



GPU Cluster Fabric Sizing: Ports, Rails & Oversubscription

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 · 2026-10-20 · gpu cluster fabric sizing · gpu cluster network design · rail-optimized topology · oversubscription ratio · fat tree sizing · leaf spine port count · bisection bandwidth · ai cluster network failure

Original article: <https://gpusmith.com/articles/en/gpu-cluster-fabric-sizing-ports-rails-oversubscription>



In this article

1. Executive Summary
 2. Introduction and Background
 3. Key Changes in AI Fabric Design That Affect Sizing
 4. The Sizing Worksheet: From Endpoints to Ports
 5. Rail, Tree and Hybrid Placement
 6. Failure and Degraded-State Capacity
 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

GPU cluster fabric sizing is port arithmetic on a frozen endpoint list, followed by explicit declarations about traffic, oversubscription and failure. The endpoint count is nodes times NICs per node times ports per NIC, counted at the link rate the switch will actually run. A **64 x 800G** switch chip such as Broadcom's Tomahawk 5 provides **51.2 Tbps** in one direction and splits into **128 x 400G** ports (investors.broadcom.com) (www.arista.com), yet Cisco's datasheet quotes a 64 x 800G switch at **102.4 Tbps**, twice the one-direction total, which matches the full-duplex convention Arista states as 51.2 Tbps of throughput (102.4 Tbps full-duplex) (www.arista.com) (www.cisco.com). Mixing those conventions is the most common way to double-count a fabric.

Published designs show how the arithmetic lands. NVIDIA's DGX B300 scalable unit (SU) is **64 nodes and 512 GPUs** served by **16 leaf and 8 spine** switches (docs.nvidia.com); a Cisco 256-GPU unit uses **8 leaf and 4 spine** switches (www.cisco.com); an Arista two-tier design for 8,192 accelerators uses **128 leaf and 64 spine** systems (www.arista.com). A two-tier non-blocking fabric of radix-r switches reaches at most r^2 divided by 2 endpoint links (derived here), and DeepSeek's published two-layer fat tree of 64-port switches matches that arithmetic at 2,048 endpoints and 96 switches ([Table 3 of the DeepSeek paper](#)). Beyond the ceiling, a third tier or additional independent planes are required.

Oversubscription is a declared design choice. This report defines it as aggregate downlink capacity divided by aggregate usable uplink capacity, and shows the inverse where sources invert it. AMD writes subscription as downstream demand to upstream capacity and recommends **1:1.16** (instinct.docs.amd.com); Meta describes an uplink that is twice the downlink as **1:2 under-subscribed** (engineering.fb.com); Alibaba's published tier-1 switch works out to **1.067:1** (derived from 128 x 200G down and 60 x 400G up) (ennanzhai.github.io); and Meta's Llama 3 cluster states **1:7** between pods without giving the direction (arxiv.org). No source supports one universal ratio.

Degraded-state capacity must be recalculated, not asserted. NVIDIA states that its two-plane B300 fabric can continue on **half of the original bandwidth** after a single switch, transceiver or cable failure (docs.nvidia.com). That statement belongs to that design; a single-plane rail fabric loses the GPUs attached to

a failed rail switch (arxiv.org). In the worked example below, losing one of four spines in a plane takes a 1:1 leaf to **1.33:1** on that plane.

Topology capacity is only an upper bound on application goodput. NCCL's test tooling defines bus bandwidth with a correction factor of $2(n-1)/n$ for all-reduce so results can be compared with hardware limits (raw.githubusercontent.com). Acceptance therefore needs a topology file checked against the discovered fabric, counter baselines, and workload-representative collective tests, repeated in each failure state. All vendor figures are dated to documents opened on 2026-10-20 and tied to the exact hardware and release each describes.

Introduction and Background

This report is written for AI-fabric architects, network engineers and procurement teams who must decide how many leaf switches, spine switches, ports, optics and cables a GPU cluster needs, and whether a proposed topology stays acceptable after a link, switch or plane failure. Every vendor design in it is treated as a worked example tied to its own hardware. No benchmark or client result is claimed; every quantity is either quoted from a dated primary document or labeled as derived arithmetic.

The scope is the **scale-out compute fabric**: the network that connects GPU NICs across nodes. The scale-up domain (NVLink-class links inside a node or rack), the storage fabric, the customer or north-south network and the out-of-band management network are sized separately. They matter here because they compete for switch ports and change locality assumptions. A shared port must never be counted twice without an explicit multiplexing assumption.

Four terms recur. A **leaf** switch attaches endpoints and has downlinks (to NICs) and uplinks (to spines). A **spine** switch connects leaves. A **rail** is the set of NICs that share the same position in every node, for example NIC 3 of every server; in a rail-optimized fabric each rail terminates on its own leaf or group of leaves. A **plane** is an independent copy of the fabric; in a multi-plane design each NIC port or breakout leg attaches to a different plane.

Independent guides already discuss bisection and blocking; one notes that same-rail traffic stays one leaf hop only when both endpoints attach to that leaf (aidatacenterguide.com), a caveat that recurs below. What is usually missing is a reproducible input sheet that converts a frozen NIC topology into port counts, ratios and surviving capacity in each failure state, with the exact reference release behind every count.

Conventions: link rates are line rates per port in one direction unless stated; "400G link" means one 400 Gb/s port or breakout leg; counts marked "derived" are computed here from cited inputs and are not vendor statements. Figures are as of October 2026. Google describes its per-VM figures as maximum egress bandwidth that cannot exceed the stated number (docs.cloud.google.com), and none of the cloud pages opened for this report states whether a figure is per direction.

Key Changes in AI Fabric Design That Affect Sizing

Change 1: Radix and breakout now set the arithmetic

Current 51.2 Tbps switch silicon exposes **64 ports of 800GbE** on a single chip (investors.broadcom.com), and systems present them as 64 x 800G or 128 x 400G in two rack units (www.delltechnologies.com). Newer 102.4 Tbps silicon is built from **1,024 SerDes lanes of 100G** (investors.broadcom.com). Breakout is not free-form: one Cisco 64 x 800G datasheet lists 2x400 and 8x100 breakout (www.cisco.com), so a design that assumes 4 x 200G from that SKU's cage must be checked against the exact SKU and software release. On the NIC side, an 800G port can be broken out into two 400G legs, which is how one NIC reaches two planes (docs.nvidia.com).

Radix also defines **stranded ports**. If a leaf has 128 usable 400G ports and a design needs 96 downlinks and 24 uplinks, 8 ports are stranded unless storage, management or customer links can use them under a declared separation.

Change 2: Planes replace single-fabric redundancy

NVIDIA's DGX B300 compute fabric is twin-planar: each GPU has 2 x 400GbE through two different planes (docs.nvidia.com), and because the planes are not joined by a core layer, twice as many nodes fit on a two-layer fabric. DeepSeek's published multi-plane fat tree goes further, with eight GPUs and eight IB NICs per node and one plane per GPU-NIC pair (arxiv.org); each plane operates independently, so cross-plane traffic must use another NIC. Arista describes NICs that attach to one plane at 800G, two at 400G or four at 200G (www.arista.com), and Alibaba keeps traffic from NIC port 0 on port 0 of the destination, isolated from port 1. The Ultra Ethernet specification adds coordinated multi-path congestion control, which matters when traffic is sprayed across planes (ultraethernet.org).

The consequence is that every count becomes a per-plane count multiplied by the number of planes, and failure means losing a fraction of every endpoint's bandwidth.

Change 3: Rail optimization is a placement contract

NVIDIA's InfiniBand bring-up guide names fat tree as its recommended topology and says an AI factory should be rail-optimized (docs.nvidia.com); the stated purpose is that a node's internal NVSwitch path reaches a NIC that is just one switch hop away. The one-hop property holds only for ranks attached to the same leaf. In the DGX B300 design, each group of 64 nodes is rail-aligned, and traffic between nodes or between rails traverses the spine layer (docs.nvidia.com). Cloud documentation uses the same vocabulary at VM level: Google describes a datacenter-wide four-way rail-aligned network (docs.cloud.google.com), and its newest shape exposes an eight-way rail-aligned topology (docs.cloud.google.com). AMD states the trade-off directly: rail networks give larger single-hop ring domains, while cross-rail traffic must traverse the spine layer or Infinity Fabric (instinct.docs.amd.com).

Change 4: Oversubscription is declared, and conventions differ

AMD defines subscription as downstream demand to upstream capacity, so 1:1.16 means 0.16 more upstream capacity than downstream demand (instinct.docs.amd.com). Cisco labels a leaf with 32 uplinks and 8 downlinks "under-subscription 4:1" (www.cisco.com), Juniper calls a leaf with twice the leaf-to-spine bandwidth of server-to-leaf bandwidth "1:2" (juniper.net), and the same vendor calls a test with leaf-to-spine bandwidth cut to half "2:1" (www.juniper.net). The classic fat-tree paper defines the ratio against worst-case achievable host bandwidth and cites typical enterprise designs of 2.5:1 to 8:1 (cseweb.ucsd.edu). This report uses one convention: **oversubscription = aggregate downlink capacity / aggregate usable uplink capacity**, with the inverse shown whenever a source differs.

The Sizing Worksheet: From Endpoints to Ports

Steps 1 and 2: Freeze the endpoint and declare the traffic

Step 1 is an input sheet frozen before any switch is chosen. Record each item and mark unknowns as unknown:

- **Nodes (N), GPUs per node (G), NICs per node (n), ports per NIC (p).** Example: a Google A3 Ultra VM has eight ConnectX-7 NICs, one per GPU, delivering 3,200 Gbps in total (docs.cloud.google.com).
- **Link mode (s).** Line rate and breakout actually used, for example 800G native or 2 x 400G.
- **Rails (R) and planes (P).** Distinct NIC positions and independent fabrics.
- **Scale-up domain.** Which GPUs communicate over NVLink-class links and never touch the scale-out fabric.
- **Separate networks.** Storage, customer and management ports, and whether any share a leaf. DeepSeek, for example, gives each node a separate 400 Gbps Ethernet RoCE NIC for storage, off the compute planes (arxiv.org).
- **Growth unit.** The repeatable SU. NVIDIA advises building the full SU fabric, including leaf switches and leaf-spine cables, even when fewer nodes are installed (docs.nvidia.com).

Step 2 declares traffic assumptions, each labeled measured, vendor-stated or unknown:

- **Collective mix.** All-reduce, all-gather and reduce-scatter shares, and the message sizes they use.
- **Rail-local share.** The fraction of traffic between ranks that sit on the same rail and leaf.
- **Any-to-any share.** Mixture-of-experts all-to-all and other patterns that cross rails.
- **Storage bursts and tenants.** Burst rates, simultaneous jobs and how they share uplinks.
- **Post-failure behavior.** The minimum bandwidth each job must retain for the cluster to count as acceptable.

A measured study of GPT-class training reported that tensor-parallel traffic, which stays inside the NVLink domain, was over 75% of transmitted data (arxiv.org). That is a workload observation, not a constant.

Step 3: Endpoint capacity and rails

The endpoint link count is $L = N \times n \times p \times P$ when each port or leg lands on one plane, and endpoint capacity is $L \times s$. For the DGX B300 SU, **64 nodes x 8 NICs x 2 planes = 1,024 links of 400G** (derived), which is **409.6 Tb/s** per direction, or 800G for each of 512 GPUs. The cited NVIDIA pages do not state whether per-GPU figures are one-way or bidirectional, so this report treats them as per-direction line rate and flags the assumption. For bandwidth calculations, note that 400 Gb/s is 50 GB/s before protocol overhead (derived), and compare measured NCCL bus bandwidth against that line rate rather than against a marketing aggregate. Cloud shapes show the same pattern at node level: Azure's ND H100 v5 gives each of 8 GPUs a dedicated 400 Gbps InfiniBand connection (learn.microsoft.com), an AWS P6-B200 instance has 8 network cards of up to 400 Gbps each (docs.aws.amazon.com), and an Oracle B300 shape lists 8 x 800 Gbps RDMA (docs.oracle.com).

Step 4: Size the leaf

For a leaf with radix r (in the same link units), the identity is $r = d + u + x + z$, where d is downlinks, u is uplinks, x is reserved or service ports and z is stranded radix. Then:

- **Oversubscription $\rho = (d \times s_{\text{down}}) / (u \times s_{\text{up}})$** , and the inverse is $(u \times s_{\text{up}}) / (d \times s_{\text{down}})$.
- **Non-blocking** requires ρ of 1 at every leaf, which Cisco defines as an equal number of downlinks to GPU nodes and uplinks to spines (www.cisco.com).

Worked example for the B300 SU (derived from published counts): the switch is an SN5600 or SN5610 class Spectrum-4 device, which belongs to the same 64 x 800G or 128 x 400G radix class as other 51.2 Tbps switches (www.arista.com). With 16 leaves per SU and 1,024 endpoint links, each leaf carries 64 endpoint links, which is one rail on one plane across 64 nodes. A non-blocking leaf then has 64 uplinks, so $64 + 64 = 128$ and nothing is left for storage or management, consistent with those networks being separate fabrics. Juniper's stripe makes the same choice at lower radix: 64 x 400G leaves split 32 down and 32 up (www.juniper.net). Arista's distributed Etherlink leaf goes the other way on purpose, with 18 host ports to 20 spine ports (a ratio of 0.9:1 in this report's convention), which Arista says absorbs an uplink or spine failure without congestion (www.arista.com).

(Hypothetical Example) A buyer wants fewer spines and accepts a **3:1** declared ratio on a 128 x 400G leaf: 96 downlinks and 32 uplinks leave 0 stranded ports and carry 38.4 Tb/s down against 12.8 Tb/s up. If the same buyer asks for 100 downlinks, 28 uplinks and 0 spare, ρ rises to 3.57:1, a 19% increase in the declared ratio that port counts alone would hide. Each declared ratio must be tied to the step 2 traffic, for example that no more than one third of endpoint bandwidth ever crosses the leaf.

Step 5: Size the spine and bound the bisection

Spine sizing uses three quantities per plane:

- **Spines per plane** = $\text{ceil}(\text{leaves} \times \text{uplinks per leaf} / \text{spine radix})$.
- **Spine ports used** = $\text{leaves} \times \text{uplinks per leaf}$, which should be checked against the spine radix to find stranded spine ports.

- **Inter-switch link (ISL) ends** = twice the leaf-to-spine link count, which is also the transceiver or cable-end count if each link has an optic at both ends.

For the B300 4-SU build: 32 leaves per plane x 64 uplinks / 128 = 16 spines per plane, 32 in total, matching the published 64 leaf and 32 spine (docs.nvidia.com). Cisco's units give 8 leaves x 32 uplinks / 64 = 4 spines for 256 GPUs (www.cisco.com) and 16 leaves with 8 spines for 512 GPUs.

The two-tier ceiling follows from radix. If every leaf has $r/2$ downlinks and $r/2$ uplinks and every spine port reaches a different leaf, the fabric reaches at most **$r \times r / 2$ endpoint links** with r leaves and $r/2$ spines, or $3r/2$ switches. For $r = 64$ that is 2,048 endpoints and 96 switches, which is exactly the FT2 two-layer fat tree in DeepSeek's published comparison (Table 3); for $r = 128$ it is 8,192 endpoints, matching Arista's statement that a 64 x 1.6T switch supports 8,192 x 800G clients in two tiers (www.arista.com). A k -ary three-tier fat tree built from k -port switches supports $k^3 / 4$ hosts (cseweb.ucsd.edu), which is 65,536 for $k = 64$. Beyond the two-tier ceiling a third tier or another plane is required.

Bisection bounds must be computed for declared partitions, never as one number:

- **Leaf-local partition (rail-local)**. If both sides of the cut are on the same leaf, the bound is the leaf's downlink capacity and no uplink is used. This is the one-hop case described above.
- **Balanced leaf partition**. Half the leaves on each side. The upper bound is $\min(\text{downlinks on one side, uplinks on one side}) = (E \times s / 2) / \max(1, \rho)$. For the B300 SU split by rails (4 against 4), that is 204.8 Tb/s per direction at 1:1 and 68.3 Tb/s at a declared 3:1 (derived).
- **Lower bound**. With static hashing, no bound above zero is guaranteed for an arbitrary pattern; fat trees are rearrangeably non-blocking, which means arbitrary patterns can be carried only with careful path assignment (cseweb.ucsd.edu).

These are topology capacity bounds. Meta reported a max-mean ratio above 1.2 for 1,000 flows on 16 links with plain ECMP (engineering.fb.com), and DeepSeek reports that training flows tend to converge on the same interconnect link (arxiv.org). Application goodput, congestion behavior and collective completion time require measured validation.

Rail, Tree and Hybrid Placement

What a rail guarantees, and what it does not

AMD distinguishes a "fat tree", in which all of a host's NICs terminate on the same switch, from a rail network in which NICs are spread across switches. A two-tier tree does not need spine switches until a second scalable unit is deployed, whereas a two-tier rail network needs spines from the first deployment to connect the rails (instinct.docs.amd.com). AMD also lists the cost: more networking hardware. Juniper's rail-optimized stripes set the leaf count by GPUs per server (juniper.net), and Arista's validated pods typically hold 64 servers and 512 GPUs with each server's NICs fanning out across eight leaf switches (www.arista.com).

Rails are valuable only when software cooperates. The [NVIDIA Collective Communications Library](https://nvidia.com) (NCCL) setting `NCCL_CROSS_NIC=0` always uses the same NIC for a ring or tree to avoid crossing rails (docs.nvidia.com), while `NCCL_CROSS_NIC=1` suits networks where a node's NICs share a switch. When a

communicator holds only a subset of GPUs, rail-aligned rings may be impossible without PXN (PCI x NVLink), and PXN moves data over NVLink to a GPU on the same rail as the destination instead of crossing the spine. Rail traffic that spans multiple SUs still uses the spine, so a rail assumption can hold in the design and still miss in the schedule.

Workload-fit matrix

Table 1 below contrasts the placements by workload. Entries mix vendor statements, published research and arithmetic from this report; "Hybrid" means a rail layout with a deliberately provisioned cross-rail or scheduled path.

Workload	Tree (all NICs on one leaf)	Rail-optimized	Hybrid / scheduled
Ring or tree all-reduce, rail-aligned ranks	Every NIC of a node shares a leaf, so locality needs scheduler support.	Same-rail ranks reach each other in one leaf hop inside a rail-aligned group (see Change 3).	Rail layout plus a deep-buffer or scheduled spine for cross-SU rings (see Implications).
All-to-all and mixture-of-experts	Strong any-to-any behavior, because every NIC of a node reaches the same leaf.	Cross-rail traffic goes to the spine or over NVLink first; one study measured all-to-all slowdowns of 8.2% to 11.2% for a rail-only design (arxiv.org).	Alibaba kept an any-to-any second tier because a rail-only one would reach over 120K GPUs per pod but not carry cross-rail traffic (enanzhai.github.io).
Storage bursts	Storage shares leaf uplinks unless separated.	Storage kept off compute rails (see Step 1).	Dedicated storage leaves with a declared ratio.
Multi-tenant contention	Tenants share uplinks; ECMP collisions grow with fewer, larger flows.	Rail isolation helps only if tenants are rail-aligned.	Meta notes cross-zone links are oversubscribed by design and uses scheduler placement to reduce cross-zone traffic (engineering.fb.com).
Locality assumption broken	Uplink load rises directly.	Traffic crosses the spine; PXN may rescue some patterns.	Depends on provisioned cross-rail capacity.

The matrix makes one rule visible: choose placement from the traffic matrix, then size the cross-rail path for the traffic that leaves the rail rather than assuming it is zero. Where all-to-all dominates, an any-to-any second tier costs more but removes a locality assumption that the scheduler would otherwise have to enforce.

Failure and Degraded-State Capacity

"Redundant" is not a number. For each failure case, recompute the surviving links, the new ratio $\rho' = d_{\text{surviving}} / u_{\text{surviving}}$, and whether every endpoint still has a path. NVIDIA's DGX B300 documentation says a single switch, transceiver or cable failure will not lead to a catastrophic job abort and that the application can continue on half of the original bandwidth (docs.nvidia.com). That is a supported-behavior statement for that design and release; it is not a property of "dual plane" in general.

Table 2 below recalculates the DGX B300 single-SU example (16 leaves, 8 spines, 64 endpoint links and 64 uplinks per leaf, two planes; derived from the published counts) under assumed even uplink distribution across the four spines of each plane.

Failure case	Surviving endpoint capacity	Leaf ratio afterwards	Connectivity	Basis
One 400G node-to-leaf link	1,023 of 1,024 links (99.9%); the affected GPU runs at 400G of 800G	Unchanged at 1:1	All GPUs connected	NVIDIA's failure statement above
One 400G leaf-to-spine link	100% of endpoint links; cross-leaf capacity for that leaf $63/64 = 98.4\%$	$64/63 = 1.016:1$	All connected	Derived
One spine (of 4 in a plane)	Cross-leaf capacity 75% on that plane, 87.5% averaged over both planes	$64/48 = 1.33:1$ on that plane	All connected	Derived; assumes even spread and working load balancing
One leaf (one rail, one plane)	960 of 1,024 links (93.75%); 64 GPUs drop to 400G	Other leaves unchanged	All connected via the other plane	NVIDIA's failure statement above
Whole plane	50% of endpoint bandwidth and 50% of bisection	1:1 on the surviving plane	All connected	Extrapolated from the half-bandwidth statement; validate
Both leaves of one rail	64 GPUs have no direct fabric path	Not applicable	Not covered by the cited statement; PXN could route through other rails' NICs	Label unsupported until tested
Single-plane rail design, one leaf lost	The rail's GPUs lose the network	Other leaves unchanged	Those GPUs unavailable	Rail-only analysis of failed rail switches (below)

The table shows why plane count dominates the failure story. With two planes, any single leaf or spine loss leaves every endpoint attached and the cost is bandwidth. With one plane, a leaf loss removes endpoints entirely unless the design adds redundant rail switches, as the rail-only analysis notes operators can (arxiv.org). In a declared-oversubscribed fabric the same event is worse: a 3:1 leaf with 8 of 32 uplinks lost becomes 4:1 (derived), and the assumption that only one third of traffic crosses the leaf must be re-checked against the surviving ratio. DeepSeek's paper describes a multi-port NIC design in which single-port failures do not disrupt connectivity (arxiv.org), which is the same plane logic applied at the NIC.

Routing and failure interact; equal-cost multipath (ECMP) hashing and adaptive routing behave differently after a link is lost. Alibaba's design has 60 aggregation switches per plane, so one failure leaves 59 equivalent paths (enanzhai.github.io). Arista documents link-failure steering in under 500 ns on its 7060X6 (www.arista.com). Meta reports that centralized traffic engineering can be outperformed by enhanced ECMP once failures push availability below a 1:1 subscription ratio (engineering.fb.com). For acceptance, a design should record, per failure state, the routing mode, the surviving ratio and the measured collective bandwidth.

Implementation Considerations and Process Changes

Reconcile the BOM with the validated design

Reference bills of materials are representative, not final: NVIDIA states that its component list must be finalized based on the actual design (docs.nvidia.com). The published tables also use different counting units. For the 4-SU B300 build, one table lists 2,048 leaf-spine cables (docs.nvidia.com) while the bill of materials lists 4,096 leaf-spine cables and 4,096 twin-port leaf-to-spine transceivers (docs.nvidia.com). The counts reconcile if the first counts twin-port assemblies (two 400G links each) and the second counts 400G legs, which is this report's inference, not a vendor statement. A worksheet must state its unit: links, cables, assemblies or transceiver ends.

Other BOM items to reconcile against the vendor design:

- **Cables and ends.** For the HGX B300 design NVIDIA recommends a dual-ported optic to simplify breakout at the ConnectX-8 OSFP port, and split cables need explicit port configuration.
- **Management and monitoring ports.** Unified Fabric Manager (UFM) nodes consume four FNM ports on the Q3400 switches in the XDR design, and the B200 design removes one node from the SU to make room for UFM connectivity.
- **Customer and storage connections.** These ports come from a different budget than compute uplinks and must be counted separately.
- **Front-panel density.** Up to 36 OSFP ports fit in a 1U front panel under the OSFP MSA (osfpmasa.org), so a 2U switch exposing 64 cages is within that figure (derived).
- **Interchangeability.** Port modes must be checked per SKU even where a vendor treats two switch models as interchangeable.

Power, rack space and optics

Power and space scale with switch and module counts. NVIDIA lists a typical 940 W for an SN5600 with passive cables (docs.nvidia.com). Cisco lists 2,270 W maximum for its 64 x 800G N9364E-SG2 (www.cisco.com), Dell lists 1,500 W typical for the 64 x 800G Z9864F-ON (www.delltechnologies.com), and Juniper lists 613 W typical for a QFX5240 measured with DACs and without transceivers (www.juniper.net). Arista's distributed Etherlink leaf draws 593 W typical and 1,840 W maximum (www.arista.com). Typical figures exclude optics and are measured at a stated load, so they cannot be compared with maximums. Arista quotes under 16 W typical per 800G port (www.arista.com) and 16 W maximum for several 800G optical module types (www.arista.com), while module power classes differ by reach and type.

(Hypothetical Example) For the 4-SU B300 build, 96 compute switches at a typical 940 W would draw about 90 kW (derived), and the 4,096 leaf-spine twin-port transceivers at an assumed 16 W each would add about 66 kW (www.arista.com); actual modules, firmware and temperature set the real figure, so substitute SKU-level values from the vendor design before using either number.

Reach decides copper versus optics. Arista states that 1 m and 2 m 800G breakout cables support both switch-to-switch and switch-to-NIC use (www.arista.com); and IEEE's 800G objectives set copper over four pairs at at least 1.0 m and eight-pair single-mode at at least 500 m (ieee802.org). Leaf-spine links across racks are therefore optical in practice.

Acceptance plan

1. **Port map and topology file.** NVIDIA's bring-up guide derives a topology file from the point-to-point plan and reuses it for validation (docs.nvidia.com). The ibdiagnet topology comparison checks that the actual fabric matches the designed topology (networking-docs.nvidia.com).
2. **Cable validation.** Compare real cabling to the plan using the vendor's validation tool.
3. **Counter baselines.** UFM telemetry collects over 120 unique counters per port (docs.nvidia.com); record them at sign-off.
4. **Log review.** Review subnet manager and switch logs; treat errors as fabric-health indicators.
5. **Collective tests.** `nccl-tests` reports algorithm and bus bandwidth, and bus bandwidth should reflect the speed of the hardware bottleneck (raw.githubusercontent.com). For all-reduce the correction factor follows from $2(n-1)$ data transfers (raw.githubusercontent.com).
6. **Failure drills.** Repeat step 5 with a link, a spine and a plane removed, and compare with the failure matrix.
7. **Host-side pinning.** Use exact HCA matching in NCCL, because prefix matching can select unintended devices (docs.nvidia.com), and supply a single-host topology file when overriding detection.

Data Analysis and Evidence

Table 3 below collects published reference designs so the derived per-leaf split can be compared. Counts are each vendor's; the ratio column is derived here as a consistency check unless a citation states otherwise. Each row describes the exact hardware named and is not a general performance promise.

Design (source)	Endpoint mix and unit	Leaf / spine counts	Stated or derived ratio
DGX B300, Spectrum-X Ethernet	64-node SU, 2 x 400G per GPU in two planes	16 leaf, 8 spine per SU (docs.nvidia.com)	Derived 64 x 400G down / 64 x 400G up per leaf, 1:1
HGX B300 AI Factory (enterprise RA)	4-node SU, 8 ConnectX-8 NICs per node	32 nodes: 8 leaf, 4 spine, both planes (docs.nvidia.com)	Derived 64 x 400G down / 64 x 400G up per leaf
RTX PRO AI Factory (enterprise RA)	4-node SU, 16 x 400G per SU	Layouts larger than 4 SUs are an exception to the standard architecture (docs.nvidia.com)	Described as a full non-blocking fat tree
Cisco 64 x 400G leaf and spine	256 GPUs, 400G per GPU	8 leaf, 4 spine; 512 GPUs: 16 leaf, 8 spine (www.cisco.com)	Non-blocking; equal downlinks and uplinks
Juniper QFX5230 stripe	256 H100 GPUs per stripe (www.juniper.net)	8 leaf, 4 spine	32 x 400G down and up per leaf
Arista 7060X6 two-tier	8,192 accelerators	128 leaf, 64 spine	Derived 1:1
Arista 7700R4 distributed Etherlink	Up to 16 accelerators per leaf (www.arista.com)	Leaf has 18 host and 20 spine ports	0.9:1 by design
DeepSeek multi-plane fat tree	8 GPUs and 8 IB NICs per node	64-port switches; up to 16,384 GPUs in theory (arxiv.org)	Independent planes

Design (source)	Endpoint mix and unit	Leaf / spine counts	Stated or derived ratio
Meta Llama 3 RoCE cluster	3,072-GPU pod with full bisection (arxiv.org)	Pods joined by an aggregation layer	1:7 between pods, direction not stated
Alibaba HPN tier 1	1,024 GPUs per segment (enanzhai.github.io)	51.2T ToR, 128 active plus 8 backup 200G down, 60 x 400G up	Derived 1.067:1

The table's main lesson is that leaves are usually built as 1:1 boxes, which matches Arista's statement that two-tier AI designs are typically provisioned at 1:1 or overprovisioned for headroom (www.arista.com), and Juniper's recommendation of a 1:1 subscription factor for its GPU backend (www.juniper.net): the NVIDIA, Cisco, Juniper and Arista 7060X6 counts above are consistent with equal downlinks and uplinks (the Arista 7700R4 is an exception at 0.9:1), and Juniper's AMD-based test topology is deliberately provisioned at 1:2 for headroom. The exceptions are documented and deliberate: AMD's recommended 1:1.16 means 16% more upstream capacity than downstream demand, or rho of 0.862 in this report's convention (derived: $1 / 1.16$); Alibaba's 1.067:1 at tier 1 pairs with 15:1 at its core layer (enanzhai.github.io); and Google states that parts of its fabrics were intentionally kept oversubscribed to save cost (conferences.sigcomm.org). Depopulated growth is also a documented pattern: Google's earlier fabrics supported deployment at 50% of maximum bisection bandwidth and in-place upgrade (conferences.sigcomm.org), the opposite of building the full SU fabric first.

Three data points qualify any cost argument. First, DeepSeek's cost comparison reports that its multi-plane fat tree reaches 16,384 endpoints at the same cost per endpoint as a two-layer tree, and below a three-layer tree ([Table 3 of the DeepSeek paper](#)). Second, a rail-only study reported network cost reductions of 38% to 77% against a full-bisection Clos (arxiv.org), a model that depends on cross-rail traffic being rare. Third, Alibaba reports that 96.3% of its in-production training jobs use fewer than 1,000 GPUs (enanzhai.github.io), which supports sizing the non-blocking unit around the segment rather than the whole pod.

Bandwidth-quoting conventions need an explicit normalization column in any worksheet. Cisco's datasheet counts 102.4 Tbps for a 64 x 800G switch, while 64 x 800G is 51.2 Tbps in one direction. Cloud providers quote node totals such as 3.2 Tb/s per VM in a non-blocking fat tree (azure.microsoft.com), and CoreWeave describes both its InfiniBand and Ethernet fabrics as non-blocking fat trees (docs.coreweave.com). Never add figures of unknown direction or unit.

Implications and Future Directions

Switch radix keeps rising, and the two-tier ceiling rises with its square. Broadcom positions its 102.4 Tbps Tomahawk 6 for two-tier scale-out networks of more than 100,000 XPU's at 200 Gbps per link (investors.broadcom.com) and lists Clos, rail-only and rail-optimized topologies (investors.broadcom.com). Treat such figures as the silicon's ceiling under a stated link rate; the worksheet still has to be run with the actual NIC mode, breakout and reserved ports. Scheduled fabrics replace the spine with deep-buffer elements to reach larger clusters; AMD describes a Jericho3-AI/Ramon3 design (instinct.docs.amd.com) and Arista reports over 27,000 x 800GbE host ports with a super-spine layer (www.arista.com). The spine then behaves as one logical switch, which changes failure modeling.

For procurement, three requests follow from the worksheet: a per-leaf port table with stranded and reserved ports, a spine and optics BOM with its counting unit, and a failure matrix signed off against the vendor's supported-behavior statement. The numbers must come from the vendor's current release, because link modes, breakout support and validated software change between releases.

Open questions remain. No public benchmark in the sources opened for this report measures goodput versus oversubscription for a representative mix of collectives, and the vendor failure statements cover single failures rather than correlated ones. Meta's Llama 3 collective library opens 16 flows between two GPUs to give ECMP more entropy (arxiv.org), a reminder that routing behavior is part of the effective ratio.

Frequently Asked Questions (FAQs)

How many switch ports does a GPU cluster need? Count endpoint links ($N \times n \times p \times P$), add an equal number of uplinks for a non-blocking leaf, then add spine ports equal to total uplinks. For 4 SU of DGX B300 that is 2,048 GPUs with 64 leaf and 32 spine switches. Add reserved ports for storage, management and customer networks separately.

Is InfiniBand or Ethernet better for an AI training cluster? The arithmetic is the same; the differences are radix, load-balancing features and support. NVIDIA validates both a Spectrum-X Ethernet B300 design and a Quantum-X800 InfiniBand design, and DeepSeek notes IB switches typically offer 64 ports against 128 commonly found on RoCE switches (arxiv.org). Choose from the validated design for your GPU, not from a protocol ranking.

How is a fat tree sized? A three-tier tree of k -port switches supports $k^3 / 4$ hosts (cseweb.ucsd.edu); two tiers support about $r^2 / 2$ endpoint links at 1:1 (derived).

What happens to bandwidth when a link fails? Recompute it. For NVIDIA's B300 design, half of the original bandwidth is documented for a single switch, transceiver or cable failure; other fabrics must be recalculated and tested.

Is a 1:1 ratio required? Not always. Reference designs are mostly 1:1, while AMD recommends 1:1.16 and some hyperscale layers are deliberately oversubscribed. The ratio must follow the declared traffic and the post-failure ratio.

Is rail-optimized always better than a tree? No. Rails help rail-aligned collectives and need spines to connect rails; trees favor any-to-any traffic.

Conclusion

Fabric sizing for a GPU cluster reduces to a short, auditable chain: freeze the endpoint list, convert it to links at a declared rate and plane count, split each leaf into downlinks, uplinks and reserved ports, size the spine from total uplinks, and bound the bisection for named traffic partitions. The ratio that results is a decision about traffic, not a property of the topology name, and it must be restated after each failure.

The reference architectures studied here are most valuable as worked examples tied to their own hardware, link modes and releases. They do not promise performance for other fabrics, and their failure statements do not transfer. The remaining uncertainty is measured behavior: topology capacity bounds achievable goodput but does not predict it. [GPU cluster acceptance testing](#), counter baselines, workload-representative collectives and failure drills turn the worksheet's assumptions into evidence.

Source links

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

1. <https://investors.broadcom.com/news-releases/news-release-details/broadcom-ships-tomahawk-5-industrys-highest-bandwidth-switch>
2. <https://www.arista.com/assets/data/pdf/Whitepapers/7060X6-Series-AI-Networking-at-800G-White-Paper.pdf>
3. <https://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/nexus-9364e-sg2-switch-ds.html>
4. <https://docs.nvidia.com/dgx-superpod/reference-architecture/scalable-infrastructure-b300/latest/dgx-superpod-architecture.html>
5. <https://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/nexus-9000-series-switches-ai-clusters-wp.html>
6. <https://arxiv.org/pdf/2505.09343>
7. <https://instinct.docs.amd.com/projects/MI3XX-reference/latest/index.html>
8. <https://engineering.fb.com/wp-content/uploads/2024/08/sigcomm24-final246.pdf>
9. <https://ennanzhai.github.io/pub/sigcomm24-hpn.pdf>
10. <https://arxiv.org/pdf/2407.21783>
11. <https://docs.nvidia.com/dgx-superpod/reference-architecture/scalable-infrastructure-b300/latest/network-fabrics.html>
12. <https://arxiv.org/pdf/2307.12169>
13. <https://raw.githubusercontent.com/NVIDIA/ncccl-tests/master/doc/PERFORMANCE.md>
14. <https://aidatacenterguide.com/part-8-networking-fabrics-and-optics/8-5-scale-out-topology-sizing-and-oversubscription>
15. <https://docs.cloud.google.com/compute/docs/gpus/gpu-network-bandwidth>
16. <https://www.delltechnologies.com/asset/en-us/products/networking/technical-support/dell-powerswitch-z9864f-on-spec-sheet.pdf>
17. <https://investors.broadcom.com/news-releases/news-release-details/broadcom-ships-tomahawk-6-worlds-first-1024-tbps-switch>
18. <https://docs.nvidia.com/enterprise-reference-architectures/hgx-ai-factory/latest/network-logical-architecture.html>
19. <https://www.arista.com/assets/data/pdf/Whitepapers/AMD-Next-Gen-AI-White-Paper.pdf>
20. <https://ultraethernet.org/wp-content/uploads/sites/20/2025/06/UEC1.0Whitepaper.pdf>
21. <https://docs.nvidia.com/networking/display/infiniband-cluster-bring-up-procedure.pdf>
22. <https://docs.cloud.google.com/compute/docs/accelerator-optimized-machines>
23. https://juniper.net/documentation/us/en/software/jvd/jvd-ai-dc-apstra-amd/solution_architecture.html
24. https://www.juniper.net/documentation/us/en/software/jvd/jvd-ai-dc-apstra-nvidia-weka/solution_architecture.html
25. <https://cseweb.ucsd.edu/%7Evahdat/papers/sigcomm08.pdf>

26. <https://learn.microsoft.com/en-us/azure/virtual-machines/sizes/gpu-accelerated/ndh100v5-series>
27. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa-acc-inst-types.html>
28. <https://docs.oracle.com/iaas/Content/Compute/References/computeshapes.htm>
29. <https://www.juniper.net/documentation/us/en/software/ncce/ai-clusters-data-center-design/ai-clusters-data-center-design.pdf>
30. <https://www.arista.com/assets/data/pdf/Datasheets/7700R4-Distributed-Etherlink-Switch-Datasheet.pdf>
31. <https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/env.html>
32. <https://docs.nvidia.com/dgx-superpod/reference-architecture/scalable-infrastructure-b300/latest/components.html>
33. <https://osfpmsa.org>
34. <https://docs.nvidia.com/networking/display/SN5000/Specifications>
35. <https://www.juniper.net/us/en/products/switches/qfx-series/qfx5240-data-center-switches.html>
36. <https://www.arista.com/assets/data/pdf/Datasheets/7060X6-Datasheet.pdf>
37. <https://www.arista.com/assets/data/pdf/Datasheets/Transceiver-Data-Sheet.pdf>
38. https://www.arista.com/assets/data/pdf/Datasheets/Arista-800G_Optics_FAQ.pdf
39. https://ieee802.org/3/df/proj_doc/objectives_P802d3df_220317.pdf
40. <https://networking-docs.nvidia.com/ibdiagnetutilityum/2220/topology-comparison>
41. <https://docs.nvidia.com/enterprise-reference-architectures/rtx-pro-ai-factory/latest/network-logical-architecture.html>
42. https://www.juniper.net/documentation/us/en/software/jvd/jvd-ai-dc-qpp-bgp-dpf/ip_fabric_gpu_backend_fabric_architecture.html
43. <https://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p183.pdf>
44. <https://azure.microsoft.com/blog/azure-previews-powerful-and-scalable-virtual-machine-to-help-customers-accelerate-ai>
45. <https://docs.coreweave.com/docs/platform/networking/hpc-interconnect>

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.