GPU and RDMA node debugging tools

One node from the training job down to the fabric. Each command is pinned to the component it reads. Find the sick component, run the commands on its row.

Linux, any vendor · **NVIDIA GPU and ConnectX** · **AMD Instinct and Pollara**

USER SPACE

```
pidstat -u -r -d -p <pid> 1   vmstat 1
cat /proc/pressure/{cpu,memory,io}
rdma resource show qp
```

```
ibv_devices
ibv_devinfo -d <dev>
```

everything below is read through these

Training job
PyTorch, JAX, MPI launcher

NCCL / RCCL
collectives

CUDA / ROCm HIP
GPU runtime

libibverbs, rdma-cm
RDMA user library

NVML, DCGM engine / AMD SMI library
management and telemetry

sysfs, procfs, netlink
the kernel's windows

```
nvidia-smi pmon -s um   dcgmi stats --pid <pid>
amd-smi process   amd-smi monitor
```

```
NCCL_DEBUG=INFO
rocminfo   NCCL_DEBUG=INFO (RCCL)
```

```
nvidia-smi -q   dcgmi discovery -l   dcgmi health -c
amd-smi static   amd-smi metric   amd-smi list -e
```

LINUX KERNEL

```
dmesg -T -l err,warn   journalctl -k -b -1
rdma dev   rdma link   lspci -k -s <bdf>
```

```
dmesg -T | grep -E "AER:|PCIe Bus Error"
cat .../<bdf>/aer_dev_correctable
ip -s -s link show   /proc/interrupts
```

GPU driver
nvidia.ko (NVRM) / amdgpu

RDMA core
ib_core, uverbs

NIC driver
mlx5 / ionic, ionic_rdma

PCI core, AER
hotplug

NUMA, IRQs
scheduler

netdev, DCB
PFC, trust

Kernel log
printk

```
dmesg | grep "NVRM: Xid"   nvidia-bug-report.sh
amdgpu RAS lines   amd-smi ras --cper  (AFID)
```

```
mlnx_qos -i <if>   cma_roce_mode -d <dev> -p 1
nicctl show qos pfc
```

HARDWARE IN THE NODE

```
numactl -H   lstopo-no-graphics
cat /sys/bus/pci/devices/<bdf>/numa_node
ipmitool sel elist   dmidecode -t bios
```

```
sudo lspci -vv -s <bdf>  (LnkCap, LnkSta)
lspci -tv   lspci -nn -d 10de: | 1002: | 15b3:
```

no vendor-neutral view: use the vendor tool

CPU 0 / NUMA node 0
cores, DIMMs

CPU 1 / NUMA node 1
cores, DIMMs

BMC and UEFI firmware
out of band, decides before Linux

PCIe: root ports, switches, retimers, slots
speed and width per link

GPU
HBM, SRAM (ECC)
GSP / PMFW firmware

GPU
HBM, SRAM (ECC)
GSP / PMFW firmware

GPU
HBM, SRAM (ECC)
GSP / PMFW firmware

GPU
HBM, SRAM (ECC)
GSP / PMFW firmware

no vendor-neutral view: use the vendor tool

Scale-up fabric: NVLink / Infinity Fabric xGMI
GPU to GPU links inside the node

```
rdma link show   rdma statistic show
ethtool -i <if>   ethtool -S <if>   ibstat
```

```
ethtool -m <if>
ethtool -I --show-fec <if>
```

RDMA NIC: ConnectX (mlx5) / Pollara 400 (ionic)
firmware, PFC, DCQCN, queue pairs

Optics and cable
PHY, FEC, lanes, transceiver

```
nvidia-smi topo -m
amd-smi static --numa
```

```
nvidia-smi --query-gpu=pcie.link.gen.current,
  pcie.link.width.current --format=csv
amd-smi metric --pcie   amd-smi static --bus
```

```
nvidia-smi -q -d ECC,ROW_REMAPPER
dcgmi diag -r 1..4   dcgmi dmon -e 230,311,395
amd-smi metric --ecc --ecc-blocks   bad-pages
rvs -c <conf>   amd-smi firmware
```

```
nvidia-smi nvlink -s   nvlink -e   dcgmi nvlink -s
amd-smi xgmi -l   xgmi -m   topology -o -t
```

```
mlxconfig -d <dev> q   mlxfwmanager --query
nicctl show card --detail   nicctl show dcqcn
nicctl show rdma queue-pair statistics
```

```
mlxlink -d <dev> -c -m   mlxlink --show_fec
nicctl show port   nicctl show port statistics
```

CLUSTER FABRIC

```
ib_write_bw --report_gbits   ib_write_lat
ping, ip route  (RoCE underlay)
```

```
lldpcli show neighbors ports <if> summary
rdma link show   ip -br link   ping -I <if>
```

Leaf and spine switches
PFC + ECN (RoCE), credits (IB)

Subnet manager
InfiniBand only

Peer node
the far end

Multiplanar: several independent planes per NIC
one leg per plane, no shared fate, a dead plane is silent

```
perfquery -x <lid> <port>   iblinkinfo   sminfo
ibdiagnet --get_phy_info  (InfiniBand)
ib_write_bw -d ionic_0 --use_rocm=<n>   rccl-tests
```

```
doca-xplane-client get-status   get-planes-summary
NCCL_DEBUG=INFO   NCCL_IB_HCA=<hca>:<port>:<rail>:<plane>
```

START HERE: THE ORDER TO CHECK

1. Host and firmware
`pidstat  /proc/pressure  ipmitool sel elist`
Is the rank starved? What did the BMC see?
clean? go to the next step

2. PCIe and NUMA
`lspci -vv  aer_dev_correctable  numactl -H`
LnkSta below LnkCap, errors at either end
clean? go to the next step

3. GPU and its memory
`Xid  nvidia-smi -q  amd-smi metric --ecc`
Xid or RAS block, remap or bad page pending
clean? go to the next step

4. Scale-up link
`nvlink -s -e  xgmi -l  topology -o`
Inactive or D link, hops 2 where 1 expected
clean? go to the next step

5. NIC, optics, planes
`rdma link  ethtool -S  mlxlink -c  lldpcli`
pauses, discards, BER, a missing plane
clean? go to the next step

6. Prove it with traffic
`ib_write_bw  then --use_cuda / --use_rocm`
below line rate, or GPU run far below host run
clean? go to the next step

How to use it: start at the component where the symptom shows, then walk down a row at a time until a number is wrong. Compare with a healthy twin node. Capture the support bundle before any reset.

Commands checked against man7.org, docs.kernel.org, docs.nvidia.com, rocm.docs.amd.com and docs.amd.com on 2026-11-15. Flags vary by version: see Appendix A.

Layout after Brendan Gregg's Linux observability tool maps (brendangregg.com). rik-kisnah.ai