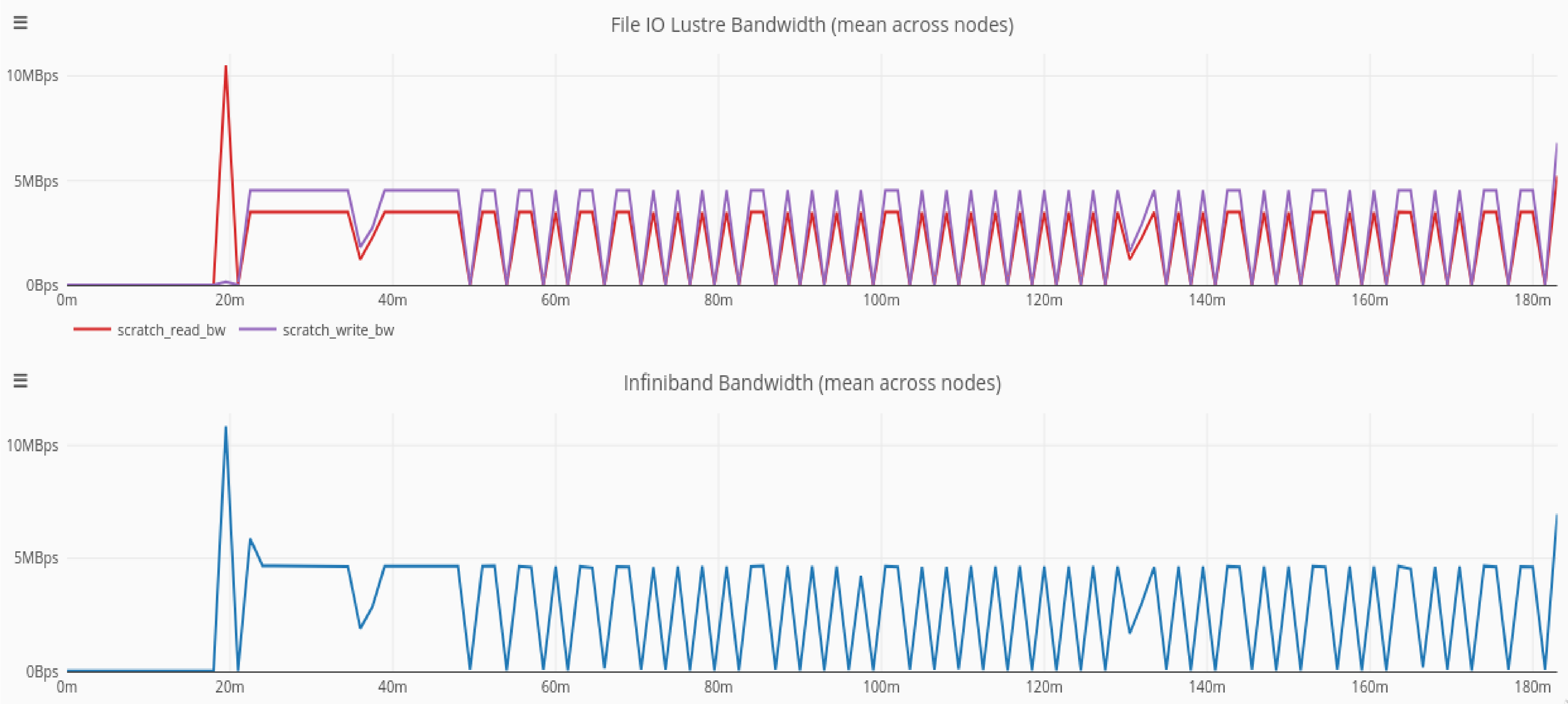
Sebastian Oeste
Center for Information Services and High Performance Computing (ZIH)

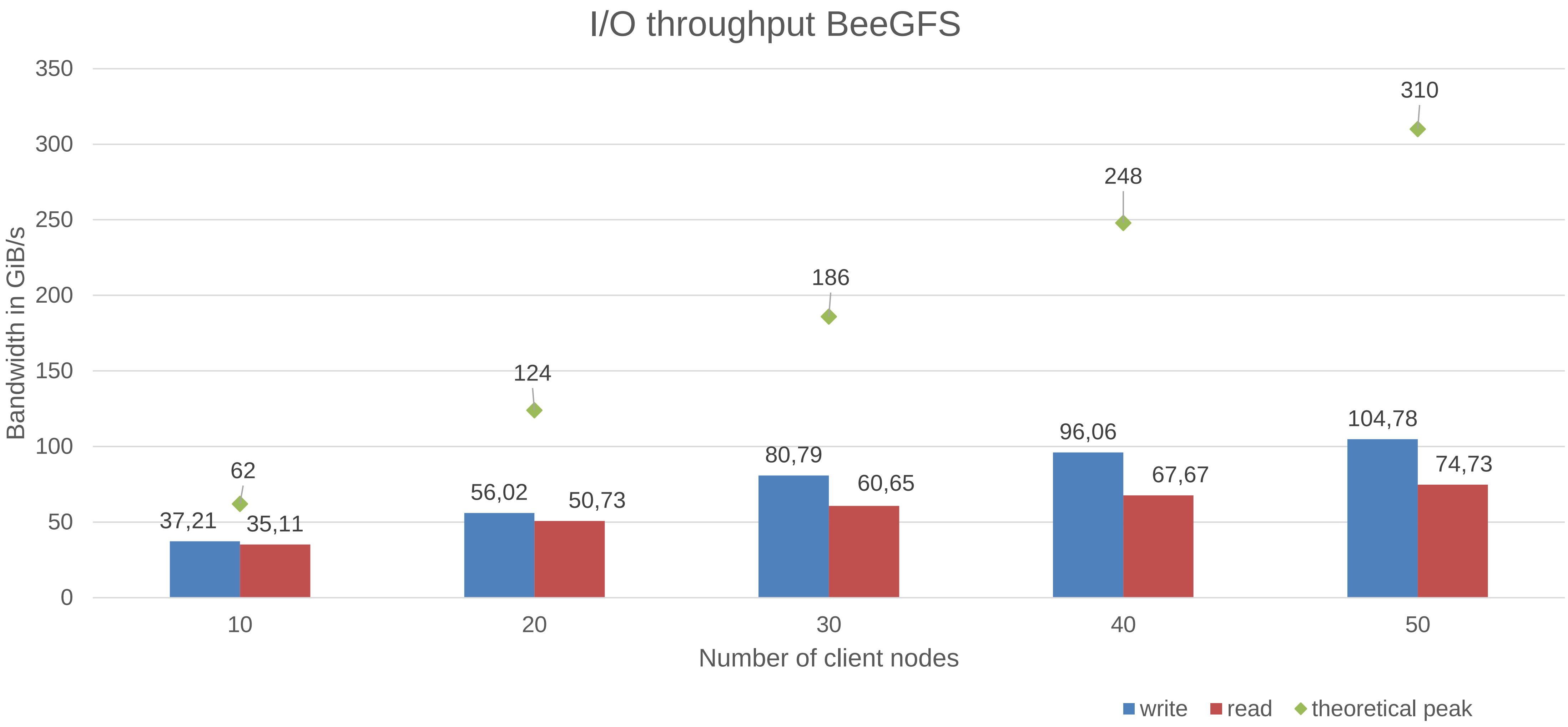
Characterization of Infiniband routes to support data intensive I/O

ISC-IODC 2022

Motivation


File IO Lustre Bandwidth (mean across nodes)
10MBps
5MBps
0Bps
0m
20m
40m
60m
80m
100m
120m
140m
160m
180m
scratch_read_bw
scratch_write_bw
Infiniband Bandwidth (mean across nodes)
10MBps
5MBps
0Bps
0m
20m
40m
60m
80m
100m
120m
140m
160m
180m

Throughput test with a sane I/O pattern



I/O performance impact factors

Application

- Request sizes
- Access pattern
- I/O operation

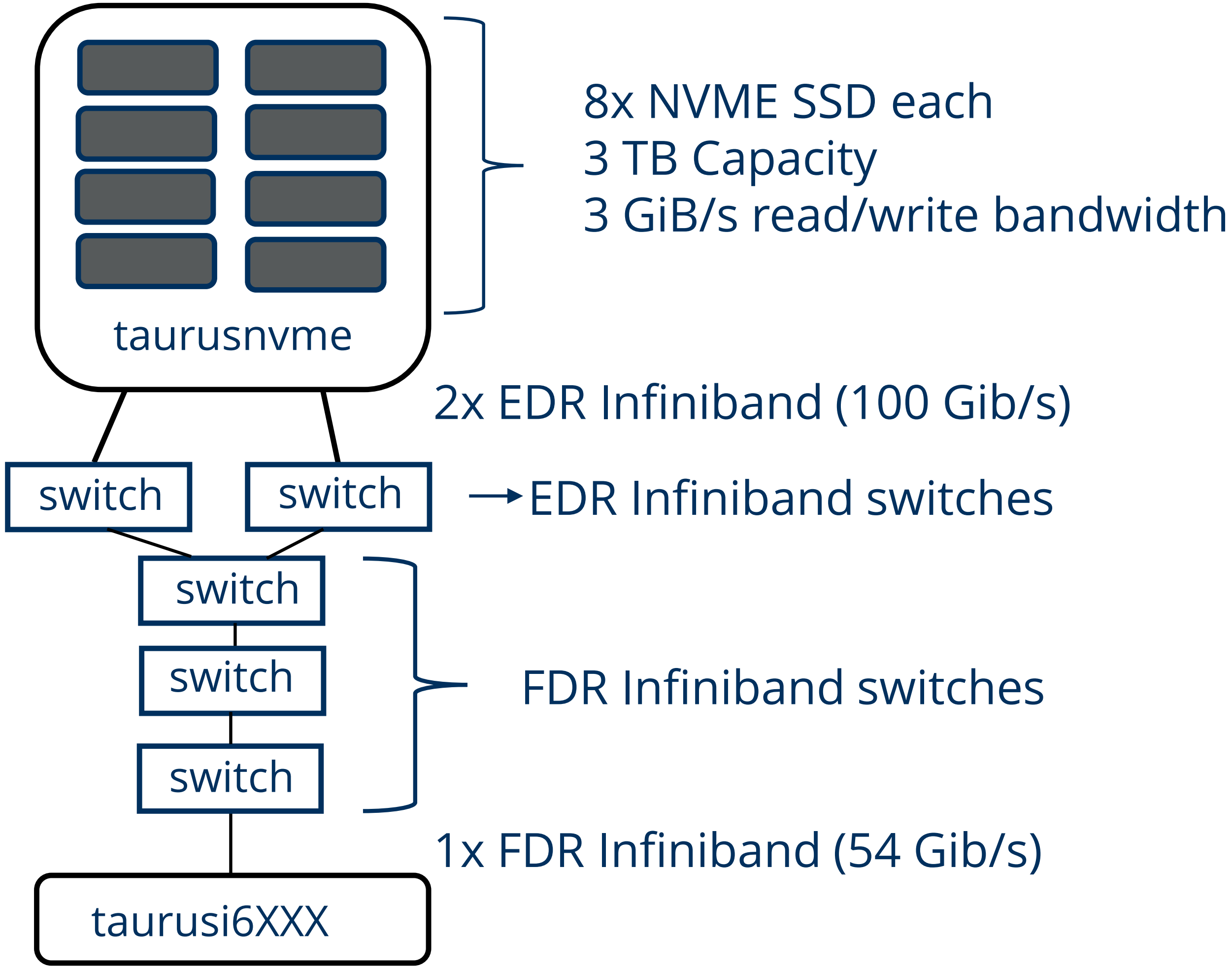
Network

- Message sizes
- Network paths

File system

- Stripe sizes
- File hierarchy

Setup



taurusnvme

8x NVME SSD each
3 TB Capacity
3 GiB/s read/write bandwidth

2x EDR Infiniband (100 Gib/s)

switch

switch

→ EDR Infiniband switches

switch

switch

switch

FDR Infiniband switches

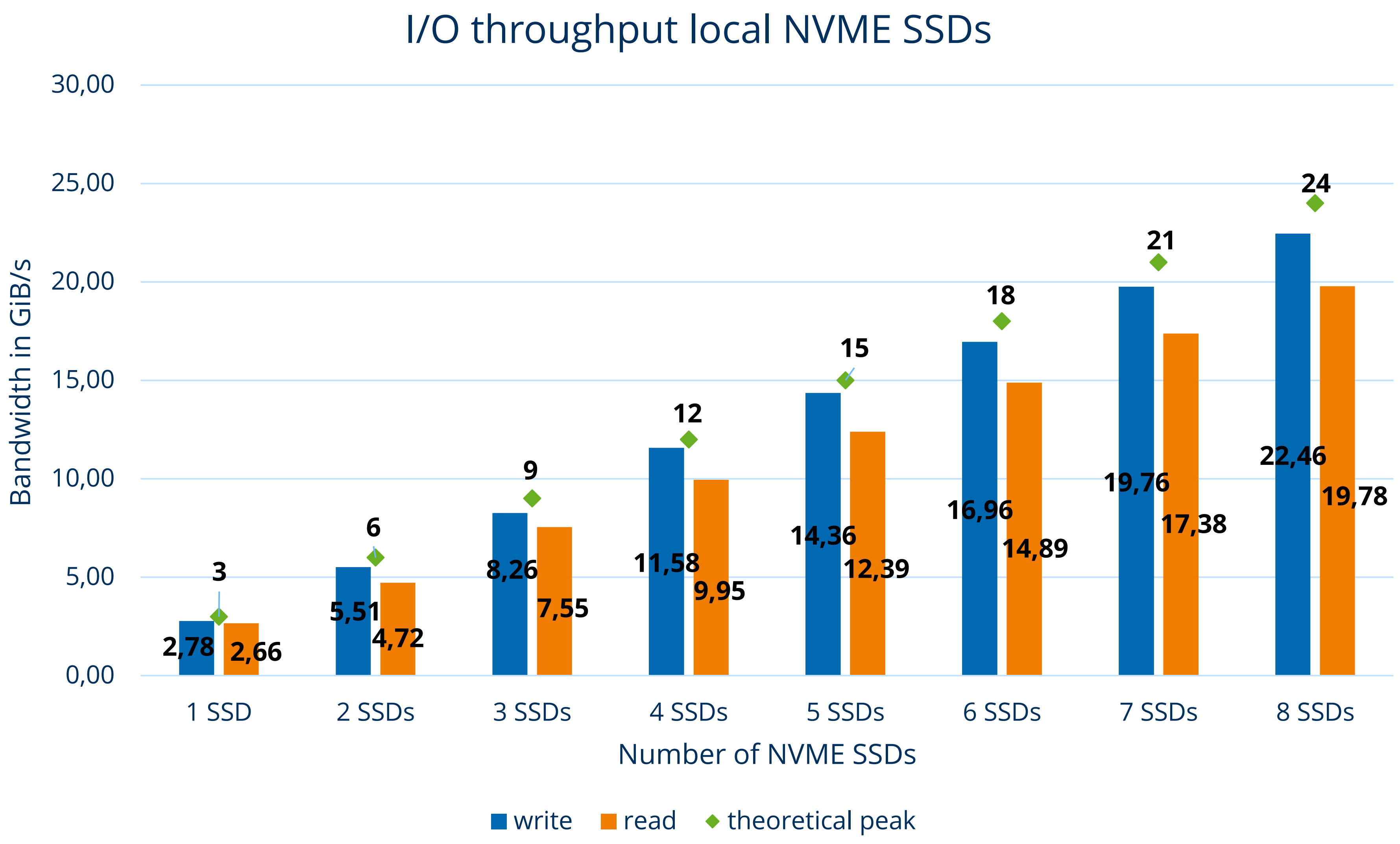
1x FDR Infiniband (54 Gib/s)

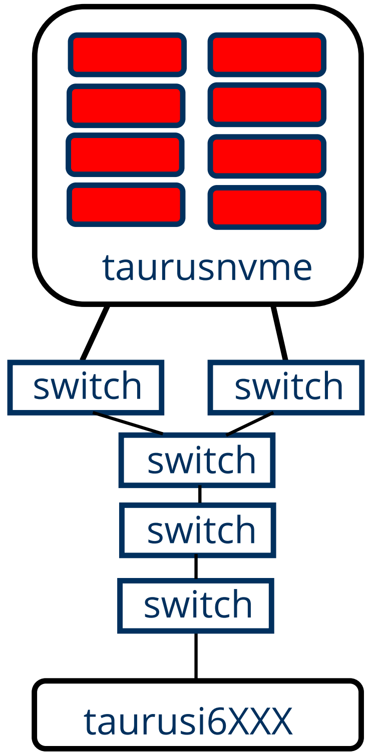
taurusi6XXX

Application: IOR

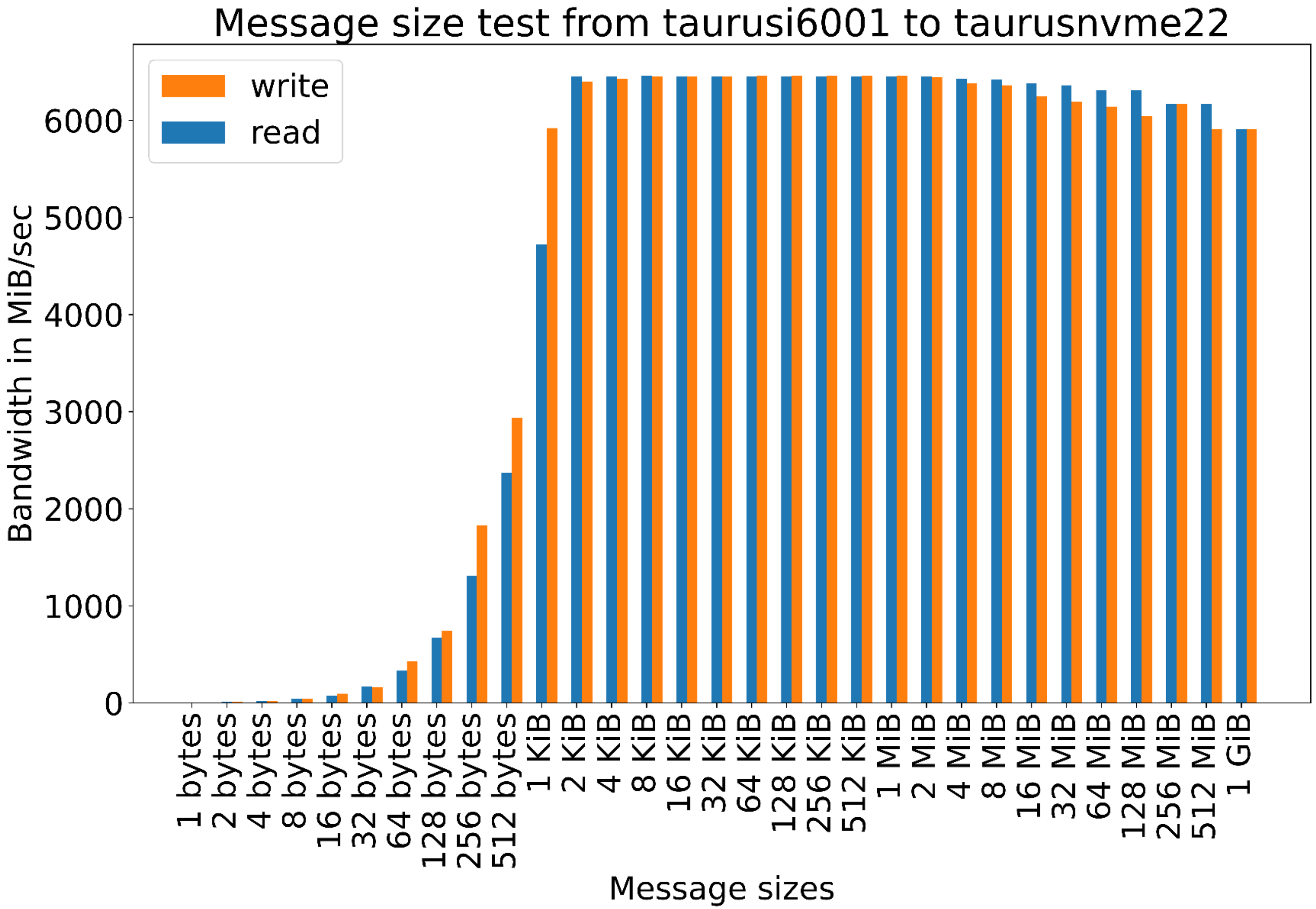
- Blocked sequential I/O pattern
- File per process
- 2 MiB request sizes

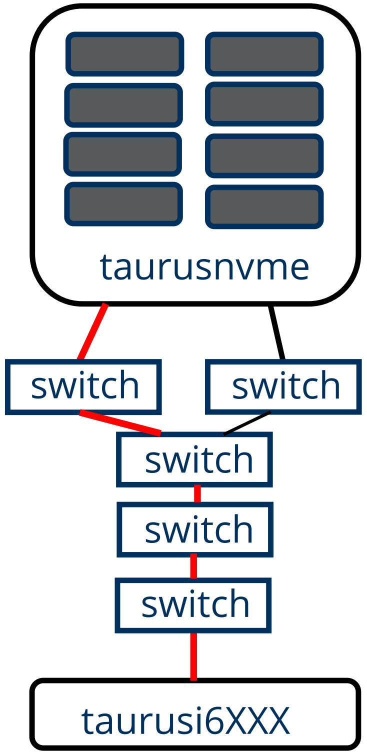
Single NVME SSD performance





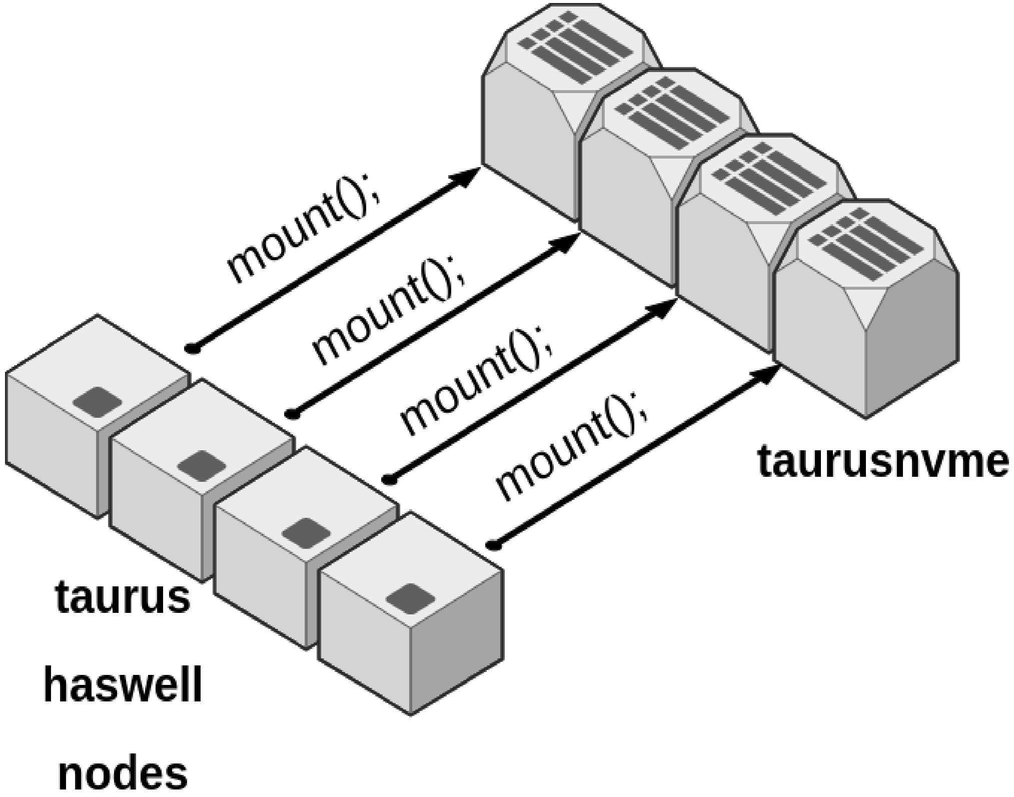
Check message sizes of network links



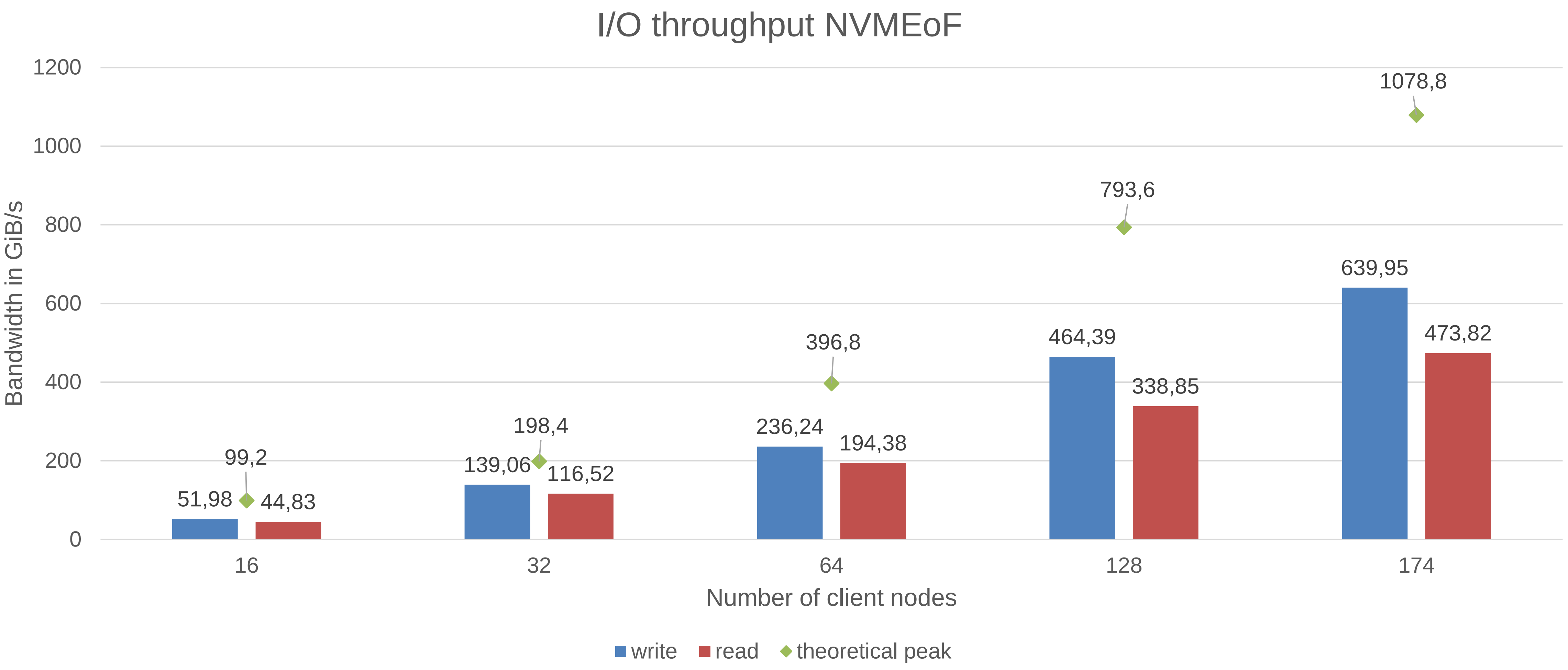


Eliminate the parallel file system as shared medium using NVME over Fabrics

- Using `nvme-connect` to connect NVME SSDs directly to compute nodes
- Server-side SSD appears as block device on compute node
- Use a local file system on that block device (e.g. ext4, XFS, ...)
- No shared view across compute nodes



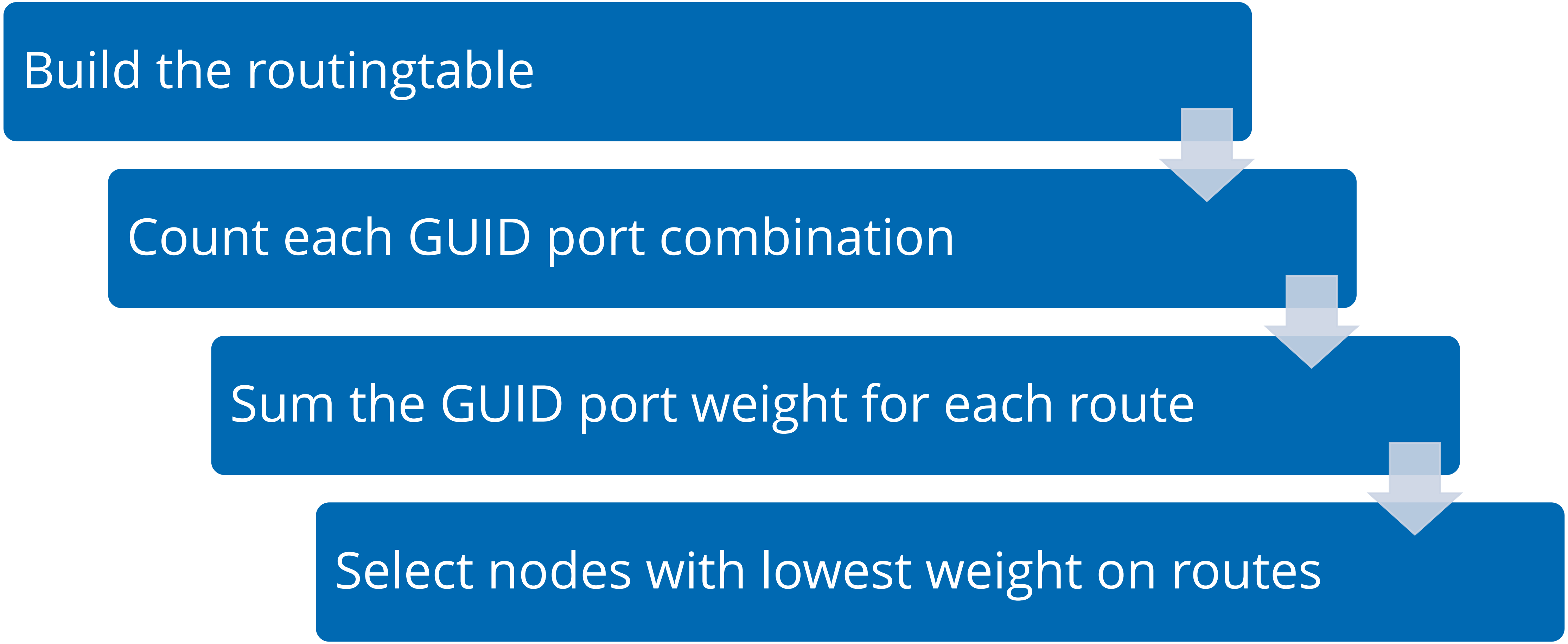
Throughput test with NVME over Fabrics



Need a measurement for the quality of network routes

- Infiniband routes are managed by the subnet manager (openSM)
- Routes may change over time (e.g. if a host crashes or a switch port becomes unavailable)
- In Infiniband networks each device get a GUID (Global Unique Identifier)
- Combination of GUID and port number refers to a unique physical link between two devices
- Multiple paths sharing the same physical link results in congestion
- No tool available

Select nodes with low overlap in paths



Build the routingtable

Count each GUID port combination

Sum the GUID port weight for each route

Select nodes with lowest weight on routes

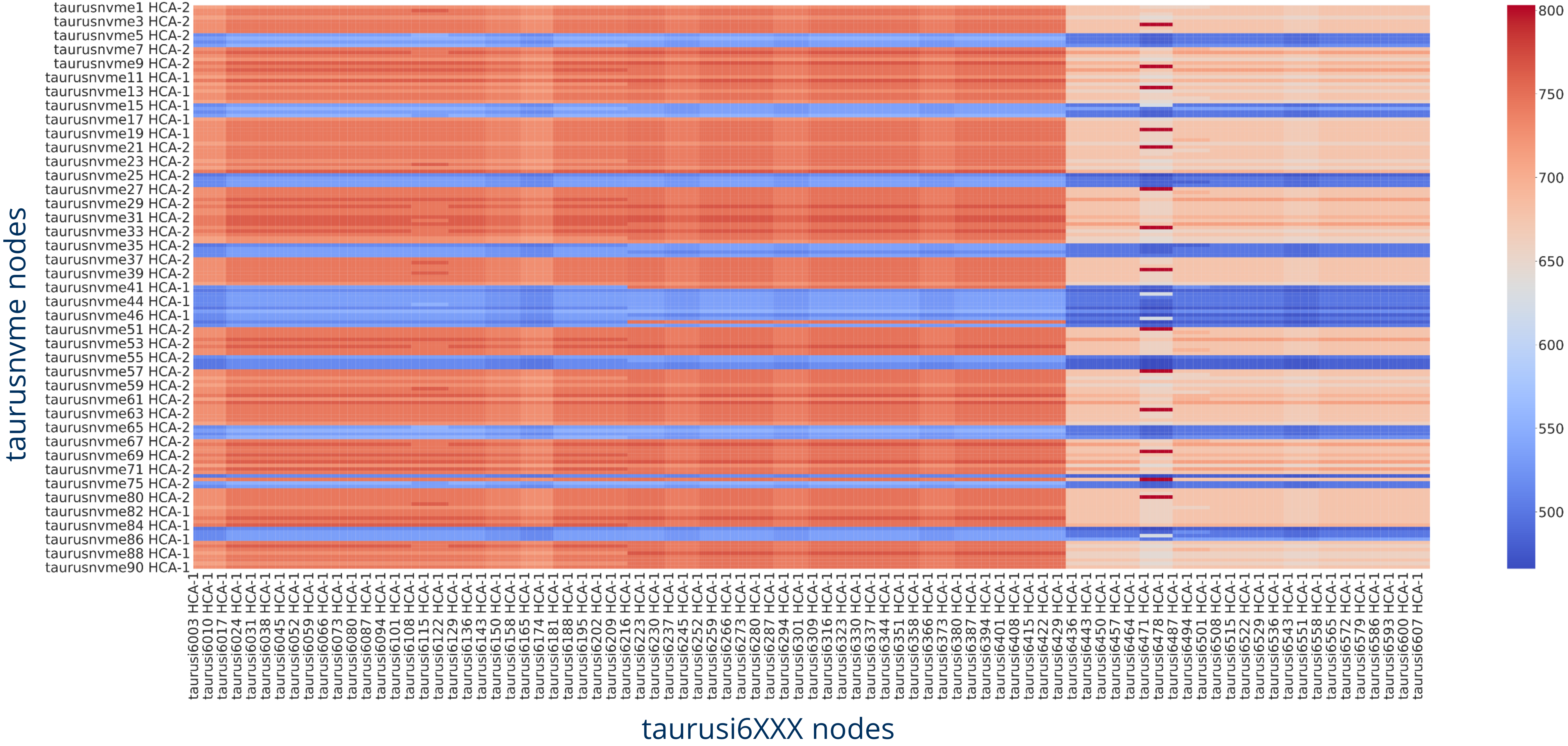
Issues with the creation of the routingtable

- OFED Tool *ibtracert* to query routes between hosts
  - Query the whole fabric takes several hours

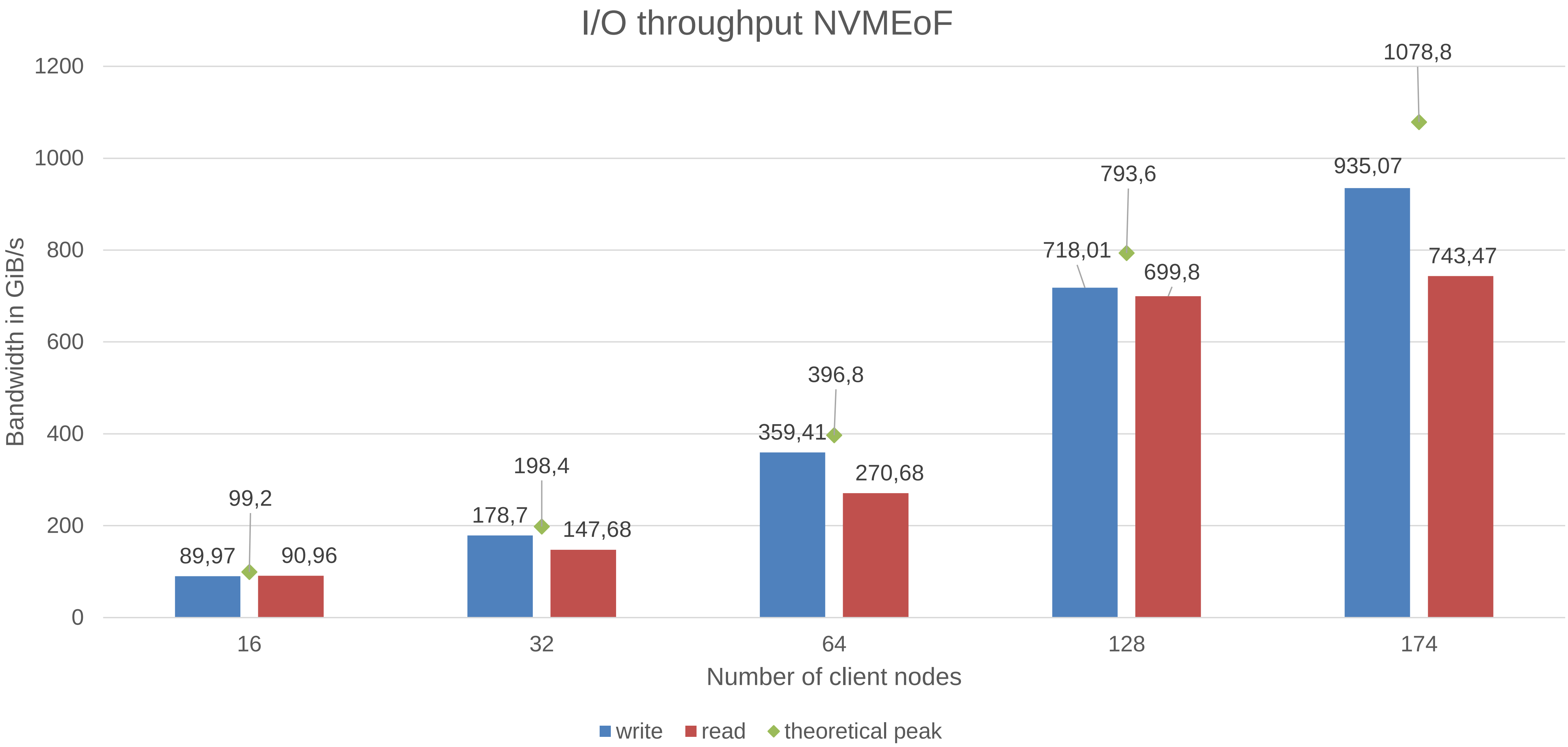
- OpenSM provides an option to dump the switch forwarding tables
  - Dump occurs every time routing changed

- Read switch fowarding tables and calculate routes manually
  - Takes ~10min for the whole fabric

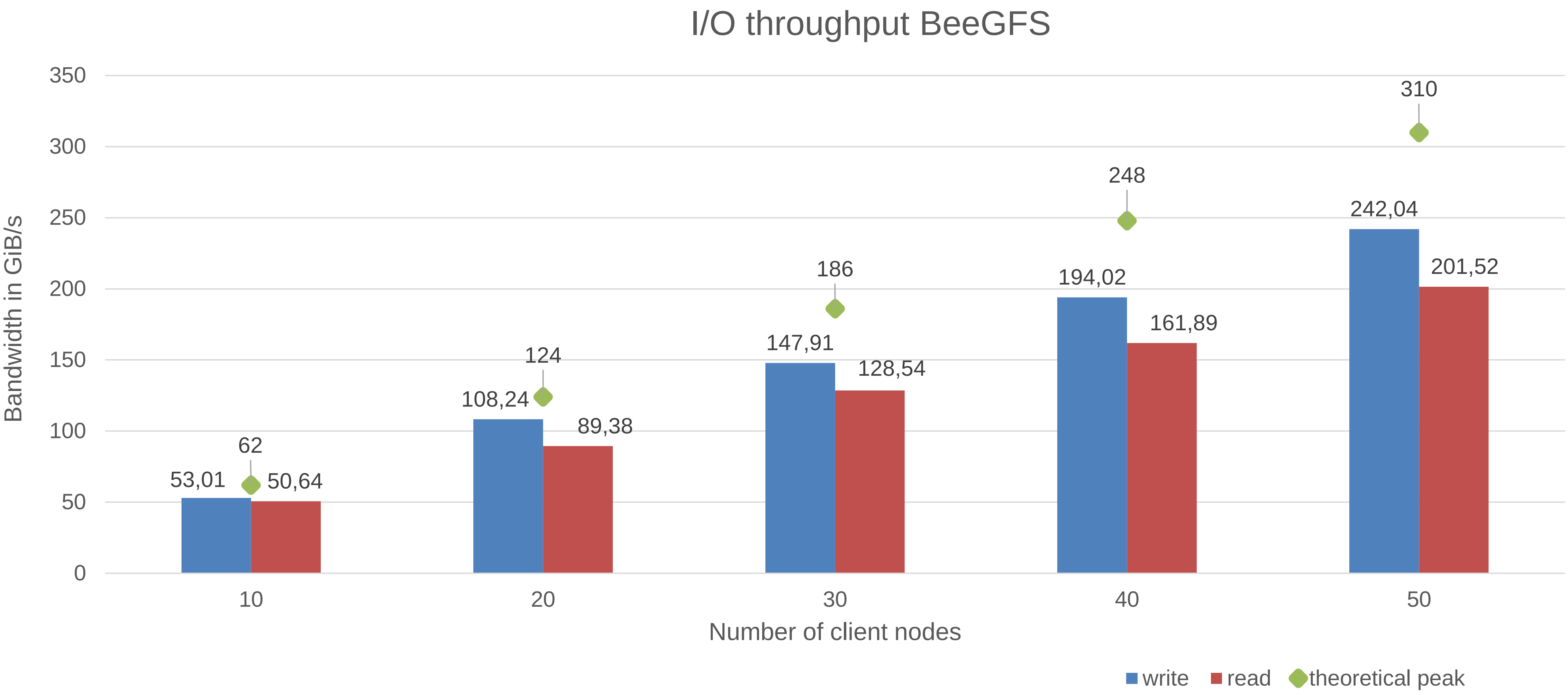
Select nodes with lowest overlap in routes



Throughput test NVME over Fabrics with selected routes

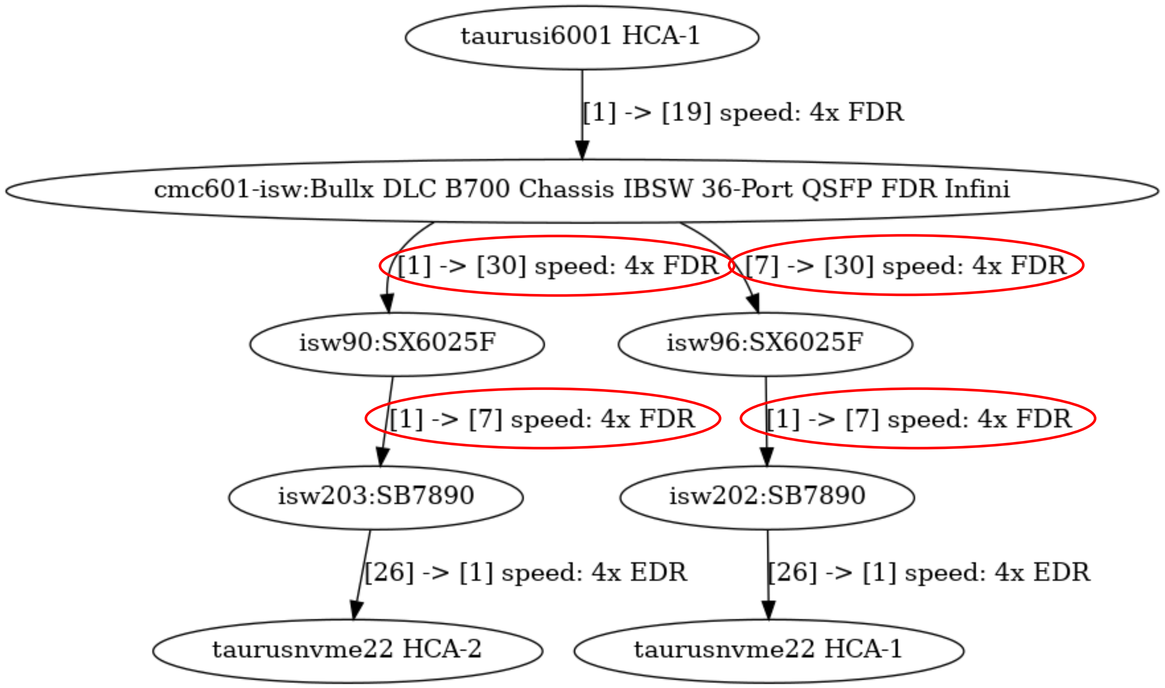


Select routes to the parallel file system



Different routes for read and write direction

Write direction



taurusi6001 HCA-1

[1] -> [19] speed: 4x FDR

cmc601-isw:Bullx DLC B700 Chassis IBSW 36-Port QSFP FDR Infini

[1] -> [30] speed: 4x FDR

[7] -> [30] speed: 4x FDR

isw90:SX6025F

isw96:SX6025F

[1] -> [7] speed: 4x FDR

[1] -> [7] speed: 4x FDR

isw203:SB7890

isw202:SB7890

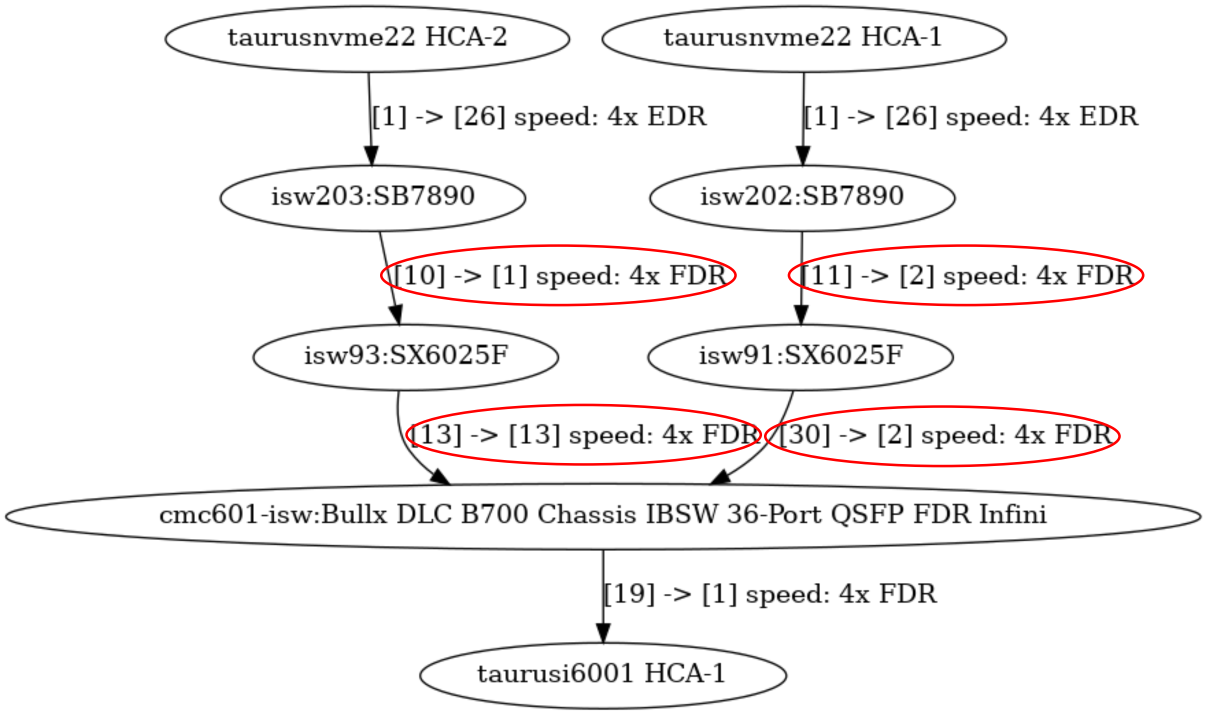
[26] -> [1] speed: 4x EDR

[26] -> [1] speed: 4x EDR

taurusnvme22 HCA-2

taurusnvme22 HCA-1

Read direction



taurusnvme22 HCA-2

taurusnvme22 HCA-1

[1] -> [26] speed: 4x EDR

[1] -> [26] speed: 4x EDR

isw203:SB7890

isw202:SB7890

[10] -> [1] speed: 4x FDR

[11] -> [2] speed: 4x FDR

isw93:SX6025F

isw91:SX6025F

[13] -> [13] speed: 4x FDR

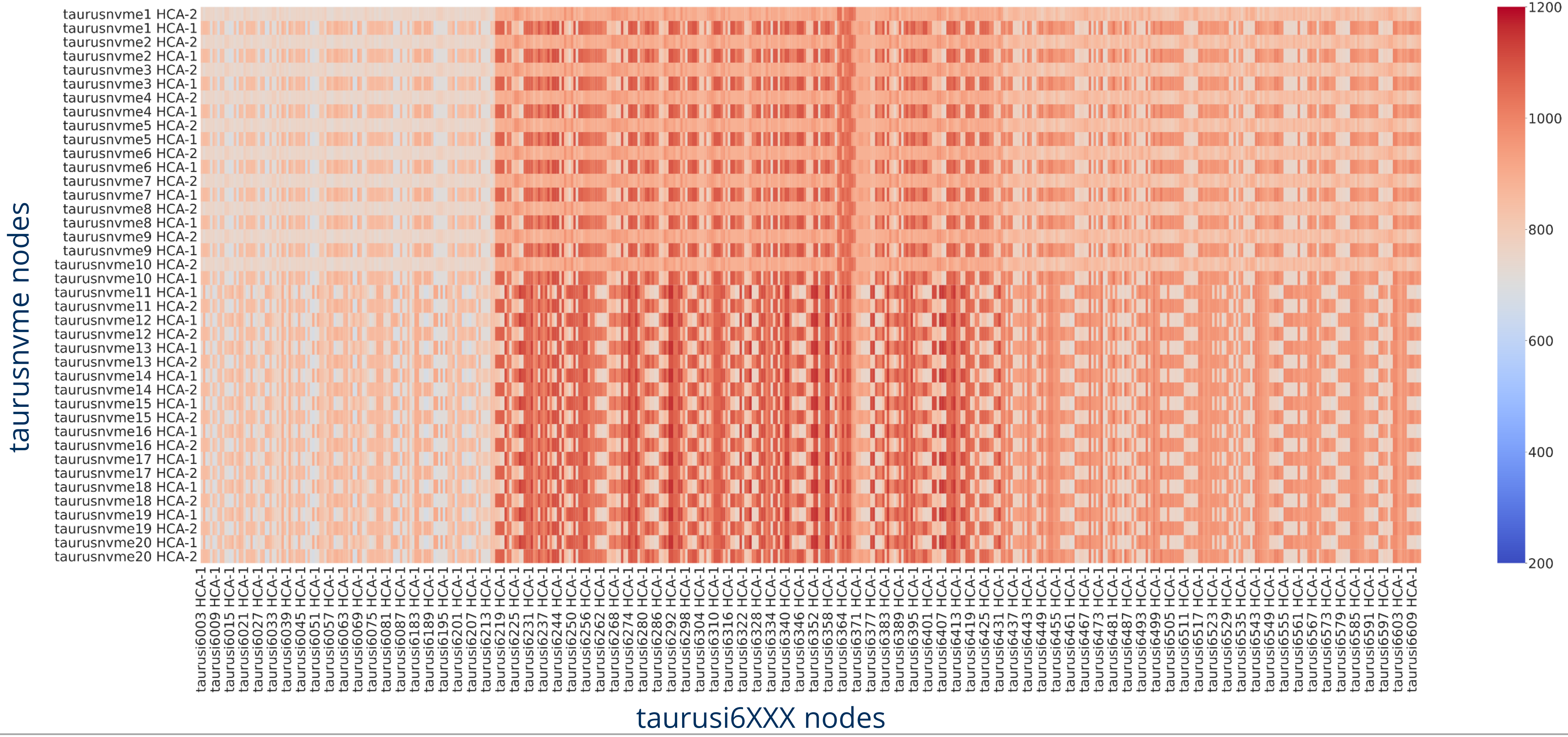
[30] -> [2] speed: 4x FDR

cmc601-isw:Bullx DLC B700 Chassis IBSW 36-Port QSFP FDR Infini

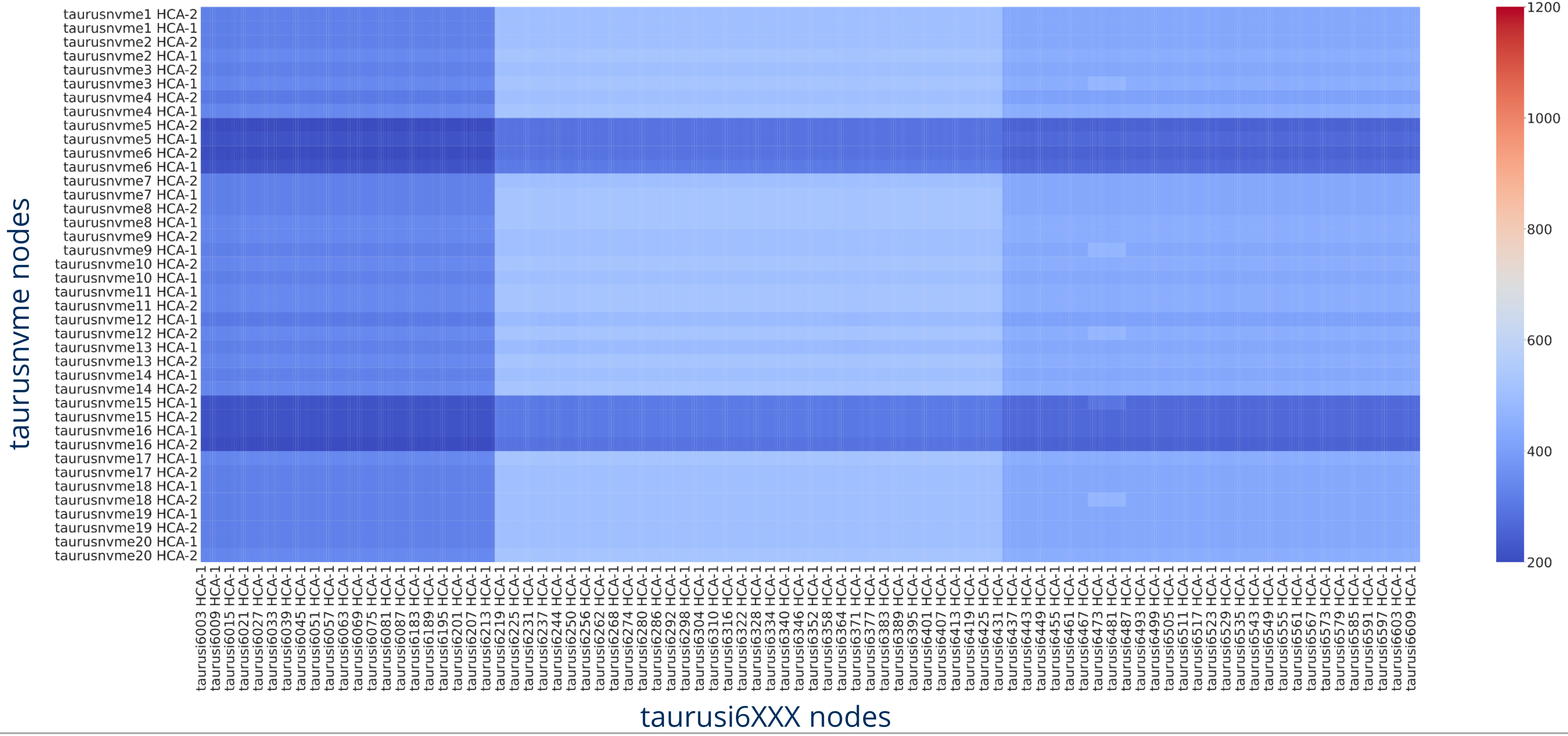
[19] -> [1] speed: 4x FDR

taurusi6001 HCA-1

Routes for read direction



Routes for write direction



Summary and further work

Summary:

- Congestion on the network has a significant impact on I/O performance
- Select nodes with lower overlap in their routes can improve I/O performance
- Tool that weights paths
  - Without producing load on the fabric
  - Able to monitor path changes over time
  - https://github.com/blastmaster/IBspy

Further work:

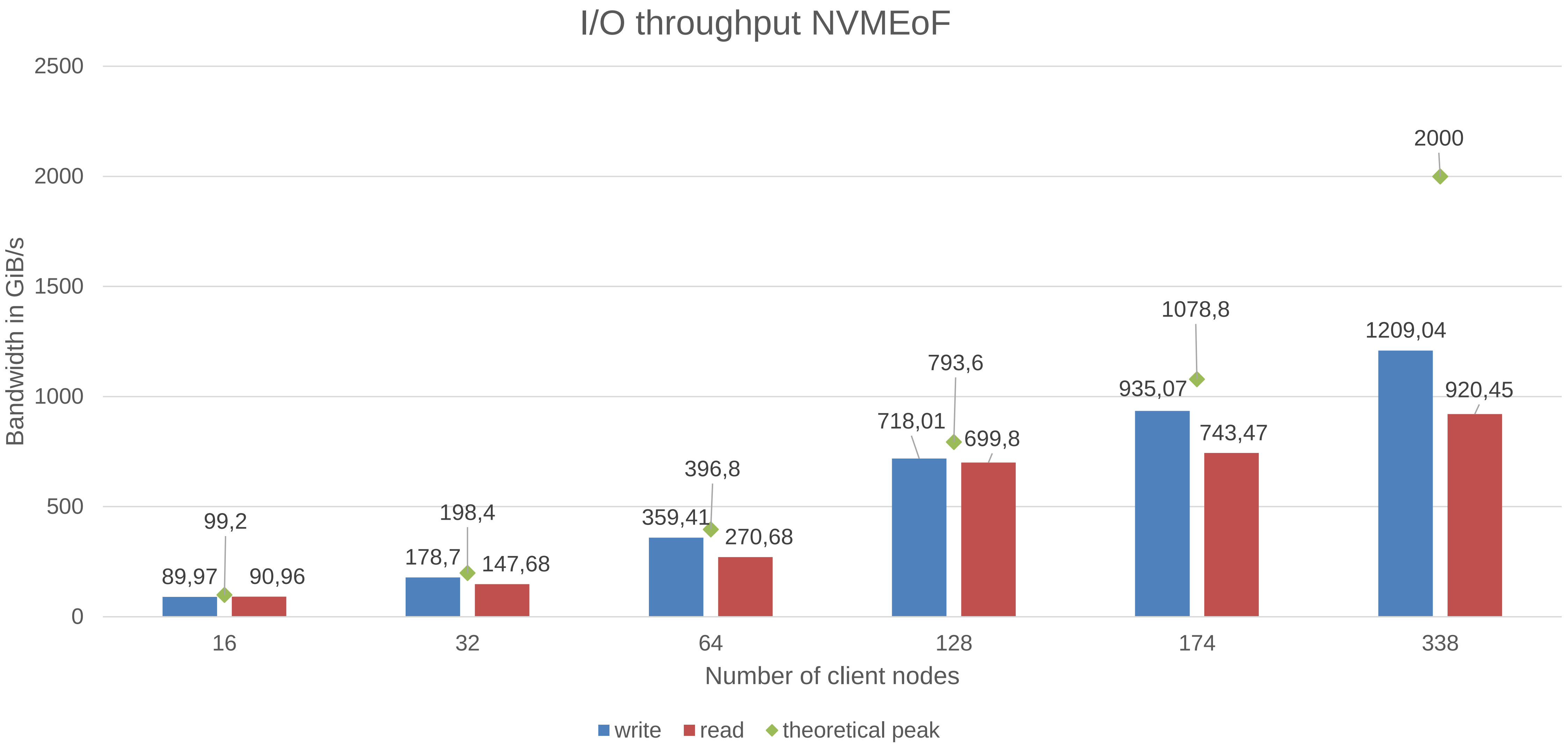
- Enable route evaluation for live jobs together with PIKA
- Slurm plugin for integration in job scheduler

Discussion

- Are there other sides that discover similiar challenges with the network between storage and compute nodes?
- How do you monitor that?
- Possible solutions?

Backup

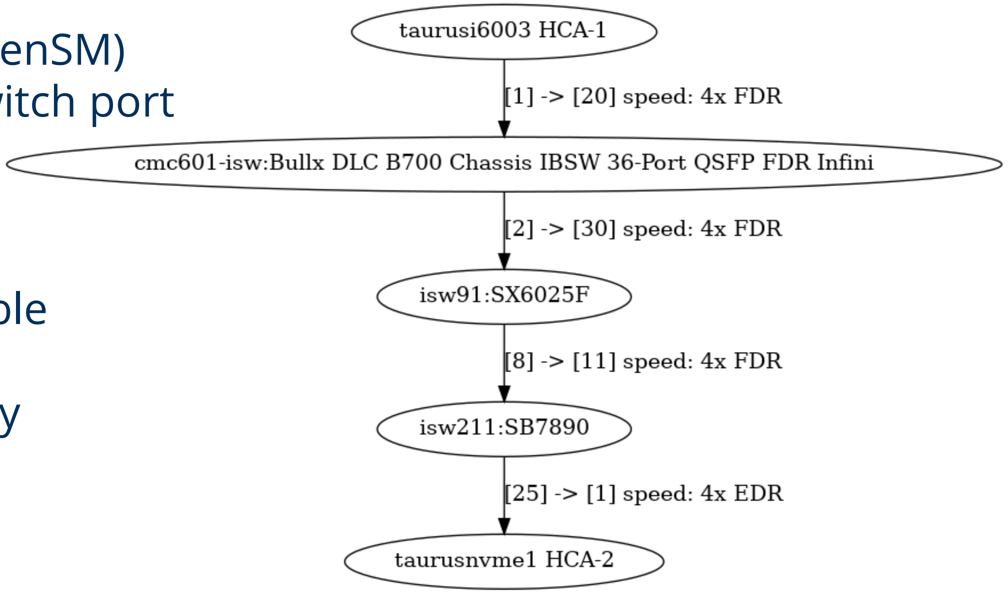
Throughput test NVME over Fabrics with selected routes



Number of client nodes	write	read	theoretical peak
16	89,97	90,96	99,2
32	178,7	147,68	198,4
64	359,41	270,68	396,8
128	718,01	699,8	793,6
174	935,07	743,47	1078,8
338	1209,04	920,45	2000

Collect the routing information

- Infiniband routes are managed by the subnet manager (openSM)
- Routes may change over time (e.g. if a host crashes or a switch port becomes unavailable)
- OFED Tool *ibtracert* to query routes between hosts
  - Query the whole fabric takes several hours
- OpenSM provides an option to dump switch forwarding table
  - Dump occurs every time routing changed
- Read switch fowarding tables and calculate routes manually
  - Takes ~10min for the whole fabric



taurusi6003 HCA-1
[1] -> [20] speed: 4x FDR
cmc601-isw:Bullx DLC B700 Chassis IBSW 36-Port QSFP FDR Infini
[2] -> [30] speed: 4x FDR
isw91:SX6025F
[8] -> [11] speed: 4x FDR
isw211:SB7890
[25] -> [1] speed: 4x EDR
taurusnvme1 HCA-2

```
> ibtracert -G 0x08003800013c368b 0x506b4b0300fbef06
From ca {0x08003800013c368b} portnum 1 lid 367-367 "taurusi6003 HCA-1"
[1] -> switch port {0x08003800023cceda}[20] lid 302-302 "cmc601-isw:Bullx DLC B700 Chassis IBSW 36-Port QSFP FDR Infini"
[2] -> switch port {0xf452140300369540}[30] lid 565-565 "isw91:SX6025F"
[8] -> switch port {0x248a070300bedb10}[11] lid 170-170 "isw211:SB7890"
[25] -> ca port {0x506b4b0300fbef06}[1] lid 73-73 "taurusnvme1 HCA-2"
```

The setup

Storage: taurusnvme nodes

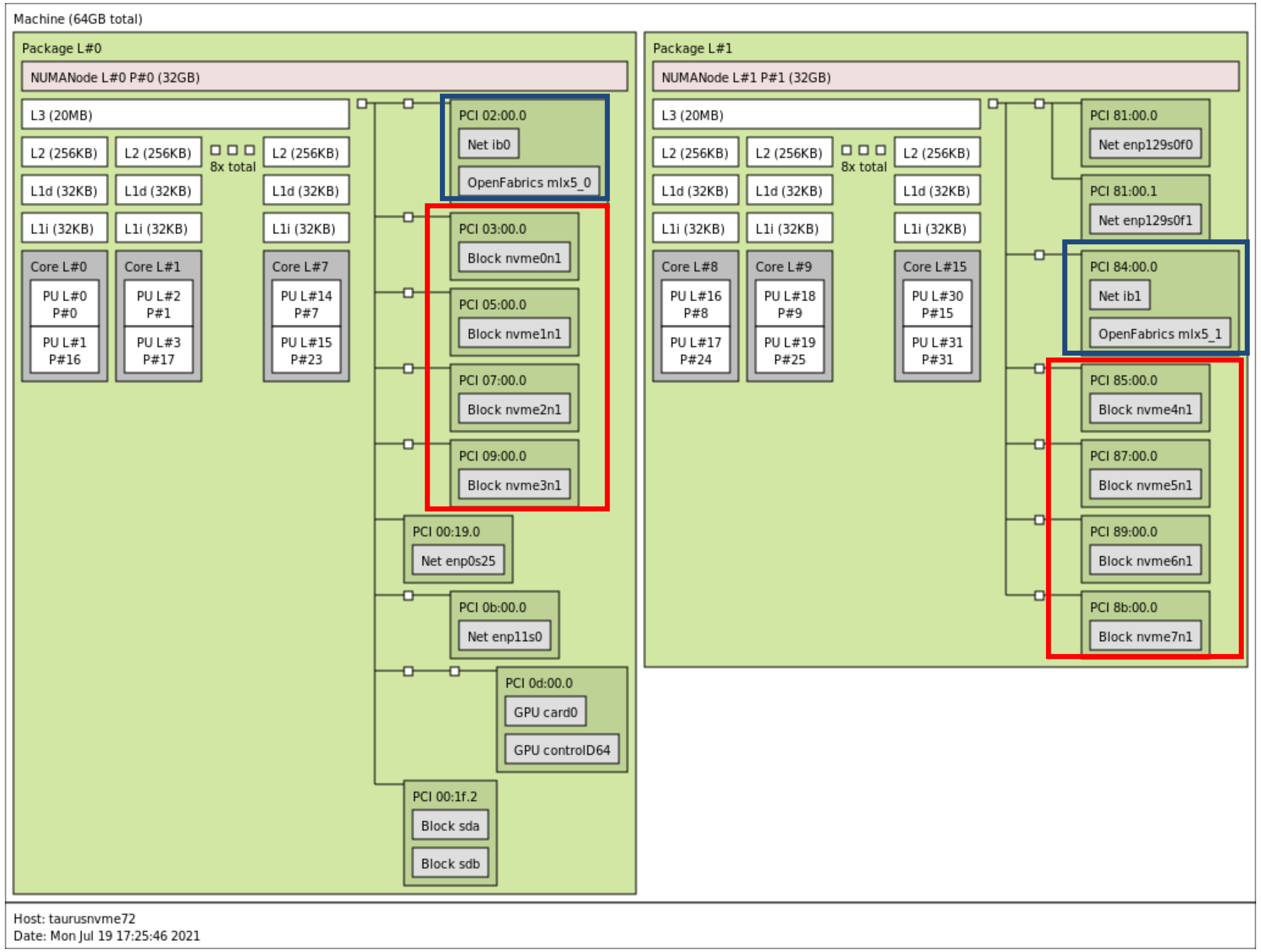
- 2x EDR Infiniband (100 Gbit/s)
- 8x NVME SSD with 3TB capacity and ~3GiB/s read/write bandwidth

Client: taurus haswell nodes

- 1x FDR Infiniband (54 Gbit/s)
- 24 cores haswell CPU

Benchmark: IOR

- Using blocked sequential I/O pattern
- File per process
- 2MiB request sizes


Machine (64GB total)
Package L#0
NUMANode L#0 P#0 (32GB)
L3 (20MB)
L2 (256KB)
L2 (256KB)
8x total
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core L#0
PU L#0 P#0
PU L#1 P#16
Core L#1
PU L#2 P#1
PU L#3 P#17
Core L#7
PU L#14 P#7
PU L#15 P#23
PCI 02:00.0
Net ib0
OpenFabrics mlx5_0
PCI 03:00.0
Block nvme0n1
PCI 05:00.0
Block nvme1n1
PCI 07:00.0
Block nvme2n1
PCI 09:00.0
Block nvme3n1
PCI 00:19.0
Net enp0s25
PCI 0b:00.0
Net enp11s0
PCI 0d:00.0
GPU card0
GPU controlD64
PCI 00:1f.2
Block sda
Block sdb
Package L#1
NUMANode L#1 P#1 (32GB)
L3 (20MB)
L2 (256KB)
L2 (256KB)
8x total
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core L#8
PU L#16 P#8
PU L#17 P#24
Core L#9
PU L#18 P#9
PU L#19 P#25
Core L#15
PU L#30 P#15
PU L#31 P#31
PCI 81:00.0
Net enp129s0f0
PCI 81:00.1
Net enp129s0f1
PCI 84:00.0
Net ib1
OpenFabrics mlx5_1
PCI 85:00.0
Block nvme4n1
PCI 87:00.0
Block nvme5n1
PCI 89:00.0
Block nvme6n1
PCI 8b:00.0
Block nvme7n1
Host: taurusnvme72
Date: Mon Jul 19 17:25:46 2021