVLink bandwidth ([github.com](https://github.com)), so the throughput on newer parts should not be extrapolated from those rows.

## Training, prefill and decode are different workloads

Training has large, regular token populations per microbatch and a backward pass that doubles the all-to-all count. DeepSeek-V3 trained with 64-way EP across 8 nodes, found cross-node EP gave a computation-to-communication ratio of about 1:1 ([arxiv.org](https://arxiv.org)), and reported using 20 of 132 SMs on each GPU for communication. Prefill batches many tokens and favors throughput: DeepSeek-V3's prefill unit is 4 nodes with 32 GPUs ([arxiv.org](https://arxiv.org)). Decode batches few tokens per expert and is latency bound: the same paper's decode unit is 40 nodes with 320 GPUs ([arxiv.org](https://arxiv.org)), with batch sizes per expert usually within 256 tokens ([arxiv.org](https://arxiv.org)). A later DeepSeek inference overview describes decode deployment units of 18 nodes ([github.com](https://github.com)), so the two documents disagree on unit size and the date of each should be checked before reusing either number.

NVIDIA states that expert dispatch and combine all-to-all can quickly saturate an already memory-bound decode phase ([developer.nvidia.com](https://developer.nvidia.com)), and that EP32 on GB200 NVL72 achieves up to 1.8x more per-GPU throughput than a smaller EP configuration in its scenario ([developer.nvidia.com](https://developer.nvidia.com)). vLLM notes that a single compute-bound prefill request can delay the forward pass of an entire EP group ([vllm.ai](https://vllm.ai)), a motivation for disaggregating prefill and decode.

A decode sizing example (Hypothetical Example): 128 tokens per rank, 9 experts per token including the shared expert, hidden size 7168, FP8 dispatch at one byte and BF16 combine at two bytes. That is 1,152 vectors, about 8.3 MB dispatch and 16.5 MB combine per layer. At 50 GB/s the dispatch would take roughly 165 microseconds if all vectors crossed one NIC, so decode is dominated by latency, kernel launch and synchronization rather than bytes. Preallocation matters here too: SGLang notes that DeepEP's low-latency decode mode requires preallocating a fixed memory size ([lmsys.org](https://lmsys.org)).

## Failure and fallback behavior

Plan the fallbacks before the pilot:

- **Flex activation:** the Megatron flex path needs both the dispatcher type and the backend selected; Bridge documents a helper function to apply the flex backend ([docs.nvidia.com](https://docs.nvidia.com)), and setting the backend flag alone does not turn on flex dispatch.
- **CUDA graphs:** MoE layers with dynamic shapes cannot be captured ([docs.nvidia.com](https://docs.nvidia.com)), which is an argument for fixed capacity in latency-critical serving.
- **Virtual-expert mode:** the HybridEP virtual-expert mode does not support token dropping or expert capacity limits ([github.com](https://github.com)).
- **Experimental paths:** the DeepEP NIXL integration is described as experimental ([github.com](https://github.com)).
- **Unsupported hardware:** if the preferred dispatcher is unsupported on the delivered hardware, fall back to alltoall, which the documentation recommends when EP is applied.

## Dispatcher Candidates and Support Matrix

No single dispatcher wins everywhere. *Table 5* below summarizes what the reviewed documentation says about each candidate; every cell should be re-verified against the exact versions in the pinned stack.

| Dispatcher                                    | Documented intent                                                                                      | Hardware as documented                                                                                                | Constraints and caveats                                                                                                                                                               | Unverified in this report          |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| allgather                                     | Better with TP only or very large top-k                                                                | Not stated                                                                                                            | Not the documented recommendation when EP is applied                                                                                                                                  | Realized bandwidth at EP above 8   |
| alltoall                                      | Recommended when EP is applied                                                                         | Not stated                                                                                                            | Required for token dropping in Bridge                                                                                                                                                 | Behavior across racks              |
| DeepEP (flex)                                 | Cross-node EP and fine-grained MoE such as DeepSeek-V3 ( <a href="https://github.com">github.com</a> ) | Ampere, Hopper, B200, B300                                                                                            | Uses NVLink in a node and RDMA across nodes ( <a href="https://github.com">github.com</a> ); main branch drops the NVSHMEM dependency ( <a href="https://github.com">github.com</a> ) | Which branch the pinned build uses |
| HybridEP (flex)                               | Fewer SMs and native multi-node NVLink                                                                 | GB200 and GB300 with NVL72, plus Ampere, Hopper, B200, B300 ( <a href="https://docs.nvidia.com">docs.nvidia.com</a> ) | Not stated in the pinned documents                                                                                                                                                    | Behavior on non-NVL72 clusters     |
| Runtime-specific (vLLM, SGLang, TensorRT-LLM) | Prefill and decode variants, NVLink options ( <a href="https://docs.vllm.ai">docs.vllm.ai</a> )        | Per runtime                                                                                                           | Wide-EP kernels in TensorRT-LLM target GB200 MNNVL ( <a href="https://nvidia.github.io">nvidia.github.io</a> )                                                                        | Cross-runtime parity               |

Table 5 is a pointer list, not a ranking. The DeepEP hybrid-ep branch documents support for both NVLink inside a node and RDMA across nodes ([github.com](https://github.com)), and the DeepEP main-branch README lists Hybrid-EP under its experimental branches, pointing to the separate hybrid-ep branch, a new backend aimed at minimal

SM usage and larger NVLink domains ([github.com](https://github.com)). NVIDIA's engineering blog states that Hybrid-EP needs about 4 SMs to approach NIC bandwidth in a four-node DGX Hopper test ([developer.nvidia.com](https://developer.nvidia.com)) and 16 SMs to fill NVLink bandwidth ([developer.nvidia.com](https://developer.nvidia.com)) on a separate GB200 NVL36 platform, a 36-GPU NVLink domain ([developer.nvidia.com](https://developer.nvidia.com)), and reports about 14% higher DeepSeek-V3 throughput than DeepEP in one Megatron Core configuration ([developer.nvidia.com](https://developer.nvidia.com)). Those are vendor-reported results on a specific platform, and the Hybrid-EP repository notes that its bandwidths are algorithm bandwidth ([github.com](https://github.com)), so they should not be compared with NCCL bus bandwidth without conversion.

Compatibility lessons from the pinned documents: the flex path and the alltoall path have different overlap and dropping constraints, and the August 2026 Megatron activity report records continuing MoE work on EP all-to-all overlap under CUDA Graphs ([github.com](https://github.com)), although the passages reviewed did not list HybridEP items. A pilot must therefore log the exact commit, container, driver, NCCL version and flags.

## Implementation Considerations and Process Changes

### Design the pilot

The pilot asks whether EP fits the scale-up domain, which dispatcher wins on the real topology, and whether quality survives the chosen capacity policy. Run alltoall first, then flex with DeepEP or HybridEP where supported, at the planned layout and at one alternative that keeps EP inside the domain.

- **Per-expert token histograms:** collect per-layer, per-expert counts for p50, p95 and maximum, including early training where routers are least balanced. Megatron documents early dropless OOM risk from router imbalance and suggests a small capacity factor such as 1.0 as one mitigation ([docs.nvidia.com](https://docs.nvidia.com)).
- **Collective bytes and time:** record dispatch and combine bytes per layer from profiler traces and NCCL logs, and compare with Table 3. Use bus bandwidth conventions consistently ([github.com](https://github.com)).
- **Link utilization:** sample NVLink and NIC counters during the all-to-all, separately from DP gradient traffic.
- **Overlap:** measure how much all-to-all time hides behind expert compute, because Megatron reports EP all-to-all can consume 30-40% of training time without optimization ([github.com](https://github.com)).
- **Tail latency:** for serving, report p50, p95 and p99 per decode step, not averages.
- **Dropped and padded tokens:** log per layer and per step, not as a run-level mean.
- **Quality and convergence:** compare loss curves or task metrics against a dropless baseline whenever a capacity factor is set. Switch Transformer reports typical dropped-token rates under 1% with sufficient balancing loss ([arxiv.org](https://arxiv.org)), but this is an anchor for expectation, not a gate.
- **Routing locality:** check whether consecutive tokens favor the same experts, since this concentrates load.

Table 6 below is the result template for each candidate run; the cells stay empty until real measurements exist.

| Metric                            | alltoall | DeepEP | HybridEP | Gate (set before the run)          |
|-----------------------------------|----------|--------|----------|------------------------------------|
| Per-expert tokens p50 / p95 / max |          |        |          | Max within agreed multiple of mean |

| Metric                               | alltoall | DeepEP | HybridEP | Gate (set before the run)               |
|--------------------------------------|----------|--------|----------|-----------------------------------------|
| Dropped tokens (% of assignments)    |          |        |          | Agreed ceiling, with quality check      |
| Padded tokens (% of slots)           |          |        |          | Reported, not hidden in throughput      |
| Dispatch + combine bytes per layer   |          |        |          | Within agreed band of Table 3 bound     |
| All-to-all time per layer, p50 / p95 |          |        |          | Below agreed share of step time         |
| Overlap with compute (%)             |          |        |          | Reported                                |
| NVLink / NIC utilization             |          |        |          | No sustained saturation on shared links |
| Step latency or tail latency         |          |        |          | Meets service-level target              |
| Quality or convergence result        |          |        |          | No regression versus baseline           |

Table 6 deliberately contains no numbers: the thresholds belong to the buyer's service-level and quality targets, not to a generic rule, and each run should come from the same trace set so variance can be computed.

## Go/no-go gates and the evidence bundle

- **Placement gate:** EP times TP fits the scale-up domain, or the cross-node share measured in the pilot is within the fabric's measured headroom with DP traffic running.
- **Traffic gate:** observed bytes are within an agreed variance of the Table 3 bound after accounting for metadata, deduplication and padding; large gaps require an explanation, not a pass.
- **Skew gate:** the hottest rank's load stays within the agreed multiple of the mean, or a load balancer such as EPLB is part of the plan. vLLM collects load statistics every forward pass and rebalances periodically ([docs.vllm.ai](https://docs.vllm.ai)), and TensorRT-LLM suggests starting with offline EPLB when workload patterns are known ([nvidia.github.io](https://nvidia.github.io)).
- **Drop gate:** if a capacity factor is set, convergence or quality matches the dropless baseline within tolerance.
- **Software gate:** the dispatcher, CUDA, PyTorch, NCCL and framework versions satisfy the documented requirements for that dispatcher.
- **Fallback gate:** a tested alltoall path exists.

Preserve the bundle: model config files, launch command and environment, parallel group tables, topology and NIC maps, NCCL and profiler traces, histograms, dropped and padded counters, and the quality results. Divisibility rules (experts by EP, world size by total parallelism) should be asserted in the launch script.

## Cross-framework notes

DeepSpeed exposes EP through an `ep_size` argument defined as the number of ranks in the expert parallel group ([deepspeed.readthedocs.io](https://deepspeed.readthedocs.io)) and a minimum capacity per expert regardless of the capacity factor ([deepspeed.readthedocs.io](https://deepspeed.readthedocs.io)). DeepSpeed-MoE also documents that coordinating all-to-all with tensor-slicing of

degree L bounds all-to-all latency by  $O(p/L)$  instead of  $O(p)$  ([arxiv.org](#)), and describes a hierarchical all-to-all. Such changes alter the shape of Table 3.

## Data Analysis and Evidence

The sources reviewed give a range of all-to-all cost estimates, none comparable with the others because model, hardware, software version and batch differ.

- **16% to 36% of MoE and transformer layer time** in GShard's study when experts grew from 128 to 2,048.
- **35.7% of end-to-end inference time** for Mixtral 8x7B in MoETuner's measurement ([arxiv.org](#)).
- **30-40% of training time without optimization** in the Megatron README, with no hardware context ([github.com](#)).
- **Up to 60% of training time** for unoptimized all-to-all in DeepSeek-V3-style architectures whose experts span multiple nodes, per the Megatron-Core report.
- **An imbalance ratio of 1.56** ((max – mean) / mean, averaged over layers and iterations at EP32 on GB200) for the hottest rank on DeepSeek R1 without load balancing ([nvidia.github.io](#)).
- **1.49x and 2.54x** EPLB speedups in SGLang prefill and decode on H100 nodes ([lmsys.org](#)), measured on a 12-node cluster of 8 H100 GPUs each ([lmsys.org](#)) and reporting 52.3k input and 22.3k output tokens per second per node in its headline configuration ([lmsys.org](#)).
- **10-30%** throughput gain claimed for token dropping in imbalanced models.
- **About 14%** Hybrid-EP gain over DeepEP on DeepSeek-V3 in one vendor configuration ([developer.nvidia.com](#)).

Two observations follow. First, the spread shows that all-to-all share is a property of the deployment and not of the model family. Second, improvements attributed to dispatchers and balancing came from named hardware and should not be copied into a business case without a pilot.

The calculated figures in this report (462 MB, 53.6 GB, 1.18 GB and others) derive from the documented formulas and published configuration files with the assumptions stated beside them. They are not measured values. The DeepSeek-V3 token-dispatch design is also a reminder that a model trained with node-limited routing and no token dropping ([arxiv.org](#)) has different traffic and quality properties than a model trained with a low capacity factor, even when both have similar parameter counts.

## Implications and Future Directions

First, the choice of scale-up domain size is an MoE decision: a 72-GPU NVLink domain can hold a wide EP group without touching the NIC, while eight-GPU nodes force a node-limited routing and deduplicating dispatcher design. Cost, power and cooling belong in separate worksheets. Second, scale-out fabric spend is justified by the measured cross-node share, not by the aggregate bisection number. Third, the dispatcher decision is revisitable per software release, which is why version pins and fallbacks are part of the acceptance criteria.

Looking ahead, DeepEP's main branch has moved to unified high-throughput and low-latency interfaces and dropped the NVSHMEM dependency ([github.com](https://github.com)), HybridEP is being exposed through multiple Megatron releases, and the IETF draft remains provisional until it is revised or lapses on its expiry date ([www.ietf.org](https://www.ietf.org)). NVIDIA's NVLink page lists a Vera Rubin NVL72 at 216 TB/s total ([www.nvidia.com](https://www.nvidia.com)), a preliminary specification that NVIDIA says may change (checked 27 October 2026), and its Spectrum-X Ethernet switch SN6600 offers 128 ports of 800G ([www.nvidia.com](https://www.nvidia.com)). Spectrum-X is advertised with a 1.6x improvement over off-the-shelf Ethernet for AI network performance ([www.nvidia.com](https://www.nvidia.com)), which is a vendor claim to be tested with the real all-to-all pattern, not assumed. Re-run the pilot at each major framework upgrade.

## Frequently Asked Questions (FAQs)

### What is expert parallel size in a mixture-of-experts model?

It is the number of ranks across which each MoE layer's experts are distributed. DeepSpeed defines `ep_size` as the number of ranks in the expert parallel group ([deepspeed.readthedocs.io](https://deepspeed.readthedocs.io)), and Megatron requires the expert count to be divisible by it.

### How is the MoE capacity factor explained, and what formula applies?

Megatron's documented relation is capacity equals tokens per rank times top-k times capacity factor divided by the number of experts. Tokens beyond capacity are dropped by probability, and the default strategy drops nothing.

### What does MoE token dropping cost in quality?

No general figure applies. Switch Transformer reported typical drop rates under 1% with adequate balancing, and DeepSeek-V3 avoided dropping entirely. Test quality against a dropless baseline.

### How much all-to-all bandwidth does MoE need?

Start with assigned vectors times hidden size times bytes per element per direction, as in Table 3, then measure. Published all-to-all time shares range from 16% to 60% across different setups.

### Expert parallelism versus tensor parallelism: which should I prefer?

For expert layers, Megatron prefers EP over TP, noting EP8TP1 beats EP4TP2 for Mixtral 8x7B, and Bridge says TP can often stay at 1 or 2 for MoE.

### What causes MoE token routing load imbalance, and what helps?

Routers concentrate tokens on some experts, producing hot ranks (an imbalance ratio of 1.56, defined as  $(\max - \text{mean}) / \text{mean}$ , in one NVIDIA measurement) ([nvidia.github.io](https://nvidia.github.io)). Auxiliary losses, capacity limits and expert load balancing such as EPLB reduce it ([docs.vllm.ai](https://docs.vllm.ai)).

## How should inference EP use NVLink and InfiniBand?

Keep prefill and decode modes separate: high-throughput kernels for prefill ([docs.sglang.io](https://docs.sglang.io)), low-latency kernels for decode ([docs.sglang.io](https://docs.sglang.io)), and NVLink-based wide EP where an NVL72 domain exists ([nvidia.github.io](https://nvidia.github.io)).

## How do I choose the expert parallel degree?

Pick the largest EP that divides the expert count, fits memory, and stays inside the scale-up domain; if it must cross nodes, use node-limited routing and a deduplicating dispatcher, then confirm in the pilot.

## What network does MoE training require?

It depends on the cross-node share of dispatch bytes. DGX B200 systems list NICs at up to 400 Gb/s ([www.nvidia.com](https://www.nvidia.com)), and newer NICs offer 800 Gb/s ([www.nvidia.com](https://www.nvidia.com)). Validate against the pilot, not line rate.

## Conclusion

Expert parallelism sizing reduces to a short, auditable chain: pin the model configuration, compute token assignments and capacity, bound dispatch and combine bytes, map each group to a physical link, choose a dispatcher the pinned software and hardware support, and then measure. The calculated bounds in this report identify where traffic is likely to concentrate and which placements deserve a pilot; they do not predict step time. The decision to keep EP inside a scale-up domain or cross nodes should rest on measured per-expert histograms, collective bytes and time, link utilization, tail latency, dropped and padded tokens, and a quality comparison against a dropless baseline, all preserved with the exact software and hardware versions. Published figures from vendors and papers are useful for framing and are conditional on their stated platforms.

## Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://docs.nvidia.com/megatron-core/developer-guide/0.15.0/api-guide/moe.html>
2. <https://docs.nvidia.com/nemo/megatron-bridge/0.6.2/parallelisms.html>
3. <https://github.com/NVIDIA/Megatron-LM/blob/main/megatron/core/transformer/moe/README.md>
4. <https://arxiv.org/pdf/2006.16668>
5. <https://arxiv.org/html/2603.07685v2>
6. <https://www.ietf.org/ietf-ftp/internet-drafts/draft-li-moe-ep-communication-optimization-00.html>
7. <https://arxiv.org/pdf/2401.04088>
8. <https://arxiv.org/html/2412.19437v2>
9. <https://www.deepspeed.ai/tutorials/mixture-of-experts>
10. <https://github.com/NVIDIA/nccl-tests/blob/master/doc/PERFORMANCE.md>
11. <https://arxiv.org/html/2504.14960v3>
12. [https://docs.vllm.ai/en/latest/serving/expert\\_parallel\\_deployment](https://docs.vllm.ai/en/latest/serving/expert_parallel_deployment)
13. [https://docs.sglang.io/advanced\\_features/expert\\_parallelism.html](https://docs.sglang.io/advanced_features/expert_parallelism.html)

14. <https://arxiv.org/pdf/2101.03961>
15. <https://huggingface.co/deepseek-ai/DeepSeek-V3/raw/main/config.json>
16. <https://huggingface.co/mistralai/Mixtral-8x7B-v0.1/raw/main/config.json>
17. [https://nvidia.github.io/TensorRT-LLM/blogs/tech\\_blog/blog04\\_Scaling\\_Expert\\_Parallelism\\_in\\_TensorRT-LLM.html](https://nvidia.github.io/TensorRT-LLM/blogs/tech_blog/blog04_Scaling_Expert_Parallelism_in_TensorRT-LLM.html)
18. <https://arxiv.org/pdf/2502.06643>
19. <https://lmsys.org/blog/2025-05-05-large-scale-ep>
20. <https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/usage/collectives.html>
21. <https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/mpi.html>
22. <https://www.nvidia.com/en-us/data-center/h100>
23. <https://docs.nvidia.com/dgx/dgxm100-user-guide/introduction-to-dgxm100.html>
24. <https://docs.nvidia.com/dgx/dgxb200-user-guide/introduction-to-dgxb200.html>
25. <https://www.nvidia.com/en-us/data-center/dgx-b200>
26. <https://www.nvidia.com/en-us/data-center/gb200-nvl72>
27. <https://docs.nvidia.com/dgx/dgxb200-user-guide/hardware.html>
28. <https://www.nvidia.com/en-us/networking/products/ethernet/supernic>
29. <https://www.nvidia.com/en-us/networking/products/infiniband/quantum-x800>
30. <https://github.com/deepseek-ai/DeepEP/tree/hybrid-ep>
31. [https://github.com/deepseek-ai/open-infra-index/blob/main/202502OpenSourceWeek/day\\_6\\_one\\_more\\_thing\\_deepseekV3R1\\_inference\\_system\\_overview.md](https://github.com/deepseek-ai/open-infra-index/blob/main/202502OpenSourceWeek/day_6_one_more_thing_deepseekV3R1_inference_system_overview.md)
32. <https://developer.nvidia.com/blog/scaling-large-moe-models-with-wide-expert-parallelism-on-nvl72-rack-scale-systems>
33. <https://vllm.ai/blog/2025-12-17-large-scale-serving>
34. <https://github.com/deepseek-ai/DeepEP>
35. <https://nvidia.github.io/TensorRT-LLM/1.3.0rc13/features/parallel-strategy.html>
36. <https://developer.nvidia.com/blog/optimizing-communication-for-mixture-of-experts-training-with-hybrid-expert-parallel>
37. [https://github.com/deepseek-ai/DeepEP/blob/hybrid-ep/docs/README\\_Hybrid-EP.md](https://github.com/deepseek-ai/DeepEP/blob/hybrid-ep/docs/README_Hybrid-EP.md)
38. <https://github.com/nvidia/megatron-lm/issues/6921>
39. <https://deepspeed.readthedocs.io/en/latest/moe.html>
40. <https://arxiv.org/pdf/2201.05596>
41. <https://www.nvidia.com/en-us/data-center/nvlink>
42. <https://www.nvidia.com/en-us/networking/spectrumx>

## THE PUBLISHER

# About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

## Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

## Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

## Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

## Work with GPU Smith

Explore the [hardware library](#), [GPU references](#), [networking references](#) and [vendor research](#). [Contact GPU Smith](#) with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

**Website:** <https://gpusmith.com>

---

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.