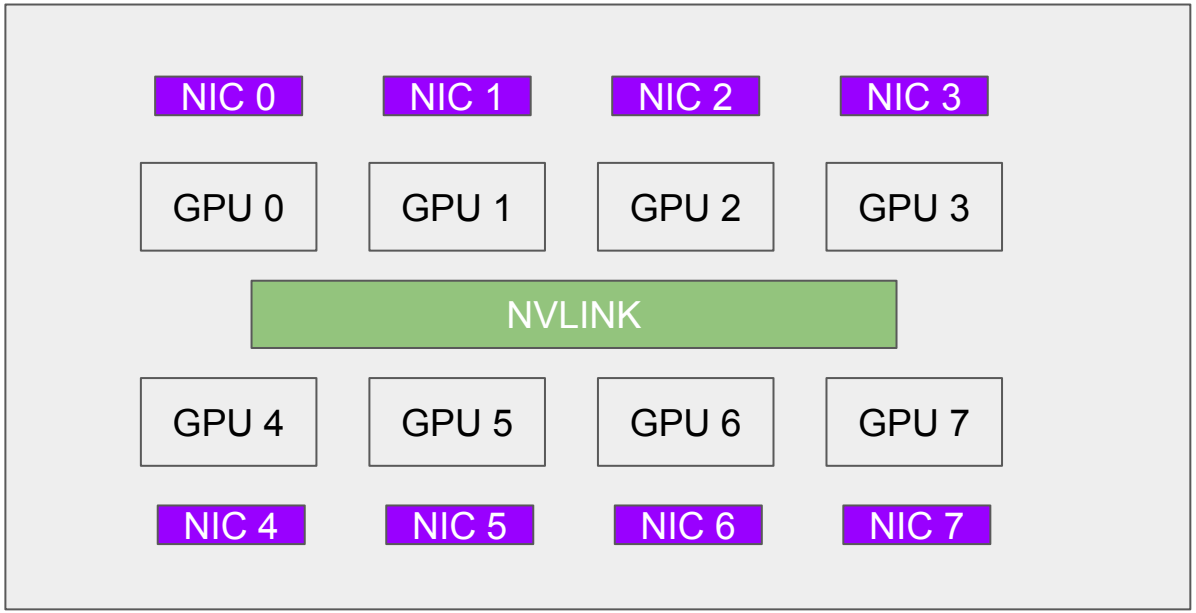
ToR: what is it good for?

And other routing stuff

Agenda

- How comm. collectives map onto network
- ToRs and Rails
- Rail-only topologies

A GPU node schematics (HGX H100 board)



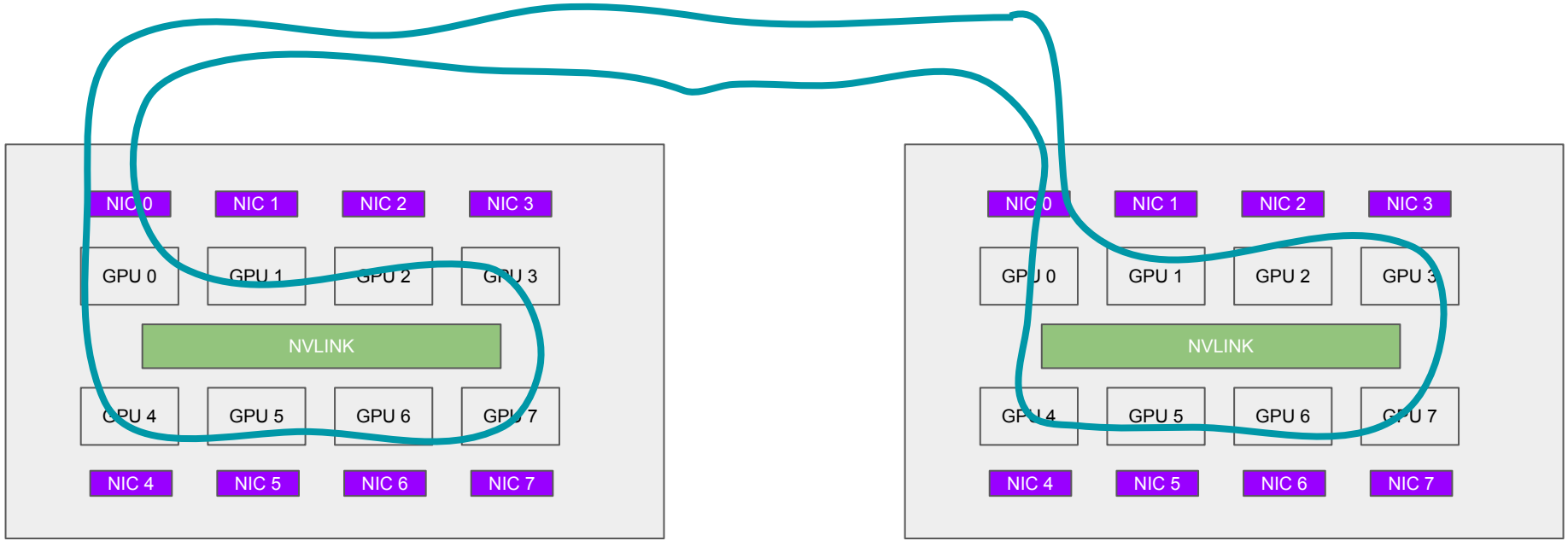
E.g. NIC = 50GB/s

—

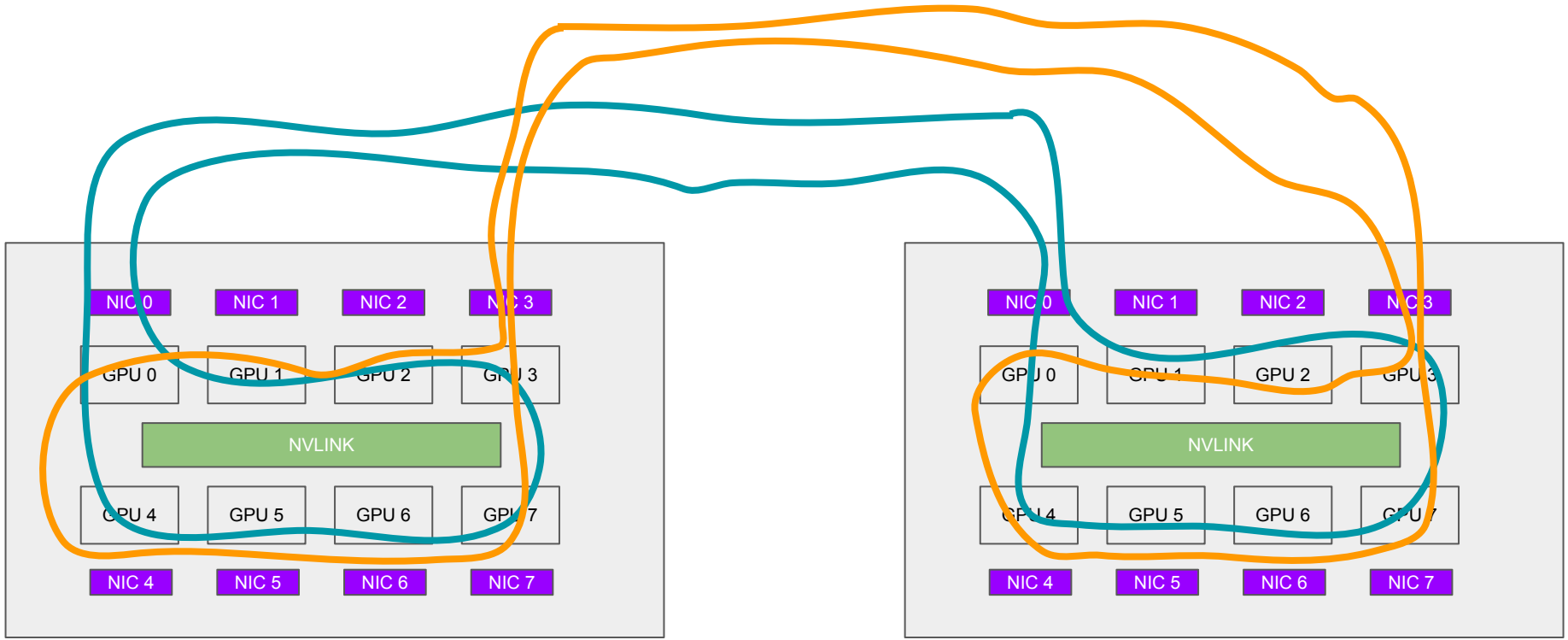
NVLINK = 450 GB/s

There are also CPUs somewhere… and PCIe switches

A collective traffic flow (ring-based) [1]



A collective traffic flow (ring-based) [2]

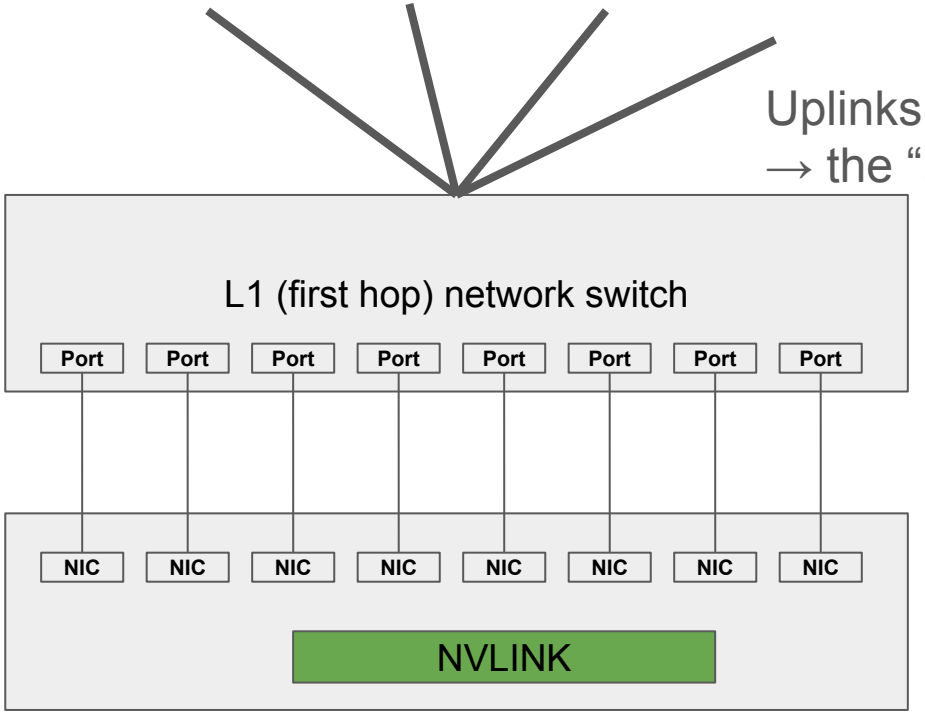


Things to pay attention to

- NVLink is part of collective communication flow!
- Ring “logical” topology by itself is bandwidth-optimal…
- … if it **maps** on the network properly (without overlaps)
- Similar logic applies to “non-ring” collectives (trees)

Node → Network connectivity: “ToR” style

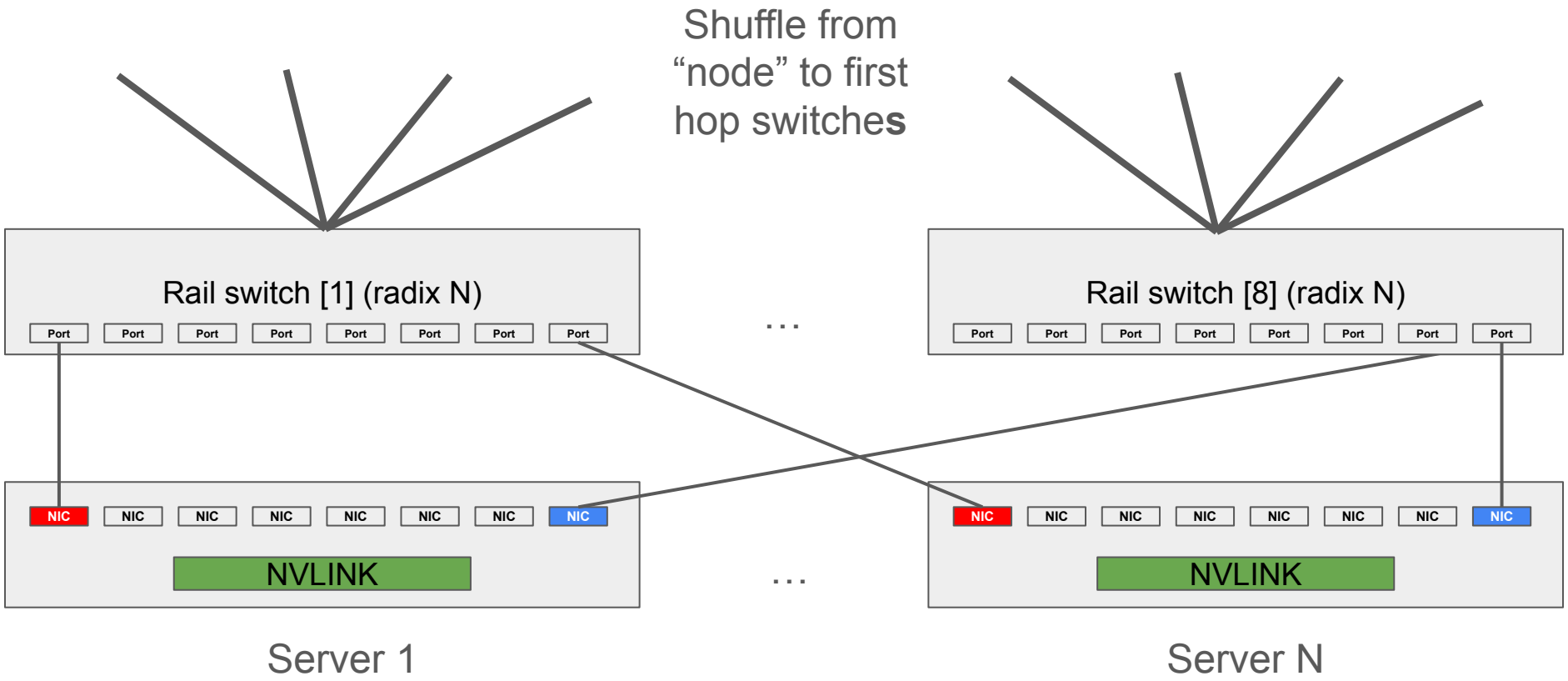
Uplinks (load-balanced)
→ the “rings” map here



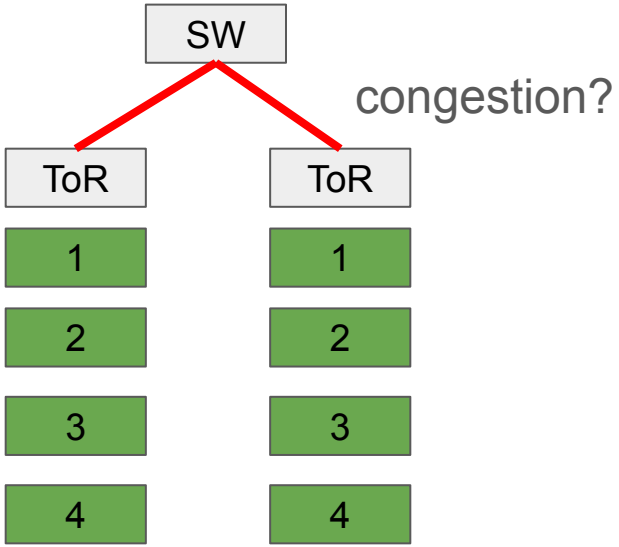
Parallel from
“node” to first
hop switch

Server

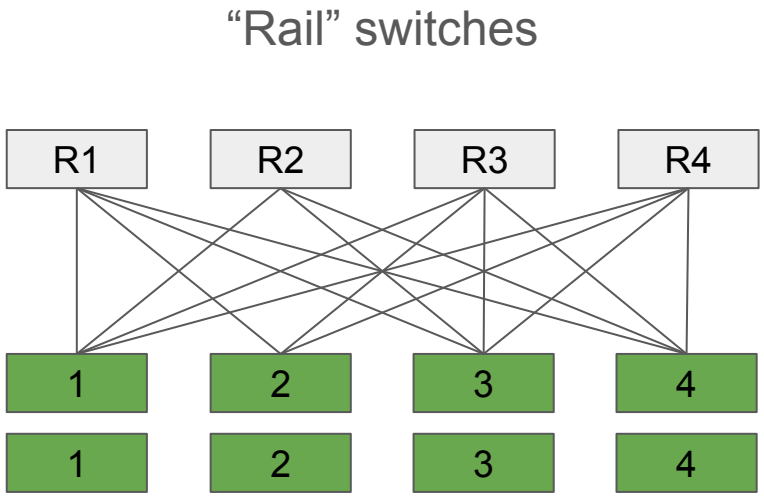
Node → Network connectivity: “Rails” style



Rail vs. ToR connectivity



Collective traffic has to cross
ToR → Next Level links



Within a “rail group” traffic does
not cross L1→L2 tree layers

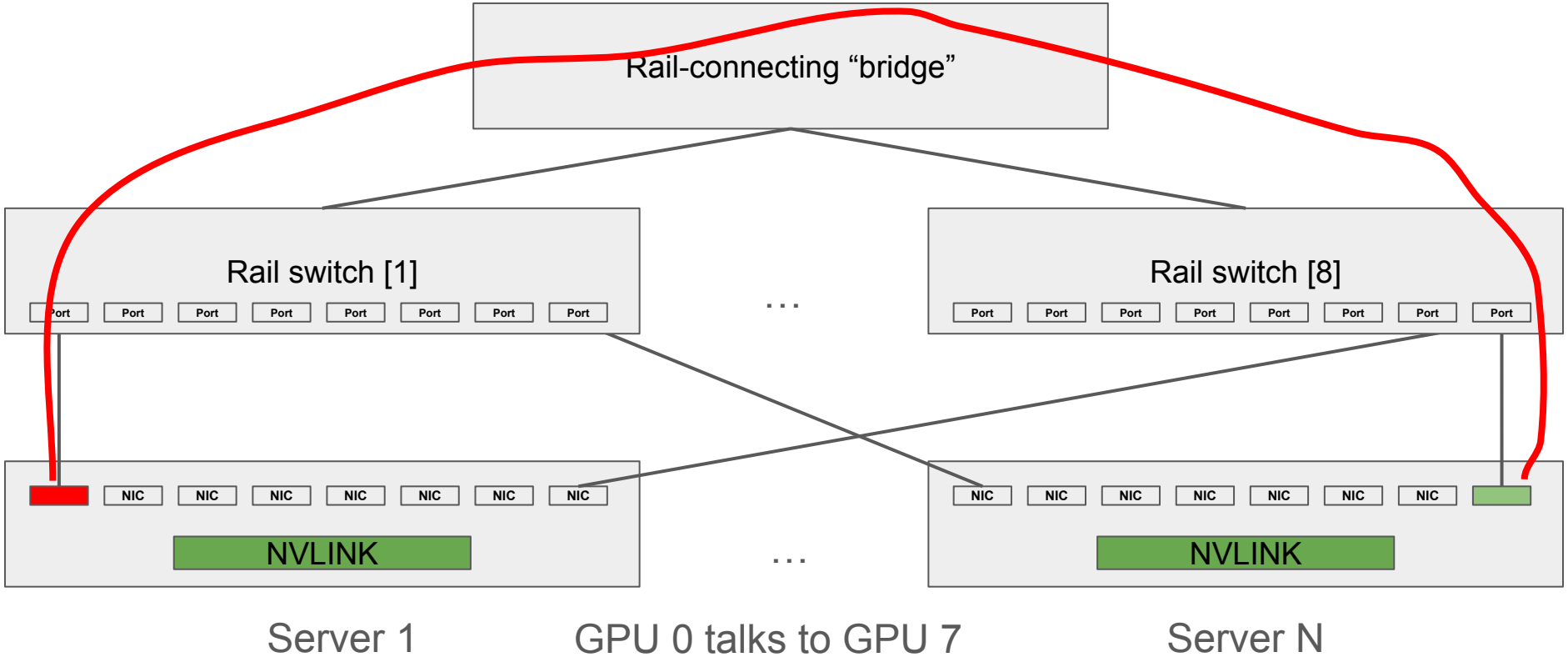
So how big is the gain?

- Depends how well “ToR” does load-balancing (adaptive routing)
- Rails benefit exists only for traffic “local” to the rial group
- With “basic ECMP” rail structure (first-hop shuffle) helps

Tradeoffs:

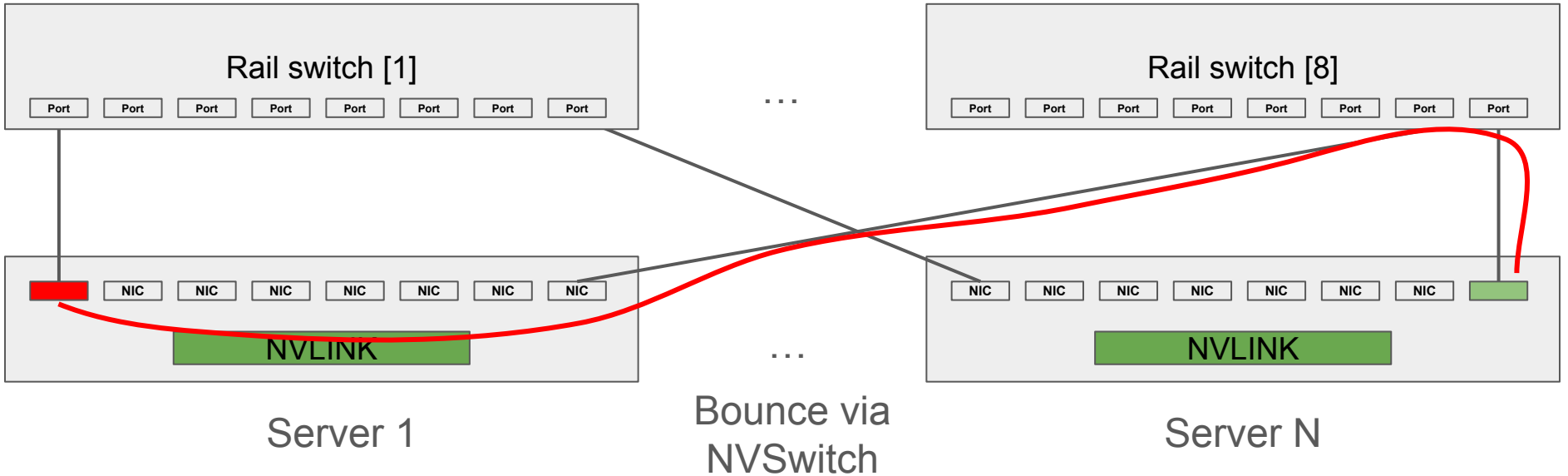
- Rail shuffle requires **optical connectivity** from NIC to the switch
- Rail switch maintenance affects large(r) group of nodes
- Switch are now “detached” from racks (testing is more complex)

Rail crossing (cross-rank talk): bridge switch layer



Rail crossing (cross-rank talk): NVLINK (PXN)

NVLINK is like a ToR!



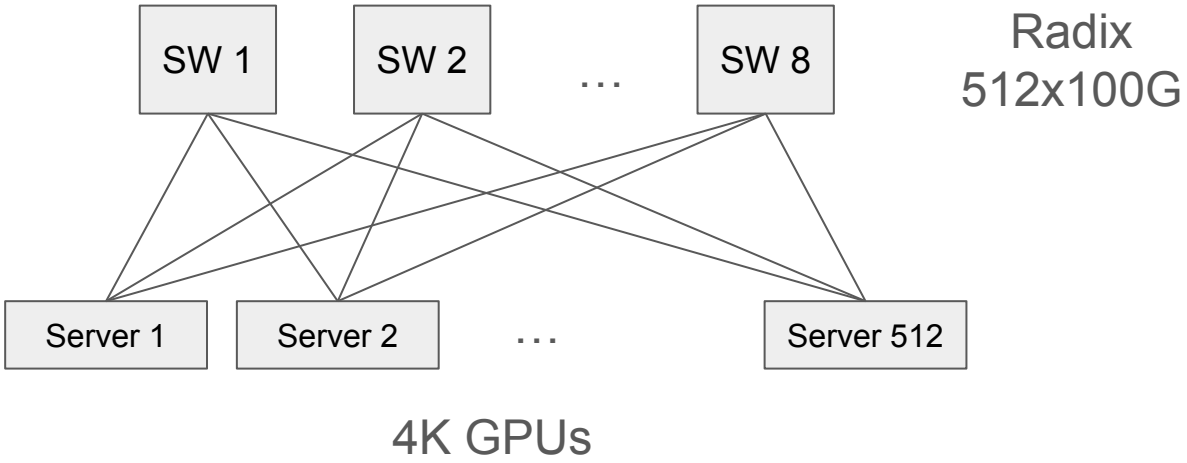
Do we even need to cross the rails?

- Depends on the application of course, duh
- If training job is “rail aligned” many collectives “stay on rails”
- But some application use all-to-all or send-recv communications
- Those must cross rails

Hint: training LLMs (transformers) has rail-aligned collectives (even MoE)

The “rail-only” network topology

- Use high-radix first-hop switches (or Clos “virtual” fabrics)
- Cross rails over PXN
- Profit?!



Is it such a capital idea?

Pros:

- You cut switch hops - avoid load-balancing
- You can layer of switches/optics - save $ and [W]
- You feel like riding a gravy train

Cons:

- You have scaling limits with one-hop switches
- Blast radius of maintenances is high
- No chance to use copper (depends if you care)
- Gets murkier with larger NVL domains (beyond node)