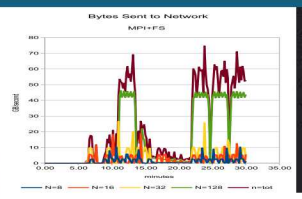
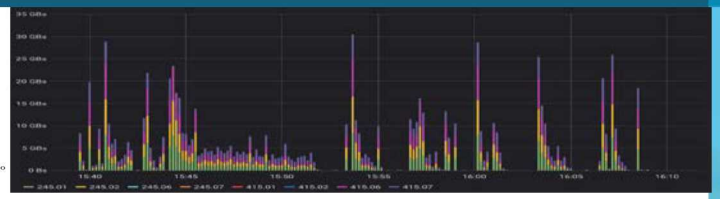
SAND2020-9599C

Sandia National Laboratories

# LDMS Monitoring of EDR InfiniBand Networks



*LDMSCON 2020*

Benjamin Allan, M. Aguilar, B. Schwaller, and S. Langer

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2020-xxxx C

# Abstract

We demonstrate high-rate (1hz), low-overhead, scalable collection of Infiniband port counter data on all ports in a network with an easily configured LDMS plugin.

This enables detailed performance analysis of network features such as adaptive routing, quality-of-service, in-network data reductions, and their impacts on HPC applications.

We present preliminary capability and performance results and some lessons learned about collecting data from switches at scale.

# Outline

- Why the ibnet sampler for LDMS?
- How do we make it manageable?
- What did we see in early production testing?
  - System analysis
  - Sampler performance
- What are we learning?
  - Recommendations for deployment
  - How to improve the sampler and ldmsd
- Available as part of OVIS-4 branch: https://github.com/ovis-hpc/ovis

# What we want

- We want to see all the network performance data all the time.
  - Not just end-points where LDMSD can be installed or applications which can be instrumented.
- We want the data at frequencies and times coherent with other LDMS data (i.e. 1/minute, 1Hz, or 0.1 Hz) for load analysis
- We want the data supported by MAD libraries from the latest hardware.
- Low overhead and scalable, like other LDMS plugins.

*Single-node subnet managers do not scale to give us what we want*

# Making it manageable

- Problems:
  - Thousands of ports on large systems, latency of MAD queries
  - Must expect failures and adapt to part replacements without daemon reconfiguration
  - Don’t turn a sampler into a subnet manager - decouple software expertise
- Solution:
  - Automatic fat-tree aware division of port query work among user supplied list of ldmsd sampler hosts.

    # (for i in $(seq 1 8); do echo admin$i; done) > ***Sampler_hosts_file***

    # (for i in $(extended xmtdisc rcverr flowctlcounters vlxmitcounters xmitcc ); do echo $i; done) > ***Subset_file***

    # ***ibnetdiscover -p*** --node-name-map ib-node-name-map > Indp_file

    # ***ldms-ibnet-sampler-gen*** --net Indp_file --samplers Sampler_hosts_file --out cluster_ib --sharp 37

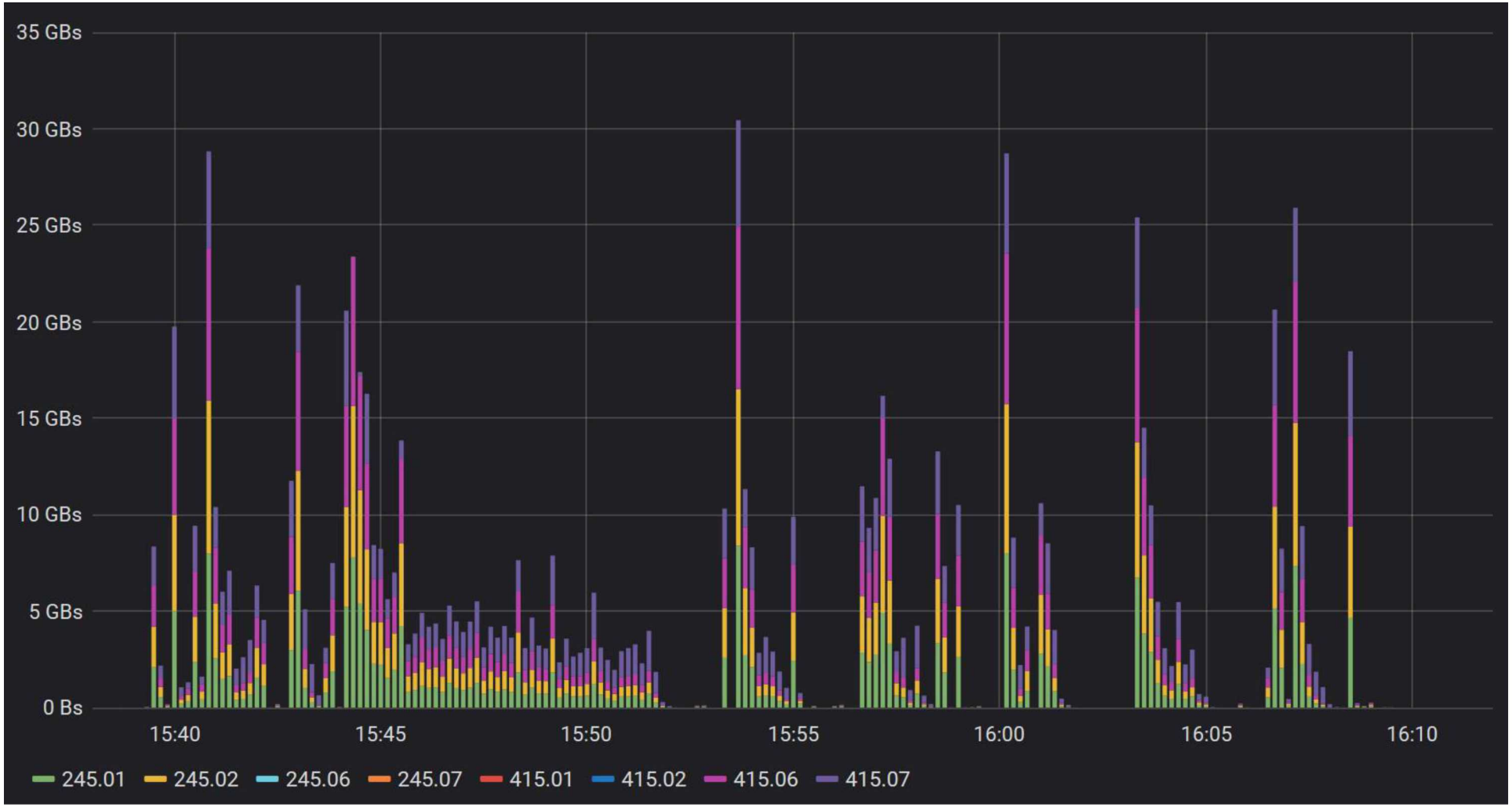
    →cluster_ib.$HOSTNAME.conf files

```
config name=ibnet source-list=cluster_ib.${HOSTNAME}.conf \
       metric-conf=Subset_file \
       node-name-map=ib-node-name-map \   ← (recent experience says eliminate this one)
       port-name=mlx5_0
```

# The test system for this work

- Sandia's Stria cluster:
  - A 300 node HPE ThunderX2 (ARM 64 bit) cluster.
  - Dual 28 core CPUs.
  - Socket-direct Connect-X 5 EDR Infiniband.
  - Dedicated Lustre servers: 2 metadata & 4 object storage.
- Stria is the production testbed for the 2594 node cluster Astra.

# System analysis (Lustre in-bound bandwidth)

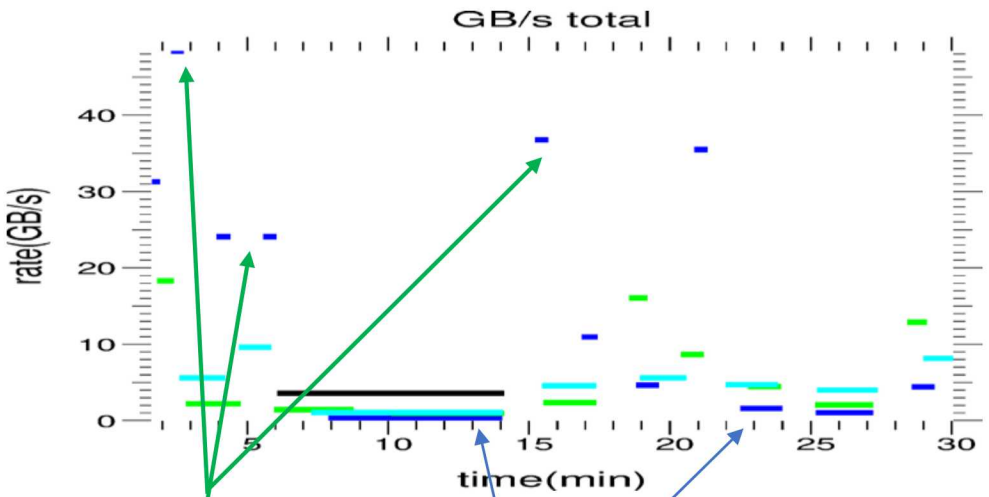



- 30 GB/s of 50 theo., 10 second average bins
- Switch ports of the Lustre OSS
- No ldmsd on clients or server

# Data for congested check point (slow job)

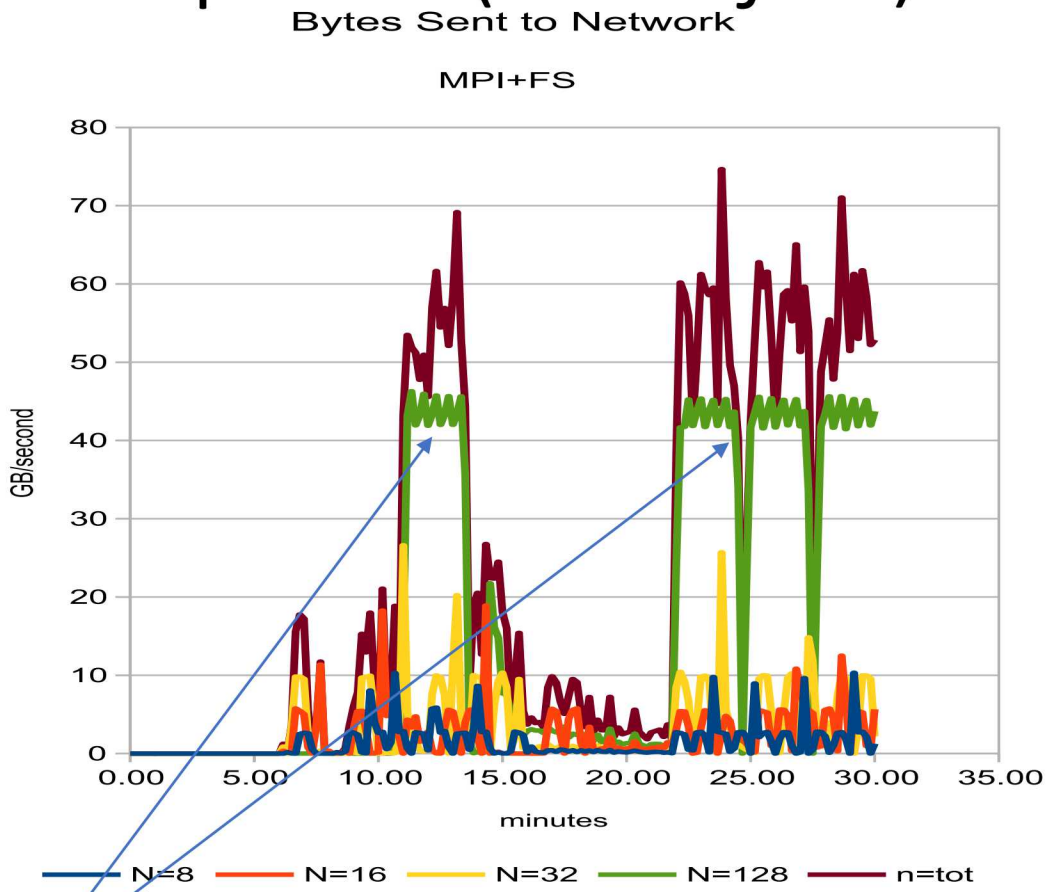
Lustre check point bandwidth

(from PF3D logs)



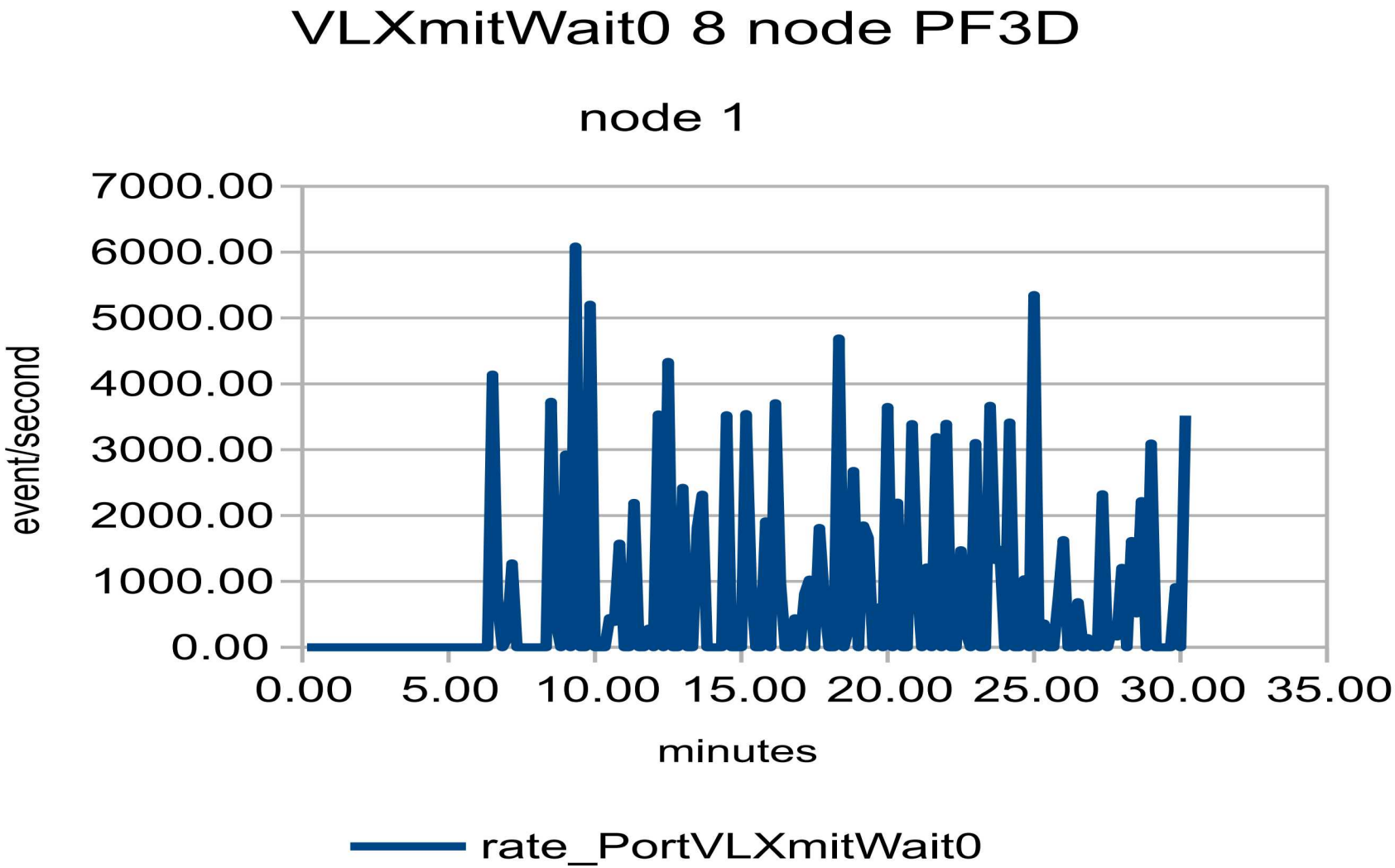

Normal N=8 checkpoints

N=8 checkpoints (blue) suppressed by FFT of N=128 (green)




Job bandwidth totals

# Data for congested check point (2)

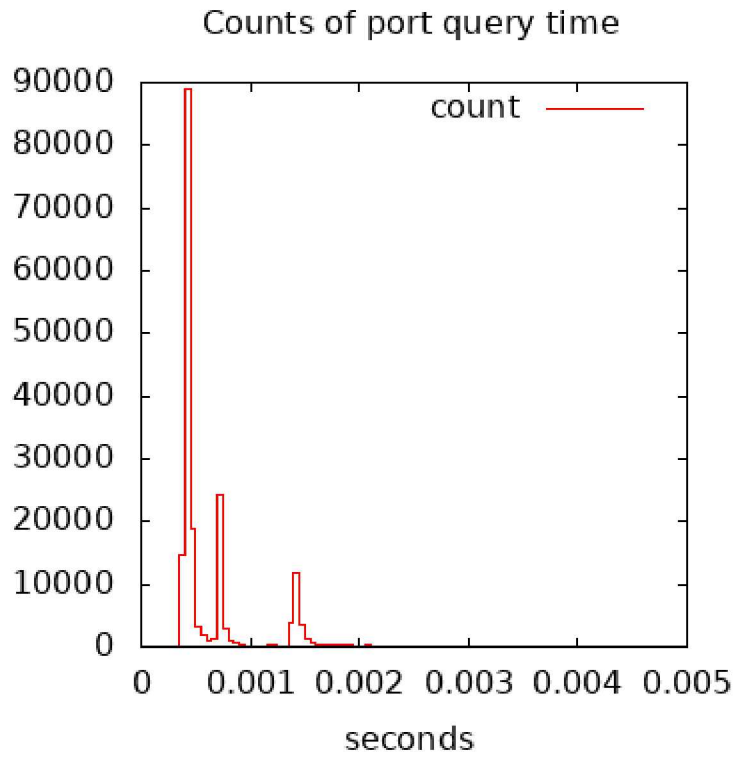


# Ibnet sampler performance (single collector)

| Time statistic | Sweep (seconds) |
|---|---|
| Minimum | 1.11 |
| Average | 1.22 |
| Maximum | 1.77 |
| Standard Deviation | 0.072 |

Sweep time statistics for 1000 ports from one host in production:

64230 sweeps (8 subsets/port) measured over 4 days.




Single-port query times from 180000 queries; bins 0.00005 seconds wide.
The sampler host here is not shared with jobs.

# PSNAP impact on co-located samplers:

A core-bound benchmark sensitive to interrupts and other OS noise.

- The test writes to node 0 NFS after the test loop finishes.

Maximum port sweep time:

- 167 milliseconds without PSNAP running.
- 400 milliseconds with PSNAP; *slower*.

Maximum single port query time:

- 5 mlx5 mad calls per port (extended(2), xmtdisc, rcverr, vlxmitcounters)
- 21 milliseconds without PSNAP running.
- 25 milliseconds with PSNAP; *slower*.

Conclusion:

- LDMS aggregation *offset must be more than maximum sweep time*.
  - E.g. 410000  if interval is 1000000.

# Sampler impact on co-located PSNAP: *none*

PSNAP:

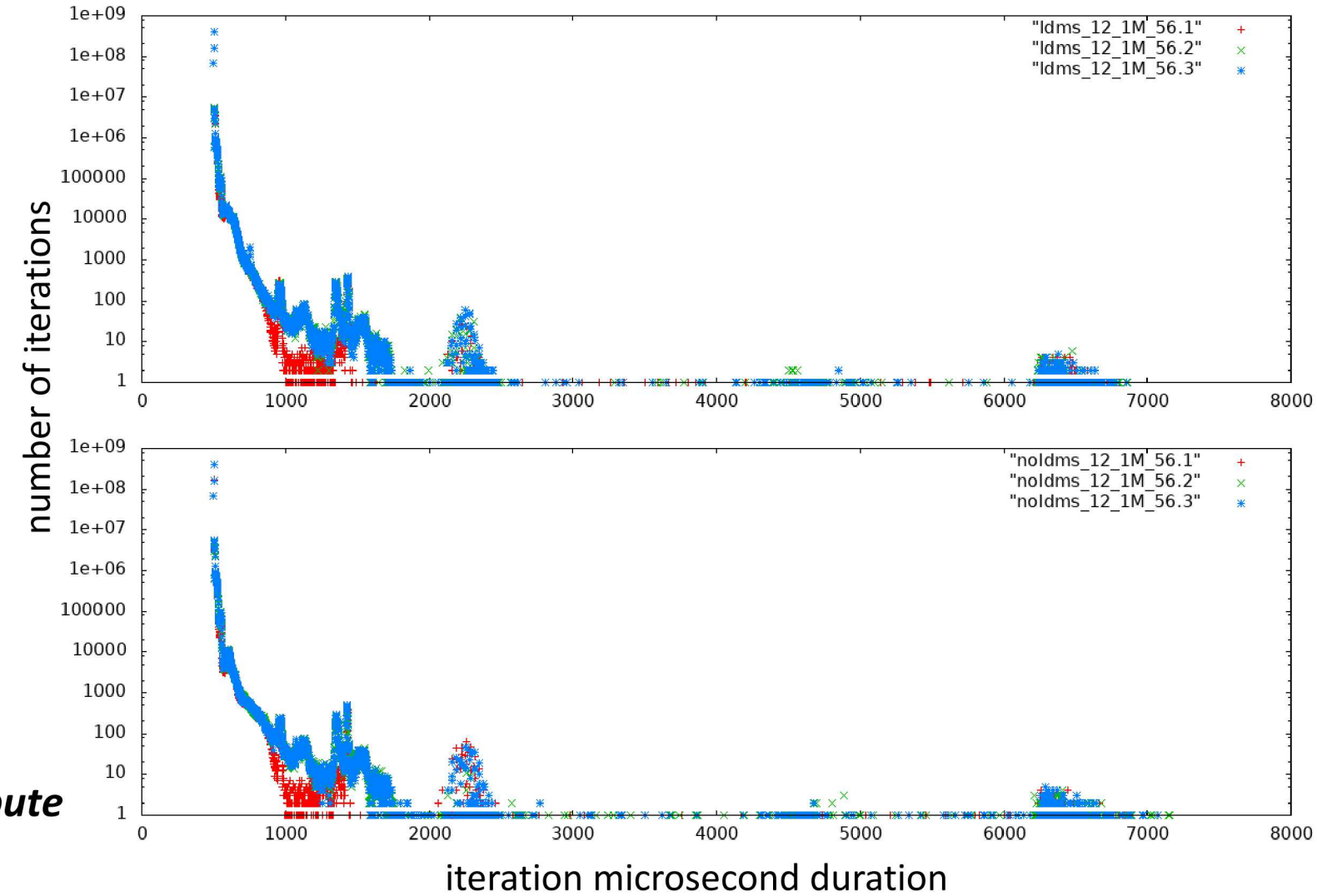
- 3 runs with and without LDMS
- All runs took 672 seconds
- 1 million iterations per core
- 56 cores/node
- 12 nodes

Ibnet samplers:

- 1 Hz data collection
- A maximum of 2 switches & 74 HCA ports per sampler
- 1008 ports total

Conclusion:

- ***Sampling the fabric from compute nodes does not impact PSNAP***



# Sampler impact on bandwidth

No immediately measurable impact at 1 or 5 second sampling intervals on:

- HPL
- IOR
- PF3D
- PF3DComm

More statistically rigorous testing needed.

# Conclusions

- LDMSD performing ibnet sampling and RDMA-based aggregation does not disturb the CPU-bound micro-benchmark PSNAP.
- Local compute jobs can stretch steps in the sweep a little.
- For mlx5 hardware, configure to collect these subsets:
  - extended, xmtdisc, rcverr, vlxmitcounters

# Lessons learned

- Must adapt to mildly dynamic hardware
  - GUIDs change on running clusters due to part swaps.
  - LIDs can change at system down times.
- Rough provisioning results for EDR:
  - 1 node sampling 1000 ports -- sweep time peak 1.75 seconds.
  - Co-locate samplers with compute nodes; collect from one switch per node.
- For a single sweep with multiple MAD calls per port, skip the rest of the MAD calls on a port after the first failure.
- The mlx5 hardware does not support all subsets of perfquery.

# Ongoing work

- Refining usability features of the sampler:
  - Sampler configuration should be via names instead of lids.
  - Detect and adapt to in-service name changes.
- Getting per-lane metrics from EDR adaptive routing and quality of service:
  - Requires recent hardware (mlx6).
- Quantitative overhead testing:
  - Bandwidth testing in presence of 1Hz, 10Hz sampling.
- Scaling to 10 Hz data collection:
  - Collect from no more than 1 switch per sampler, and use nodes close to switches.
  - Collect endpoint HCA metrics locally.
  - Transport multiple samples together at 1 Hz.
- Adaptation to Omnipath port queries