Studying a 40 Tb/s data acquisition system for the LHCb experiment

› **29/01/2018**

Sébastien VALAT – CERN

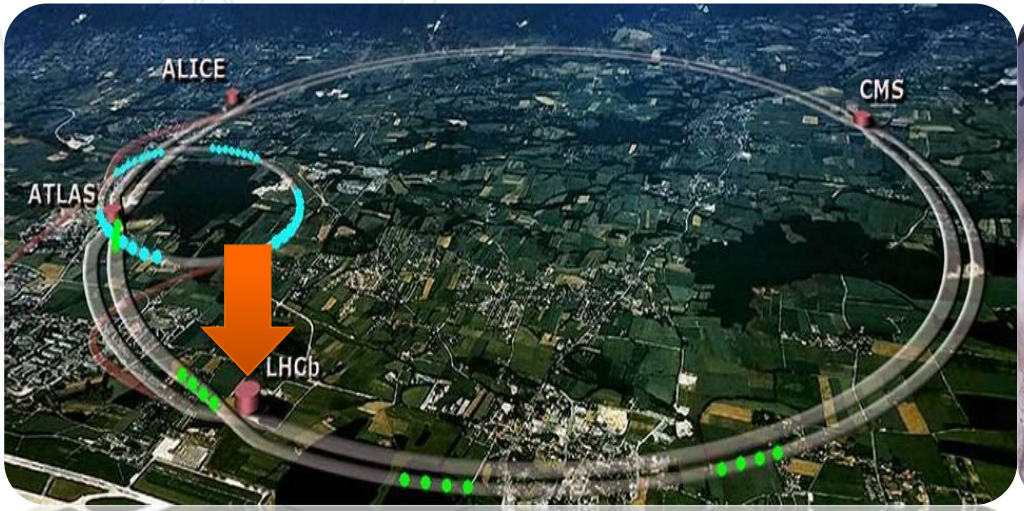
Plan

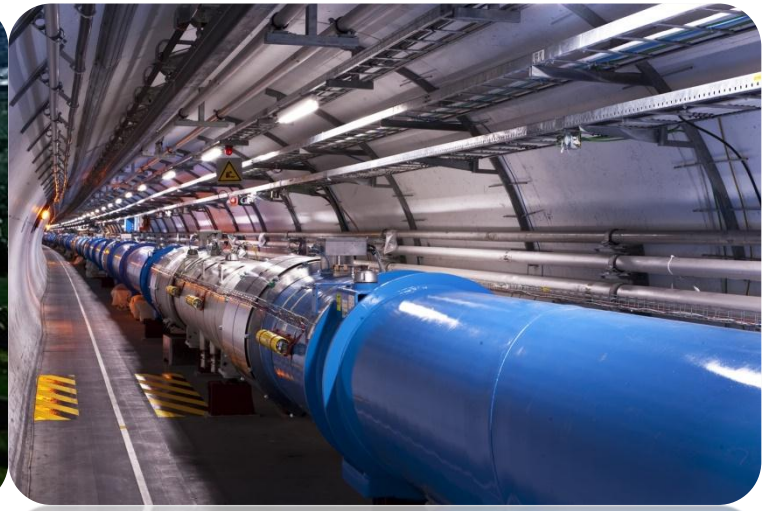
- Context
- Event building benchmark
- First test at scale : hitting the wall ☹
- Routing and cabling
- A word on failure recovery & storage
- Conclusion

Context

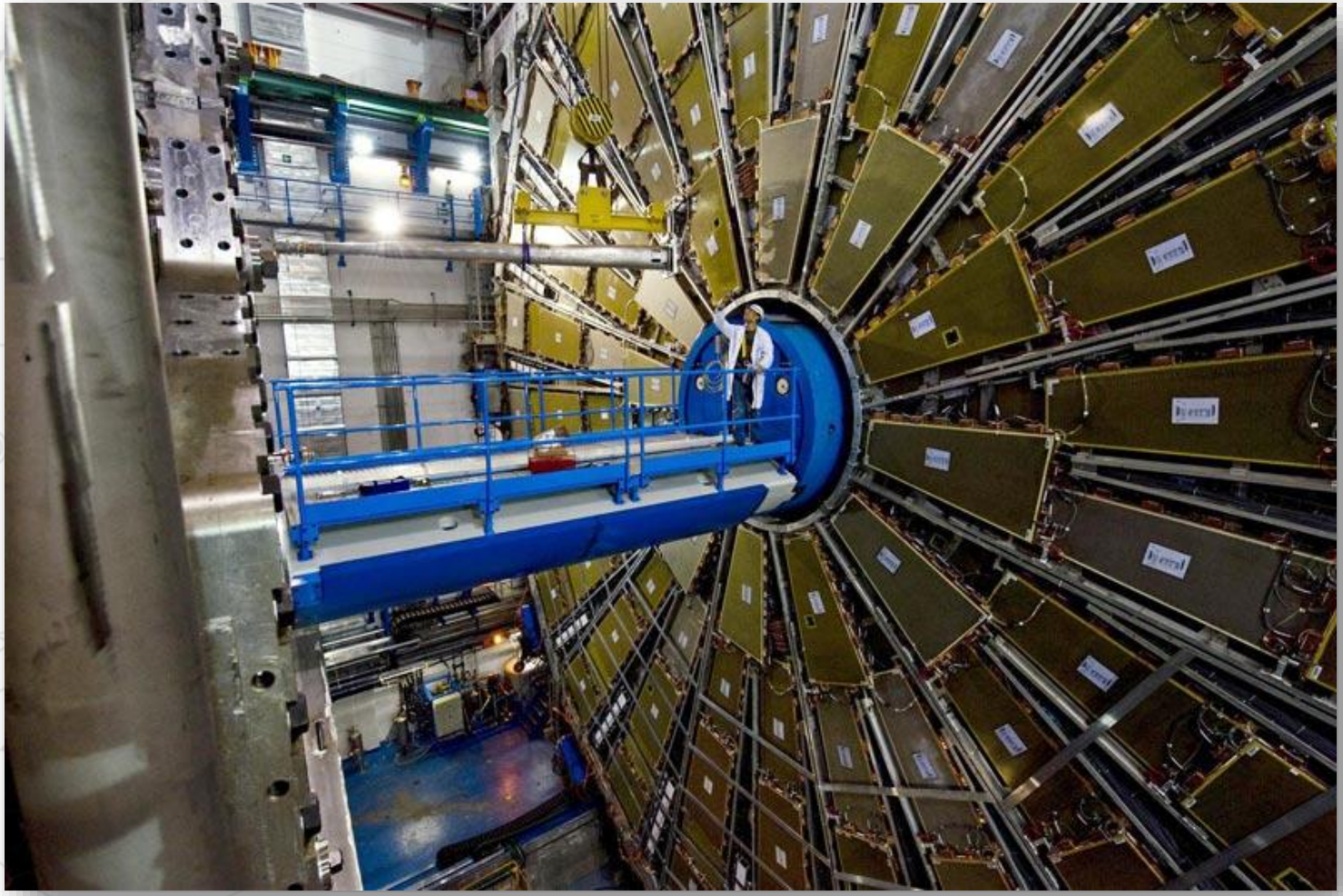
Reminder on LHC

- Accelerator of 27 km
- 10 000 superconductive magnets
- Collision energy up to 14 TeV
- Proton-Proton collisions, but also heavy-ions
- 4 BIG experiments :
  - ALICE, ATLAS, CMS, **LHCb**

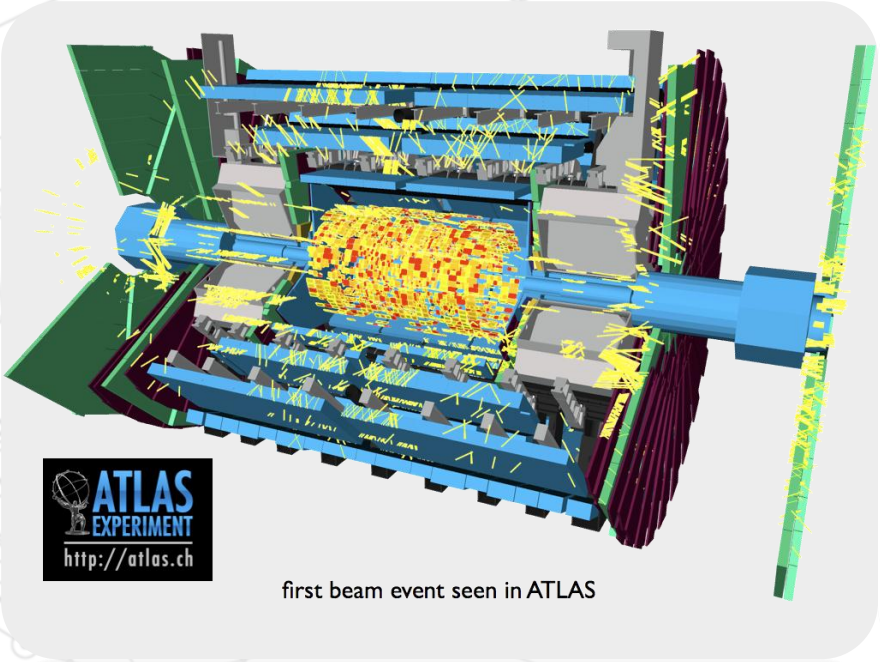




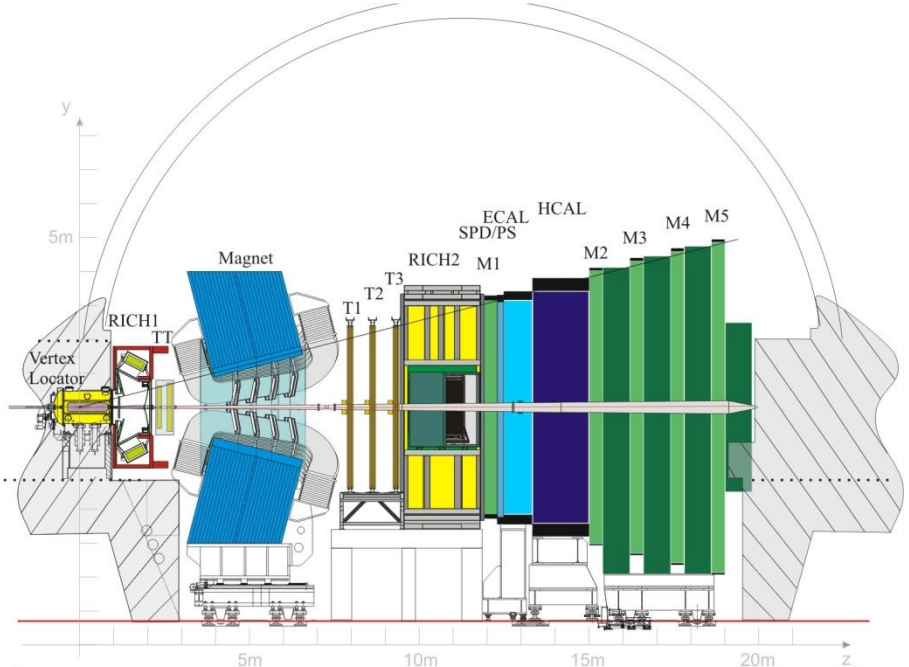
What is BIG ? ATLAS



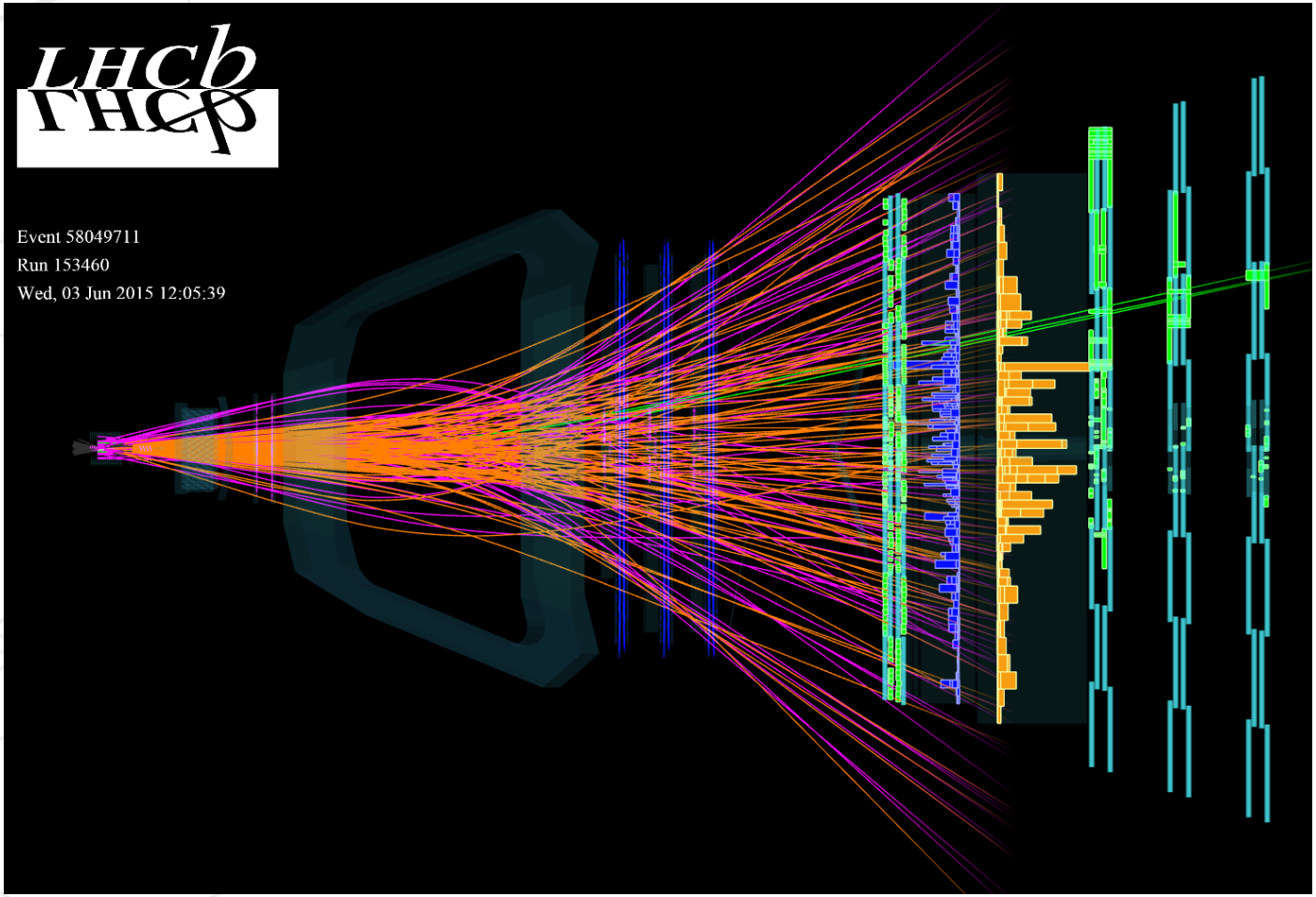
Big detectors as onions



first beam event seen in ATLAS



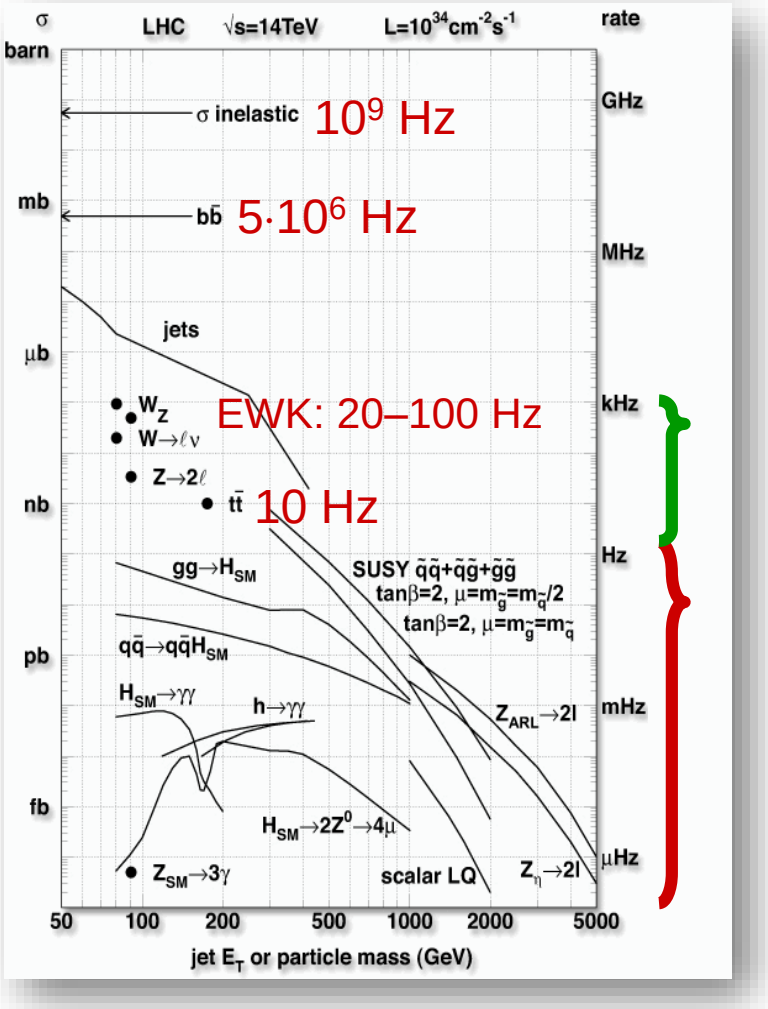
What is a collision



Background image: Shutterstock

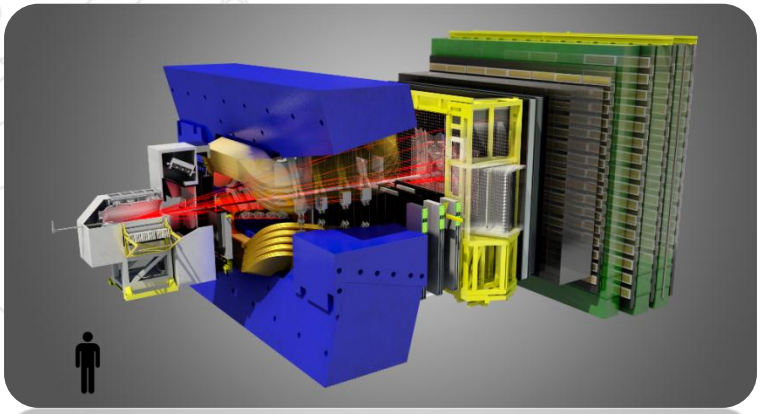
Most collisions are…. « useless »

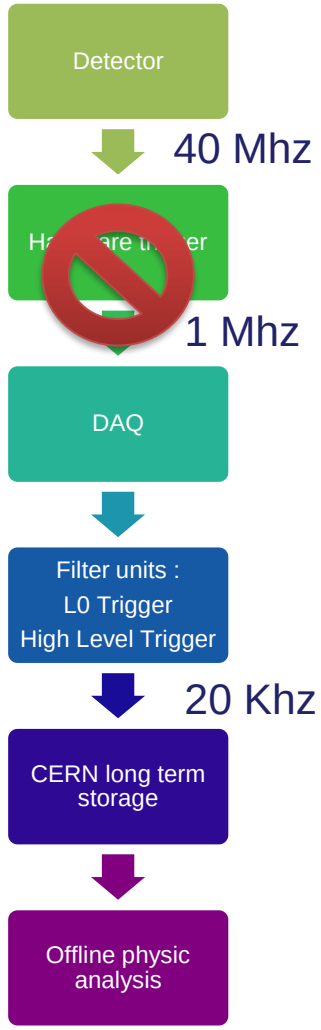
- A collision every **25ns**
- Meaning **40 Mhz**
- Most collisions are **well known physics**
- New physics is **rare**



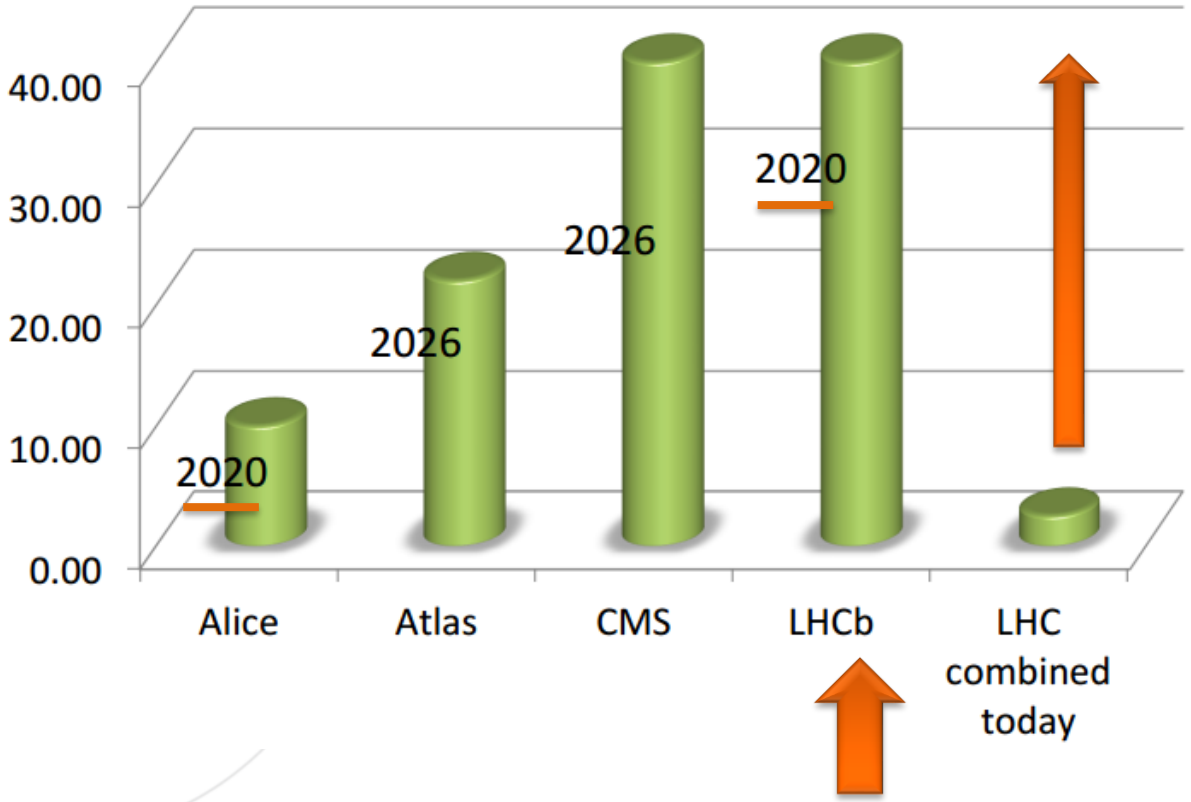
LHCb, an upgrade for 2018-2020

- Update of sub-detectors
- **Removal of hardware trigger**
  - Currently in **FPGA** and **analogic** components
  - Have to answer in **real time**
  - **Hard to maintain** and update
- **New status :**
  - Larger **event rate** (1 Mhz to 40 Mhz)
  - Larger **event size** (50 KB to ~100 KB)
- **Much more data for DAQ & Trigger (x80 => 40 Tb/s)**

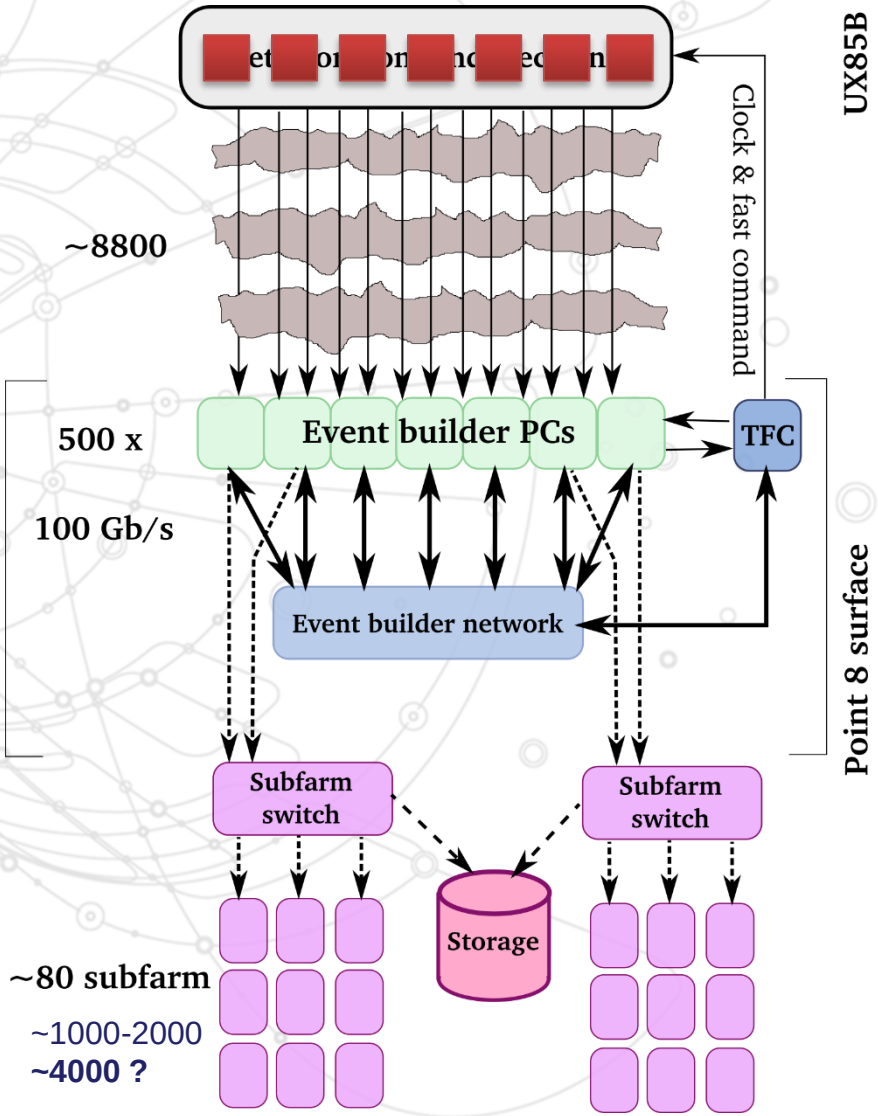




40 Tb/s for LHCb in 2020

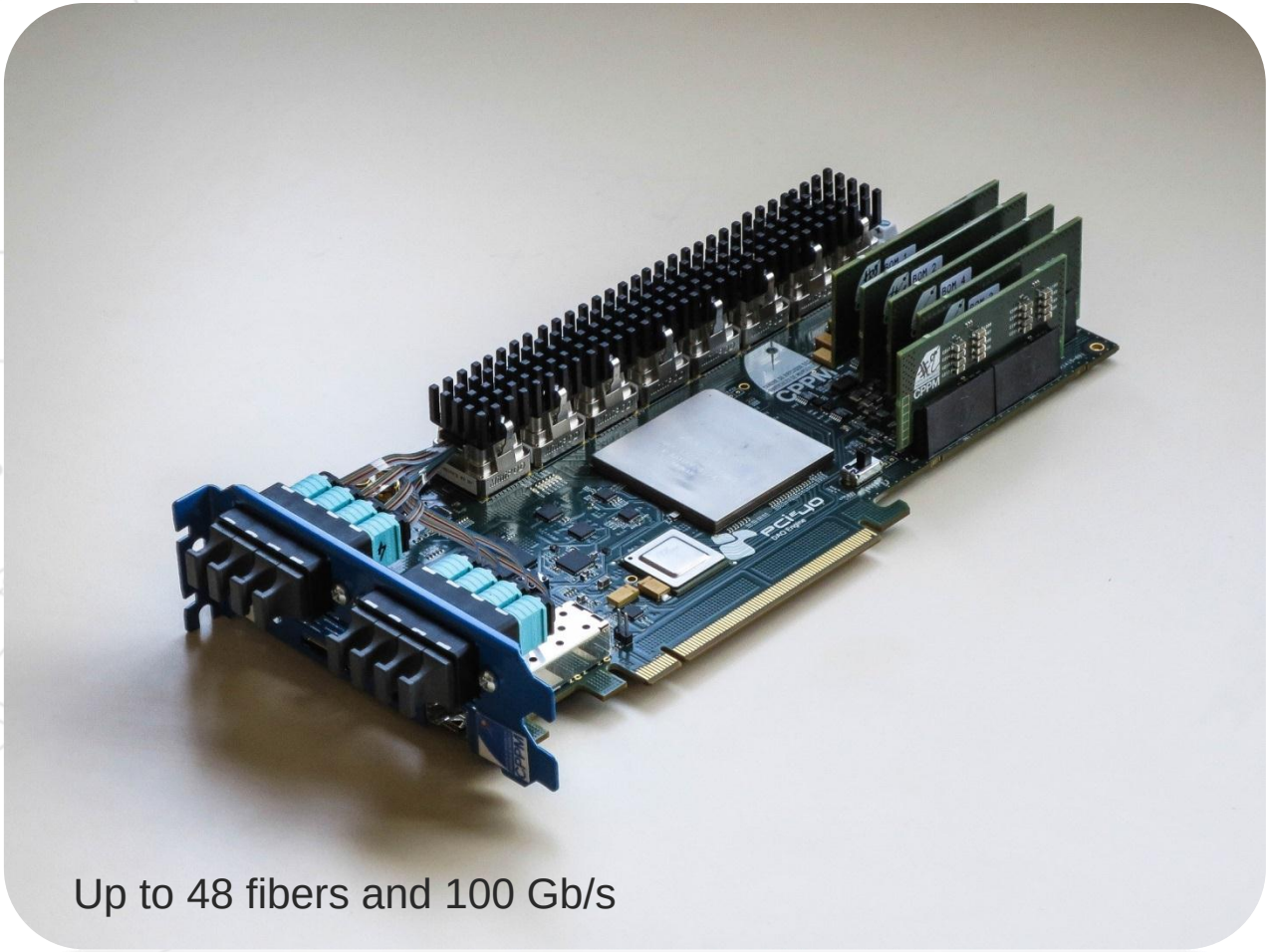


Data flow



- Numbers
  - **~8800** optical links going out from detector to the surface (**~100 m)** and up to **~4.8 Gb/s** each.
  - **~500** readout nodes (up to 24 input links each)
  - Filter farm of **O(1000)** nodes
- **40 Tb/s** to transport
  - Can run over **512 nodes**
  - Need a **100 Gb/s** network
  - Considering net. load at **80%**
  - So **80 Gb/s**
  - Current estimation is **70 Gb/s**

Starting point : the acquisition board



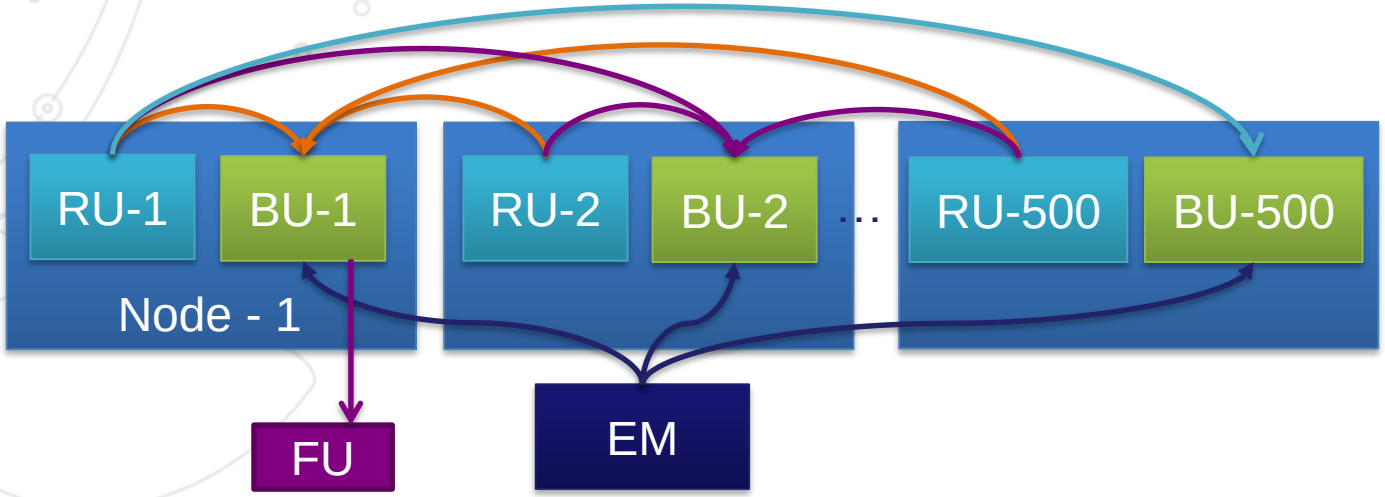
Up to 48 fibers and 100 Gb/s

Event building network benchmark

Event building network topology

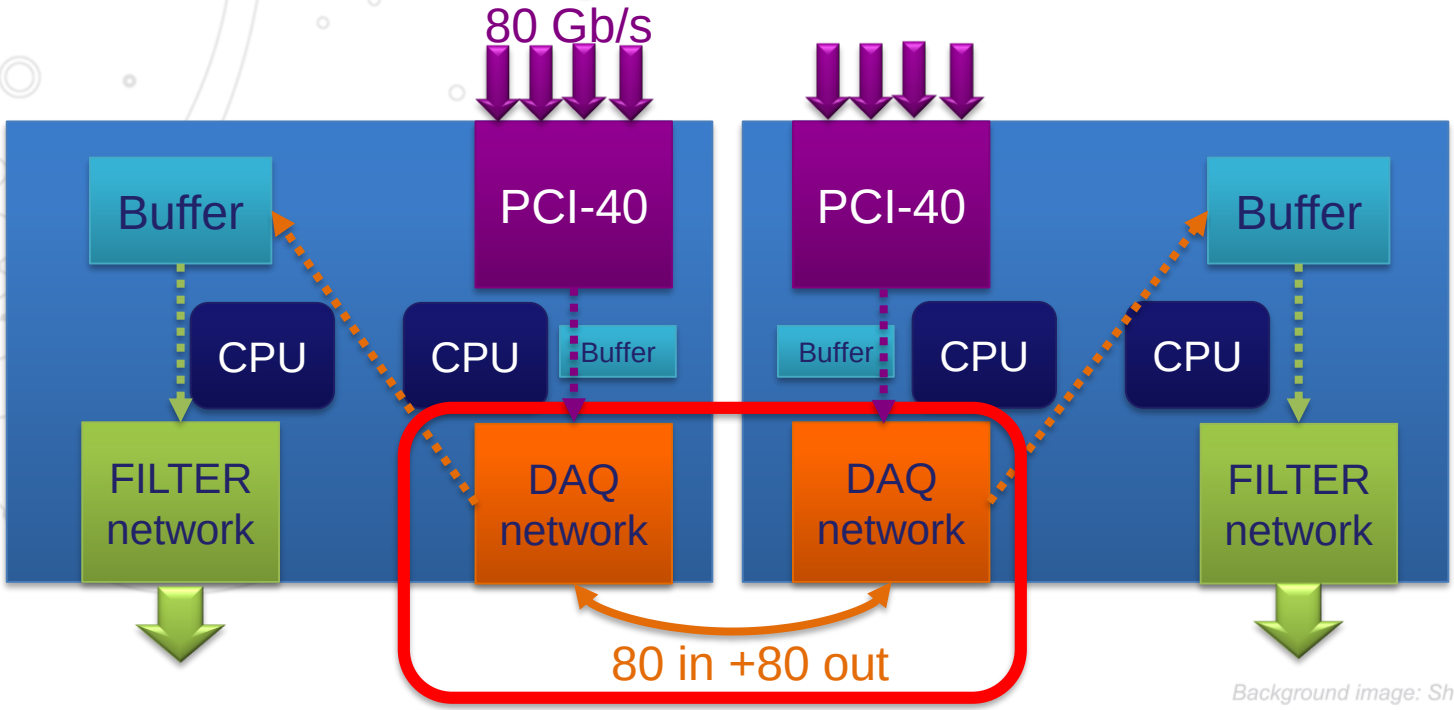
- Work with **4 units:**
  - **Readout unit (RU)** to **read** data from PCI-E readout board
  - **Builder unit (BU)** to **merge** data from **Readout unit** and send to **Filter unit.**
  - **Filter unit (FU)** to select the interesting data (**not considered here**)
  - **Event manager (EM)** to dispatch the work over **Builder unit** (credits)

- **RU/BU mostly does a "all-to-all"**
  - To **aggregate** the data chunks from each collision



IO nodes hardware

- **Three IO boards at 100 Gb/s per node:**
  - PCI-40 for fiber input
  - Event building network
  - Output to filter farm
- **Also stress the memory (320 Gb/s of total traffic)**
  - 80 in + 80 out

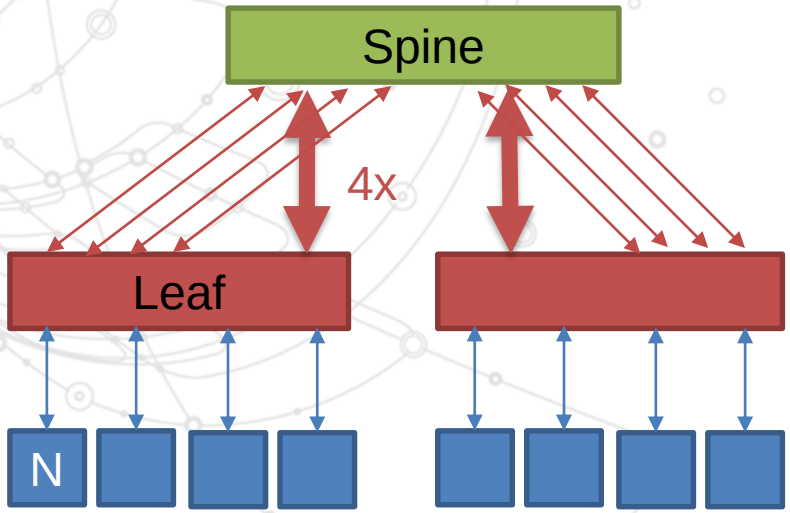


The DAQ network technologies

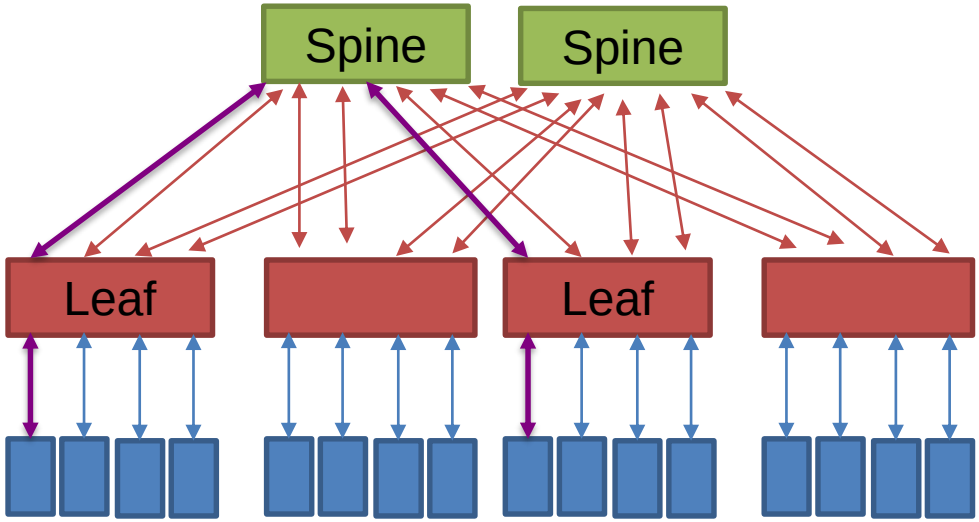
- **We expect to use HPC networks**
  - 100 Gb/s (expect running at 80 Gb/s)
- **Technologies we looked on:**
  - **Infiniband** EDR (HDR 200 Gb/s ?)
  - Intel **Omni-path** (~~version 2, 200 Gb/s ?~~)
  - 100 Gb/s **Ethernet**
- **Not as HPC apps**
  - We need to **fully fill the network continuously**
  - In **full duplex**
  - Bad pattern : **many gathers** (all-to-all) !
- **Think using a fat-tree topology**

Fat tree

- **Fat tree, theory**
  - Based on **switches**
  - **More bandwidth** at **each level**



- **Fat tree : reality**
  - Switch **radix** is currently **36** or **48**
  - Count **cables** for **512 nodes** !



Switch and cables

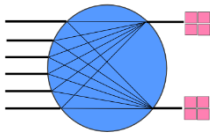
- **Example of Mellanox switch (36 ports)**

- **Can use director switch for spine**
  - Provide **648 ports** (Intel has up to **754**)
- **Cables**
  - **Capper** : up to **3 meters,** price O(**150 CHF**)
  - **Optics** : up to **100 meters,** price O(**500 CHF**)

Interfaces to exploit them

- **MPI**
  - **Support all networks**
  - **Optimized** for **IB/Omni-path**
  - **Ease test on HPC sites**
  - How to support **fault tolerance ?**
    - We need to run 24h/24h and 7d/7
- **Infiniband VERBS**
  - For **IB** and **Ethernet** (RoCE / softRoCE / iwrap)
  - Low level
  - **OK** for **fault tolerance**
- **Libfabric**
  - For **IB**, **Omni-path**, and TCP
  - Low level
  - Node **failure** and **recovery** support **to be studied**.
- **TCP/UPD** (we don't depend on latency)
  - Issue : **CPU usage** and memory **bandwidth**

CERN openlab

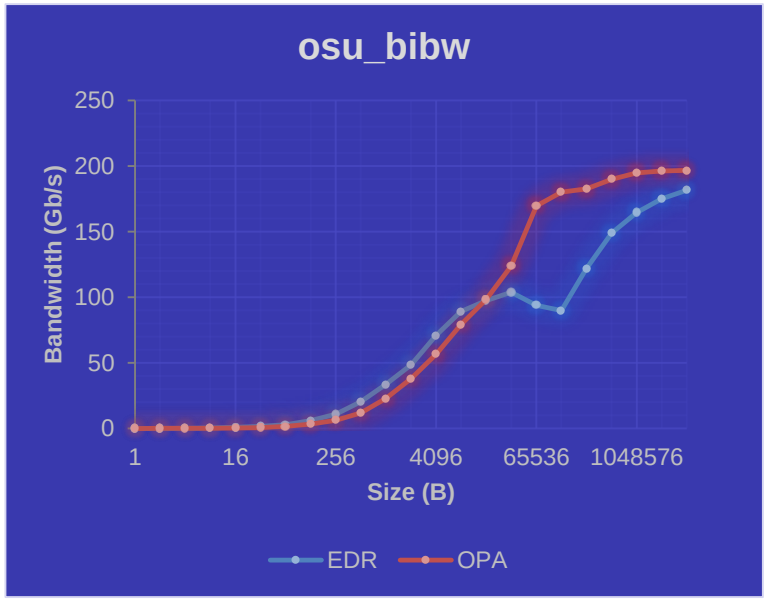


DAQPIPE

- **A benchmark to evaluate event building solutions**
  - DAQ Protocol Independent Performance Evaluator
  - Provides EM/RU/BU units
- **Support various APIs**
  - **MPI**
  - Libfabric
  - PSM2
  - Verbs
  - TCP / UDP
  - RapidIO
- **Multiple protocols**
  - PUSH
  - **PULL**

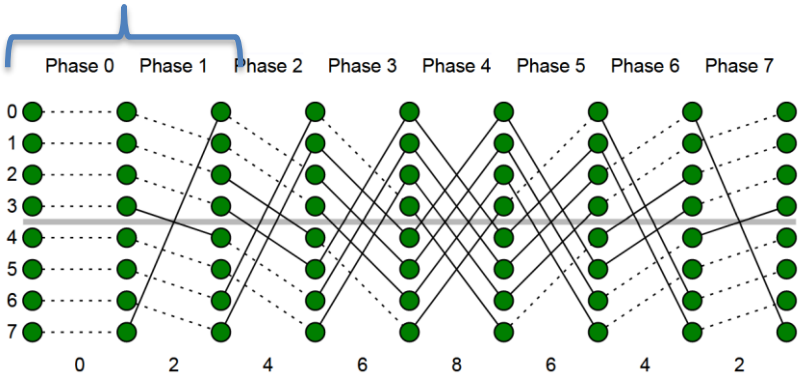
DAQPIPE

- **We need to pack events**
  - 100 KB / 512 = **~200 bytes**
- **Three message size on the network**
  - Command : **~64 B**
  - Meta-data : **~10 KB**
  - Data : **~512 KB** or **1 MB**



Communication control parameters

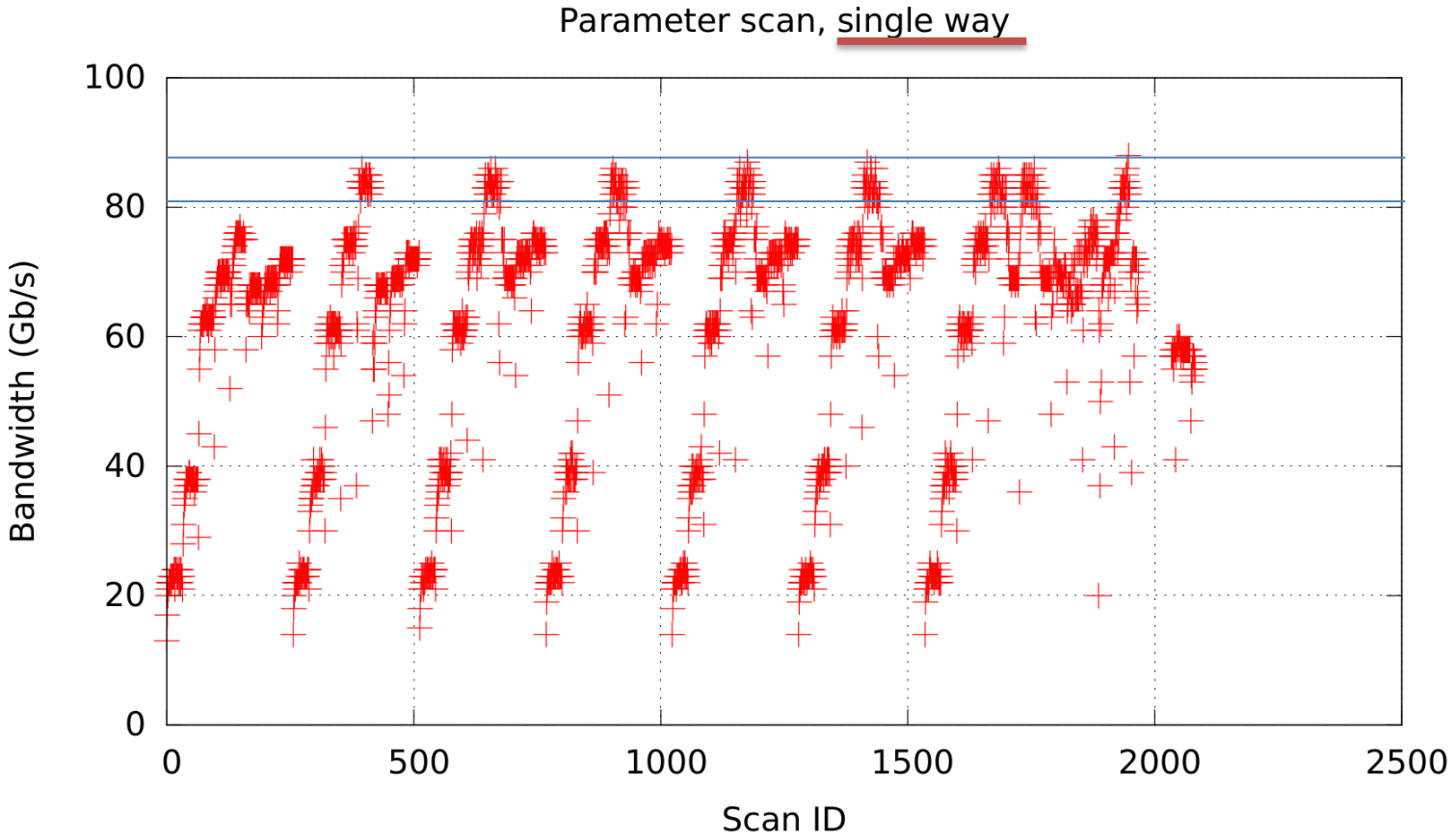
- **Manage communication scheduling models**
  - **Barrel shift** ordering (with N on-fly)
  - **Random** ordering (with N on-fly)
  - Send **all in one go**
- **Messages size**
  - best **~512 KB**
- **Parallel gathers (credits)**
  - best **~2**
- **Parallel communication per gather**
  - With how many nodes we communicate
  - best **~8**
  - So in total **2*8** communication in flight
- **Process per nodes, 1 or 2**
  - best **2**



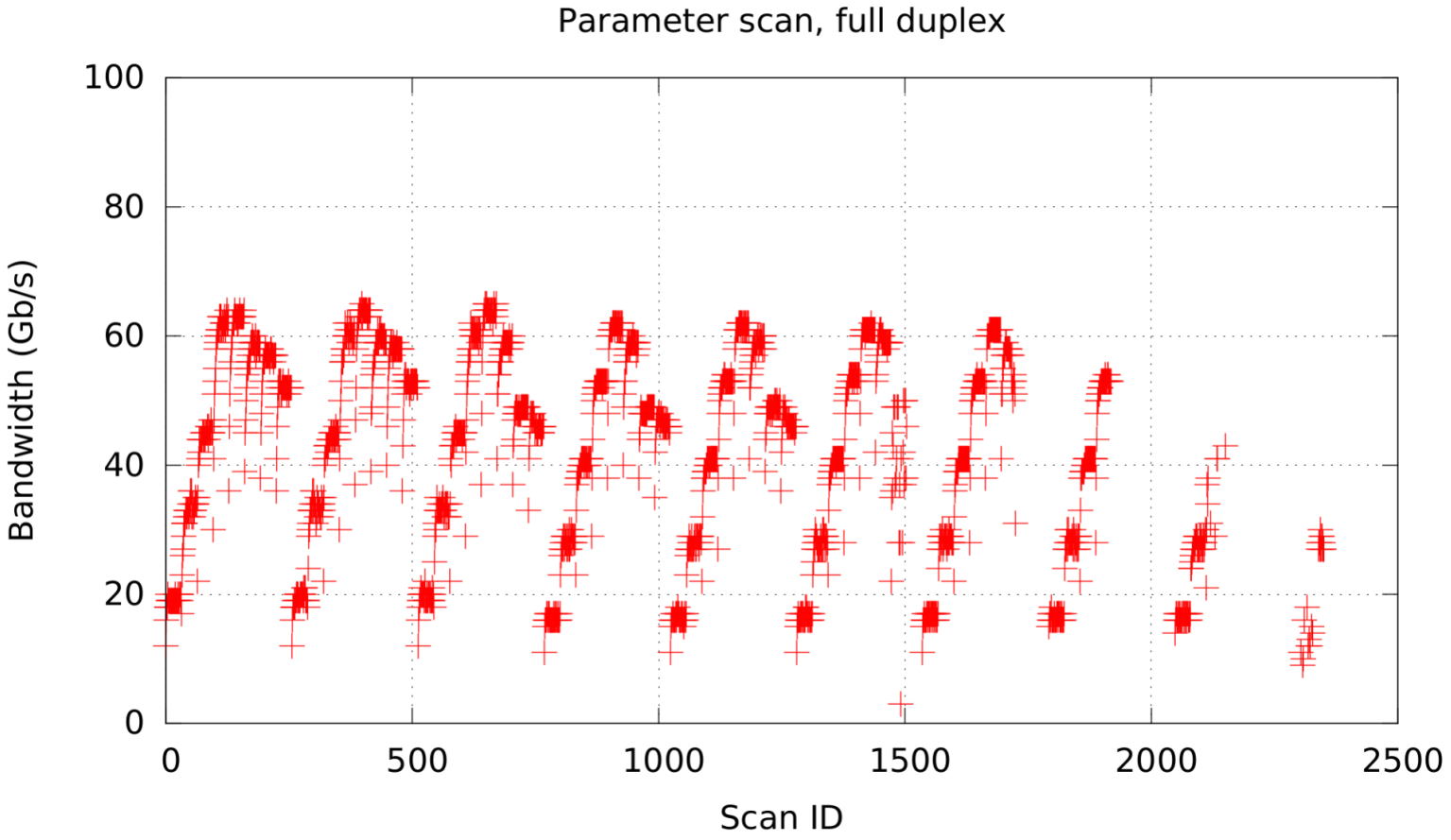
* From Bandwidth-optimal all-to-all exchanges in fat tree networks

What we have seen here on our cluster with 16 nodes OPA

87 Gb/s = **2%** of the points



Performance stability over parameters (16 nodes OPA)

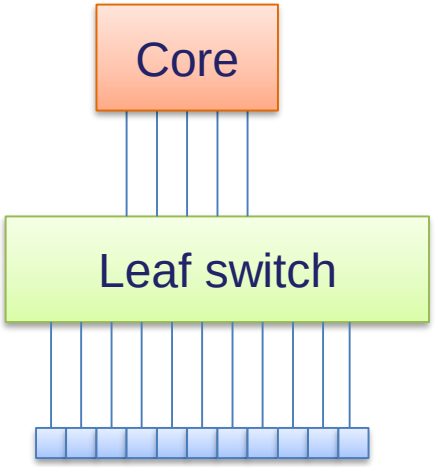


From http://www.flickriver.com/photos/tags/supercomputer/interesting/

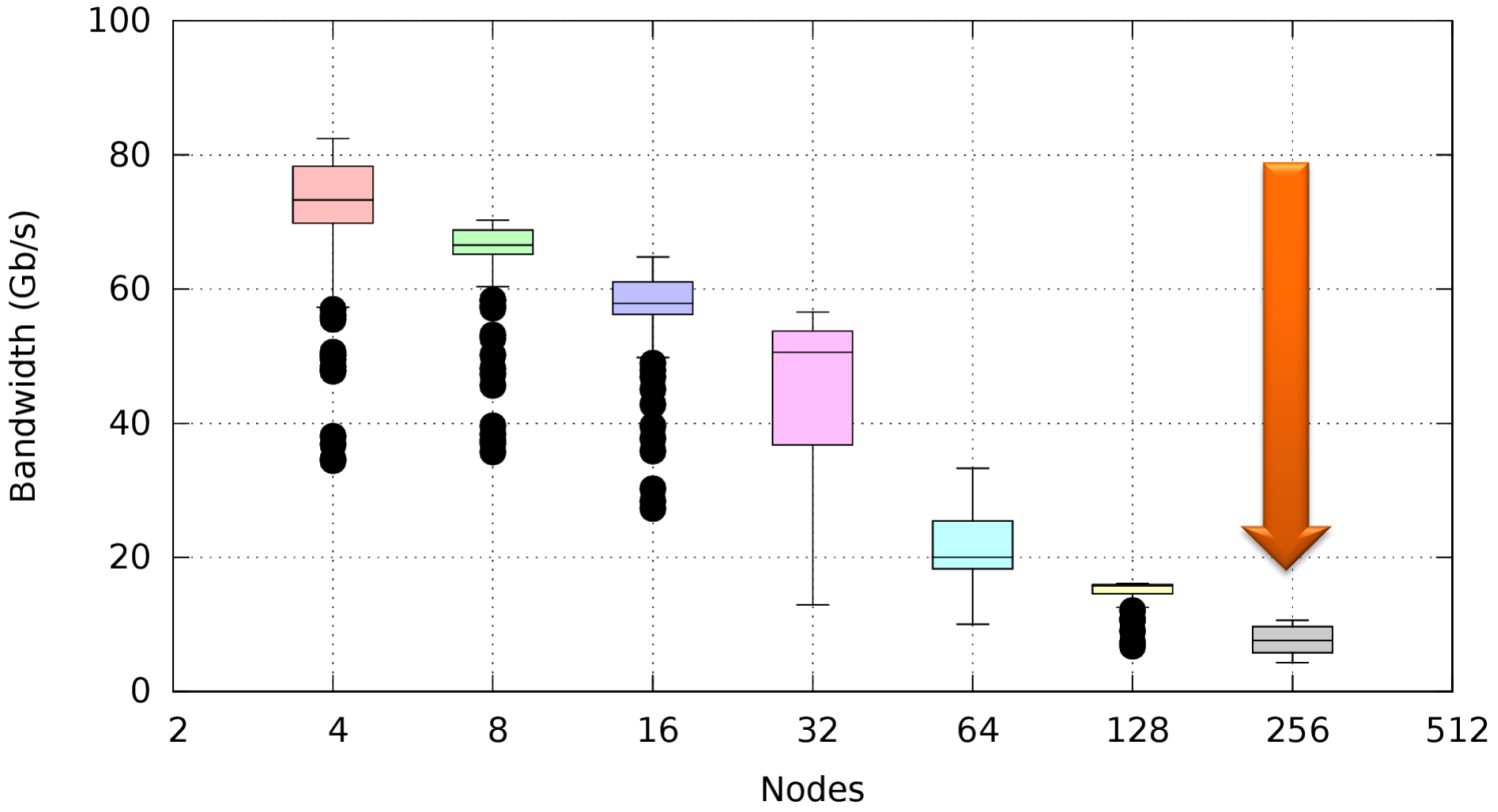
First test at scale (512): hitting the wall ☹

Mapping

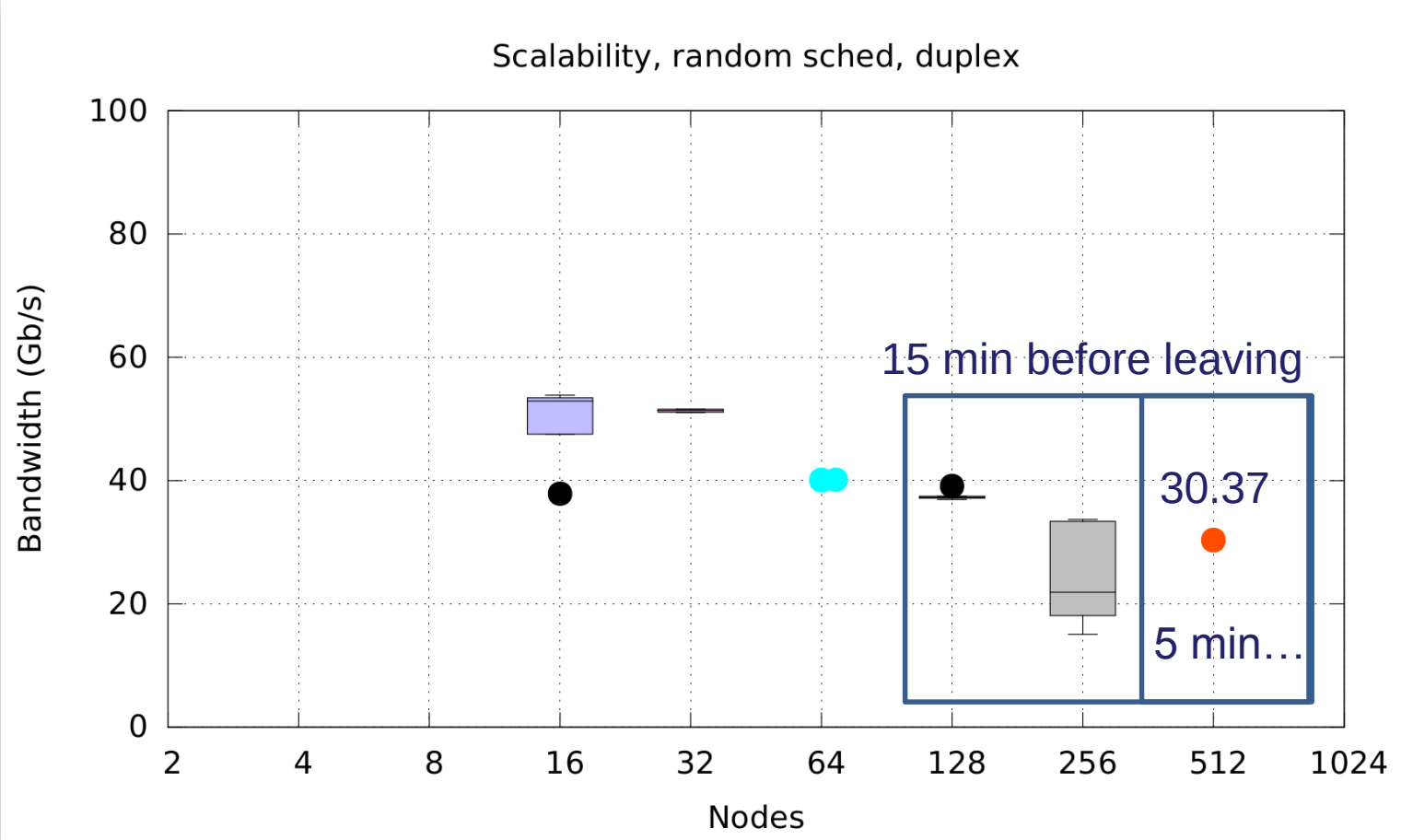
- This cluster had a 2:1 mapping
- 15 link up for 32 nodes
- We selected 15 nodes per leaf switch
- “Problem” I don’t know which nodes they gave us on each sub switches



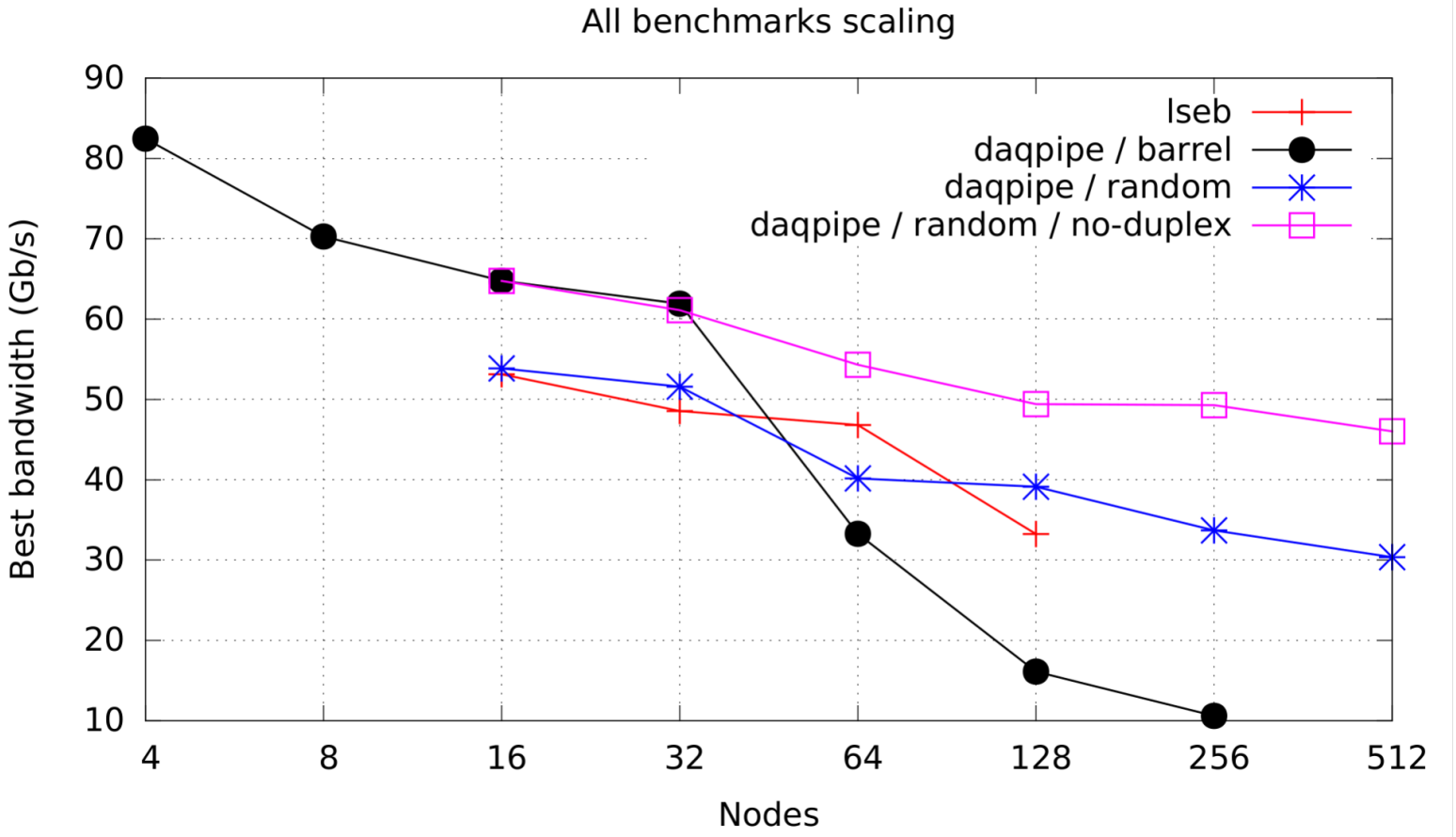
Summary scalability on Day-1 …. ☹



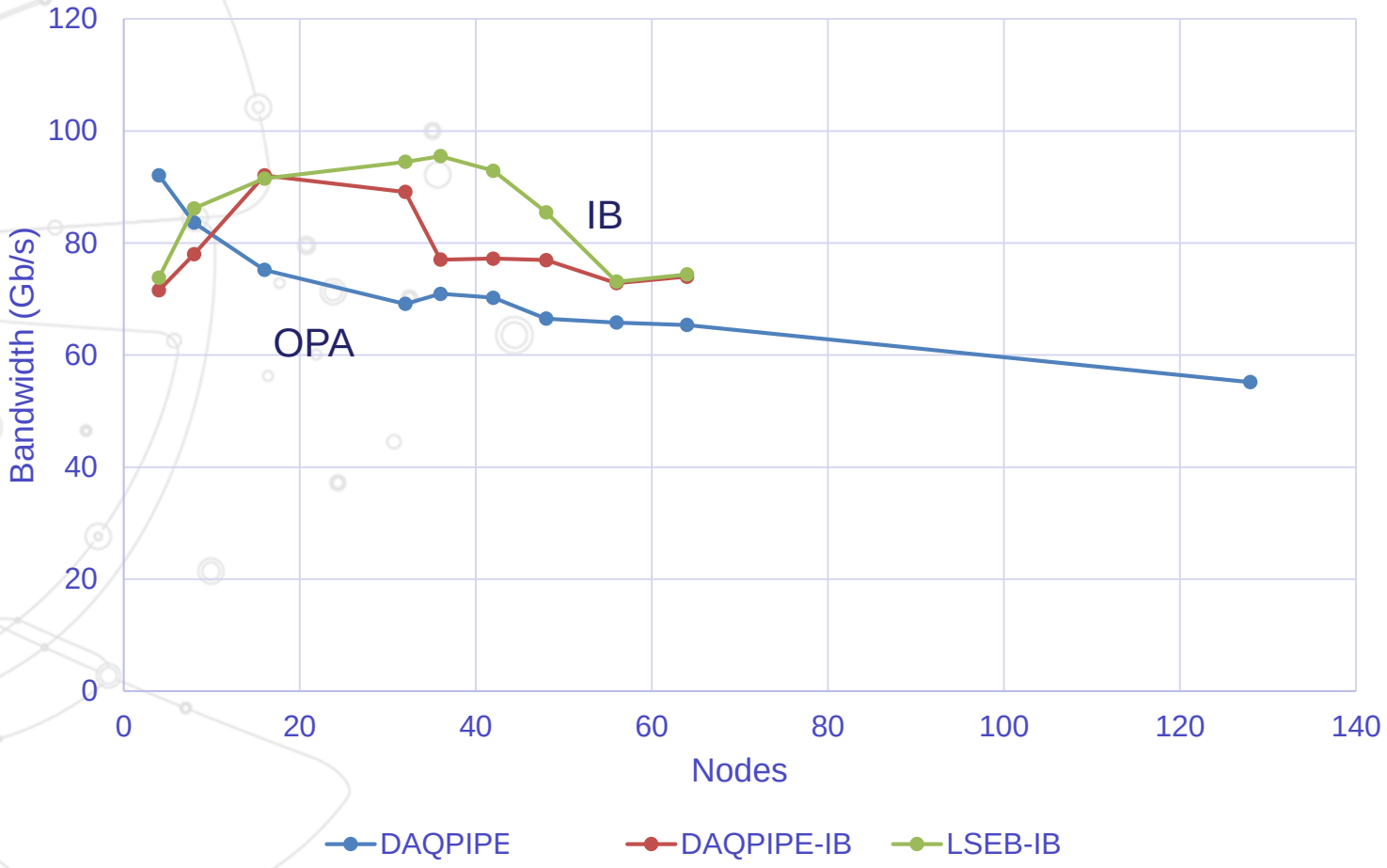
Day-2, 2 hours before leaving try random scheduling



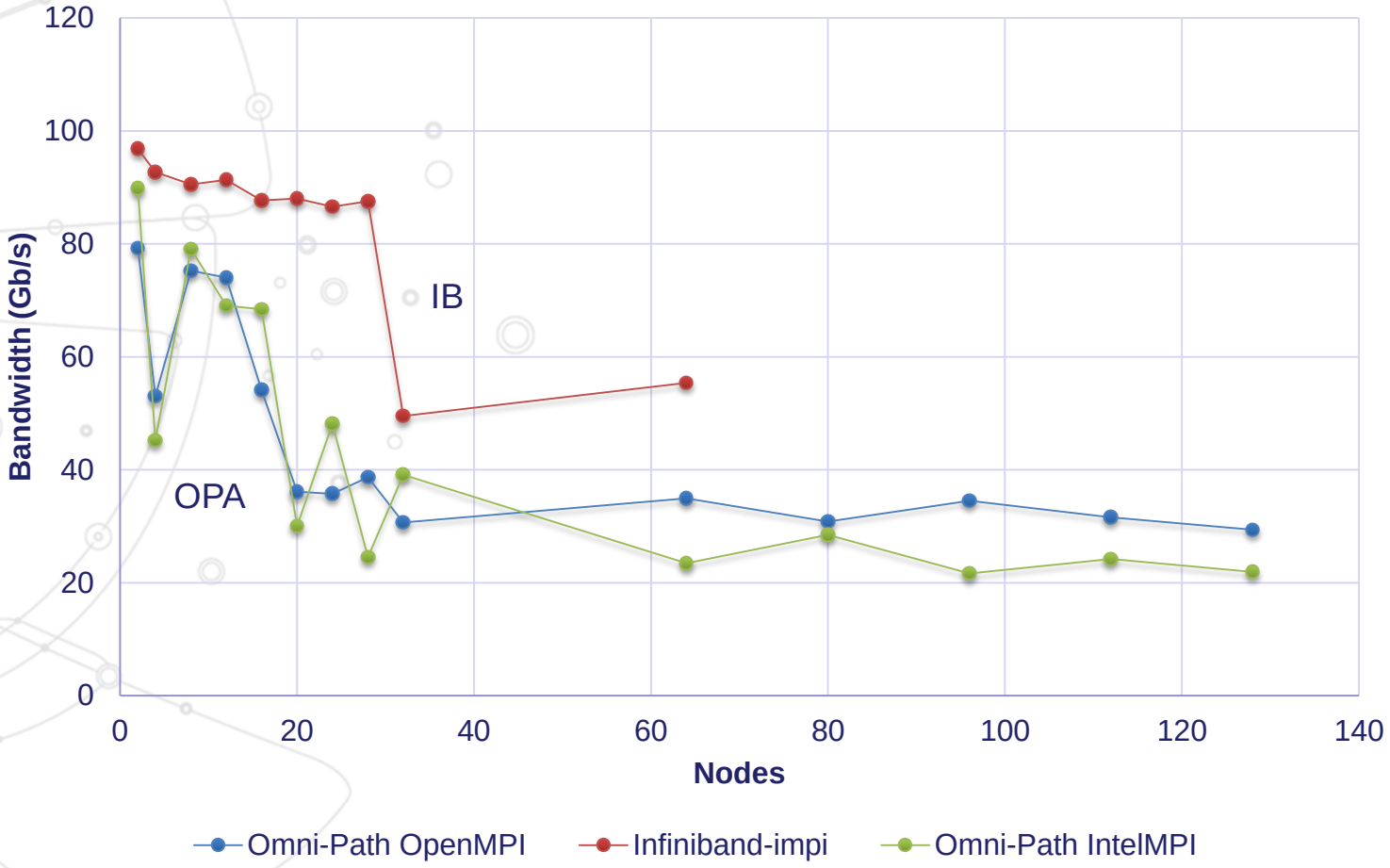
Final summary



Another test on two clusters

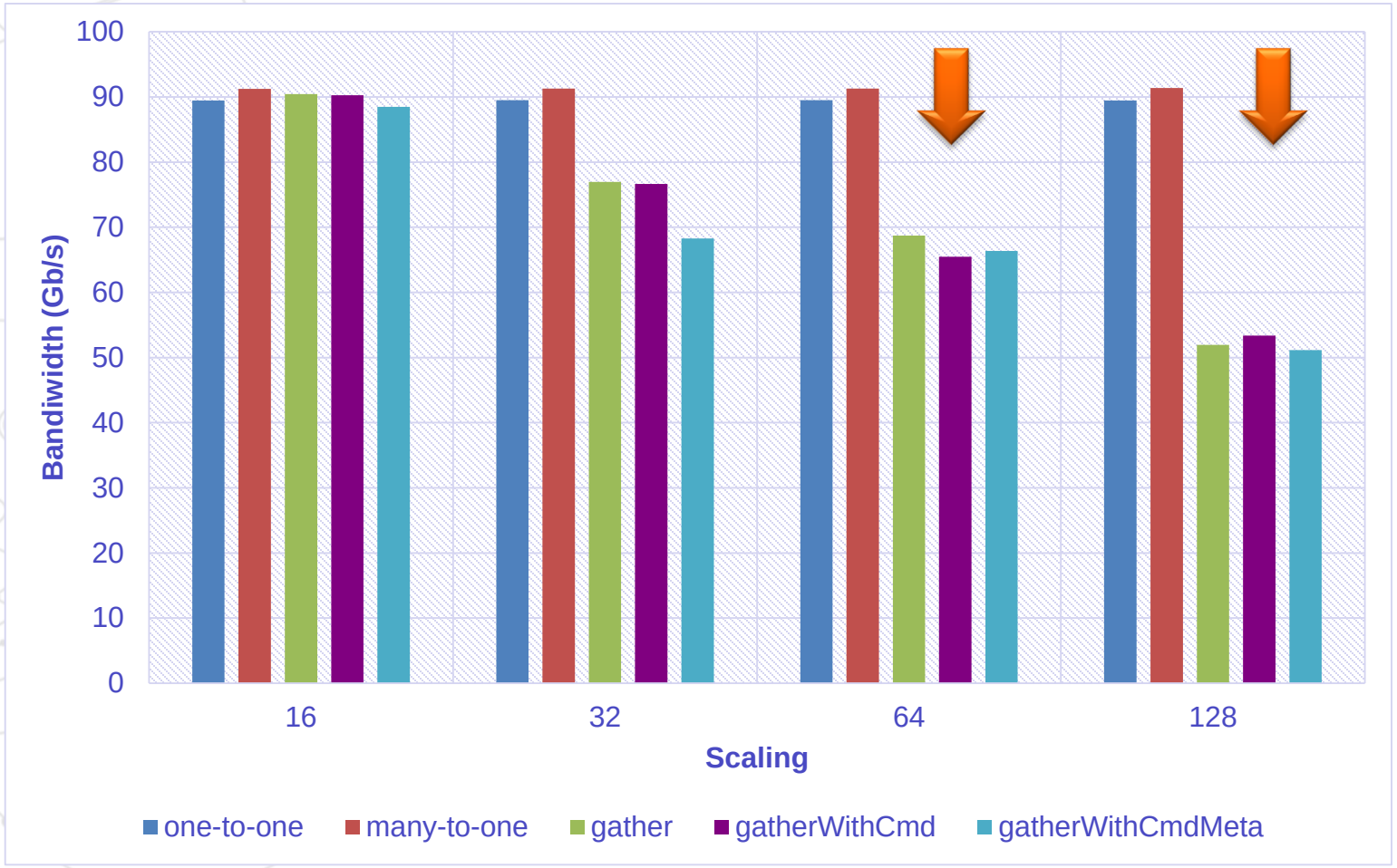


MPI_Alltoall



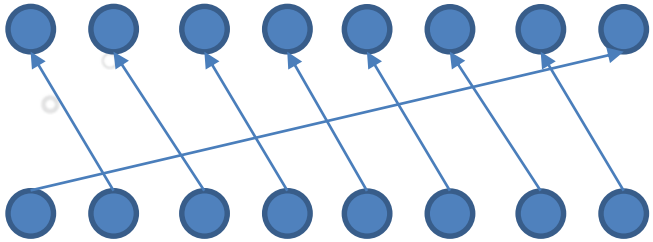
Routing & cabling

Micro-benchmarks on OPA

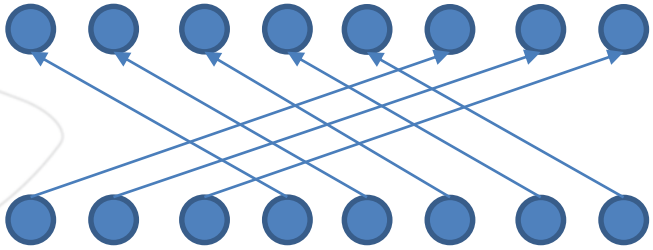


Performance of one-to-one

- Default one to one send to the neighbor node
- One variant is to **shift**
- Each represent **a step** in DAQPIPE (**barrel shifting**)
- Similar discussion than [1] but with experiment measurements



Shift = 1

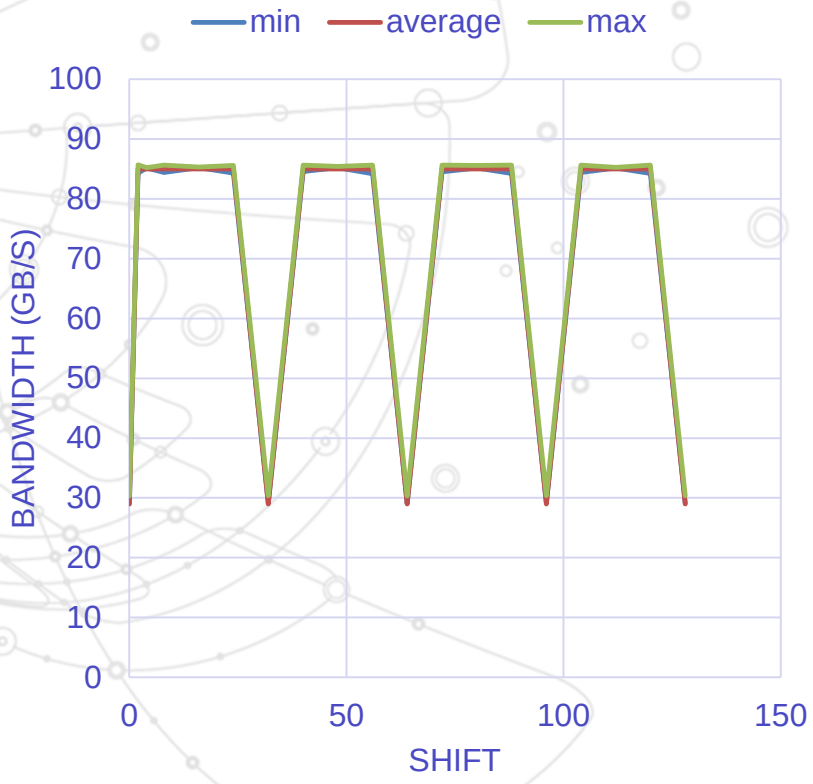


Shift = 3

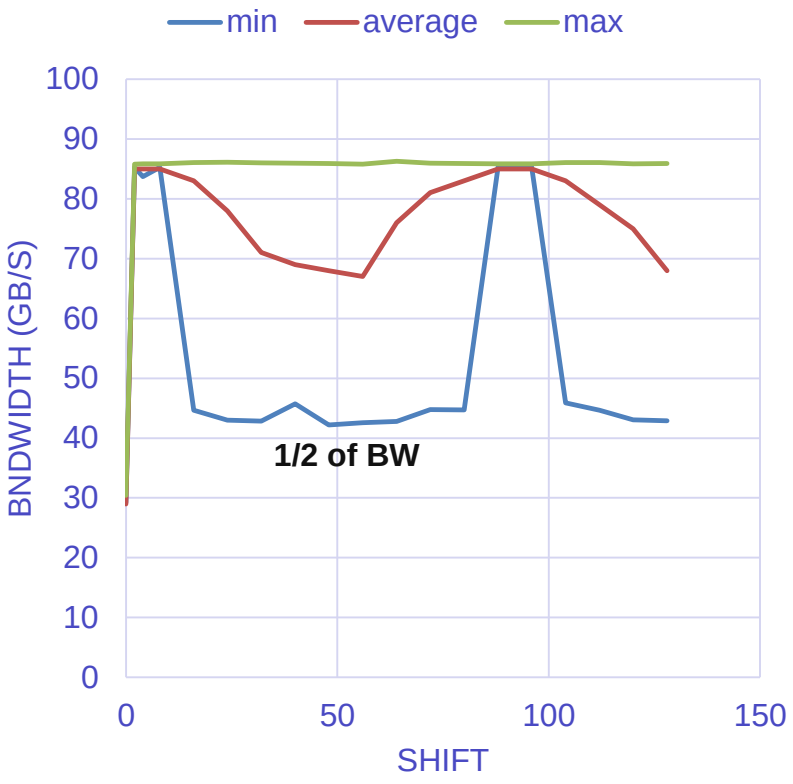
[1] Bandwidth-optimal All-to-all Exchanges in Fat Tree Networks, B. Prisacari

One-to-one with shifts InfiniBand

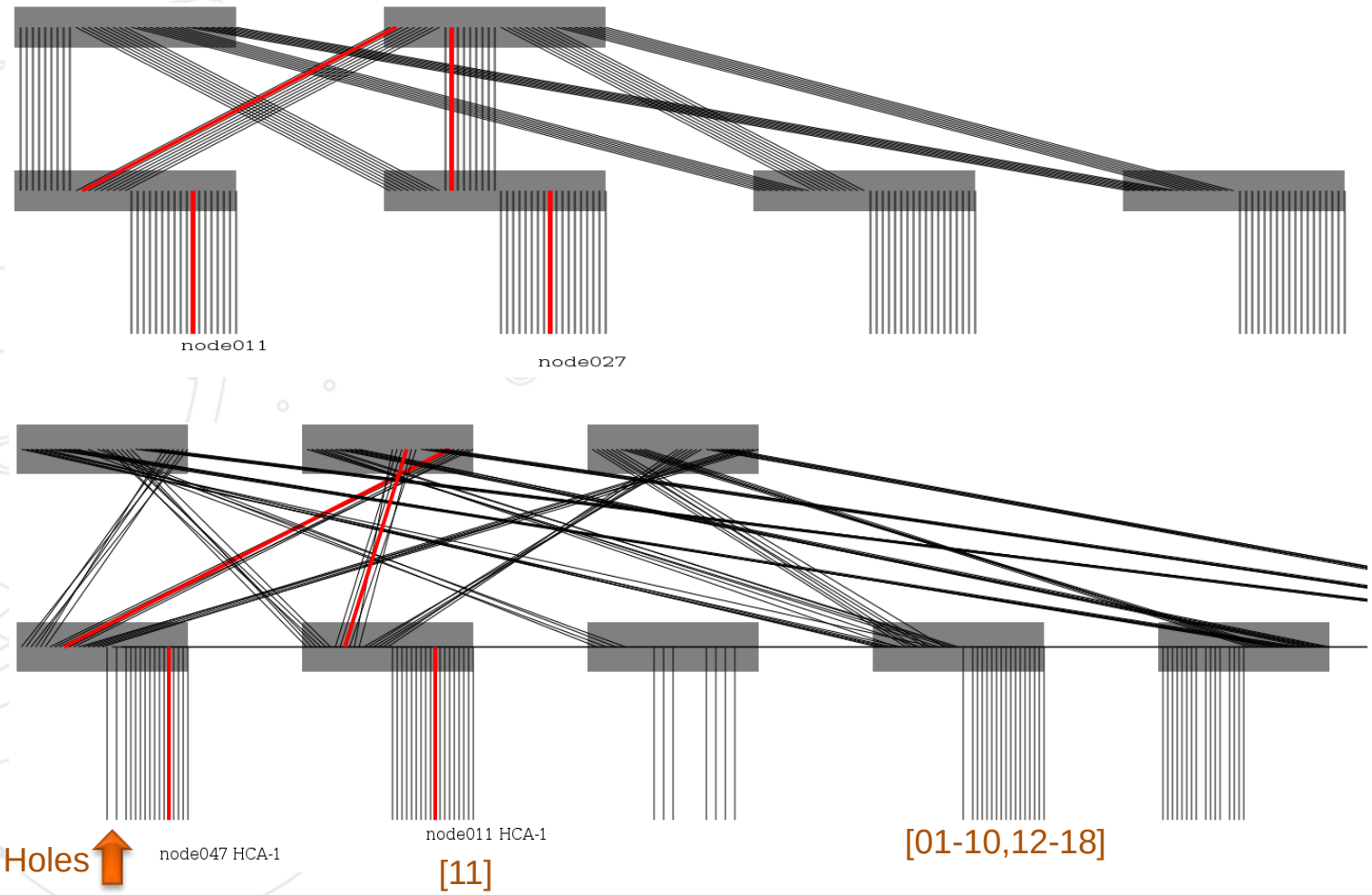
Full-duplex 16 nodes



Full-duplex, 46 nodes



Ideal and real topologies



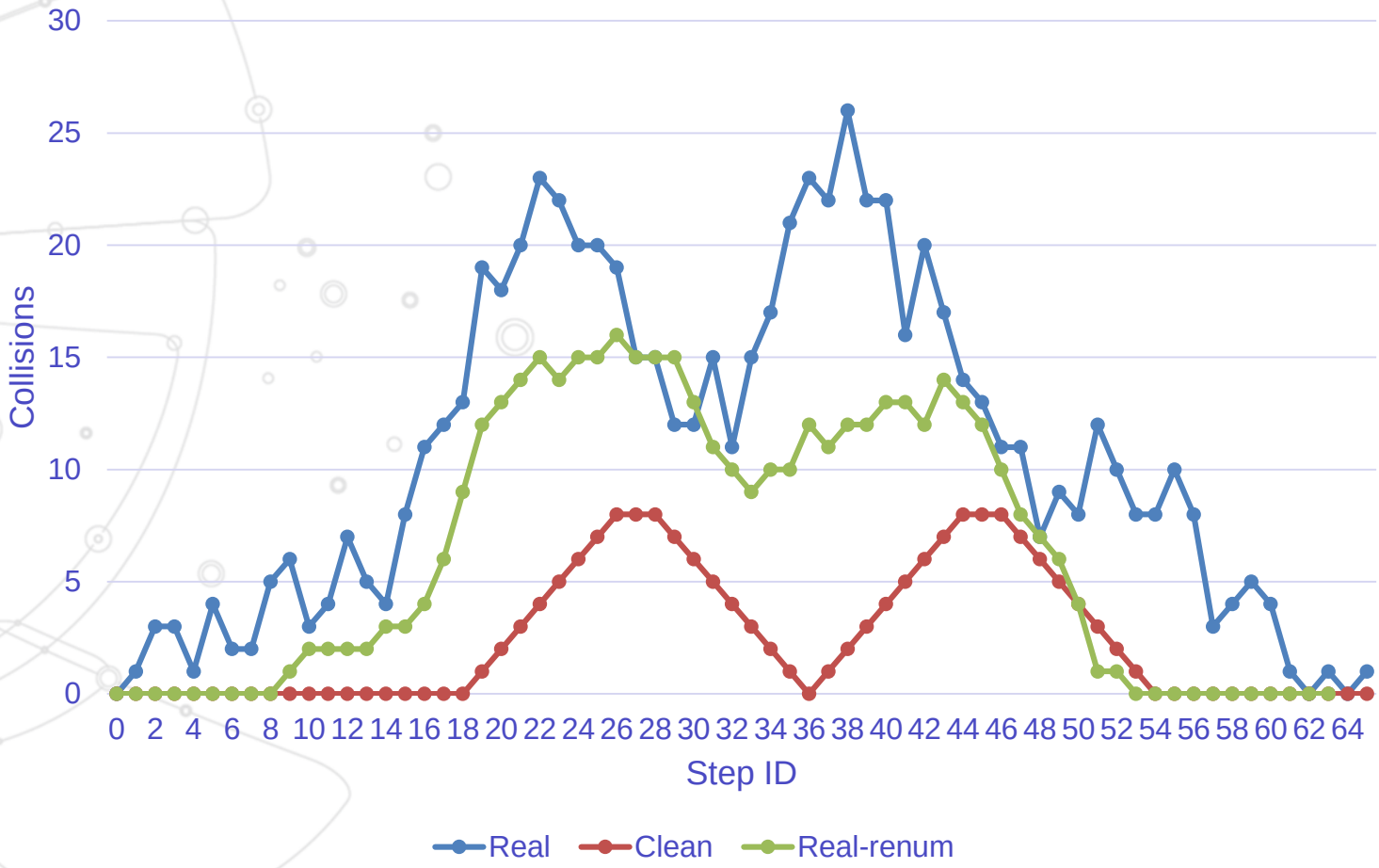
Cabling on director switch Is it better ?

```
{ 'OmniPth00117501fa000404 L110B':
  { '1': 'node027 hfi1_0',
    '2': 'node030 hfi1_0',
    '3': 'node126 hfi1_0',
    '5': 'node036 hfi1_0',
    '6': 'node031 hfi1_0',
    '7': 'node123 hfi1_0',
    '8': 'node122 hfi1_0',
    '9': 'node006 hfi1_0',
    '10': 'node033 hfi1_0',
    '11': 'node034 hfi1_0',
    '12': 'node016 hfi1_0',
    '13': 'node004 hfi1_0',
    '14': 'node002 hfi1_0',
    '15': 'node125 hfi1_0' },
```

```
'OmniPth00117501fa000404 L118A':
  { '2': 'node116 hfi1_0',
    '7': 'node008 hfi1_0',
    '11': 'node021 hfi1_0',
    '15': 'node012 hfi1_0' },
 'OmniPth00117501fa000404 L118B':
  { '2': 'node010 hfi1_0',
    '3': 'node009 hfi1_0',
    '5': 'node121 hfi1_0',
    '6': 'node022 hfi1_0',
    '7': 'node120 hfi1_0',
    '8': 'node117 hfi1_0',
    '9': 'node020 hfi1_0',
    '10': 'node011 hfi1_0',
    '13': 'node115 hfi1_0' },
```

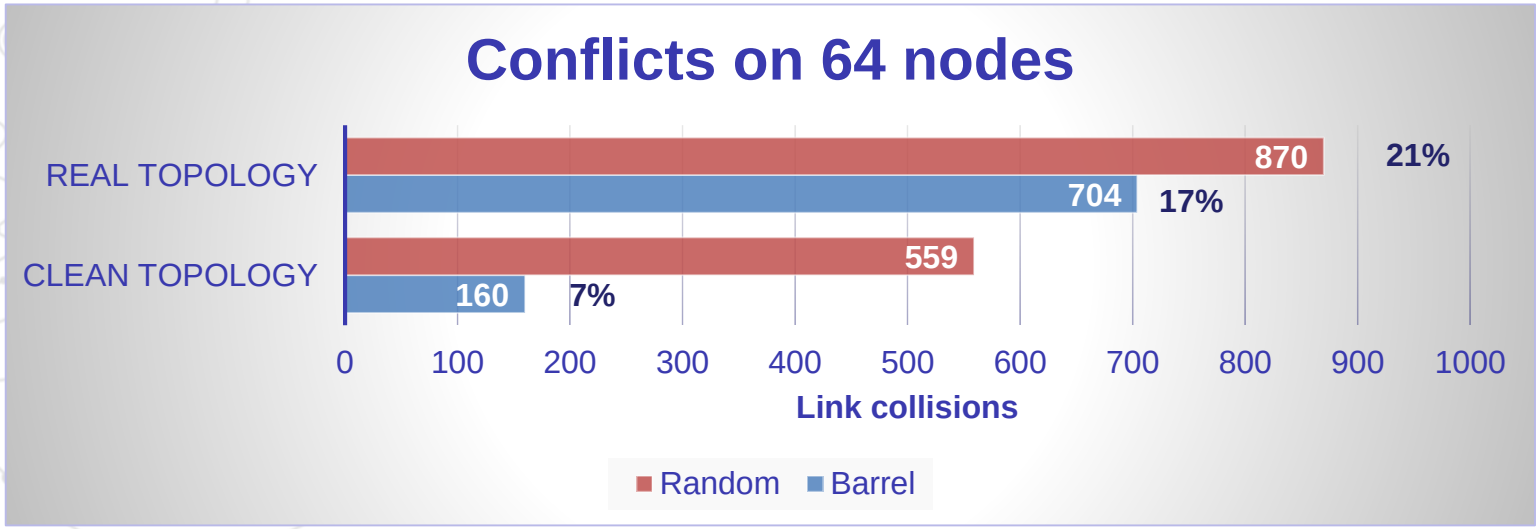
Problem: Slurm numberd then **ranks** by **ordering node names**

Collisions over barrel steps On 64 InfiniBand nodes

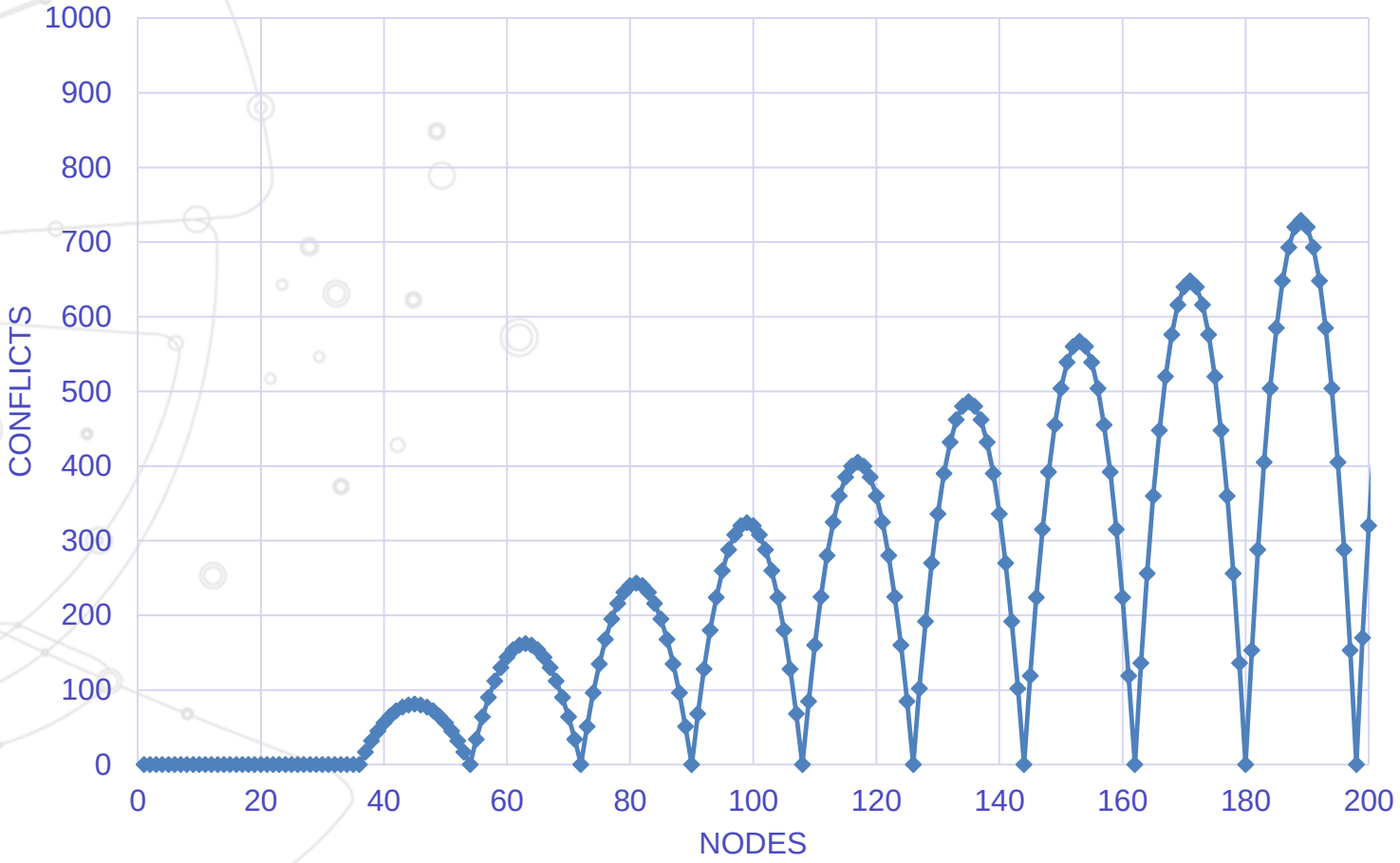


Conflicts

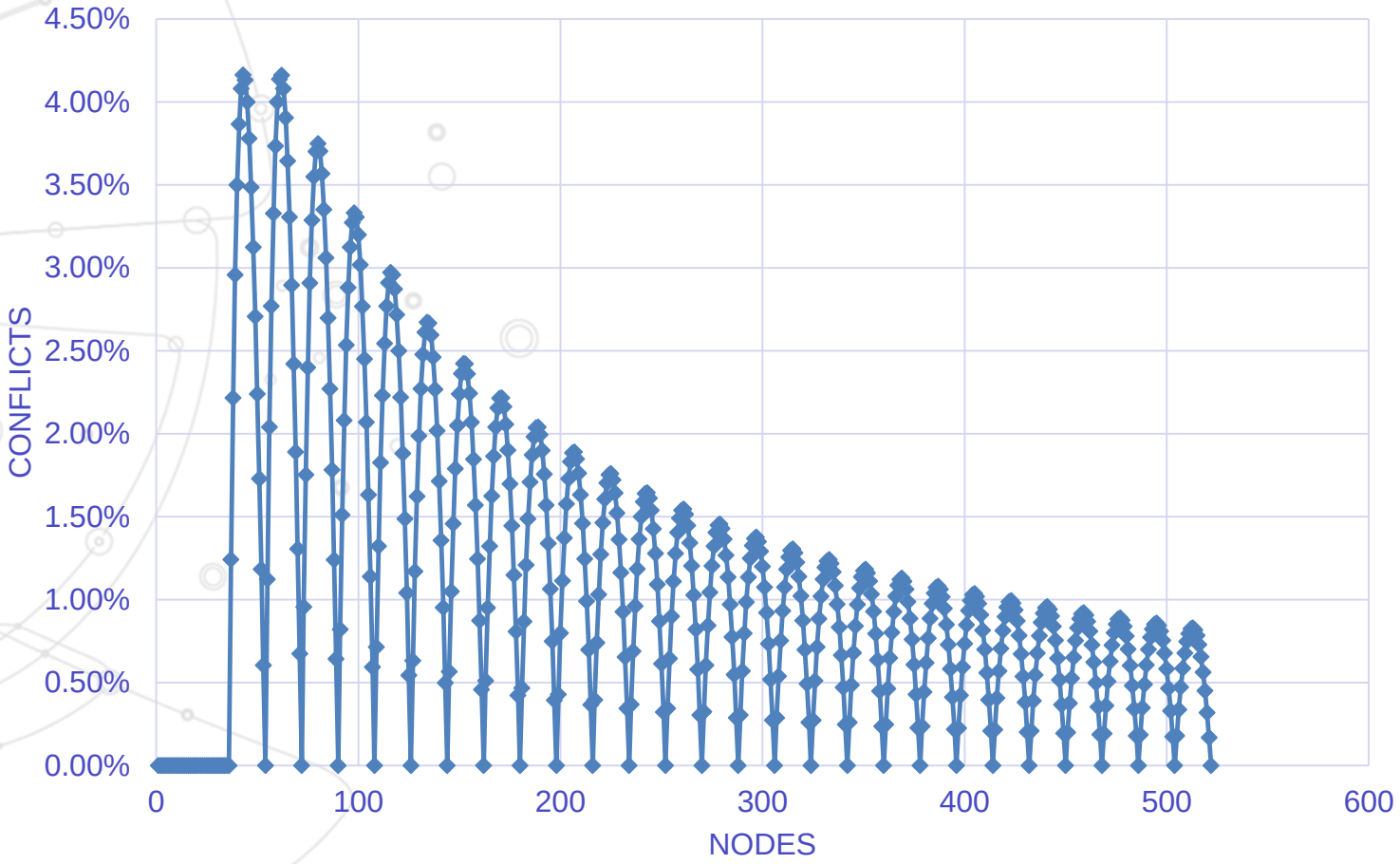
- **The real topology implies many conflicts**
- **Due to :**
  - routing table
  - missing nodes in leaf switches
- **64 nodes imply 4096 communications**
  - 870 conflicts => **21%**, 704 conflicts => **17%**, 160 conflicts => **3%**



Conflicts on clean topology

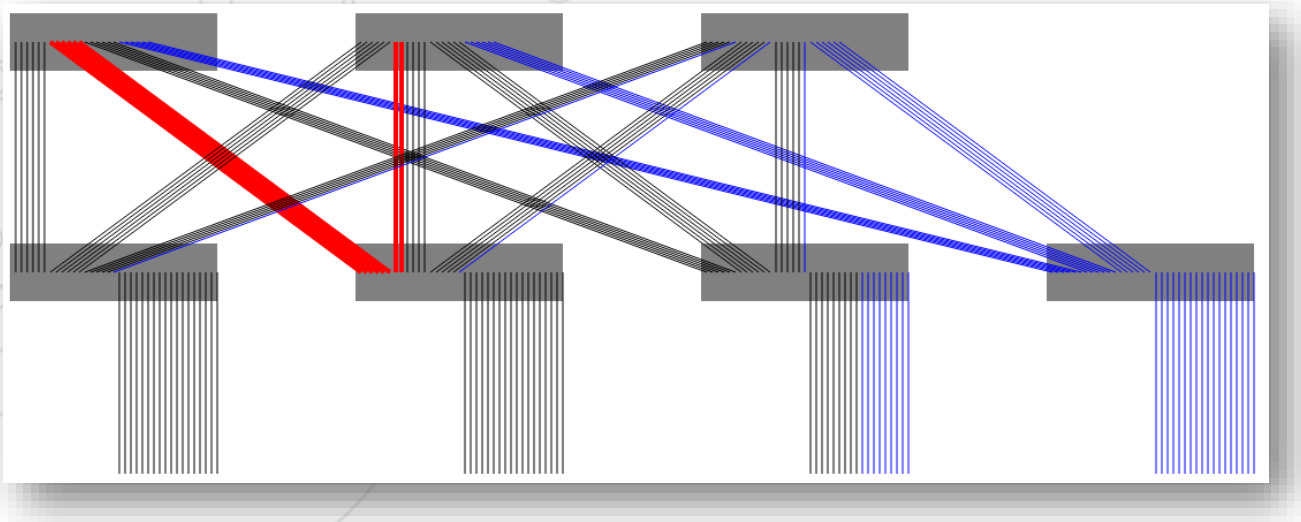


Conflicts on clean topology Pourcentage…

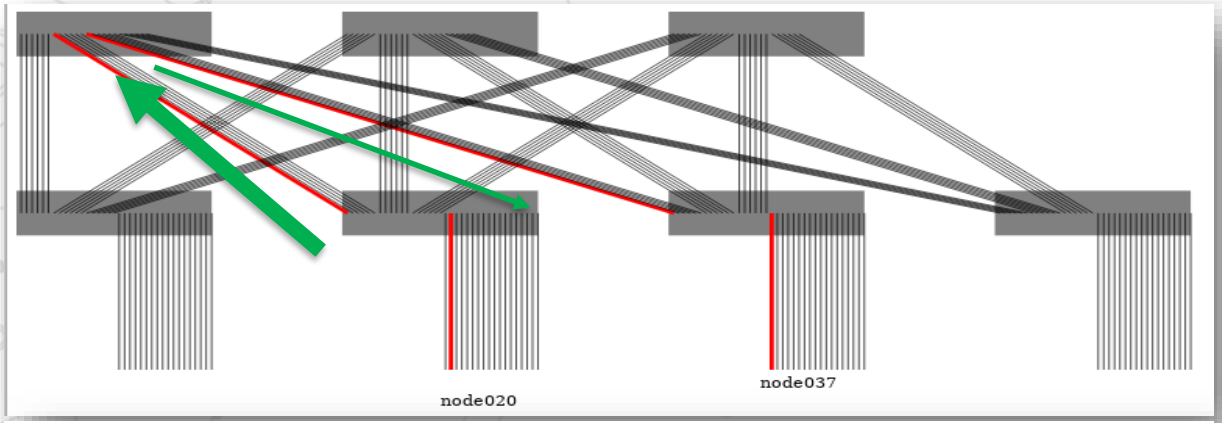


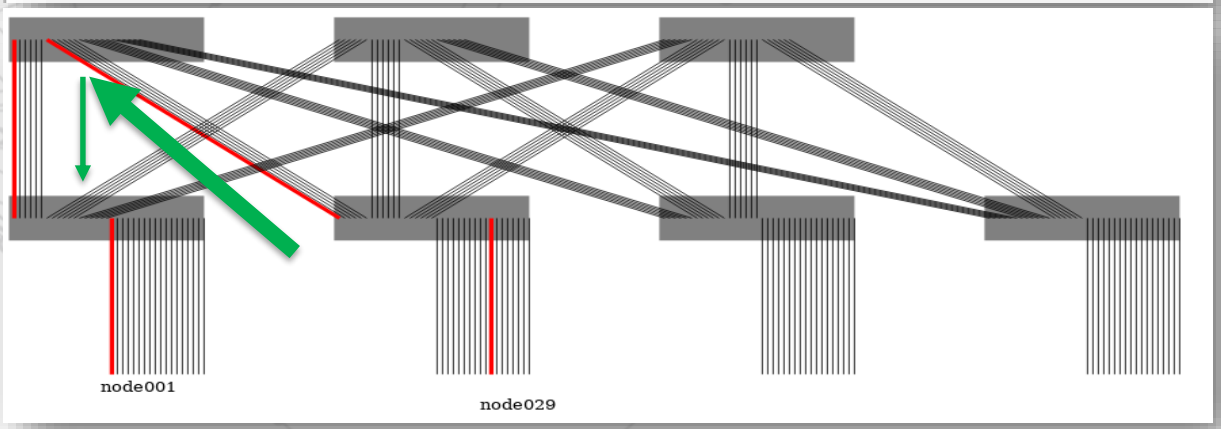
Example on clean topology

- Switch radix : 36
- Run on 45 nodes (2 * 18 + 9)
- Look on step 28 => 9 conflicts



Extract two conflicts to understand

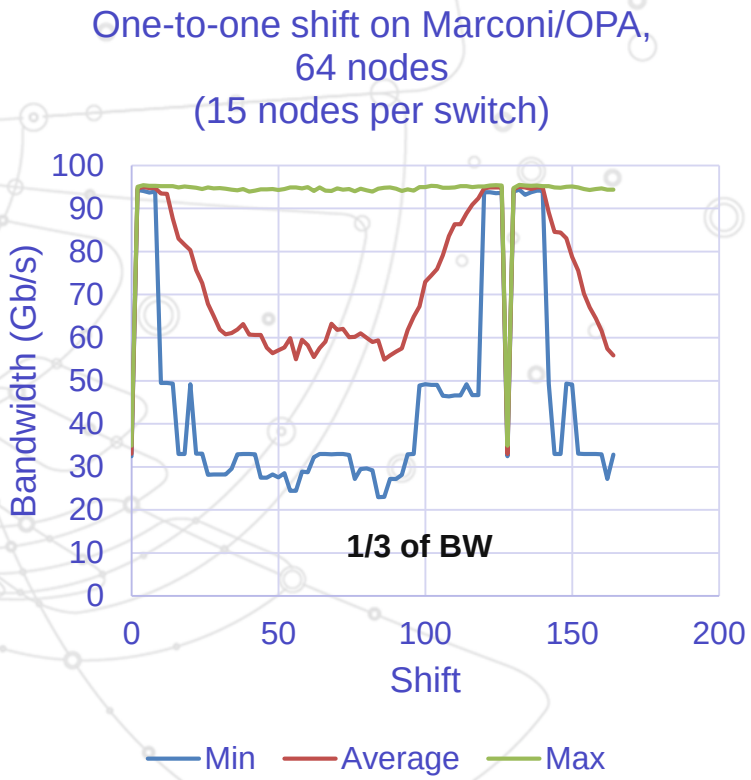


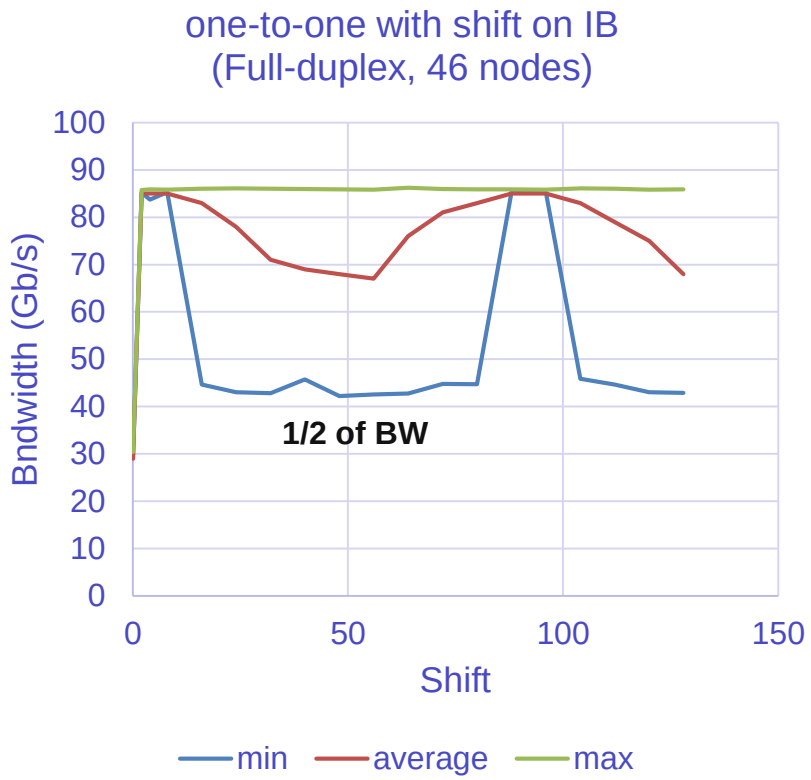


- Example, 2 conflicts of step 28
  - 20 -> 37
  - 29 -> 1
- They use the same UP link

Access again to Maronci

From 07/2017 (Tech A) & 02/2017 (Tech B)



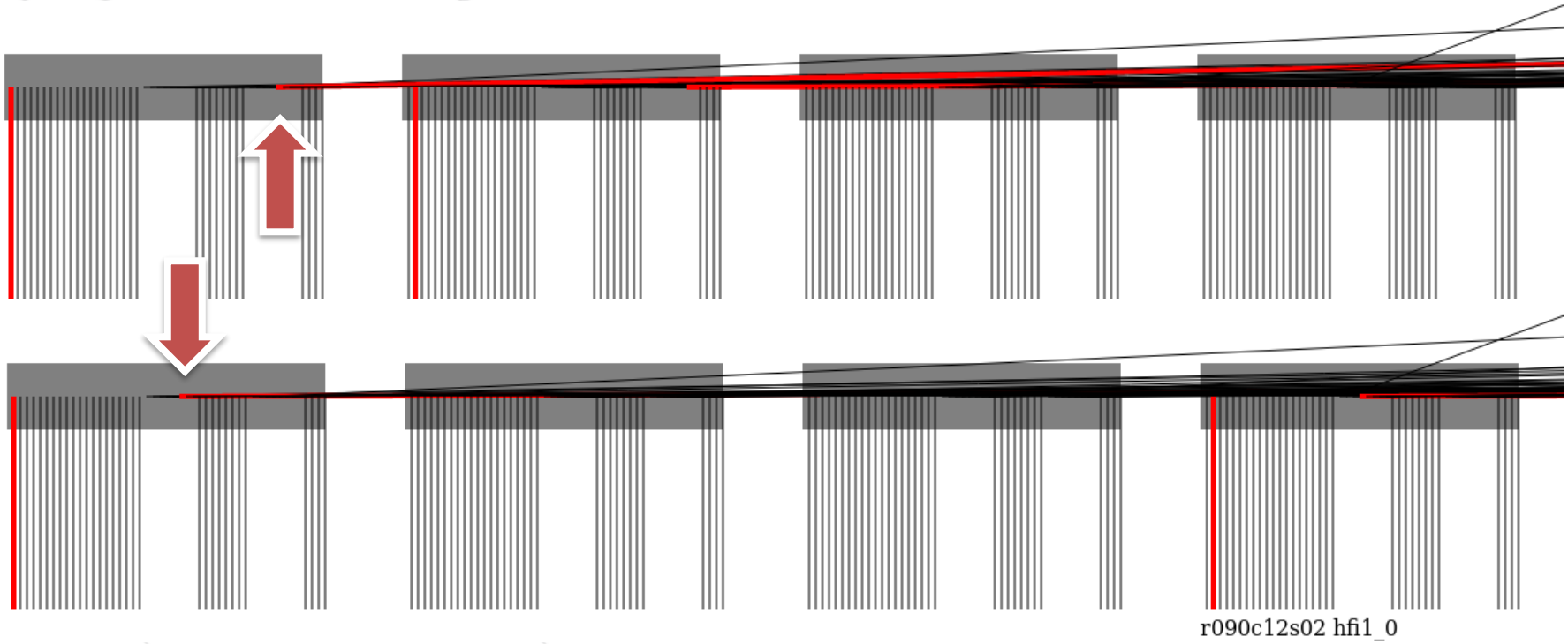


Looking on routing tables

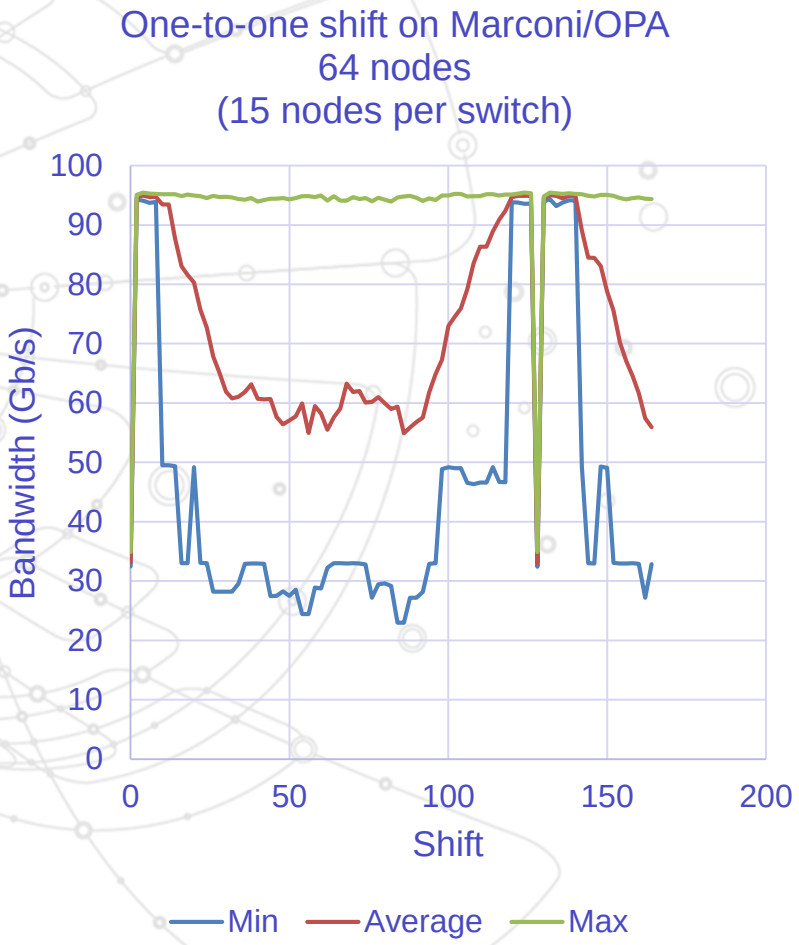
- **This time I "well" selected the nodes based on the real topology**
- **BUT, looking on routing tables :**
  - There is no regular pattern for up-links
  - Each leaf switch has a different ordering on the uplinks

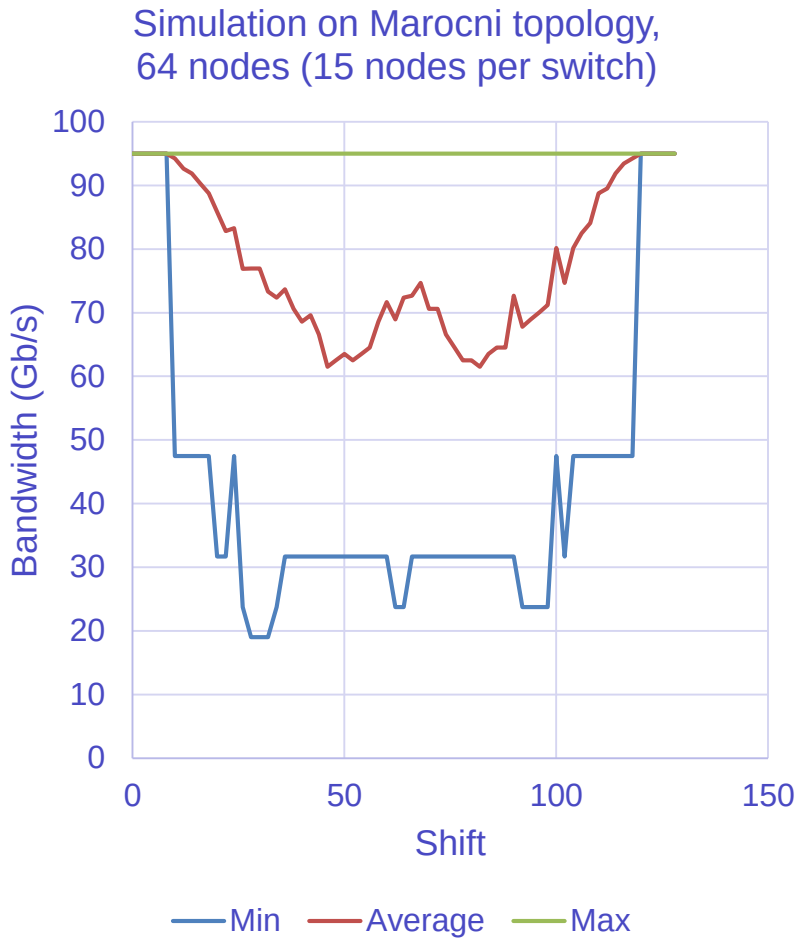
Access second node of target

…. Not same link in use depending on leaf switches

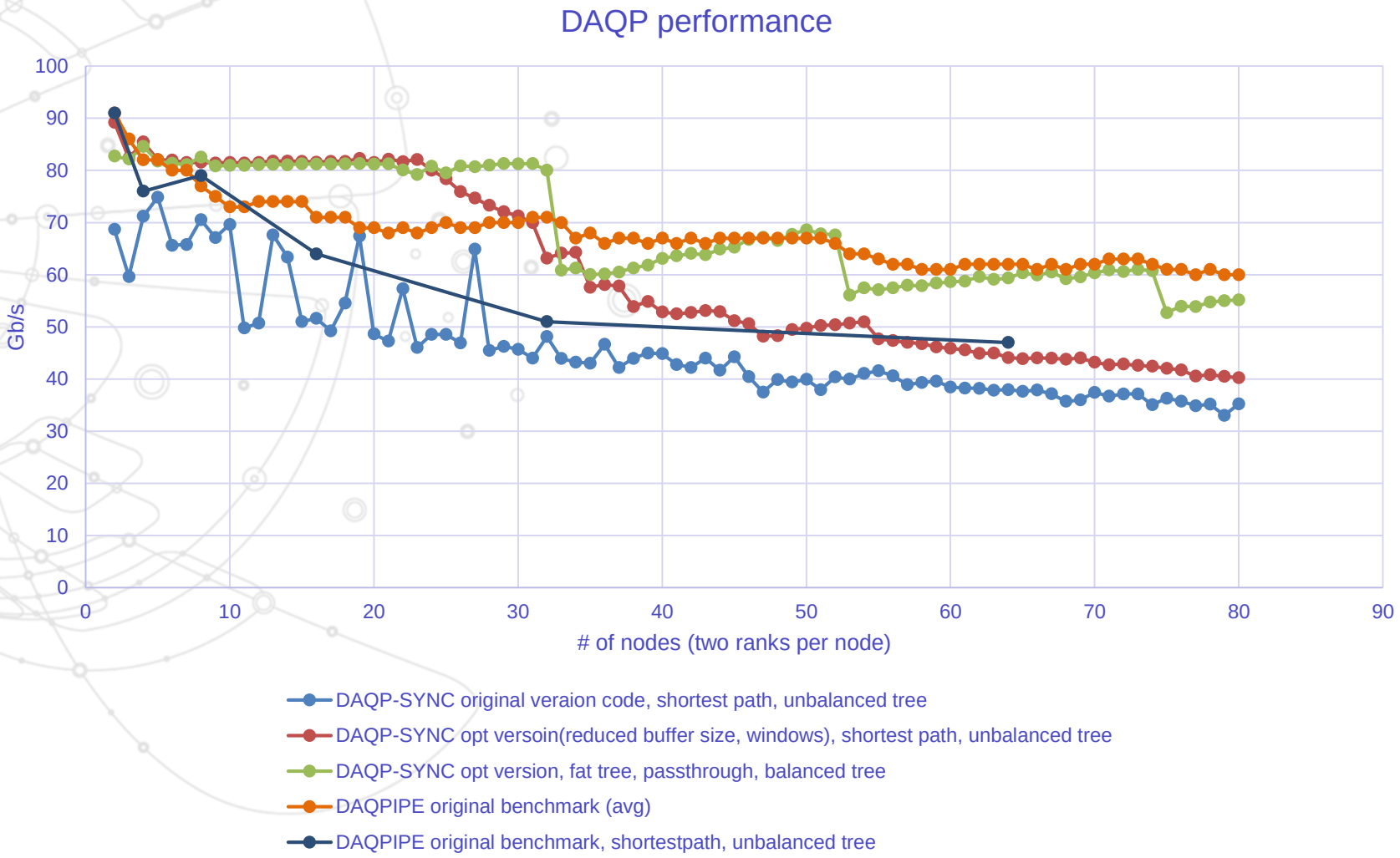


Simulation matching

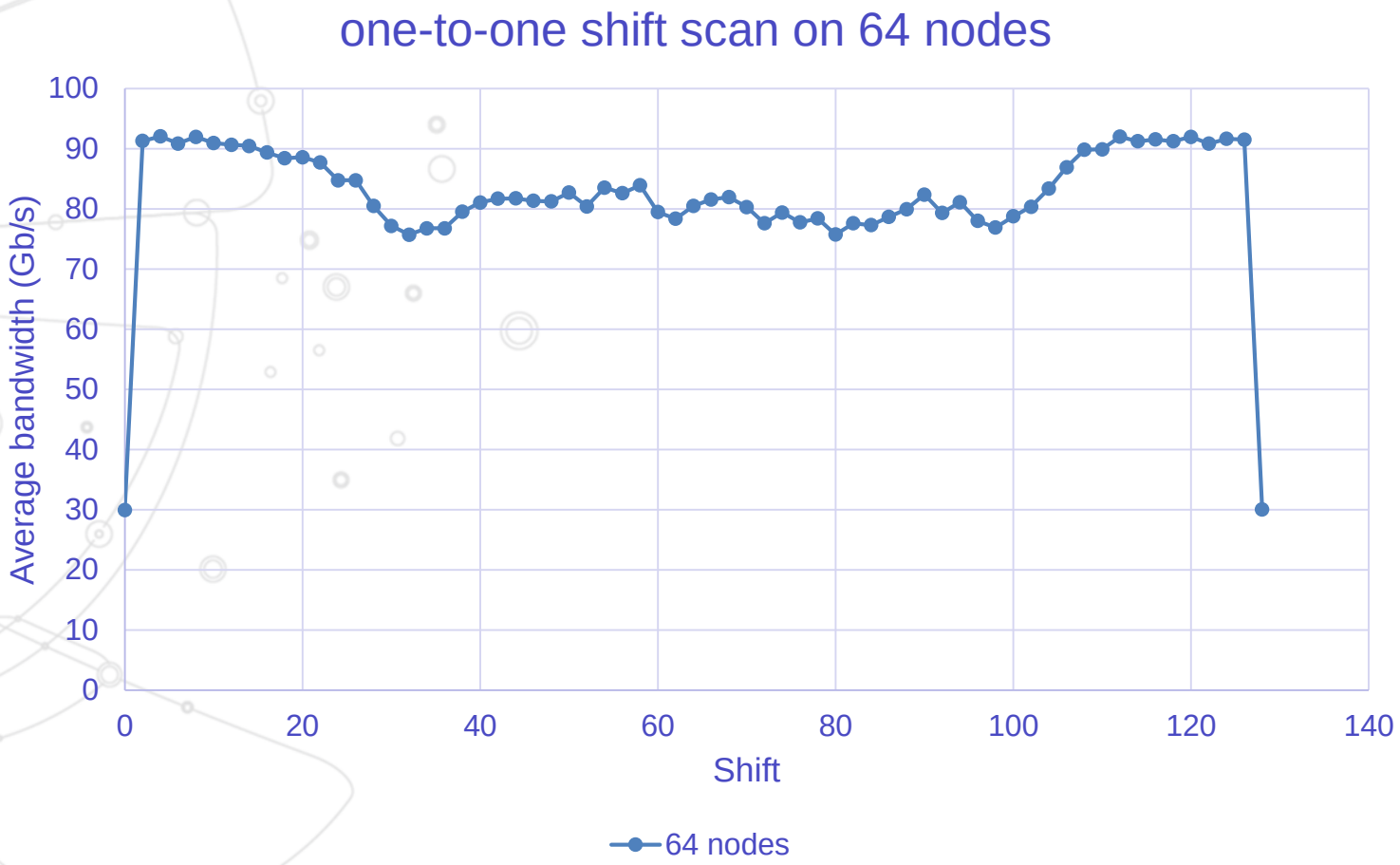




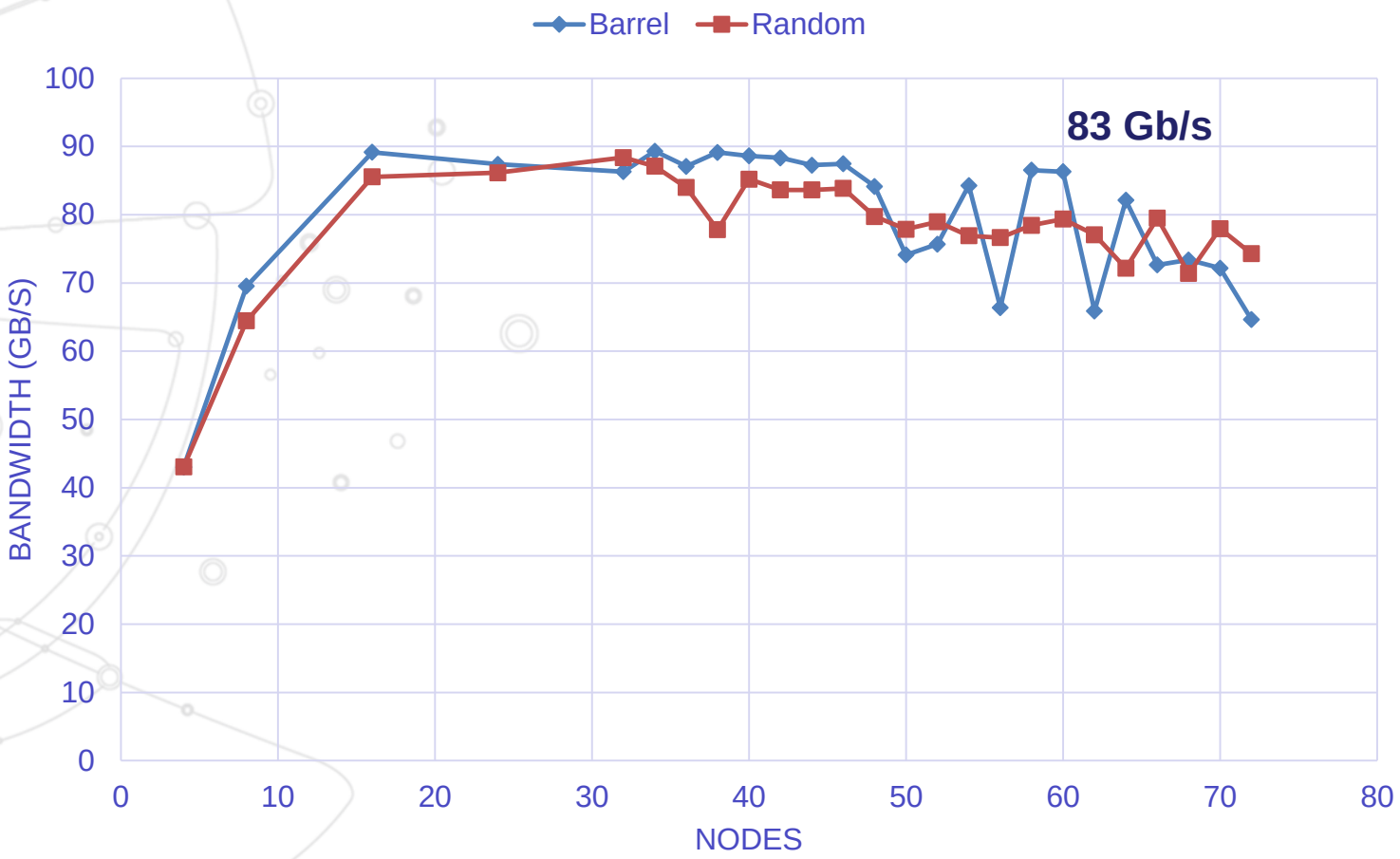
Performance improvement with optimized routing on OPA



One-to-one scan on Lenovo
This time with close to perfect cabling



Last results on Tech B (LENOVO)
This time with close to perfect cabling



Failure recovery

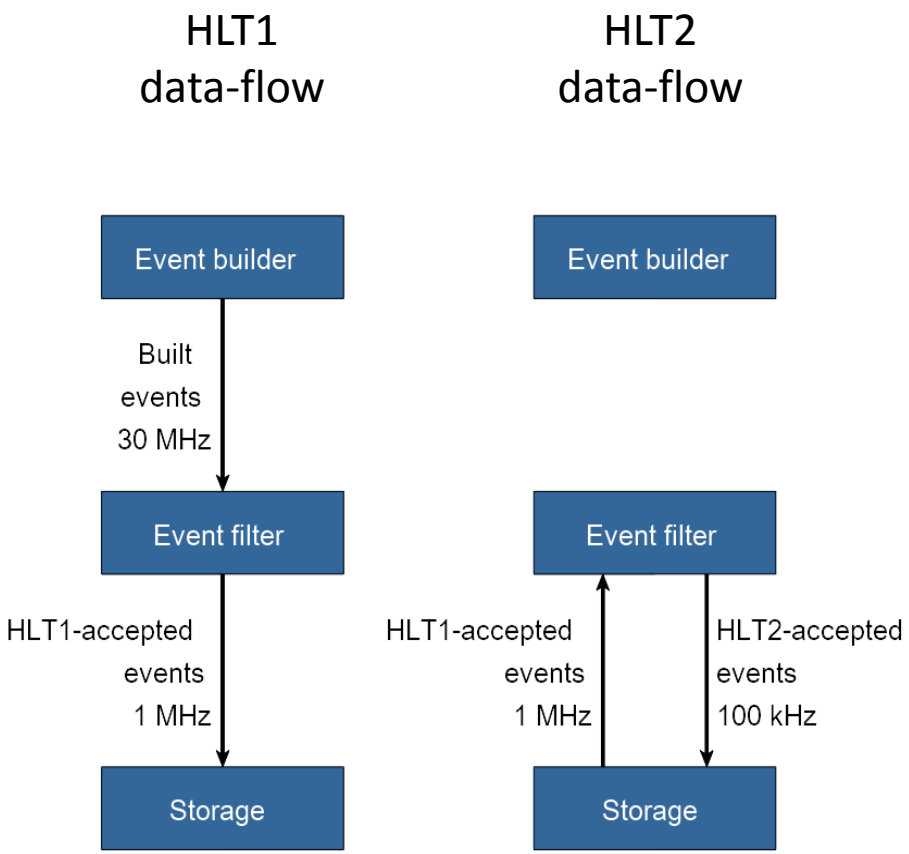
A word on failure-recovery

- **Support done for IB-verbs**
  - Was “easy”
  - Disconnection detection (delay of ~1s)
  - Reconnection
- **Tried for Omni-Path PSM2**
  - Encountered issues
  - Sometime call exit() when remote node stop
  - This propagate errors to other nodes
  - Also had to twik to reset the connection state

Storage

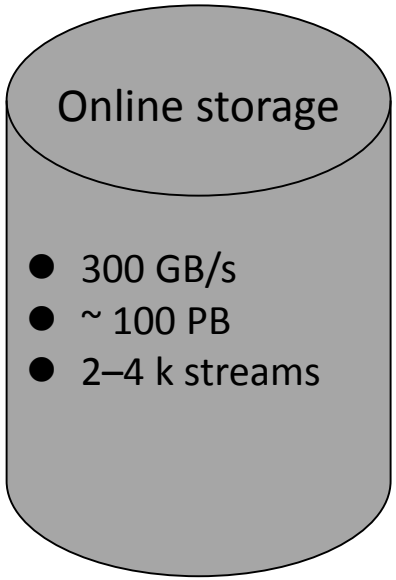
Basic strategy, same as run 2

- First filter: **HLT1**
  - **Fast reconstruction** and selection
  - **Synchronous** with DAQ at 30 MHz
  - Output: **~1 MHz**
  - Output : **~1 Tb/s => 128 GB/s**
- **Disk buffer** for HLT1-accepted events
- Second filter: **HLT2**
  - **Full reconstruction** and selection
  - **Asynchronous** (events from disk)
  - Output: **~100 kHz => ~12 Gb/s**



From Tommaso Colombo

- The disk buffer allows exploiting LHC downtime
  - Maximize event filter farm utilization
  - Need large buffer to absorb long LHC runs:
    ~100 PB for **a week's** worth of data
- Currently **investigating** both **centralized** and **distributed** solutions
- Requirements:
  - Must sustain a total of: **~150 GB/s input** + **~150 GB/s output**
  - **I/O pattern**:
    1 sequential read stream + 1 sequential write stream per filter node
  - No need for a file-system: an **object store is enough**
  - Minimal **redundancy**: some data **loss is acceptable**
  - Non-uniform data access costs is acceptable:
    filter nodes should process "local" data first
  - A global name-space is desirable for ease of operation and monitoring



Background image: Shutterstock

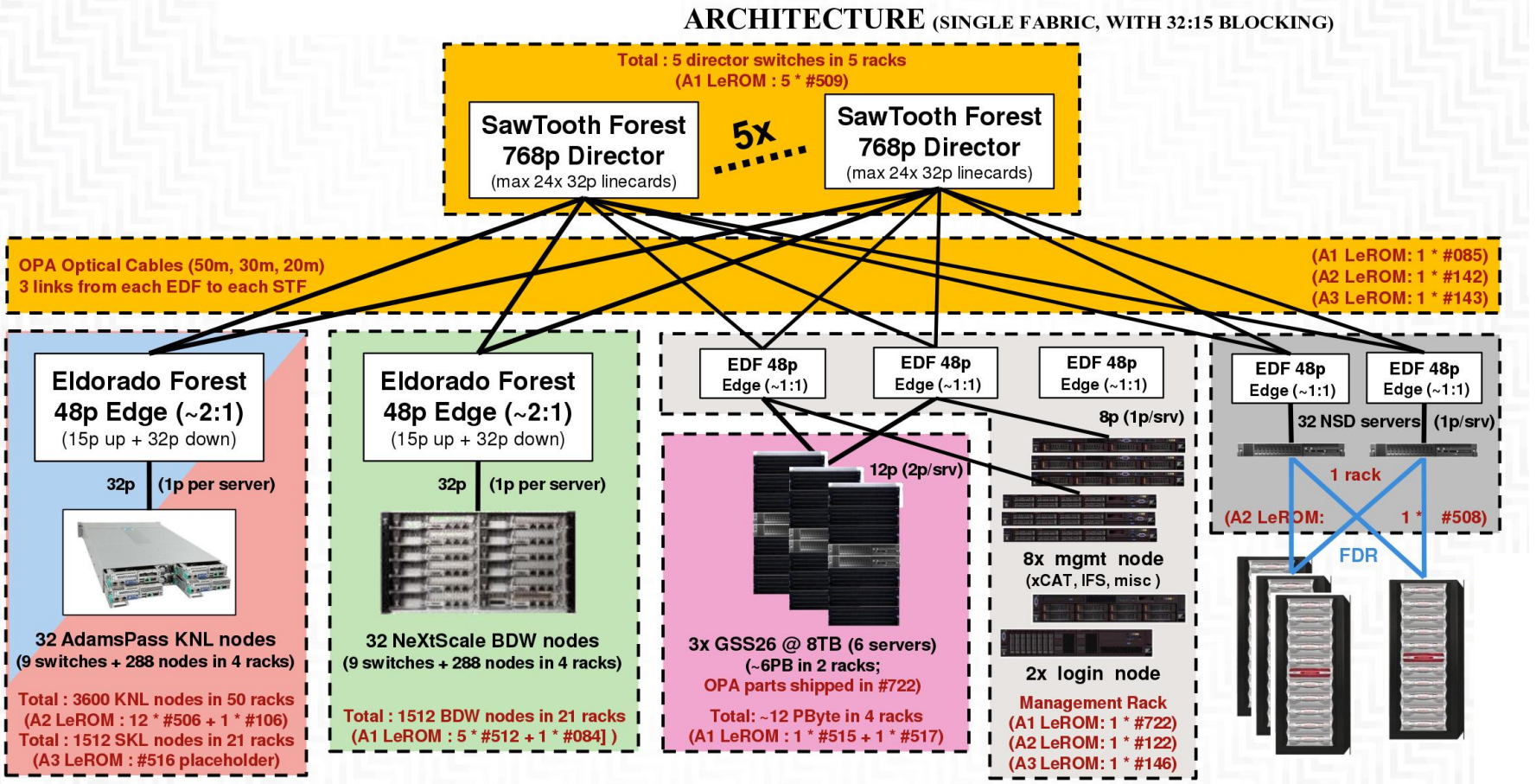
Conclusion

Conclusion

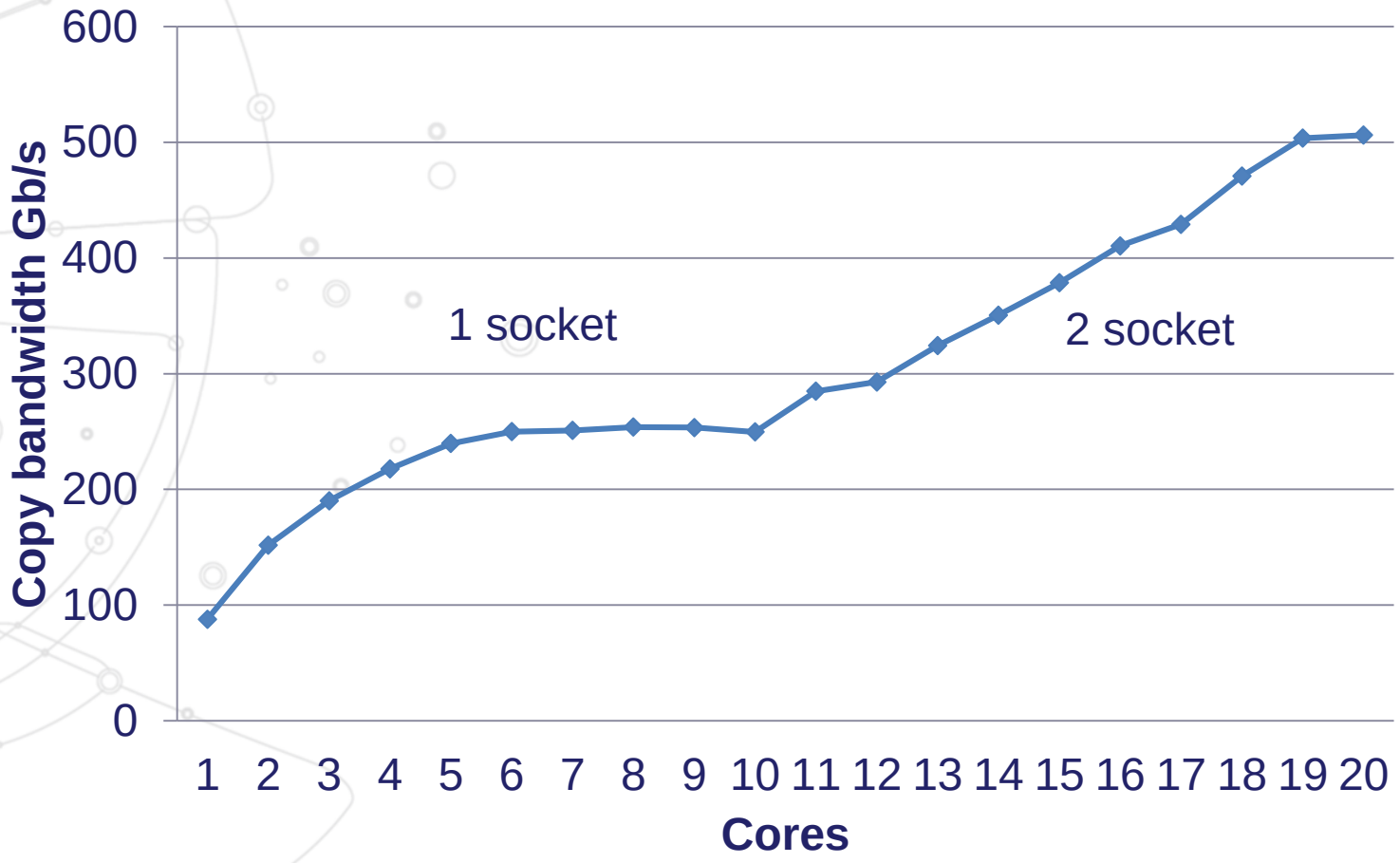
- **We are really dependent on the cabling & routing**
  - An issue when testing on existing supercomputer we do not manage
  - Now better understanding the issue
- **We are discussing with Intel for OPA issues**
  - Already get improvement validated up to **64 nodes**
  - Thanks to some driver patches
- **InfiniBand results are very good**
  - Seen **83 Gb/s** on **64 nodes** with non ideal cabling.
  - Here we have **a chance** to get the bandwidth at **512 nodes**.
  - In last resort there will be **HDR** (200 Gb/s)
- **Future work**
  - Still have to **prove 80 Gb/s at 512** nodes

Backup

Network topology

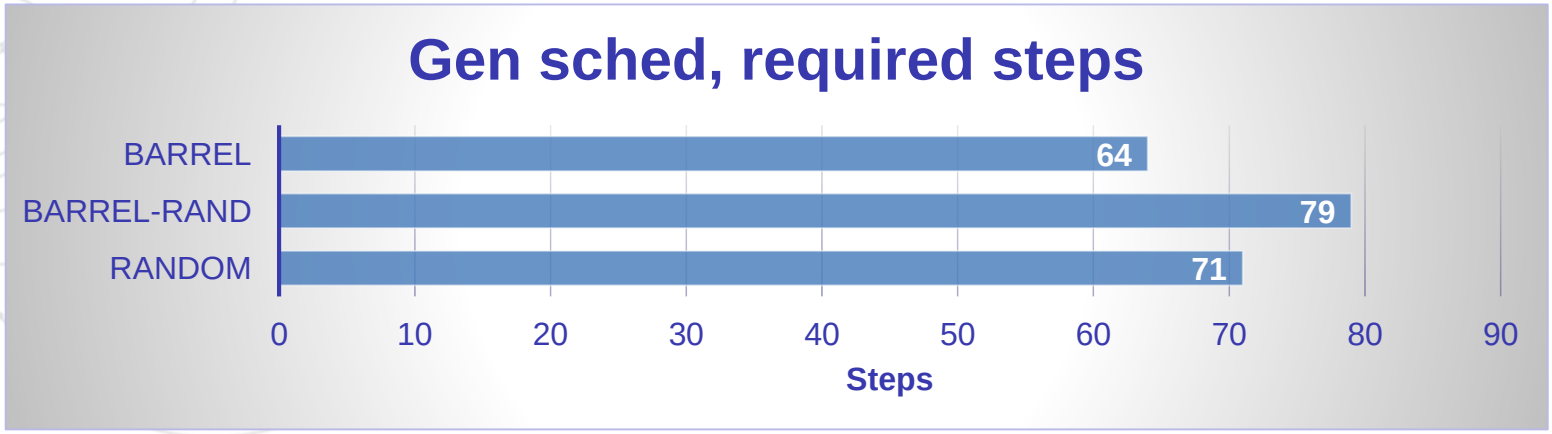


Memory bandwdith margins on 2 Xeon E5-2630 v4

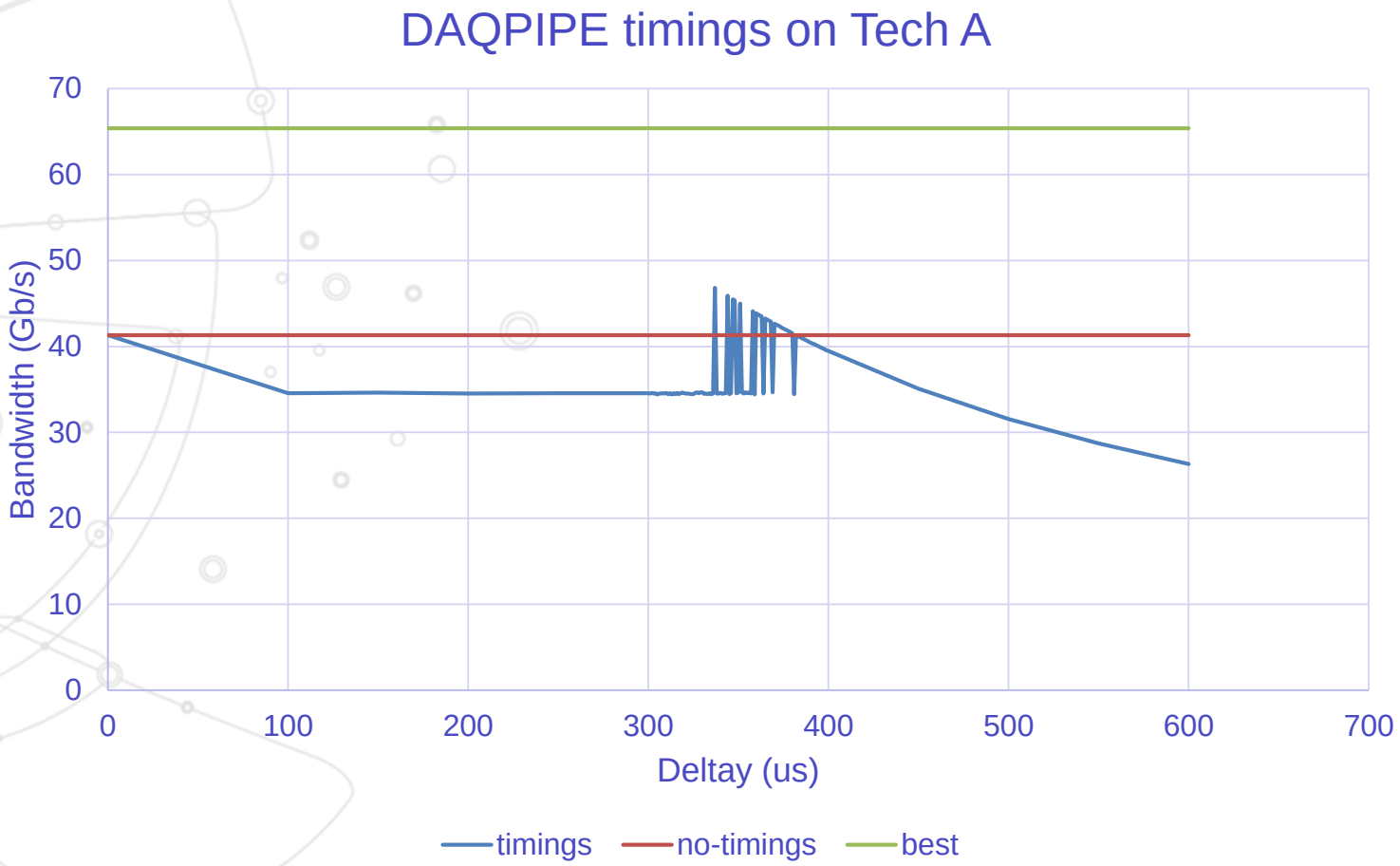


Try to generate more steps

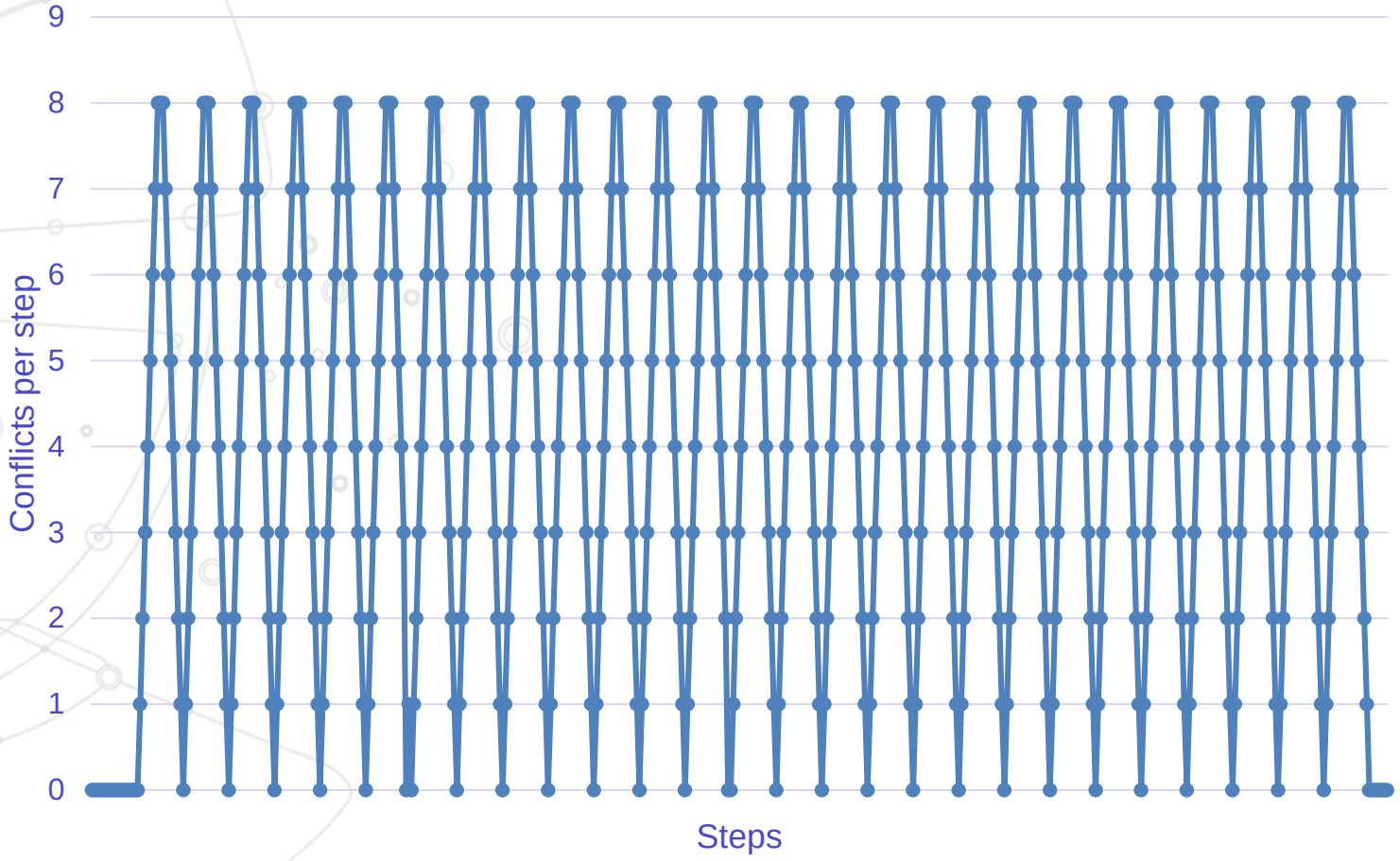
- **Random**
  - We randomly selected a non conflicting step (+10%)
- **Barrel rand :**
  - Move conflicts to another step (+23%)
- **Test for 64 nodes**



Example with timings

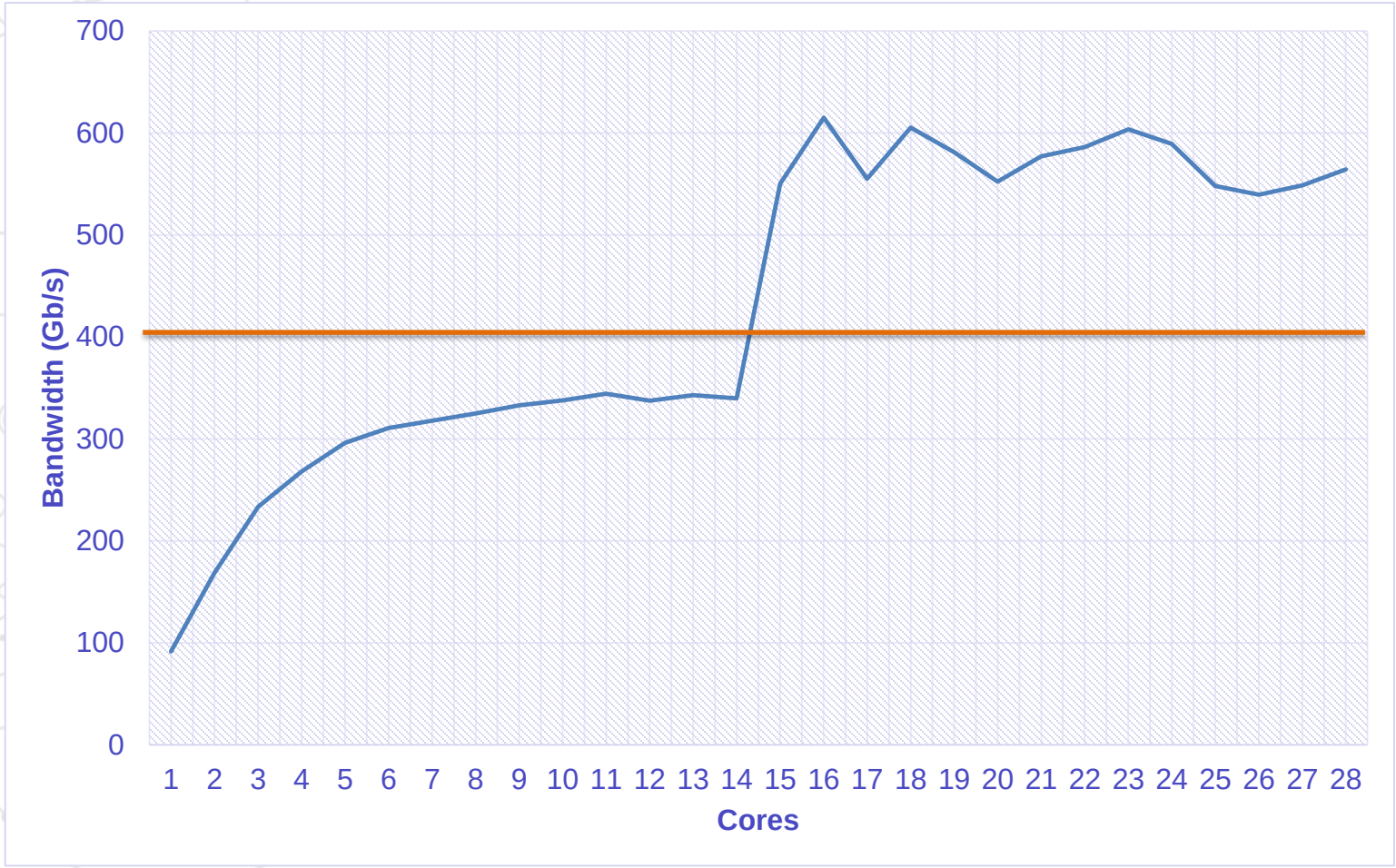


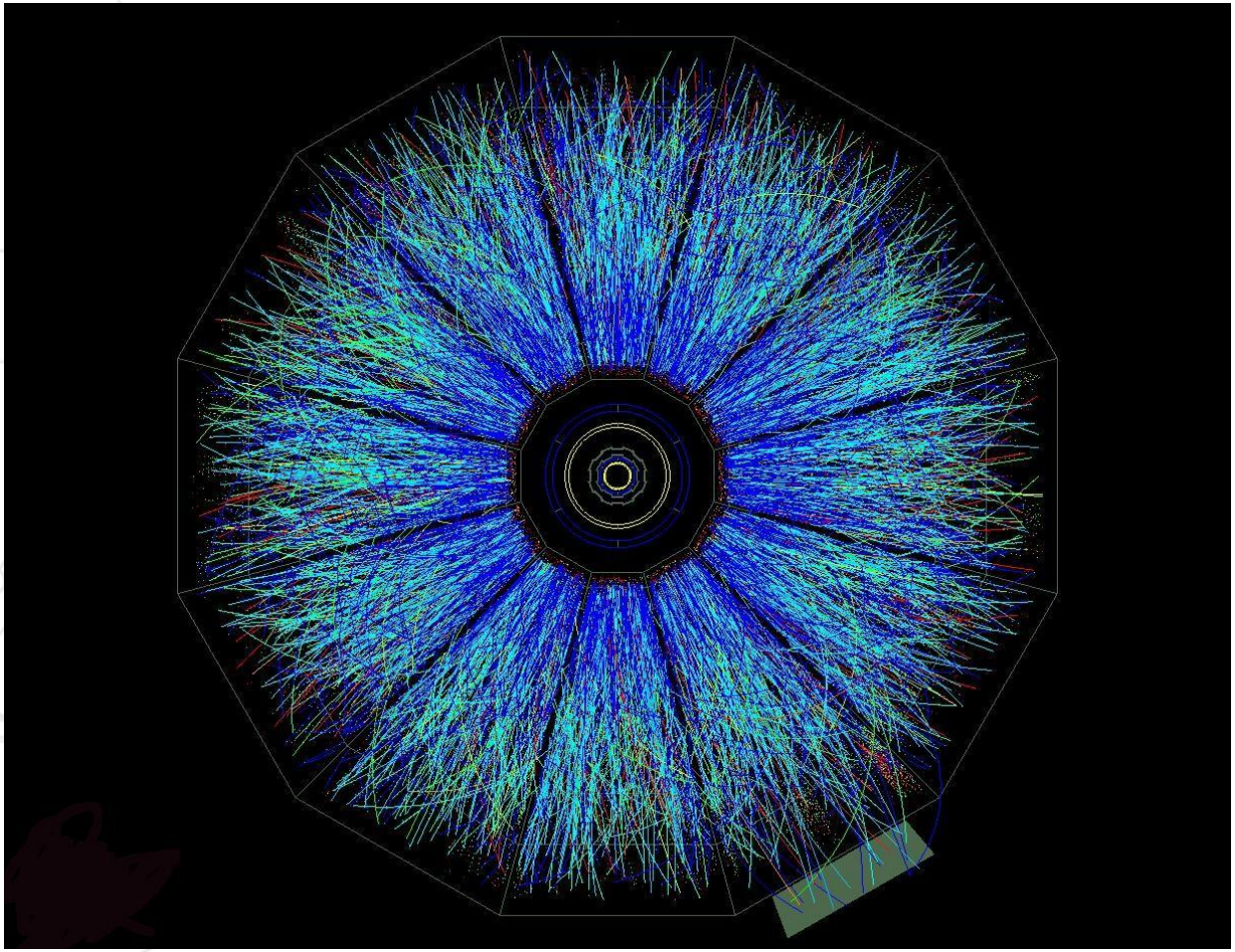
Scaling ideal topology to 512 nodes
2157 conflicts (0,8%)



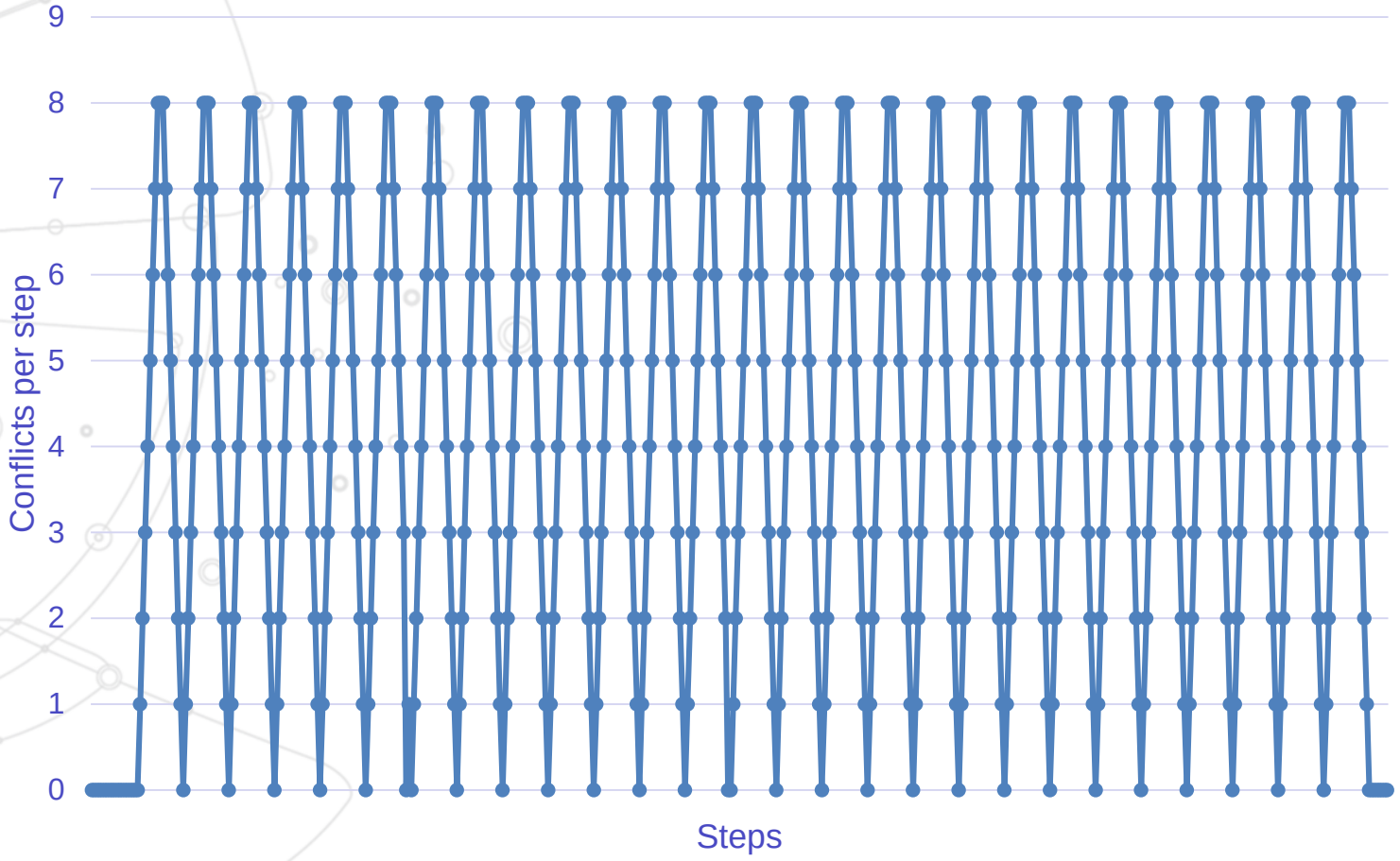
Htopml integration
Communication scheduling

Memory bandwidth (stream)





Scaling ideal topology to 512 nodes
2157 conflicts (0,8%)



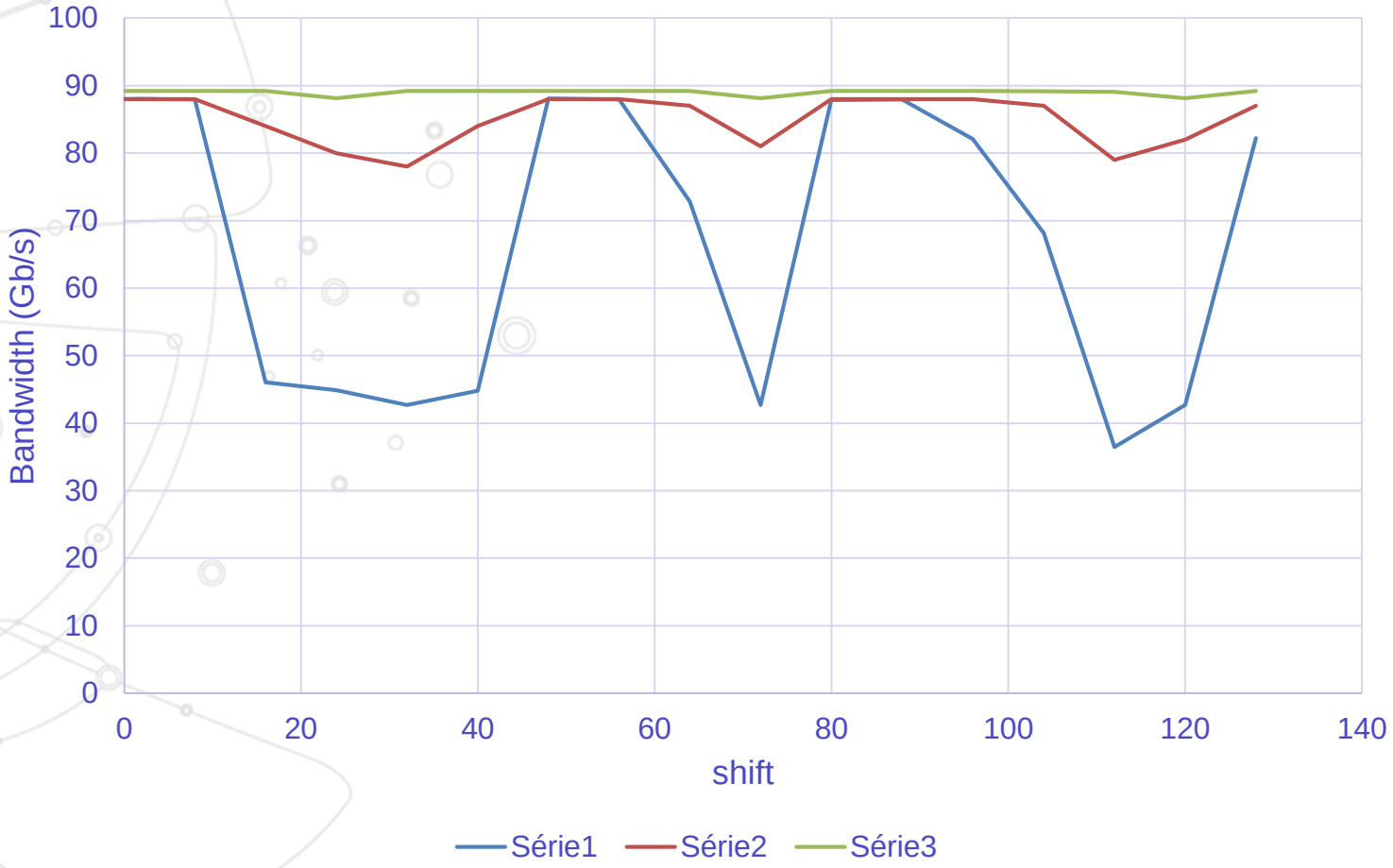
Fully populating the leaf switches

- **Barrel shifting shows no-conflict if:**
  - Ideal topology
  - Good routing table
  - Fully populating the leaf switches
- **It implies with 36 port switches to run with**
  - **72** nodes, not 64 (+12%)
  - **540** nodes, not 512 (+5%)
- **With 48 port switches :**
  - **96** nodes, not 64 (+50%)
  - **528** nodes, not 512 (+3%)

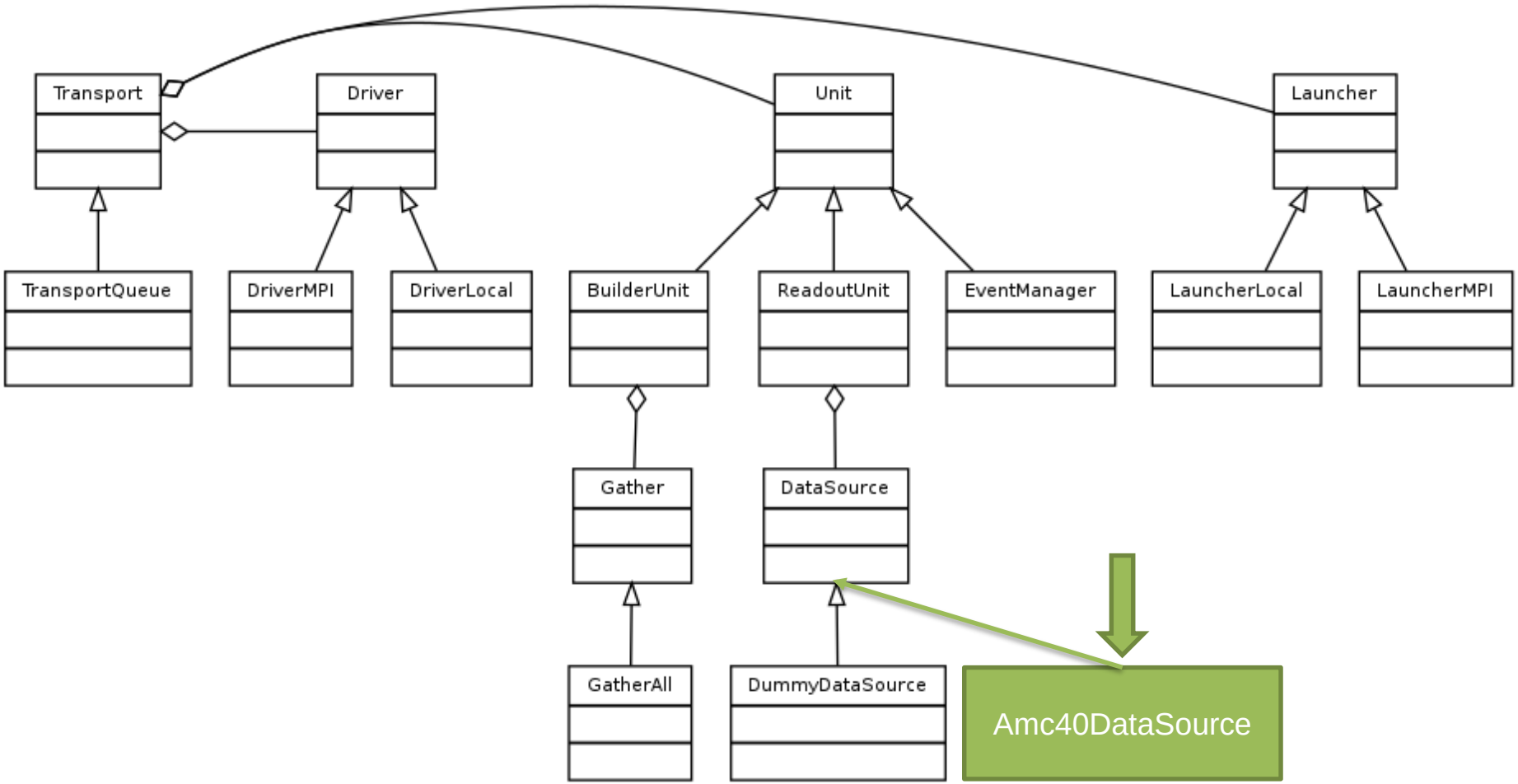
Slurm numbering

- Slurm can use a topology to improve
- It provides the list of nodes on same switch
- But still, inside a switch it uses the node name to order ranks

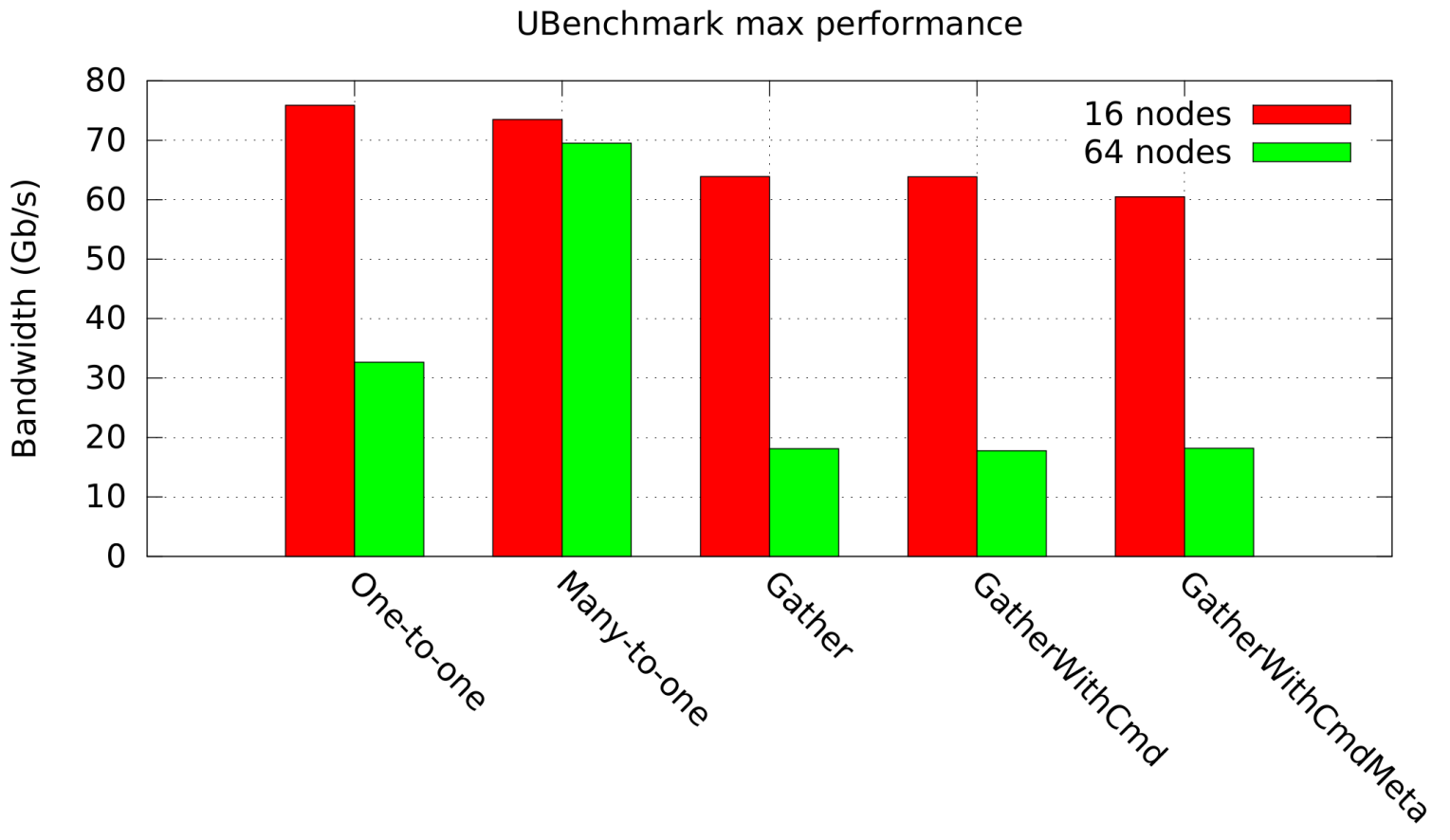
IB Single-way, 46 nodes



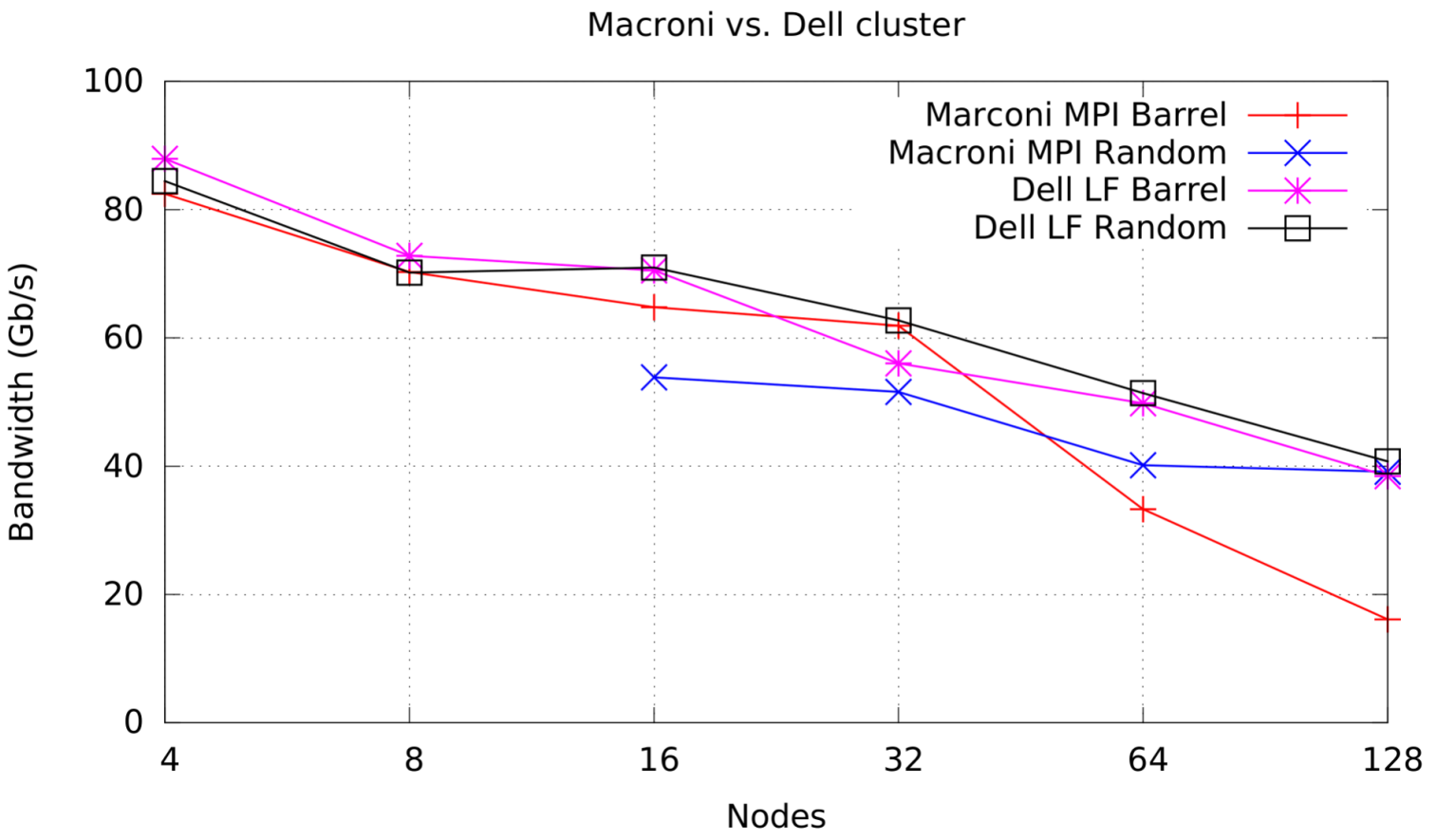
Support of AMC40



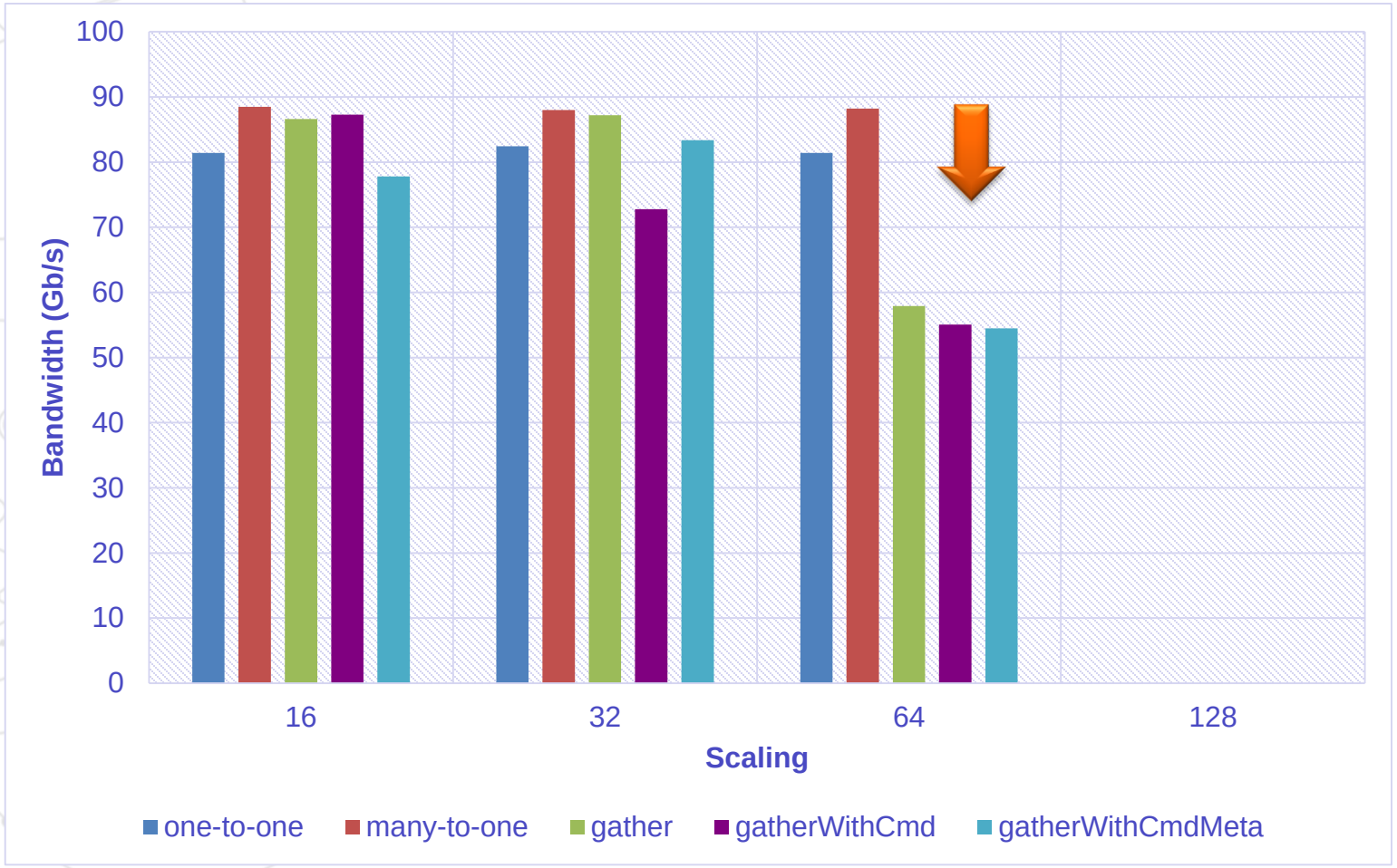
During the night micro-benchmarks



Compared to DELL cluster Full duplex



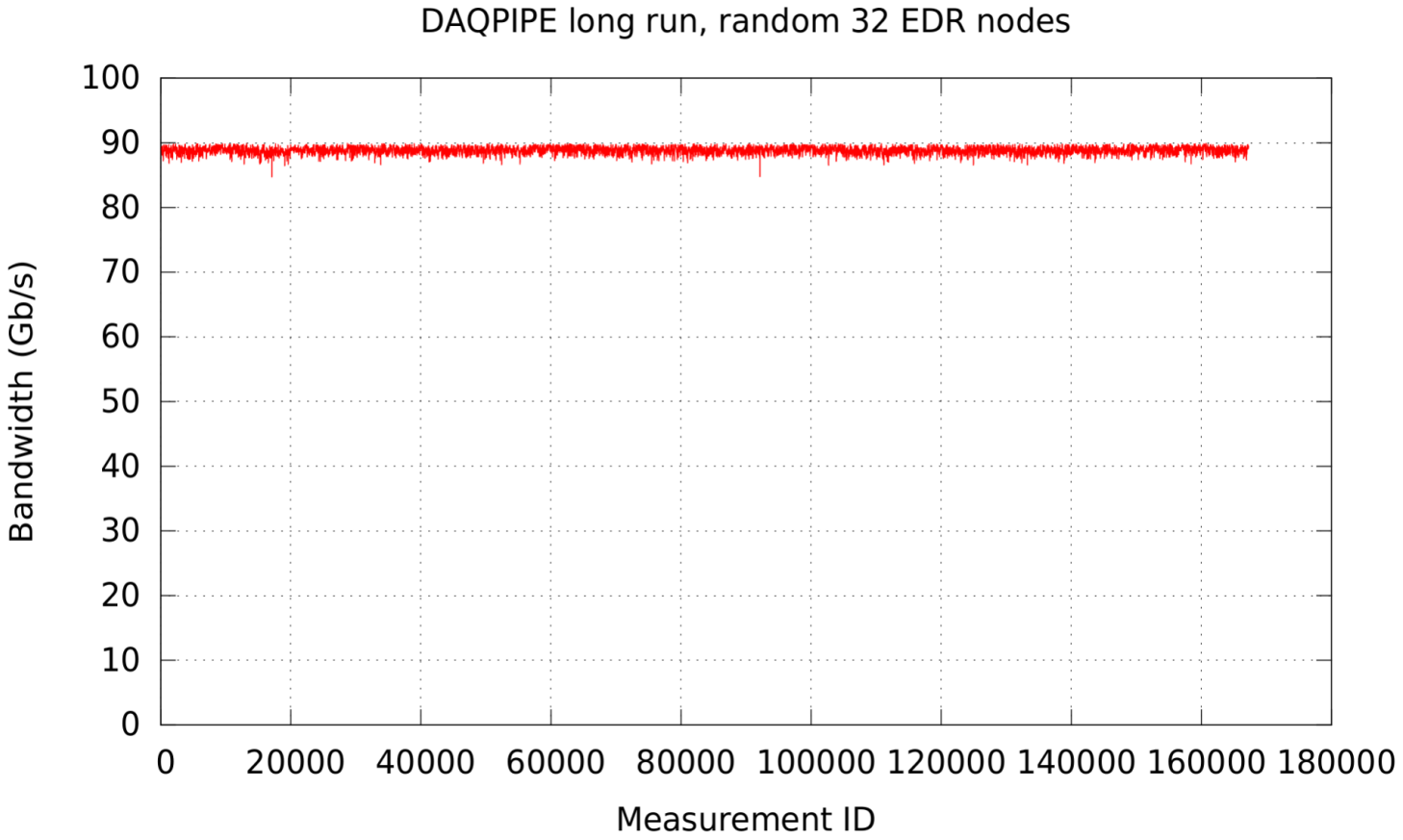
Micro-benchmarks on IB



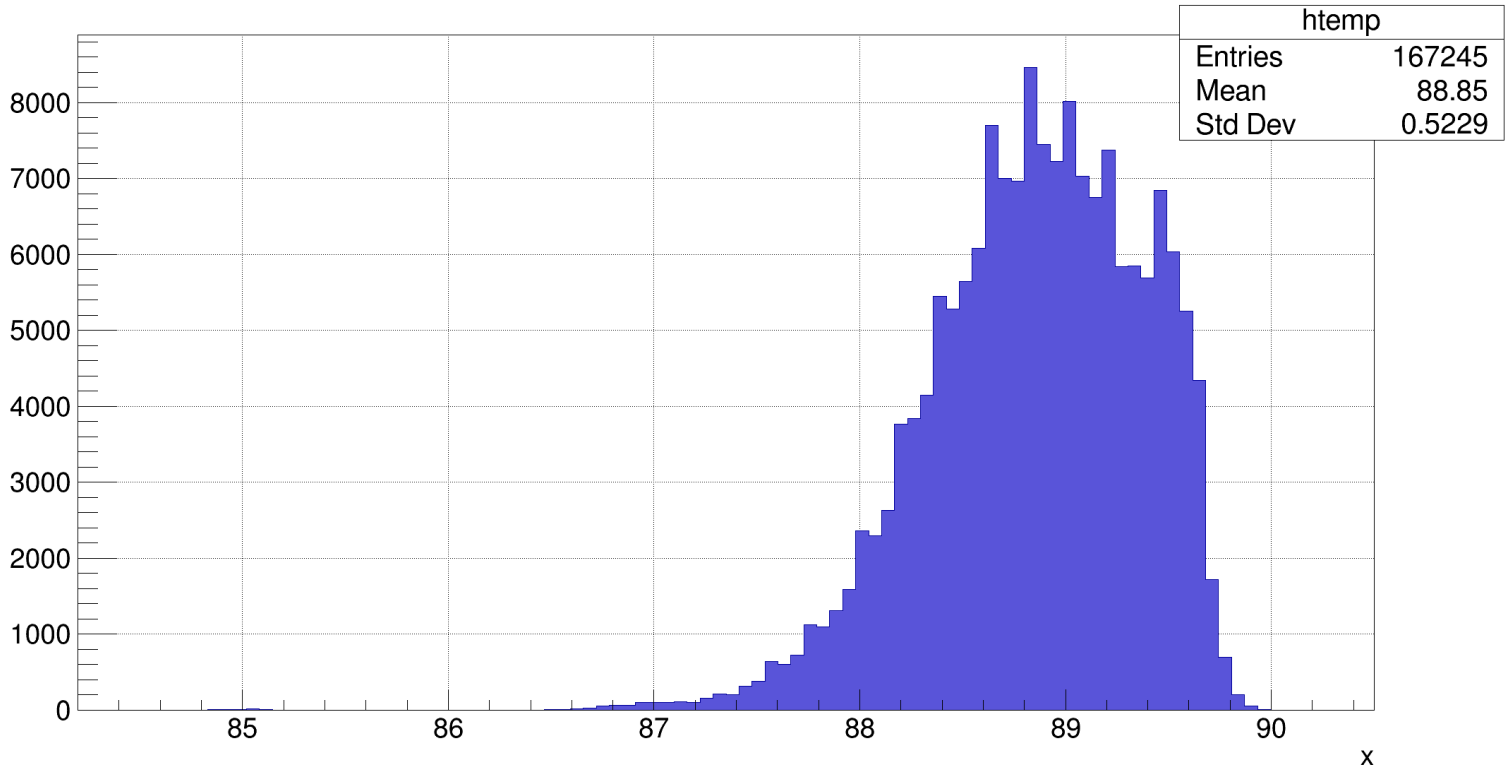
Why triggering

- **We cannot stored all the collisions !**
  - Far too much data !
- **Most collisions produce already well known physics**
- **We keep only interesting events for new physics**
- **Challenge for upgrade: need to trigger in software only**
  - Need to improve current software performance
- **For some costly functions**
  - Looks on **GPU**
  - Looks on possible **CPU embedded FPGA** for some costly functions

Stability over time (3 hours)



Stability over time (3 hours)



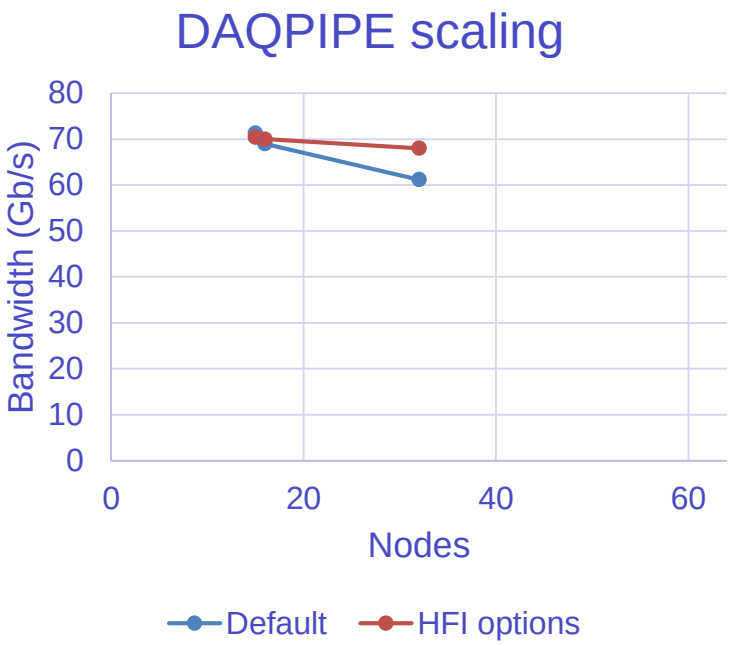
Last access to Marconi

From 07/2017

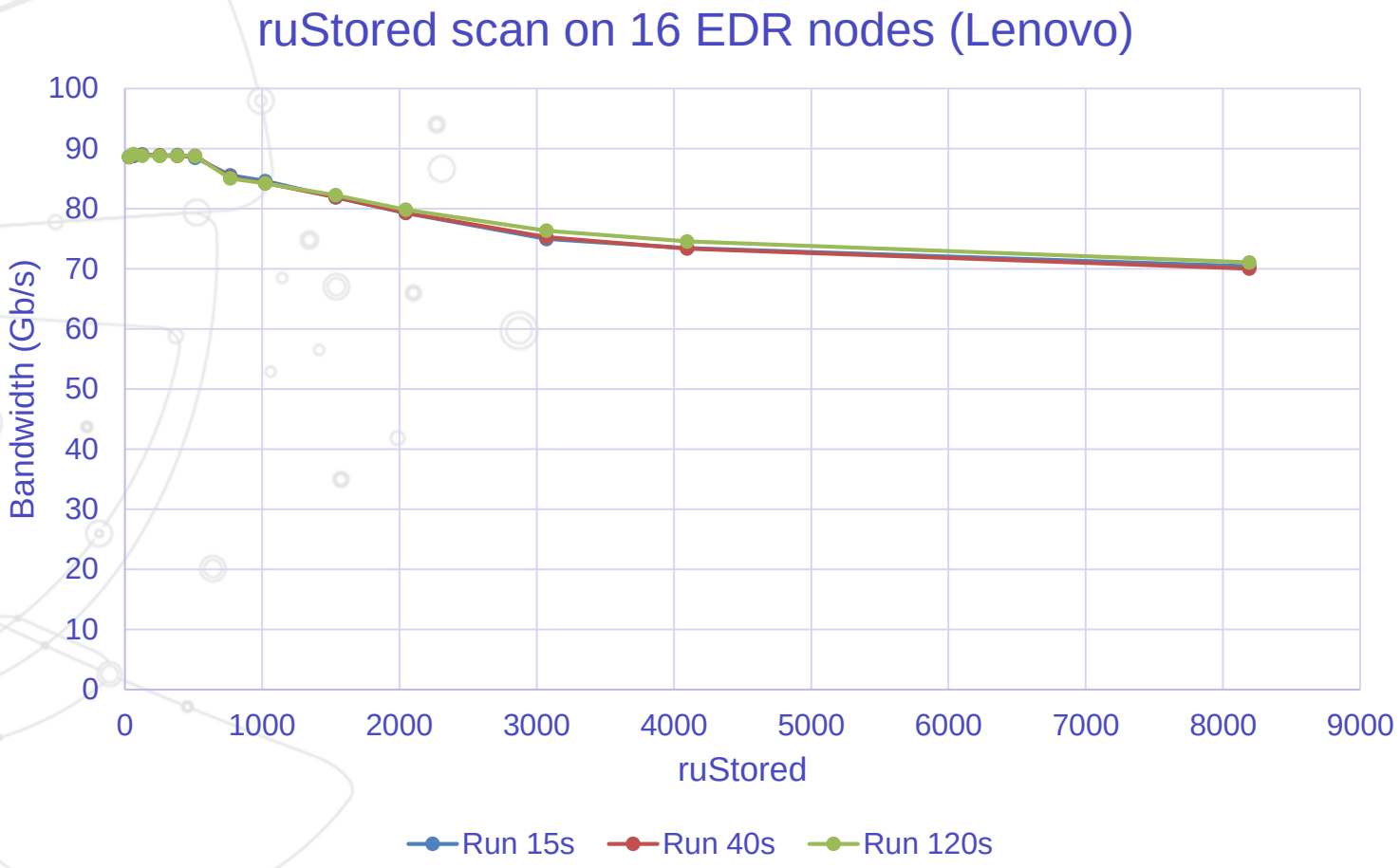
- **Tried PSM options proposed by Intel**

```
export PSM2_RTS_CTS_INTERLEAVE=1
export PSM2_MQ_RNDV_HFI_WINDOW=1048576
export TMI_CONFIG=$I_MPI_ROOT/etc64/tmi.conf
```

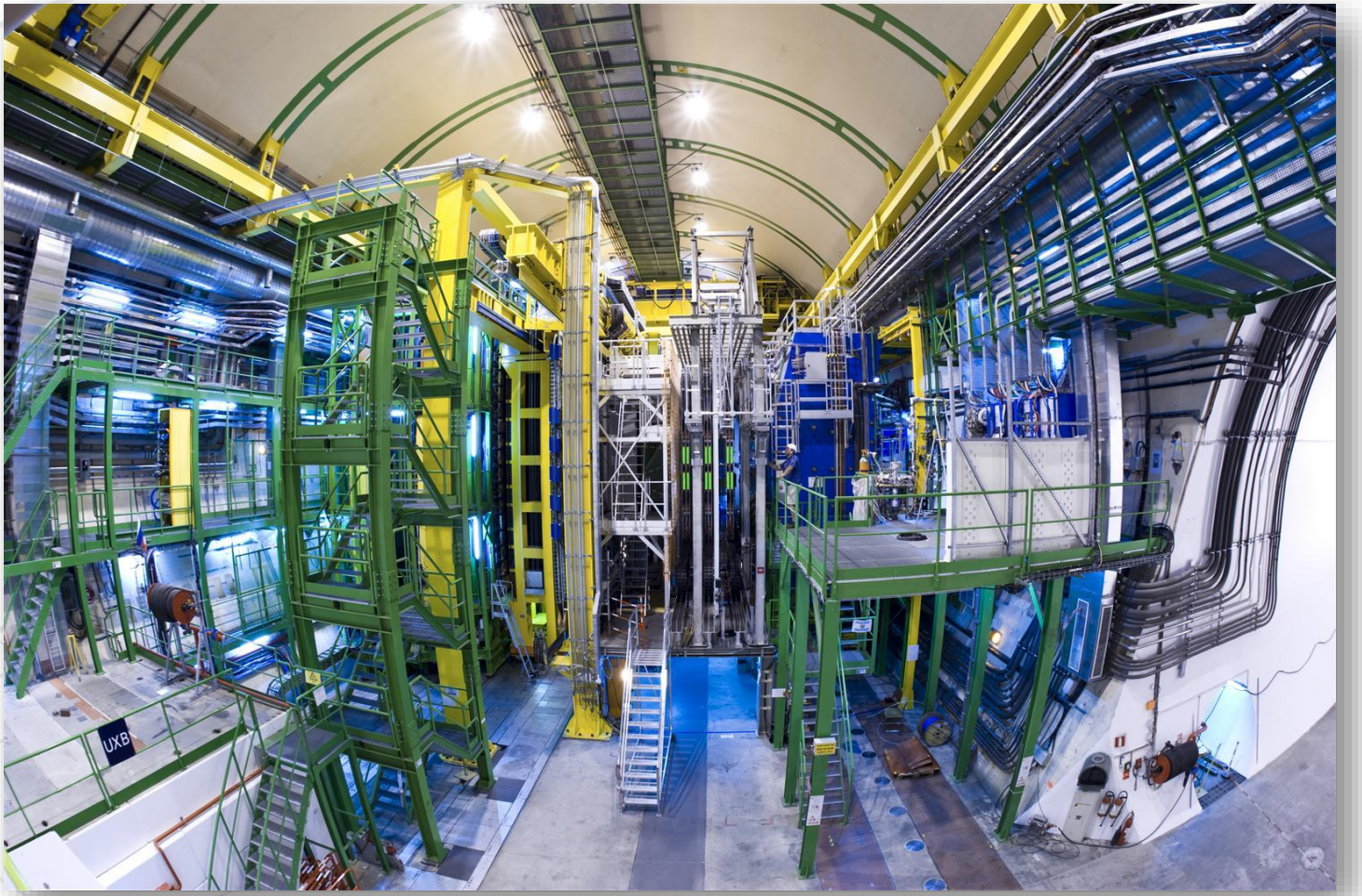
- **It shoed some improvements**
- **But we are still at 70 Gb/s, not 80…**



Buffering size issue

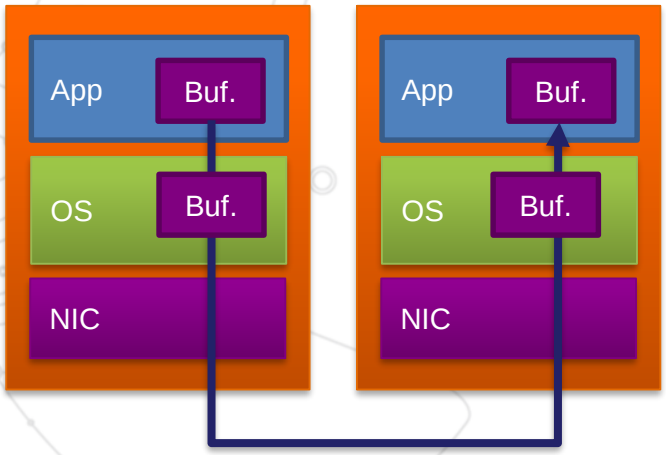


LHCb

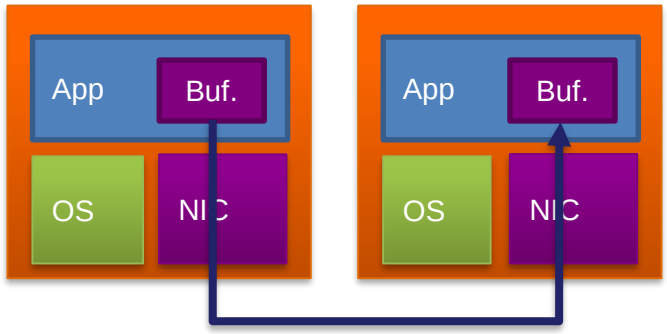


RDMA
Remote Direct Memory Access

- **Communication with Ethernet**
  - Traverse all **IP stack** in **OS**
  - Internal **memory copies** (use mem. **bandwidth** & **CPU**)
  - Higher latency



- **OS by pass in InfiniBand or Omni-Path**
  - **RDMA** : Remote Direct Memory Access
  - No memory copies
  - **Lower latency**
  - **CPU not involved**



DAQPIPE

- **A benchmark to evaluate event building solutions**
  - DAQ Protocol Independent Performance Evaluator
  - Provides EM/RU/BU units

- **Support various APIs**
  - **MPI**
  - Libfabric
  - PSM2
  - Verbs
  - TCP / UDP
  - RapidIO

- **Three message size on the network**
  - Command : **~64 B**
  - Meta-data : **~10 KB**
  - Data : **~1 MB**