Netbench

Testing network devices with *real-life traffic patterns*

HEPiX Fall/Autumn 2017 Workshop

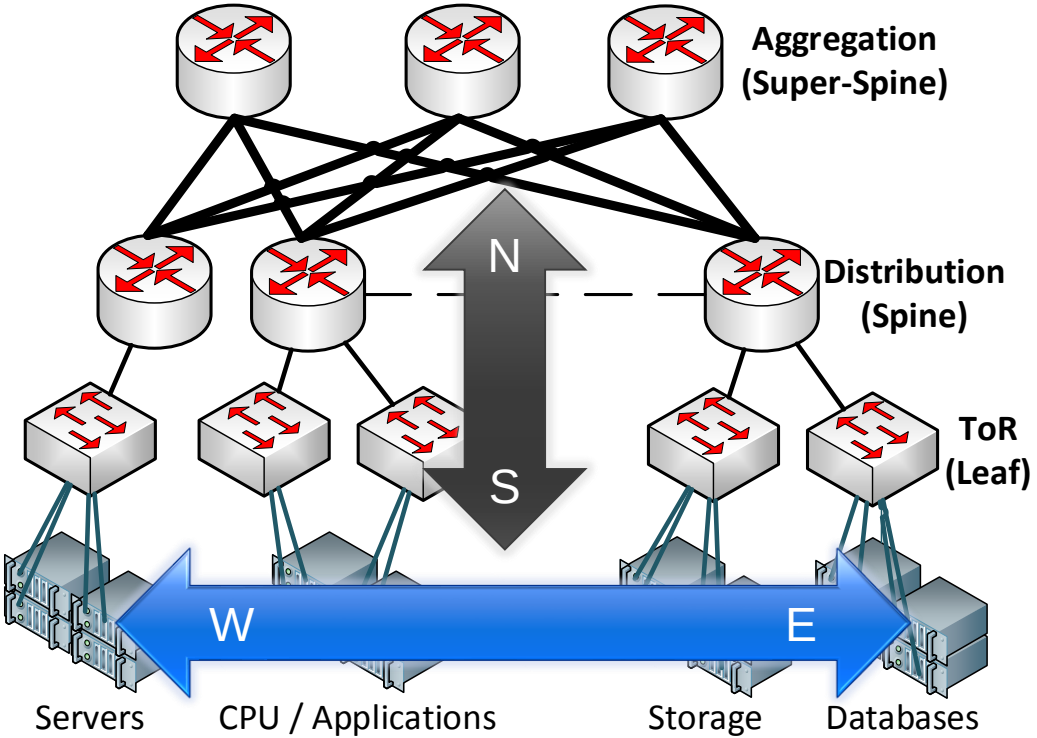
stefan.stancu@cern.ch

Outline

- The problem:
  evaluate network devices

- Test approaches
- Netbench
  - Design
  - Statistics visualization
  - Sample results

The problem



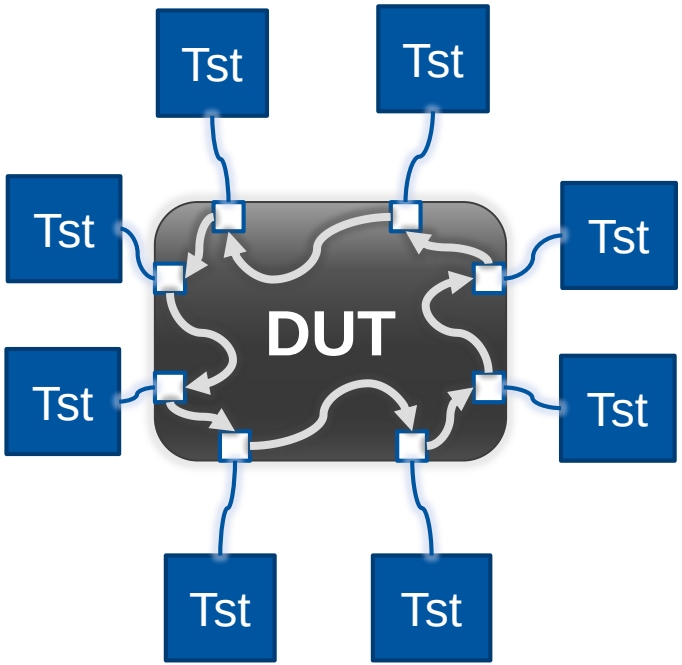
Network performance is key

- Datacentre
- Campus
- Etc.

Selecting new devices → **evaluation** is crucial:

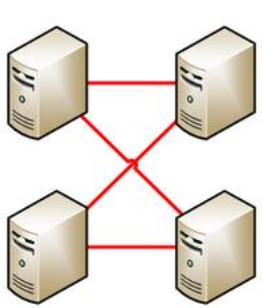
- Required features
- Required **performance**
  - Traffic patterns

Luxurious approach

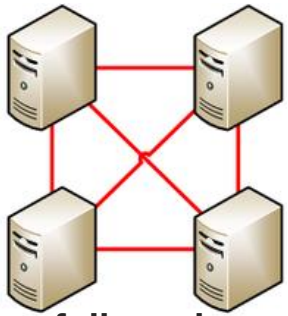


- Use one tester port for each device port
- ✘ Cost explosion (tester port $$$)
  - N tester ports (tens or few hundreds)
- ✓ Exercises all ports
  - Line-rate
  - Packet size scan
- ✓ Forwarding of traffic with complex distributions [3]
  - Partial mesh
  - Full mesh
- ✘ Buffering only li
  - RFC tests [1][
  - Synchronized
- ✘ Such tests are
  - Manufacturers
  - Third party tes

producibl
th *minima*
and are l
demand



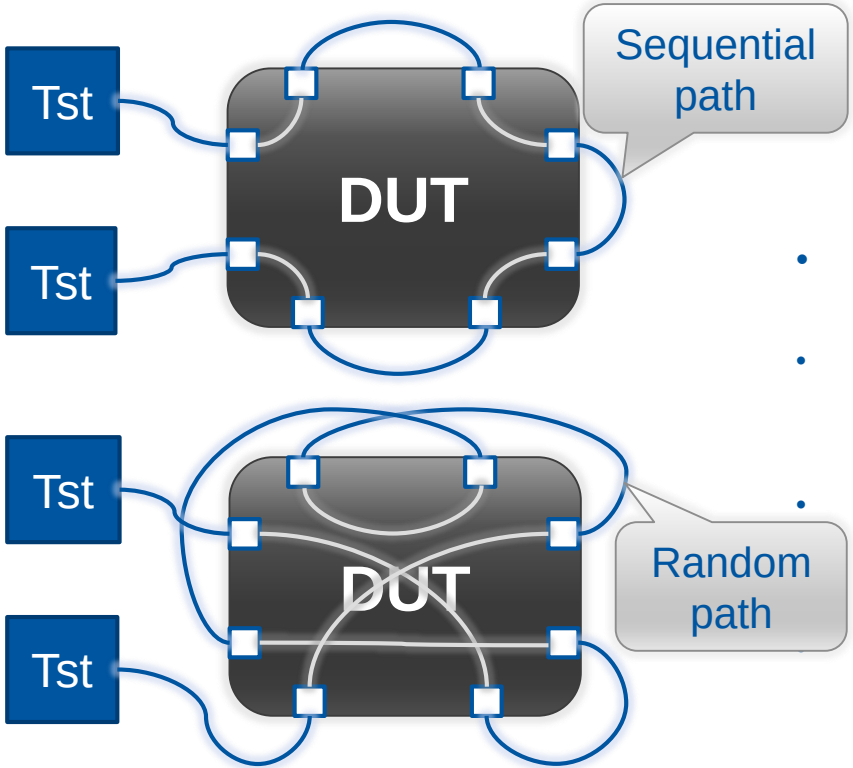
partial mesh



full mesh

Tst = Tester
DUT = Device Under Test

Typical approach (affordable) – snake test

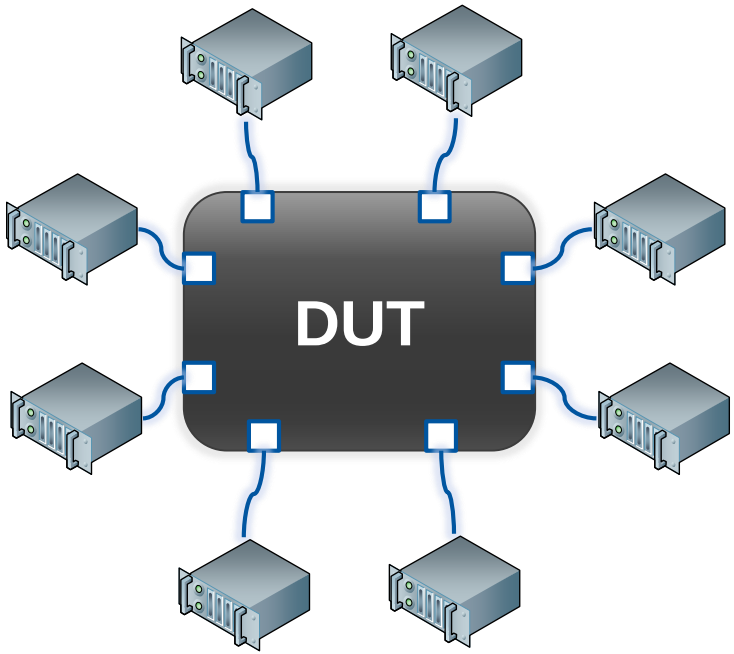


Snake test
- Use 2 tester ports
- Loop back traffic

- ✓ Contained cost (tester port $$$):
  - only 2 tester ports
- ✓ Exercises all ports
  - Line-rate
  - Packet size scan
- ✘ Forwarding on a simple linear paths
  - Sequential paths → Easy to predict & optimize
  - Random path → “Impossible” to predict
- ✘ Buffering is not exercised
  - No congestion due to linear path

Tst = Tester
DUT = Device Under Test

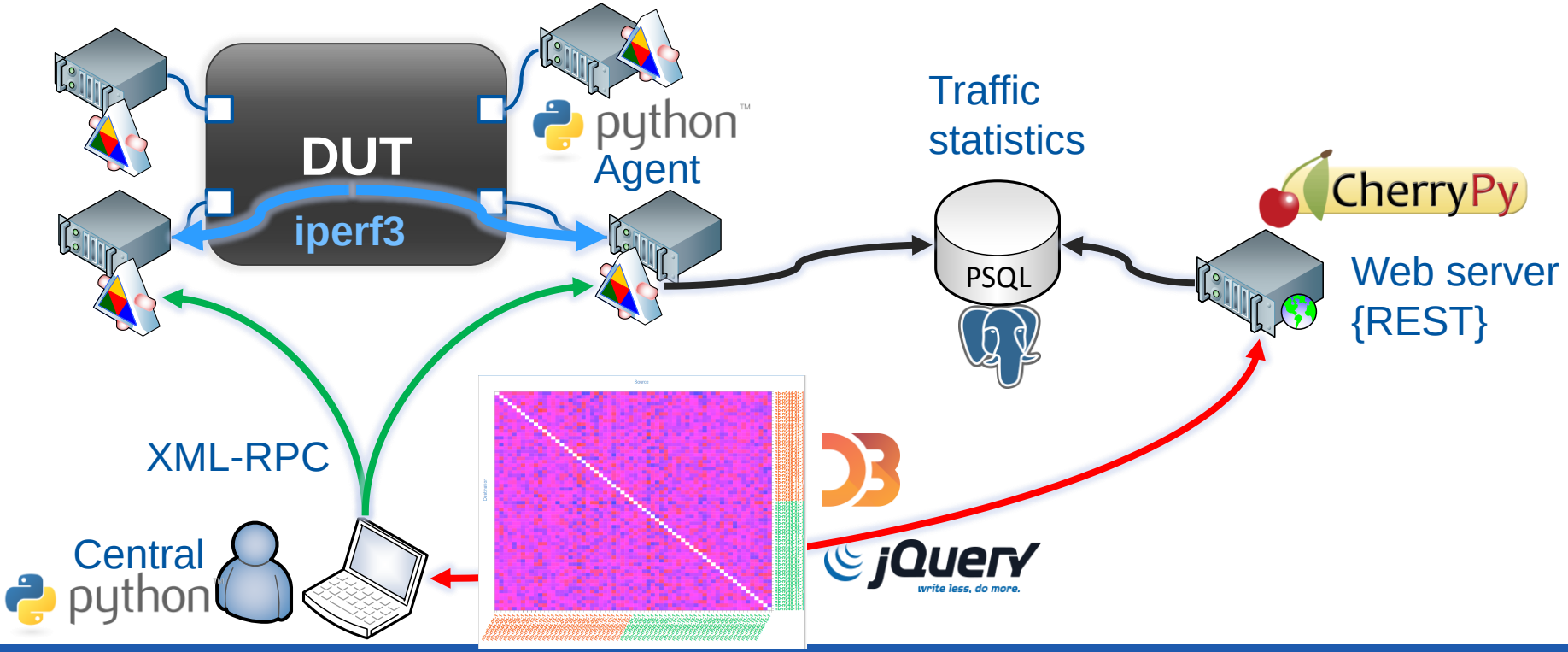
Netbench



- Use commodity servers and NICs
  - Orchestrate TCP flows (e.g. iperf3)
- ✓ Manageable cost
  - N server NIC ports – (tens or few hundreds)
  - Time-share the servers
- ✓ Exercises all ports
  - Mostly maximum size packets, similar with real-life
- ✓ Forwarding of traffic with complex distributions [3]
  - Partial mesh
  - Full mesh
- ✓ Buffering exercised
  - Multiple TCP flows, similar with real-life traffic
  - Congestion due to competing TCP flows
- *A reasonable size testbed becomes affordable*

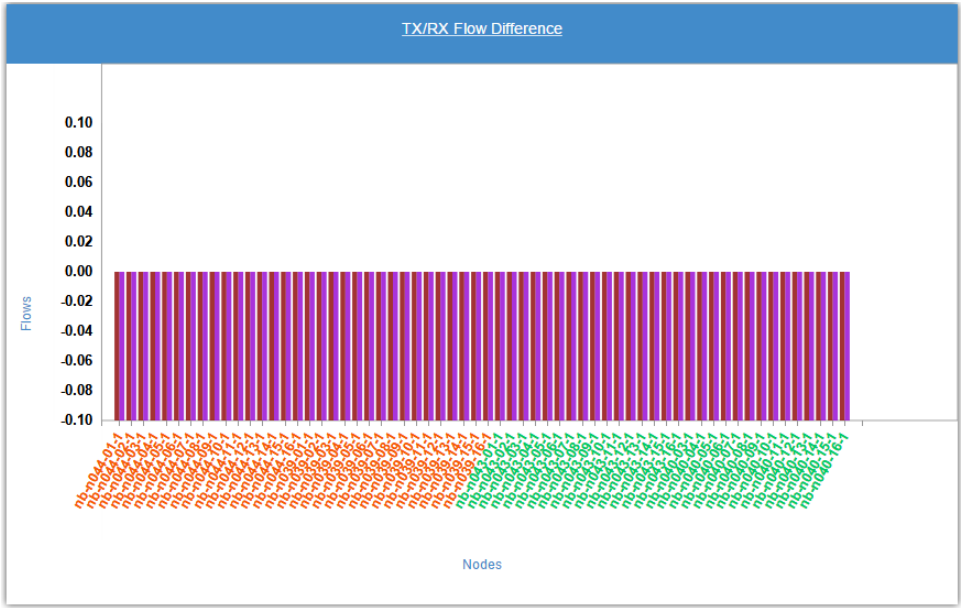
Tst = Tester
DUT = Device Under Test

Netbench design



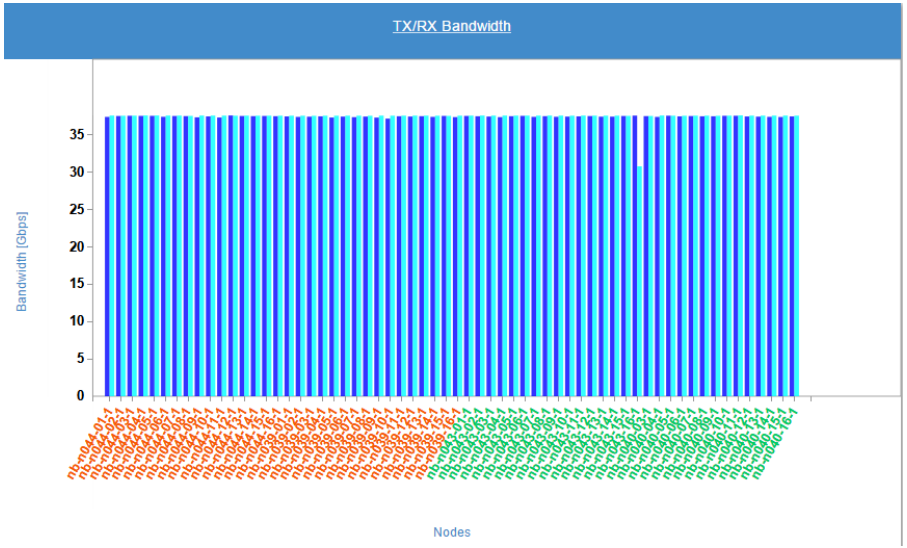
Graphs/plots – expected flows

- Plot the diff between expected and seen flows:
  - Goal: flat 0 (i.e. all flows have correctly started)



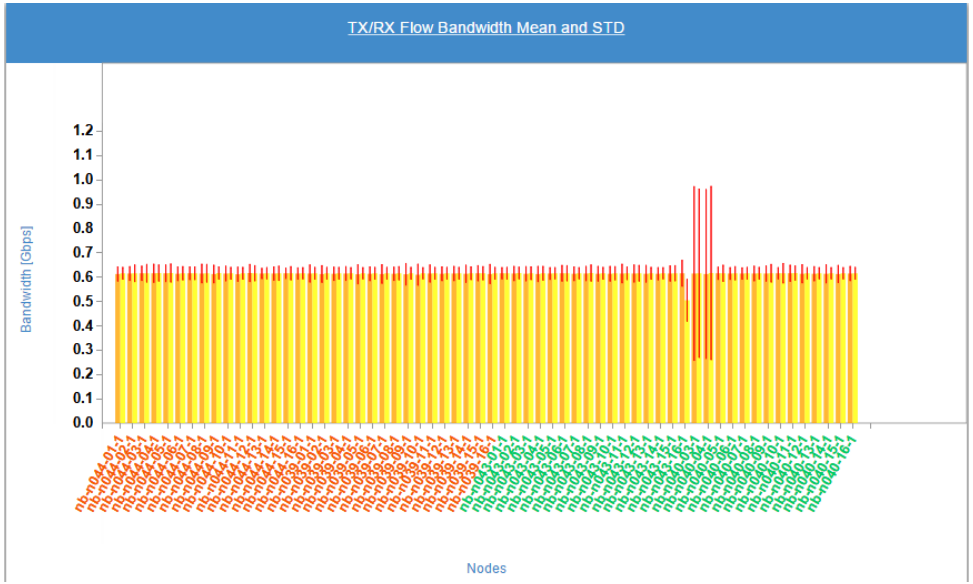
Graphs/plots – per node BW

- Plot the per node overall TX/RX bandwidth
  - Goal: flat on all nodes (fair treatment of all the nodes)



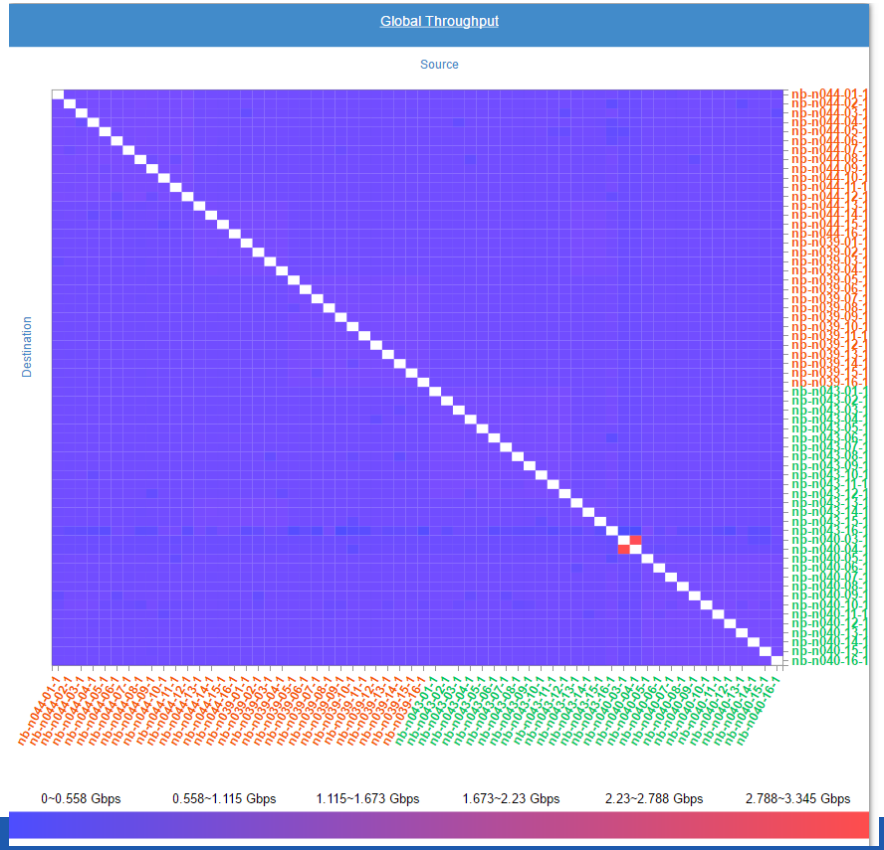
Graphs/plots – per-node flow BW

- Plot the per node average bandwidth (and stdev)
  - Goal: flat on all nodes, and small stdev



Graphs/plots – per-pair flow BW

- Plot the per pair average bandwidth
  - Goal: flat (same colour) image (no hot/cold spots)

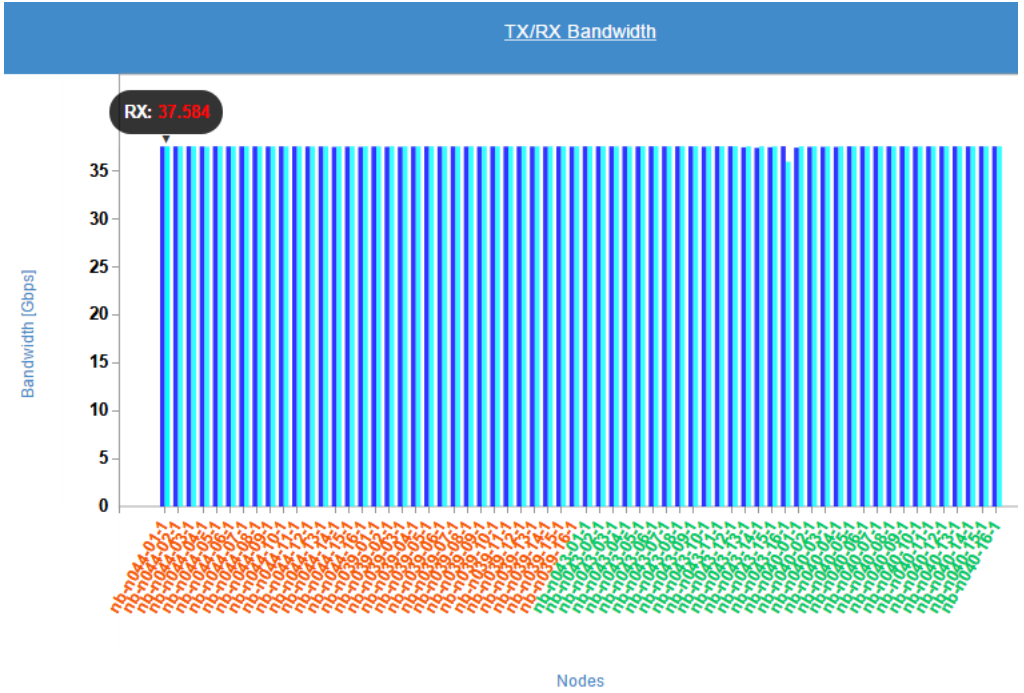


Sample results

- CERN has tendered (RFP) for datacentre routers
- TCP flow fairness evaluation
  - Netbench with 64x 40G NICs

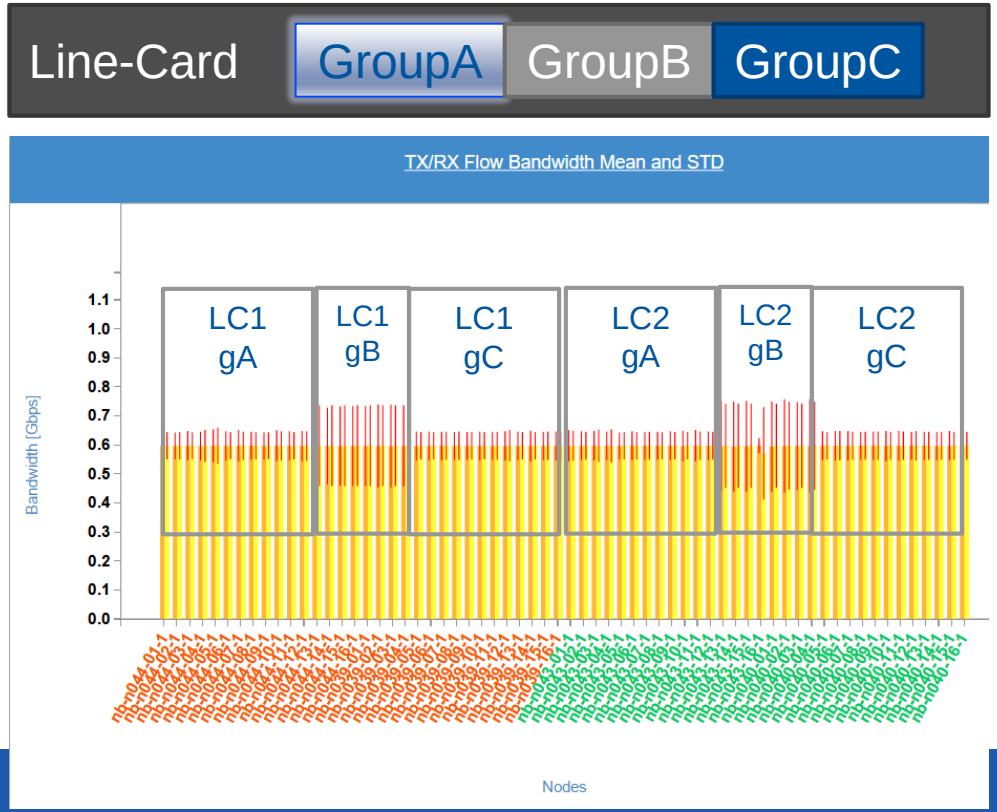
Free running TCP (1)

- Per node BW (Tx/Rx) nice and flat on all nodes ✓



Free running TCP (2)

- Groups A and C
  - all ports used
  - Small stdev
- Group B has only
  - 8/12 ports used
  - Bigger stdev

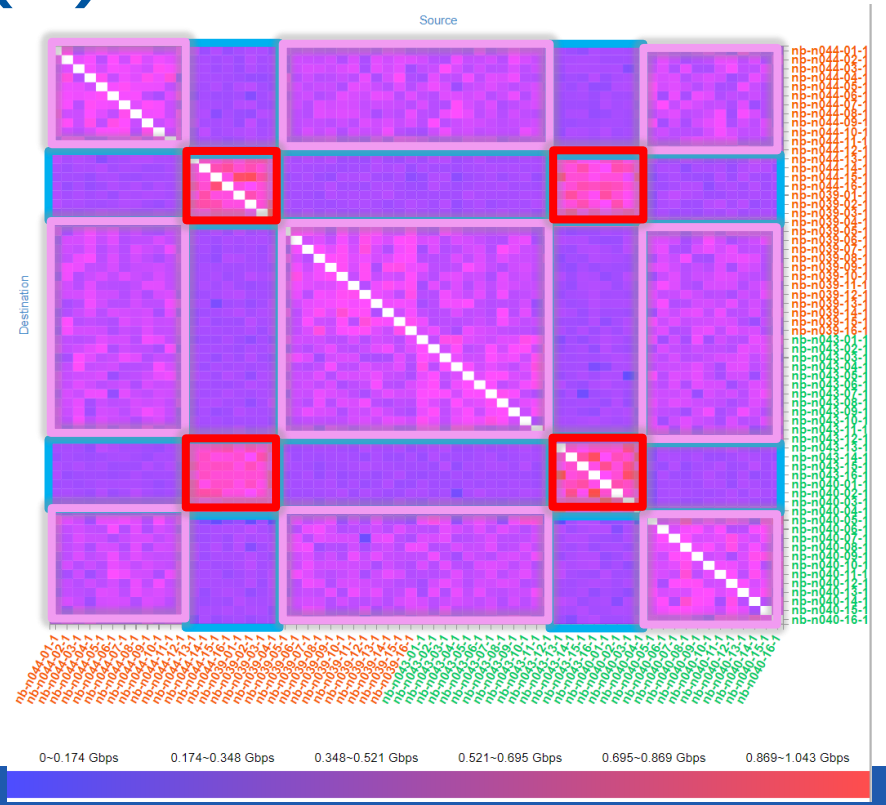


Free running TCP (3)

- Clear pattern of the unfairness
- RTT Latency (ping):
  - ~42ms regions
  - ~35ms regions
  - 0.2 – 25ms regions

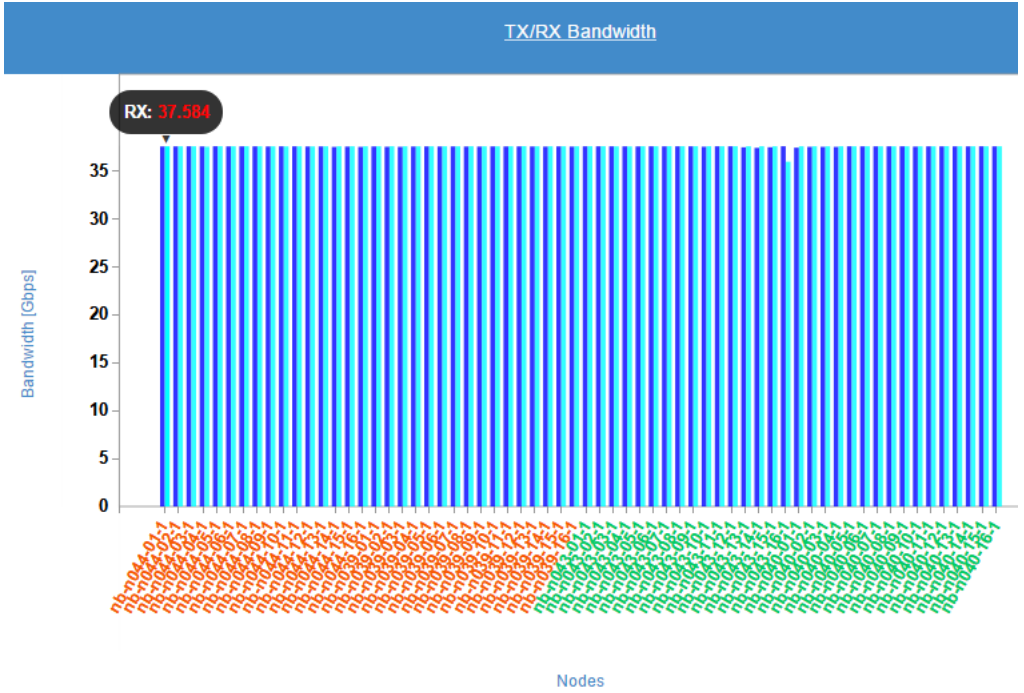
- TCP streams compete freely → device buffers fully exercised
  - Ports fully congested (most of them)
  - TCP needs to see drops to back-off

- *Good measure for the DUT buffers!*



Capped TCP window* (1)

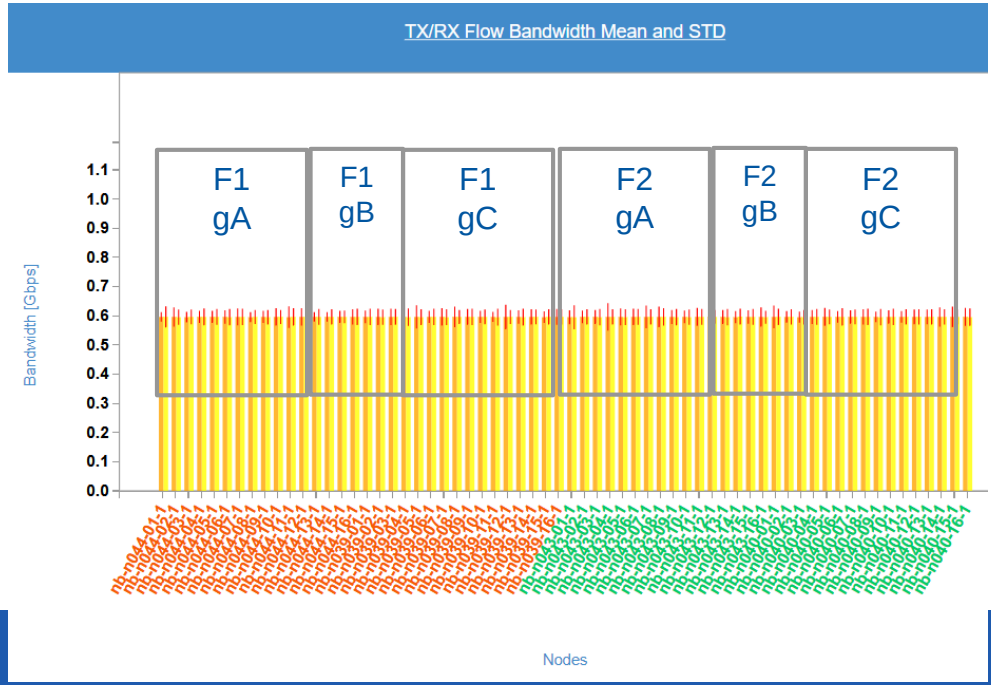
- Per node BW (Tx/Rx) nice and flat on all nodes for all tests ✓



* Capped TCP window: `iperf3 -w 64k`

Capped TCP window (2)

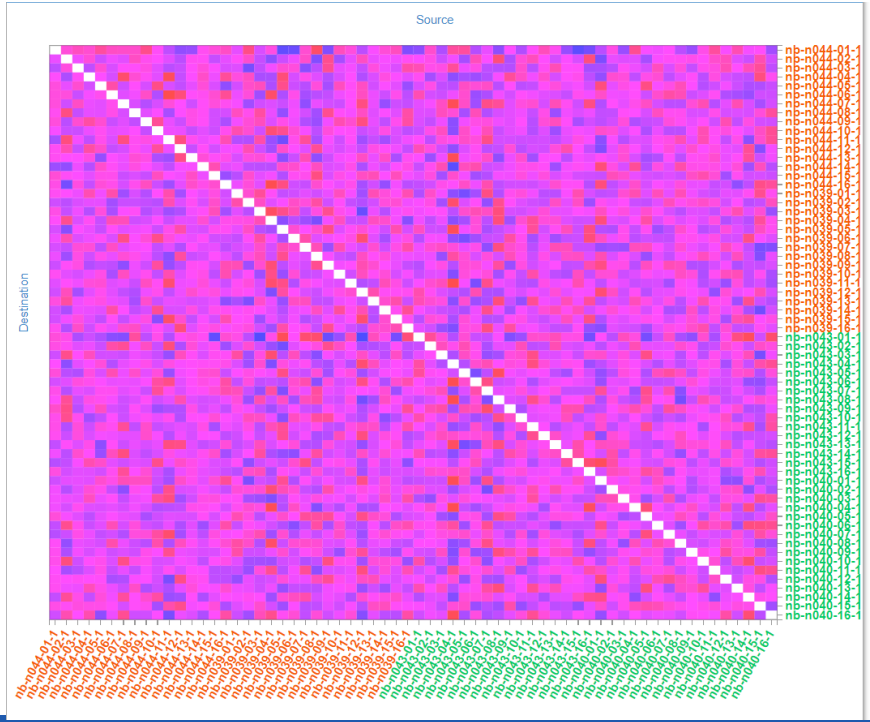
- Uniform distribution
- Small stdev



Capped TCP window (3)

- All nodes achieve ~line-rate
- Stable flat flow distribution
  - small stdev
  - latency < 1.1ms

- *Good measure for TCP flow fairness*
  - When network congestion is controlled



Summary

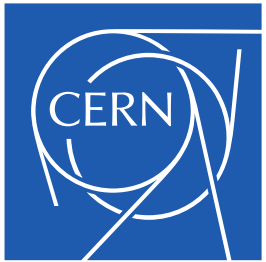
- Netbench – affordable, large-scale testing of network devices
  - traffic patterns closely resemble real-life conditions.

	Specialized HW	Specialized HW “snake”	Netbench
Test @line-rate	✓	✓	✓
Packet size scan	✓	✓	✗
Full mesh traffic	✓	✗	✓
Test buffering	–	✗	✓
Cost (large scale)	prohibitive	compromise	affordable

- Some manufacturers expressed strong *interest* in the tool
  - Anybody else?

References

- [1] RFC 2544, Bradner, S. and McQuaid J., "Benchmarking Methodology for Network Interconnect Devices"
- [2] RFC 2889, Mandeville, R. and Perser J., "Benchmarking Methodology for LAN Switching Devices“
- [3] RFC 2285, Mandeville, R., "Benchmarking Terminology for LAN Switching Devices"
- [4] iperf3 http://software.es.net/iperf/

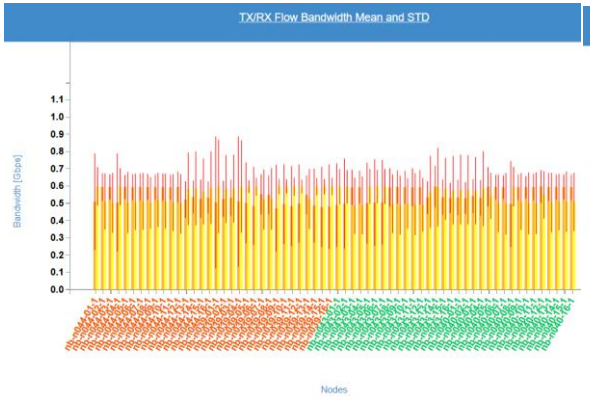

CERN

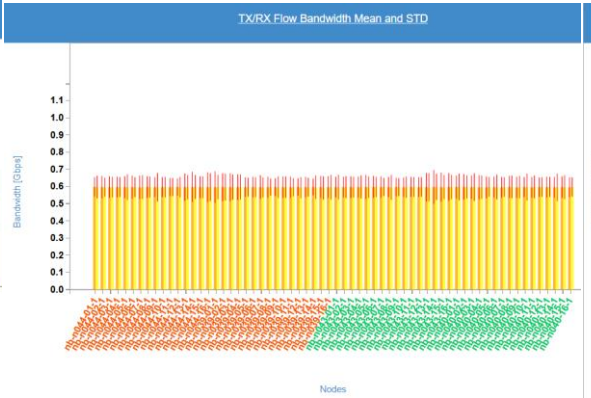
Backup material

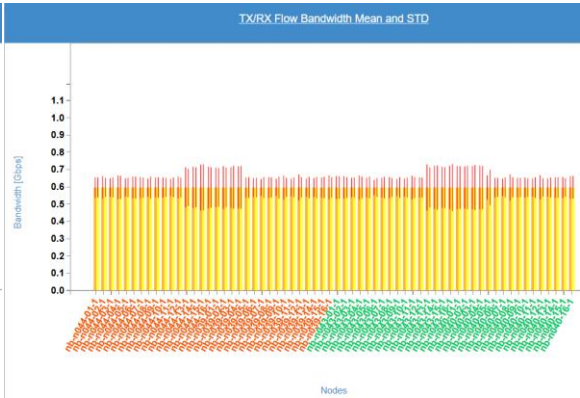
Servers tuning for fair flows

Iperf and irq affinity [1]

CPU affinity	No	Yes	**Yes** ✓
IRQ affinity	No	No	**Yes** ✓







Iperf and irq affinity [2]

CPU affinity	No	Yes	**Yes** ✓
IRQ affinity	No	No	**Yes** ✓