PANTHEON.tech

SONiC ♥ FD.io: Better Together

January 22, 9 a.m. PT

Presenters:

Miroslav Mikluš
Chief Product Officer, PANTHEON.tech

Maciek Konstantynowicz
FD.io CSIT Technical Project Lead,
Distinguished Engineer, Cisco

intel partner
Gold

What To Expect

- **SONiC**
  - History
  - Architecture
- **SONiC Users**
  - Hyperscalers
  - Enterprise
  - Telco

- **FD.io VPP and CSIT**
  - FD.io History
  - VPP and CSIT Core Projects
  - VPP -The “Magic” of Vectors
  - FD.io CSIT and Dashboard
  - CSIT Labs and Benchmarks
  - FD.io CSIT Benchmark Examples

Motivation

Orchestration & Automation

ANSIBLE

OPEN DAYLIGHT

SandWork

Network operating system

SONiC

Dataplane

cisco

BROADCOM.

NVIDIA.

Celestica

DELL Technologies

Edge-corE NETWORKS

.io

intel.

AMD

OpenPOWER

POWER8

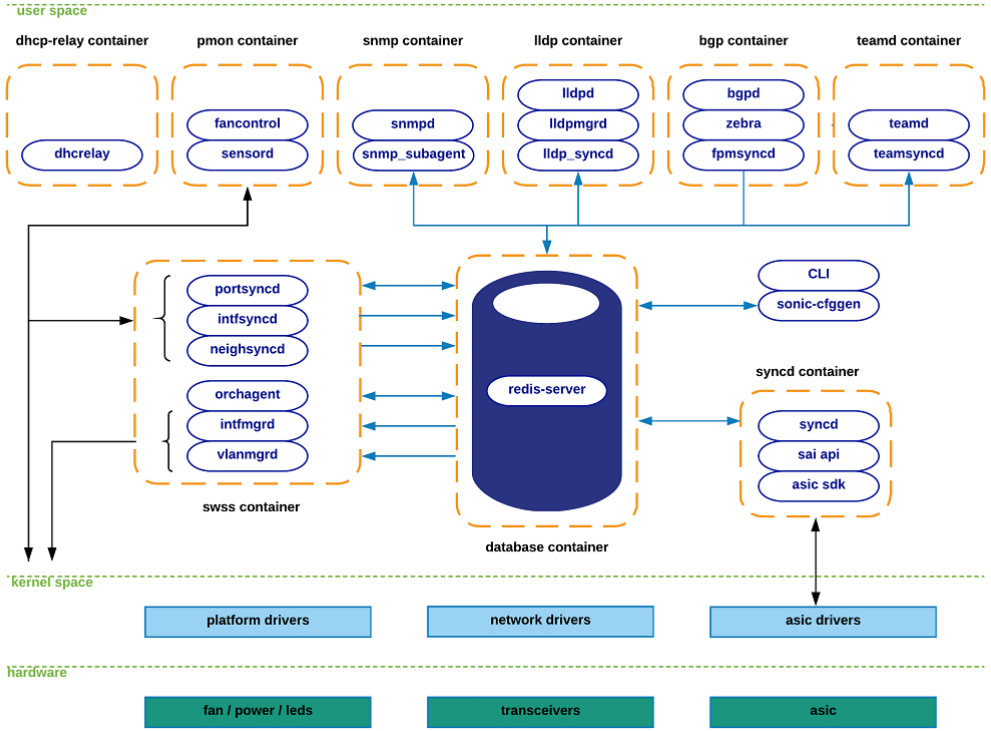
ARM

History of SONiC

- **2015**: The Switch Abstraction Interface (SAI) emerged as a critical component.
- **2016**: Microsoft developed and open-sourced SONiC.
- **2017–2018**: SONiC gained traction among major cloud service providers and enterprises, including Alibaba, Tencent, and LinkedIn, for their data center operations
- **2020**: Hardware vendors contribute to SONiC, expanding compatibility and feature set.
- **2022**: The Linux Foundation announced the incorporation of SONiC into its portfolio of open networking projects.

SONiC Architecture

- Open-Source NOS
- Scalable
- Hardware-Agnostic
- Community-Driven



SONiC Users: **Hyperscalers** 

Hyperscaler: Microsoft

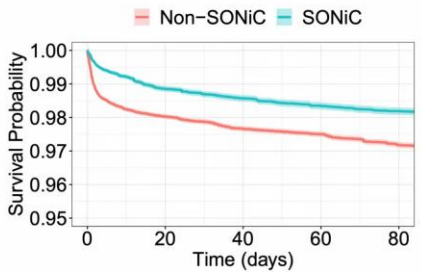
Scalability: Microsoft built SONiC to meet the scaling and operational requirements of Azure's data center networks.

Modular Design: SONiC's containerized architecture allows for independent upgrades of features, enhancing flexibility and innovation while reducing issue resolution time.

Vendor-Agnostic: SONiC supports multivendor hardware environments via the Switch Abstraction Interface (SAI), lowering dependency on single suppliers.

Key Achievements:

- Deployed across **tens of thousands of switches** in data centers globally.
- Demonstrates robustness, scalability, and multivendor interoperability.
- “Replacing vendor switch OSs with SONiC has been beneficial in improving the **resilience of data center switches**”



src: Singh, R., Lillibridge, M., & Ma, M. (2020). *SONiC: Software for Open Networking in the Cloud.* SIGCOMM Computer Communication Review.

Hyperscaler: Alibaba

Alibaba began **adopting SONiC** for its data center network devices **in 2017**.

High Flexibility: SONiC powers Alibaba's AI and cloud compute/storage infrastructure.

Key Achievements:

- Deployed **over 100,000 white-box** network devices, operating a SONiC-based software stack.
- Substantial **cost savings**, seamless infrastructure management, and significant **scaling potential.**
- **Saved 80–90% of time** of testing time by focusing efforts on features actively used on devices, rather than performing regression tests on all functionalities.

src: Alibaba Cloud. (2024). *SONiC Use Case: Alibaba Data Center Network.* SONiC Foundation.

SONiC Users: **Enterprises** 

Enterprise: Walmart

The world's largest retailer operates over **100,000 routers and switches**.

Key requirements:

- Visibility
- Reduction of operational overhead
- Vendor agnosticism

Achievements:

- Successfully deployed Enterprise **SONiC** at scale across its **data centers**.

Future plans:

- Expanding SONiC deployment to stores, distribution centers, campuses, and beyond..

src: Walmart. (2024). *How Walmart Leverages Open Source for Operational Excellence at Scale.* SONiC Foundation.

Enterprise: Broadcom

Broadcom & VMWare enterprise datacenters: ~9 DCs worldwide

Challenges Addressed:

- Improved **network visibility** and **troubleshooting** capabilities.
- **Orchestration** resulted in a significant **reduction** in operational **overhead**.
- Migration from a traditional 3-tier architecture to SONiC, **spine-leaf design**.
- SONiC rolled out in **campus** and **WAN environments**.

src: Broadcom. (2024). *Broadcom Data Center Network Migration to SONiC – VMware Infrastructure*. YouTube.

SONiC Users: **Telco** ☎

Telco: Orange

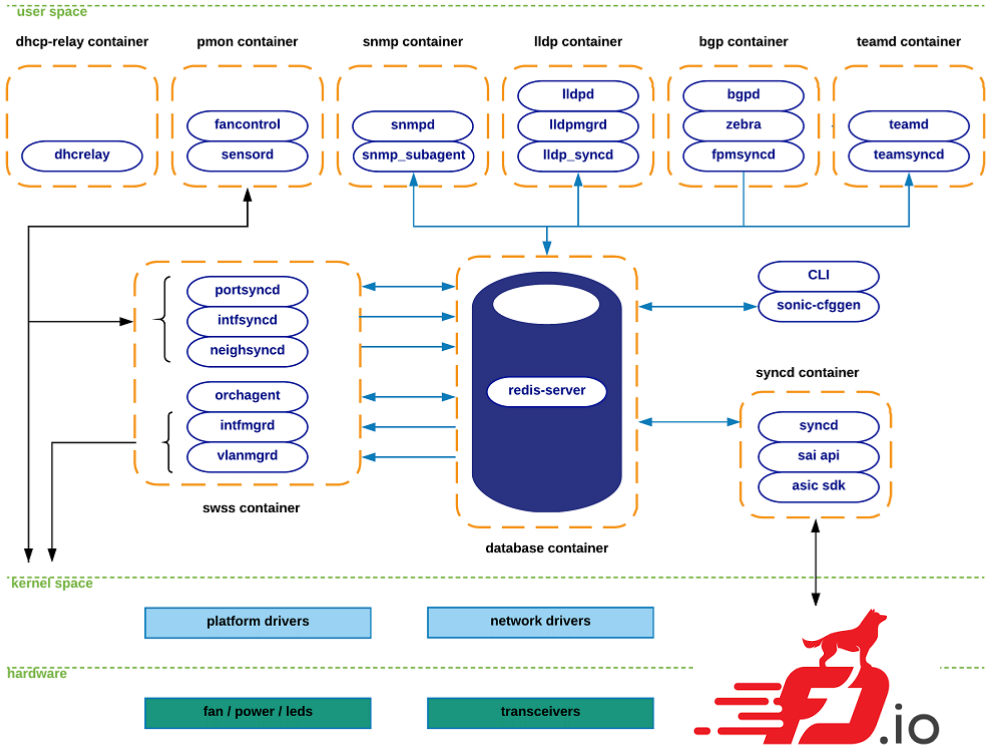
- **Challenge**: To reduce dependency on fully integrated products.
- **Solution**: Using SONiC to address the **fiber access** Telco Use case.

- **Key Achievements:**
  - Deployed **90 disaggregated** open switches.
  - Gained control of the **features roadmap**, **avoiding lock-in** and dependencies.
  - **Reduced costs,** both in terms of products and power consumption.
  - **Attracted** and upskilled **talents** to work in an open ecosystem.

src Orange. (2024). *SONiC Use Case: Orange Network Modernization.* SONiC Foundation.

SONiC – FD.io

- Open-Source NOS
- Scalable
- Hardware-Agnostic
- Community-Driven
- FD.io – vpp



FD.io

VPP Vector Packet Processing - Code and CSIT Benchmarking

History of FD.io

- **2002: VPP Code Created and Used Within Cisco**

  The high performance network forwarding code designed and authored by Dave Barach.

- **2004: Key Patent Contribution**

  Dave Barach and Eliot Dresselhaus filed a patent related to efficient packet processing.

- **2016: VPP Goes Open Source***

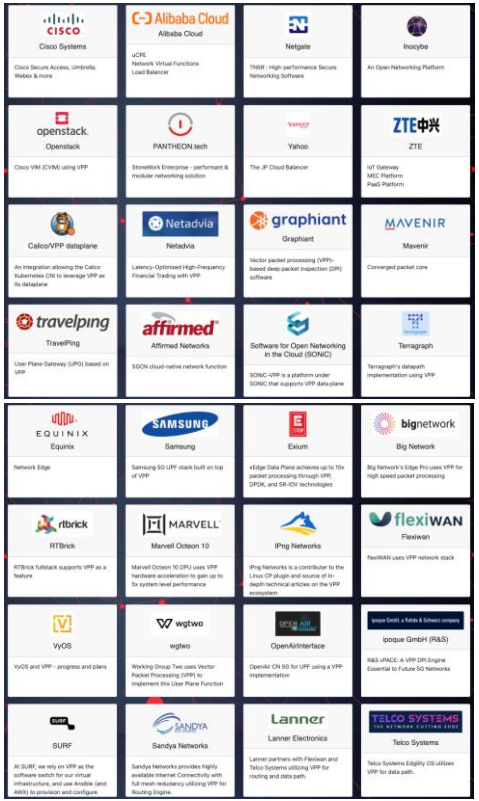
  The Linux Foundation announced the **Fast Data Project (FD.io)** with VPP as a core component.

  Cisco contributed VPP to the open-source community, enabling collaborative network innovation.

  FD.io CSIT benchmarking core project created to continuously verify performance.

- **2016+: Substantial Industry Adoption**

  Adoption of VPP code and CSIT benchmarking across the SW networking and multiple industry verticals



* Cisco Blog “A BIGGER HELPING OF INTERNET PLEASE!”, April 4, 2016, https://blogs.cisco.com/sp/a-bigger-helping-of-internet-please

FD.io VPP and CSIT

Core Projects in LFN FastData.io

	VPP Vector Packet Processing	CSIT Continuous System Integration and Testing
Performance	Feature rich networking and host stack. VPP on COTS servers in many cases **outperforms** packet processing HW[1].	Continuous **benchmarking** of VPP and DPDK.
Portability	VPP runs on COTS hardware: AMD, ARM, intel, OpenPOWER POWER8 VPP runs in any environment: **bare-metal, VM, containers**.	Performance testbeds with **Xeon**, **Arm**, **AMD**, **Atom** HW[1]. **Bare-Metal**, **VM**, **Container, Cloud** test environments.
Efficiency	Allows ability to **upscale** and **downscale.**	Executing **>3k benchmarking tests** daily.
SDN	Software **programmable**, **extendable** and **flexible.**	Open-source **CI/CD infrastructure for benchmarking** of SW data-planes, **test data analytics** and **presentation**.

[1] VPP optimally leverages relevant HW accelerators in NIC/DPU and in CPU socket for packet processing, I/O and memory copy operations. Some of these accelerations are benchmarked in CSIT e.g. IPsec, mem-copy, GTPU encap/decap.

Running Networking Functions
on ASICs/NPUs and COTS* CPUs/Servers

- Purpose built ASICs/NPUs vs COTS CPUs/servers
  - The former always win in terms of performance and power efficiency, if they implement what is needed …
  - But network and security requirements are moving fast and some things are hard to get right in ASICs/NPUs
- Welcome to the world of software networking on COTS CPUs/servers
  - With some hard problems to solve, one of them being the time budget per packet:

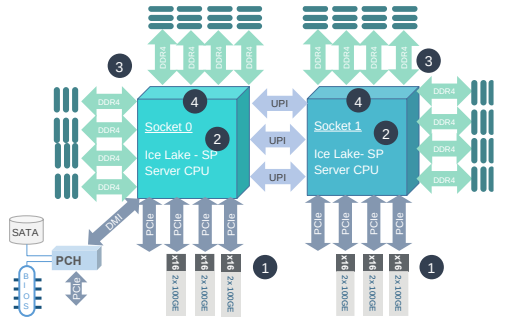
2

```
Processing time budget per packet: nsec
@1518B  10GE line-rate .8127 Mpps: 1230     multiple
@1518B 100GE line-rate 8.127 Mpps: 123.0    memory accesses
  @64B  10GE line-rate 14.88 Mpps: 67       are needed
  @64B 100GE line-rate 148.8 Mpps: 6.7      per packet …
```

3

```
Intel Xeon® memory single access time: nsec
                        At cache miss: ~90   … to fetch
                              LLC hit: ~20   instructions
                            L2i/d hit: ~7    and
                               L1 hit: ~3    data
```

1



1. Network I/O PCIe Gen4: **600 Gbps**
2. Per Core Caches: L1d **48 KB**, L2 **1.25 MB**
3. Memory: **8-channels** 3200 MHz
4. Max power: **250 W** (TDP)

Intel® Xeon® 8358
32 Cores, 2.4 GHz, 54 MB Cache

4

Simple and fast stuff used all the time goes into ASICs/NPUs.
Complex, stateful and new service functions go into Software.

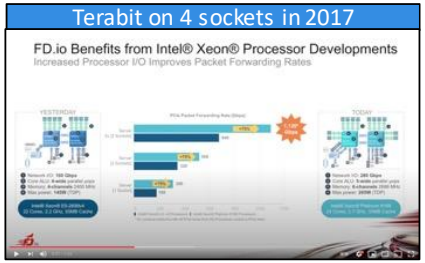
* COTS – Common Off The Shelf

Networking in software on CPUs/Servers – some LFN FD.io references:
- Benchmarking and Analysis of Software Data Planes, 2017, https://fd.io/docs/whitepapers/performance_analysis_sw_data_planes_dec21_2017.pdf
- Benchmarking Software Data Planes Intel® Xeon® Skylake vs. Broadwell, 2019, https://www.lfnetworking.org/wp-content/uploads/sites/55/2019/03/benchmarking_sw_data_planes_skx_bdx_mar07_2019.pdf

New “Moore’s law” for Software Networking
Performance Doubles Every 3 Years …

Intel® Xeon® Skylake 14nm
PCIe Gen 3.0 x16 lanes
for 100 GbE per NIC slot

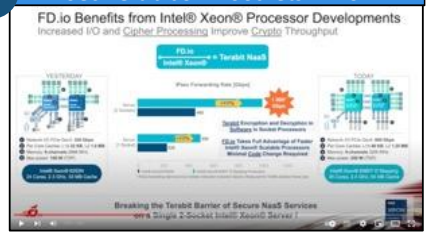
Terabit on 4 sockets in 2017



https://youtu.be/aLJ0XLeV3V4

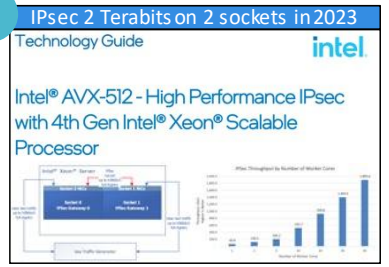
Intel® Xeon® Icelake 10nm
PCIe Gen 4.0 x16 lanes
for 200 GbE per NIC slot

1 IPsec Terabit on 2 sockets in 2021



https://youtu.be/ipQQmjzE_g0

2 IPsec 2 Terabits on 2 sockets in 2023



https://bit.ly/3PiLv9j

Intel® Xeon® Sapphire Rapids 7nm
PCIe Gen 5.0 x16 lanes
for 400 GbE per NIC slot

3 Factors driving increased network throughput in servers:
- Improving ISA, increasing PCIe I/O, higher gates density (lower nm).
- Targeted Intel Xeon optimizations e.g. AES-NI crypto in Icelake[(1)] onwards.

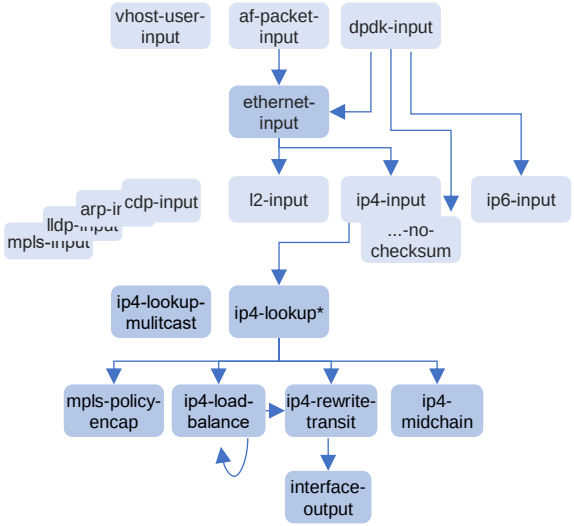
4 And importantly optimizing software for network packets:
- **Exploring and bending the laws of physics with VPP[(2)].**
- Enriching VPP functionality and lowering cycles per packet.

[(1)] Icelake at HotChips2020, https://hc32.hotchips.org/assets/program/conference/day1/HotChips2020_Server_Processors_Intel_Irma_ICX-CPU-final3.pdf
[(2)] VPP vector packet processing, cleverly batching packets into vectors for processing and solving von Neumann bottleneck.

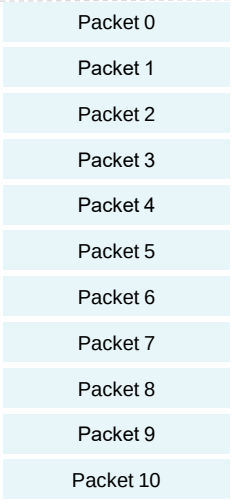
FD.io VPP – The “Magic” of Vectors

Compute Optimized SW Network Platform

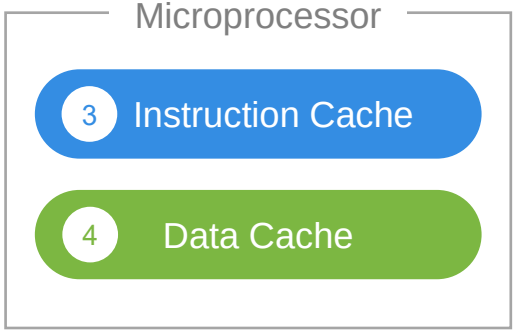
1 Packet processing is decomposed into a directed graph of nodes …



2 … packets move through graph nodes in vector …



3 … graph nodes are optimized to fit inside the instruction cache …



4 … packets are pre-fetched into the data cache.

* Each graph node implements a “micro-NF”, a “micro-NetworkFunction” processing packets.

Makes use of modern Processor micro-architectures.

Instruction cache & data cache always hot ➔ Minimized memory latency and usage.

FD.io CSIT and Dashboard

11+6

ETL pipelines

JSON

models

~3k

tests daily

~64k

performance tests per release

...

rls23xx

rls24xx

rls2502 (next)

optimization

32GB RAM

Parquet

output

~18k

tests weekly

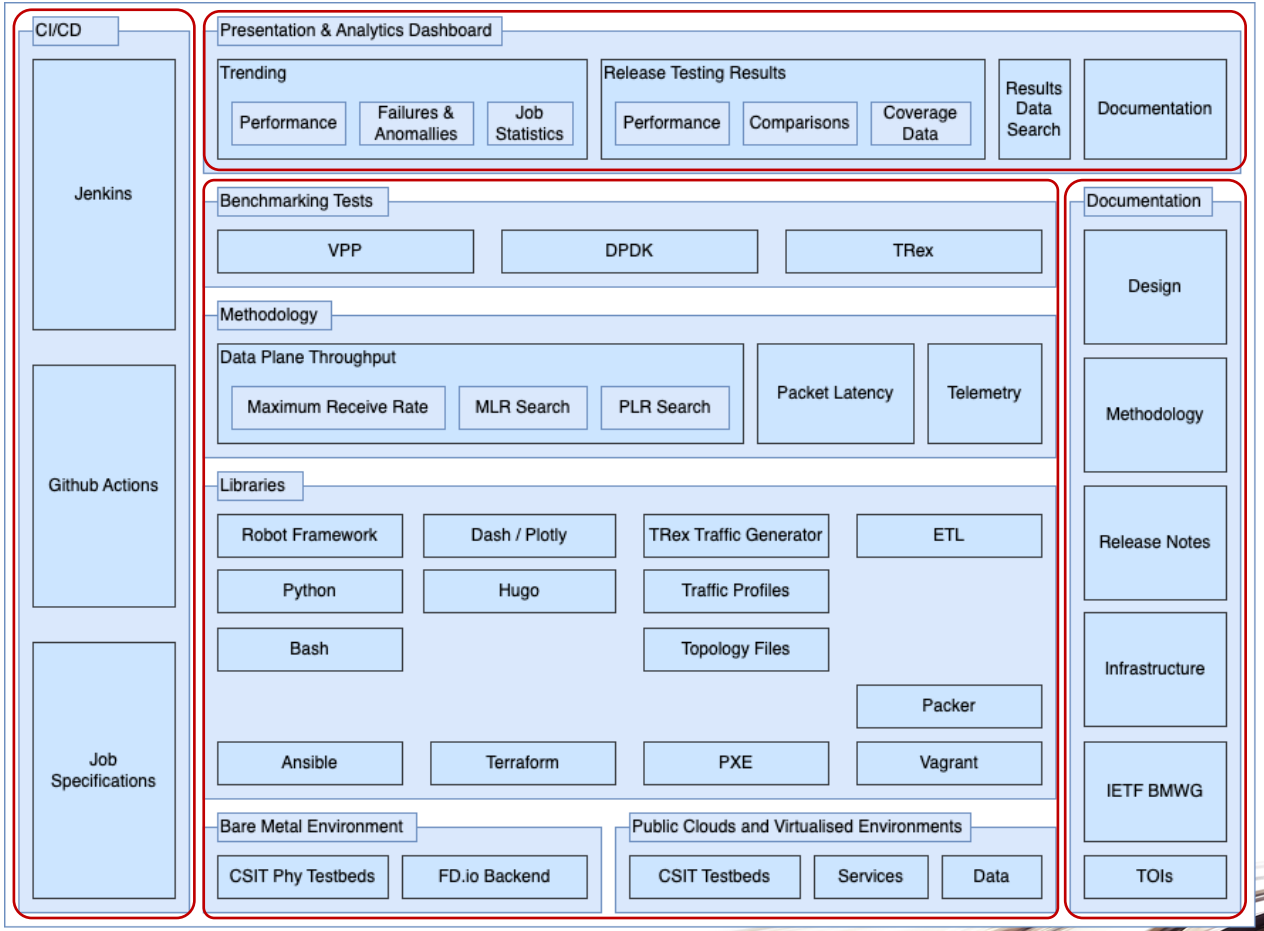
2021 ...

250 days

sliding window

... 2025

FD.io CSIT System Design



CSIT Benchmark Areas and Methodologies

Tests

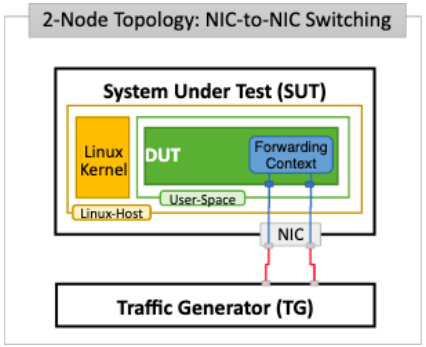
Benchmark Test Areas

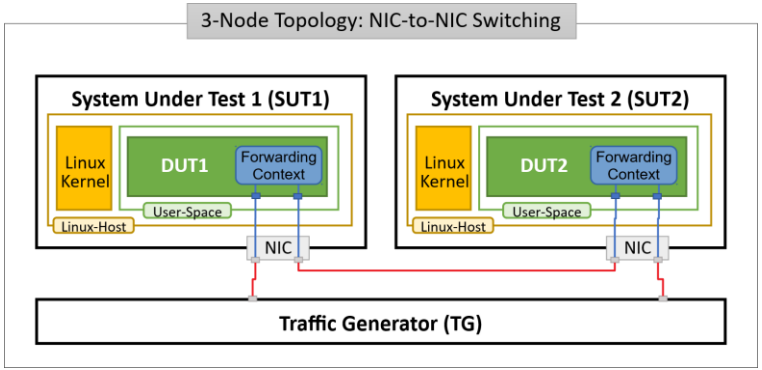
- L2 Ethernet Switching
- IPv4, IPv6 Routing
- IPsec, Wireguard with IPv4 Routing
- SRv6 Routing
- Features: ACLs, NAT44-EI/ED, Policer, …
- IPv4, IPv6 Tunnels
- KVM VMs vHost-user
- Docker Container Memif
- Drivers: DPDK, AVF, RDMA, AF_XDP

Test Methodologies

- Packet Throughput and Latency
- Stateful NAT44ed
- Stateful Host-stack
- Speedup Multi-Core
- Soak Tests
- Reconfiguration Tests

Test Topologies





Performance Testbed Variants*

~~2n–clx~~
2n–icx
2n–spr
~~2n–tx2~~
2n–zn2
2n–grc
2n–emr

3n–icx
3na–spr
3nb–spr
3n–alt
3n–snr
3n–icxd
~~3n–tsh~~
3n–emr
3n–oct

*Testbed Topology – SUT Processor Model

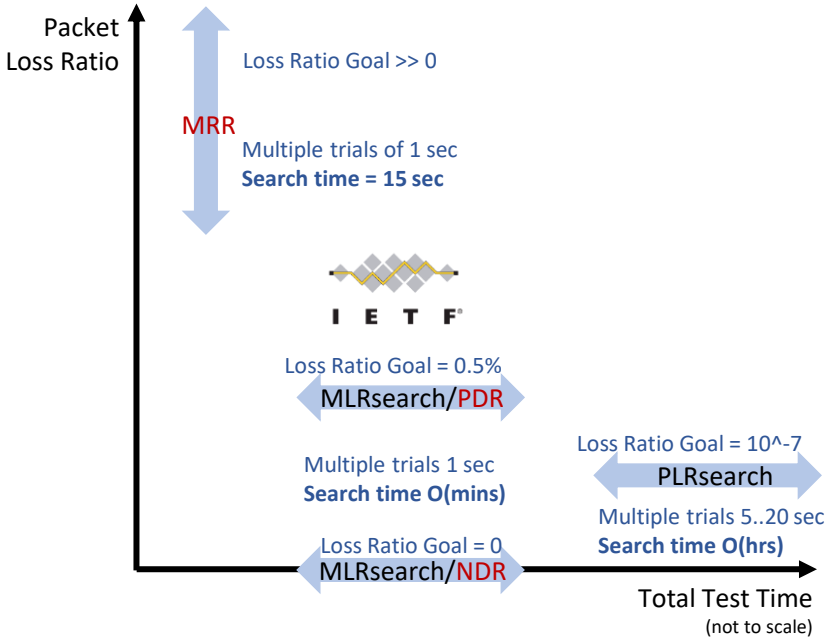
LF NETWORKING

Performance testing: List of compute platforms

Processor Family	Model	Cores per Socket	Base Frequency GHz	L3 Cache (LLC) MB	Testbeds	NICs
EMR Intel Xeon Emerald Rapids* (on-chip HW accelerators: QAT, DSA)	8568Y+	48	2.3	300	2 x 2n3n-emr	e810-2p100GE
ICX Intel Xeon Ice Lake	8358	32	2.6	48	4 x 2n-icx 2 x 3n-icx	xxv710-2p25GE, e810-4p25GE, e810-2p100GE, cx7-2p200GE xxv710-2p25GE, e810-4p25GE, e810-2p100GE, cx6-2p100GE
SPR Intel Sapphire Rapids (on-chip HW accelerators: QAT, DSA)	8462Y+	32	2.8	60	2 x 2n-spr 1 x 3na-spr 1 x 3nb-spr	e810-4p25GE, e810-2p100GE, cx7-2p200GE cx7-2p200GE e810-4p25GE, e810-2p100GE
ZN2 AMD EPYC Zen2	7532	32	2.4	256	1 x 2n-zn2	x710-4p10GE, xxv710-2p25GE, cx5-2p100GE
Marvell Octeon10 DPU** (Neoverse N2 cores)	CN106xx	24	2.5	48	1 x 3n-oct	CN106 GSERM0 2p100GbE
ALT Ampere Altra (Neoverse N1 cores)	Q80-30	80	3.0	32	1 x 3n-alt	xl710-2p40GE cx6-2p200GE
SNR Intel Atom Snowridge (on-chip HW accelerators: QAT)	P5362B	24	2.2	27	1 x 3n-snr	e810-4p25GE
Nvidia Grace C1* (Neoverse V2 cores)	C1	72	3.3	114	1 x 2n-grc	cx5-2p100GE, cx7-2p200GbE**
ICX-D Intel Xeon Ice Lake (on-chip HW accelerators: QAT)	D-2796NT	20	2.0	30	1 x 3n-icxd	e810-XXVDA2-2p25GE

* Just added
** Being added

Innovation in Test Methodologies: MRR, MLRsearch, PLRsearch



- **Maximum Receive Rate (MRR)**
  - Measures packet throughput under maximum load regardless of packet loss.
  - Used for daily trending and automated anomaly detection in FD.io CSIT.
- **Multiple Loss Ratio search (MLRsearch)**
  - Discovers multiple packet rates (NDR, PDR[1]) in a single search.
  - Discovers performance spectrum, filtering out SUT noise.
  - IETF BMWG – workgroup draft in Last Call: draft-ietf-bmwg-mlrsearch.
- **Probabilistic Loss Ratio search (PLRsearch)**
  - Soak[2] testing for data planes over an extended period of time.
  - IETF BMWG – named draft to be updated: draft-vpolak-bmwg-plrsearch.
- **All used in production CI for stateless and stateful benchmarks**
  - Stateless L2, IP4, IP6, clear-text and crypto tunnels tests for BPS, PPS.
  - Stateful UDP and TCP tests for BPS, CPS, PPS, TPS.
- **Why does one need MRR, PDR, NDR, Soak?**
  - A well behaving system should have MRR, PDR, NDR and Soak rates close to each other!

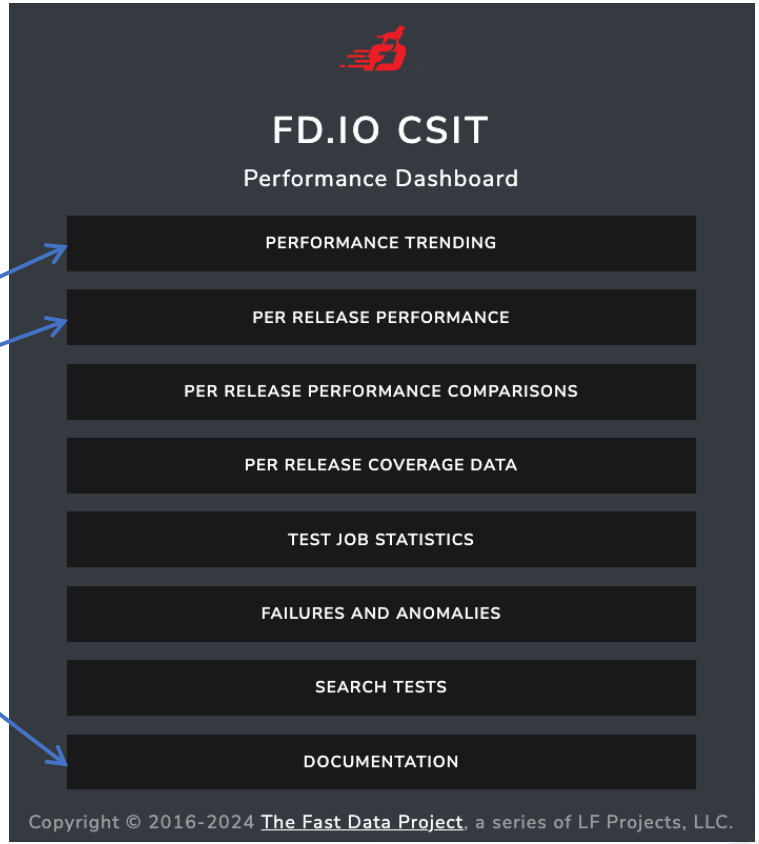
Beyond RFC2544: Innovating in benchmarking methodologies

[1] Packet rate and bandwidth measurements: Non-Drop Rate (NDR, with zero packet loss), ii) Partial Drop Rate (PDR, with packet loss rate not greater than configured non-zero packet-loss-ratio).
[2] CSIT Soak tests use PLRsearch with packet-loss-ratio set to 10^-7.

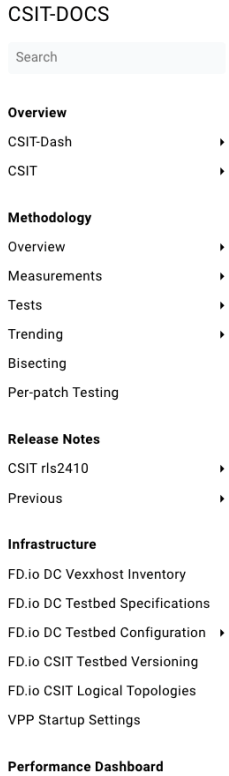
FD.io CSIT Dashboard

https://csit.fd.io/

Next slides provide content and result examples



FD.io CSIT – Documentation and Settings

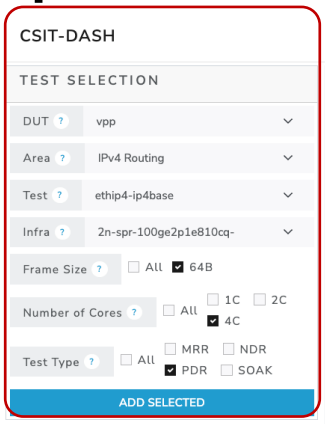


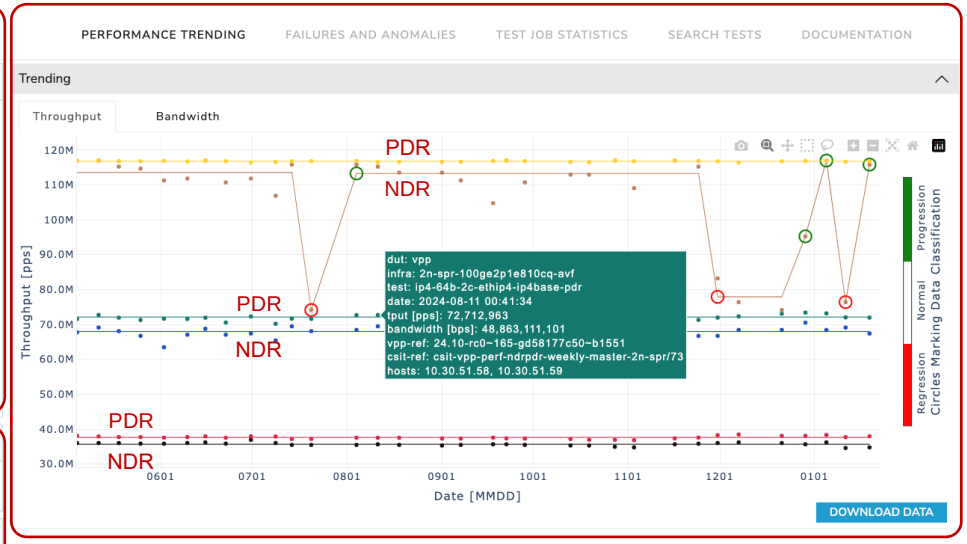
Documentation Structure

1. OVERVIEW: General introduction to CSIT Performance Dashboard and CSIT itself.
   - **CSIT-Dash**: The design and the structure of CSIT-Dash dashboard.
   - **CSIT**: The design of the FD.io CSIT system, and the description of the test scenarios, test naming and test tags.
2. METHODOLOGY
   - **Overview**: Terminology, per-thread resources, multi-core speedup, VPP forwarding modes and DUT state considerations.
   - **Measurement**: Data plane throughput, packet latency and the telemetry.
   - **Test**: Methodology of all tests used in CSIT.
   - **Trending**: A high-level design of a system for continuous performance measuring, trending and change detection for FD.io VPP SW data plane (and other performance tests run within CSIT sub-project).
   - **Per-patch Testing**: A methodology similar to trending analysis is used for comparing performance before a DUT code change is merged.
3. RELEASE NOTES: Performance tests executed in physical FD.io testbeds.
   - **CSIT 24.10**: The release notes of the current CSIT release.
   - **Previous**: Archived release notes for the past releases.
4. INFRASTRUCTURE
   - **FD.io DC Vexxhost Inventory**: Physical testbeds location.
   - **FD.io DC Testbed Specifications**: Specification of the physical testbed infrastructure.
   - **FD.io DC Testbed Configuration**: Configuration of the physical testbed infrastructure.
   - **FD.io CSIT Testbed Versioning**: CSIT test environment versioning to track modifications of the test environment.
   - **FD.io CSIT Logical Topologies**: CSIT performance tests are executed on physical testbeds. Based on the packet path thru server SUTs, three distinct logical topology types are used for VPP DUT data plane testing.
   - **VPP Startup Settings**: List of common settings applied to all tests and test dependent settings.
5. PERFORMANCE DASHBOARD
   - **Performance Trending**
   - **Per Release Performance**
   - **Per Release Performance Comparisons**
   - **Per Release Coverage Data**
   - **Test Jobs Statistics**
   - **Failures and Anomalies**
   - **Search Tests**

https://csit.fd.io/cdocs/

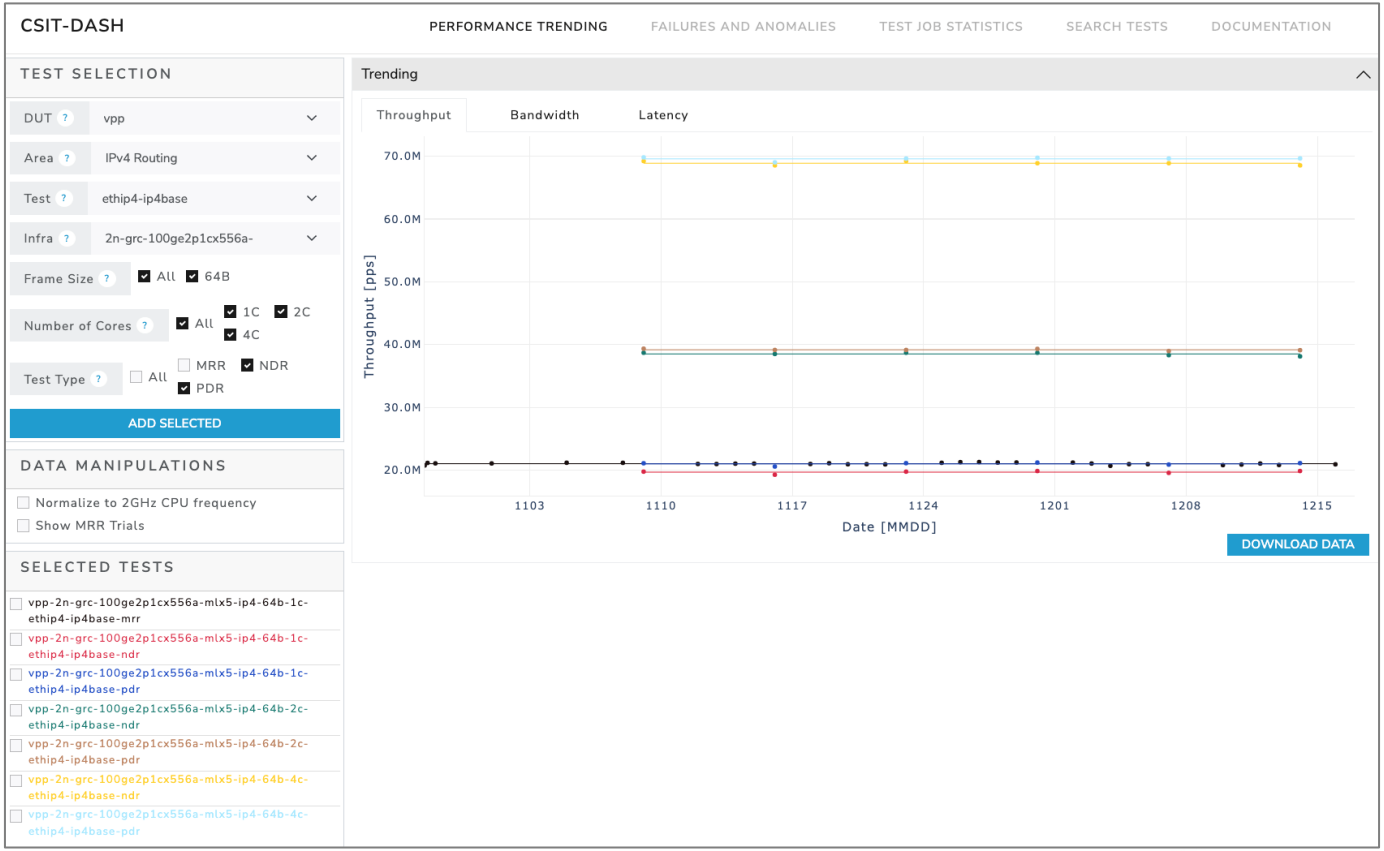
2n-spr – IPv4 baseline – 1, 2, 4 core





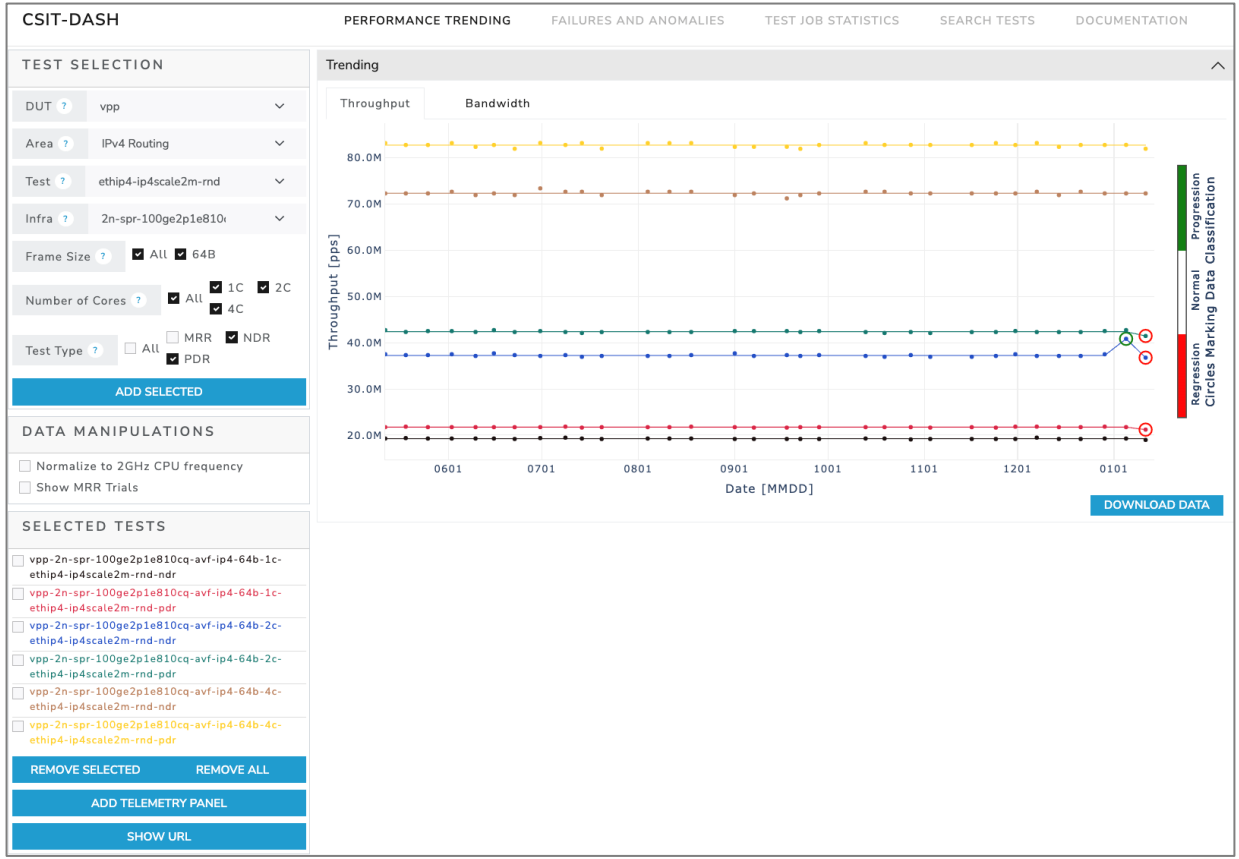
https://csit.fd.io/trending/#eNrlVcsOgjAQ_Bq8mE1oLeLFg8h_mAqrkPBY24rBr7cYk5WDiQe54KHPmc62k0lqXWvwYLHaBlESxEkg4zL3XbDaLf3QEYFswJlBEYZnlCRwl8LsAro7QUkK1uoIlgN0xbDy7agtQpObQUTuB5H86kaKjFDRM_KxDvO1Qc0HfDGGHNq3KuPrMOtkdI22vCNT_QMYz7wbDllsrO96ekNfb4zTJ-Nn1tEfWEe_sE7ONHVy-tTJmaZOTp86NdPUqelTp2aaOvVl6qJ00bSmfv60fm6L9gbOlLqyr60HG8EFlg

2n-grc – IPv4 baseline – 1, 2, 4 core



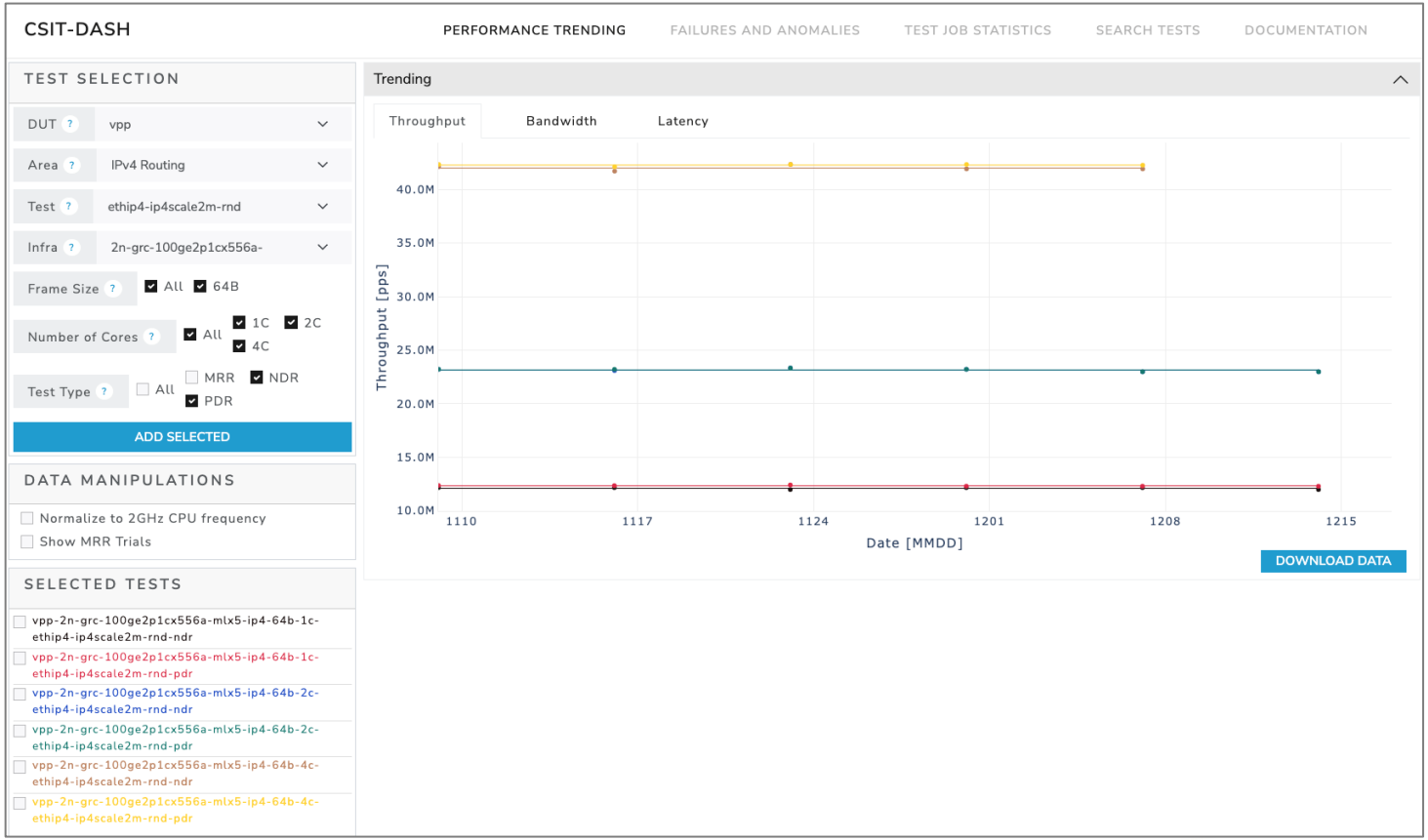
https://csit.fd.io/trending/#eNrlVssOgjAQ_Bq8mE2ktnLyoPlfppQVSHg0bVXw6y2GZOVg4oFwgEOfszvbncyh1jUGrxbLYyDOQXQOWFSkfgr2p61fHloDqyEzCsLdLkOmQ9UKcZBQla2AQnM48ARCBejy_uRHli1CZUzPwi49S3p3l0pCdN4R8rsQJUiDkjJ8NYlc2q8y4_dQ1M3lCm3xQgr1HRCuvB4EhWrM7zr9hQ5NRvEnYjrx6nQF4g1NTi6eXoN4ehLx2FKdx2ZwHluq89gMzuNLdR6fwXl8qc7jfzpPxJu6MdXny-L3Nm-e4EwhSztcvQGYblw7

2n-spr – IPv4 scale2m-rnd – 1, 2, 4 core



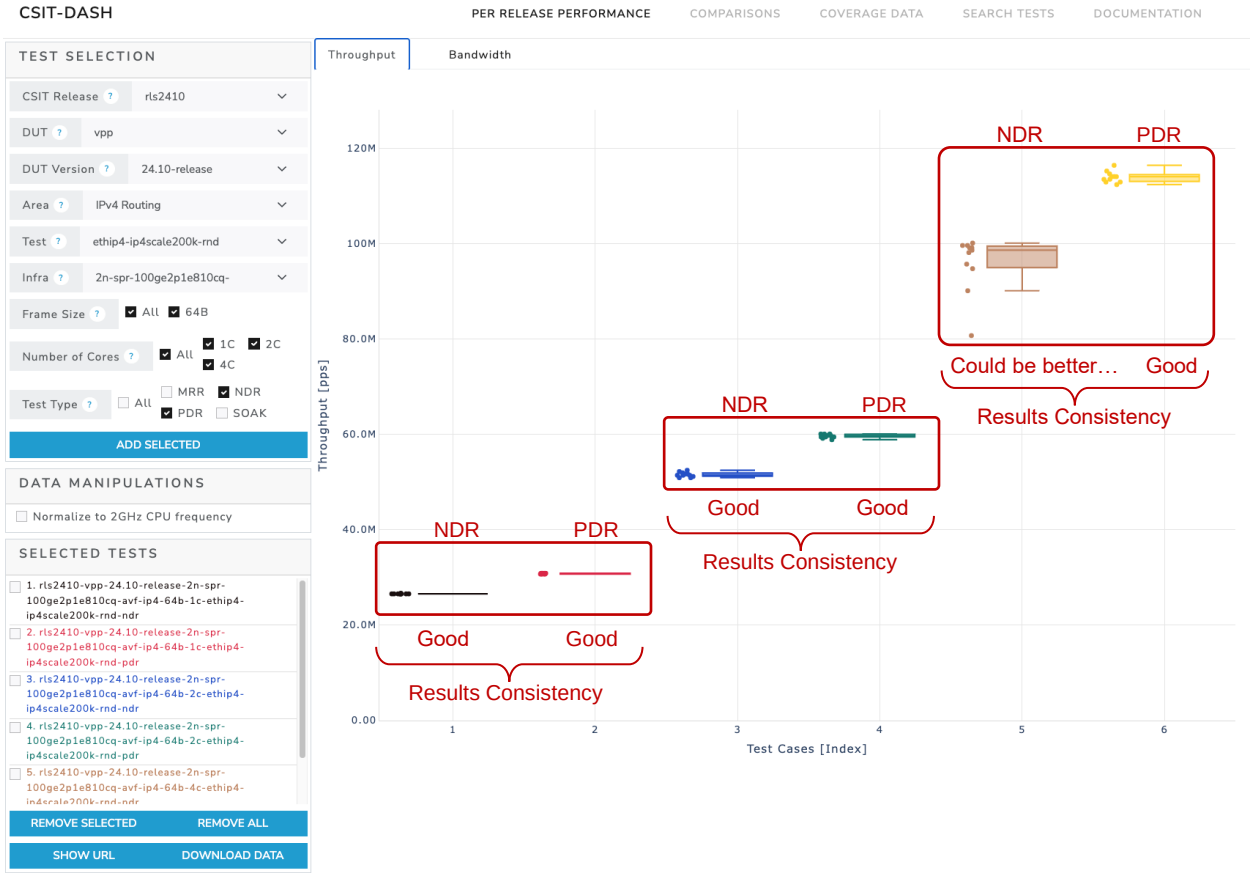
https://csit.fd.io/trending/#eNrlVcuOwjAM_JruZWWp8YYtFw7L8h8oplZW6sMkoavu15MiJIMEEgd6yiHPsT3xaKT40DvaempW2WKdFesMi7qMU_b18xmXgRmwA88OVJ4fCFnRUuX2CGbYQ80avvUOlAUK1XSKw1vTELbguhK60k218HeqVZ7CXWFBuBoFeUon8caRkYTIKVAgf8Py8FUSvHemJV__k2TEdgS3URuBlL2nCSPfoNdWi80l4t1CcjpC8juExCQcifM7EpNwJM7vSJ2EI_X8jtRJOFK_6MjF5qPrXXv5vePeV_0fBFebxl-vztUmlv4

2n-grc – IPv4 scale2m-rnd – 1, 2, 4 core



https://csit.fd.io/trending/#eNrlVcsKwjAQ_Jp6kYV2TezJg9r_kJiuttBHSOKjfr2pCKug4EF6ySHP2d3JDgNxvre0c9SsErU8k2CeV2GKVms52E5GwPYwdFqyNL0SGgyfZVyqaBtrhJqI2Ap9pBplF-NpzCcVg1hC7YroSvtWAy3Y7Hy5N8qM2KqgZHvfJygLCnOCKQMeXlvNB-fxcEHq1py9Y04l_TDuA7qMJTpdxo_mBf02WtePCL-LqWJSErzFykxDlfiBK7EOFyJE7hSxOFKMYErRRyuFD-6Uhazrrft4ycPe1f1F_C2Vo17Xt0BTXAltg

2n-spr – IPv4 scale200k-rnd – 1, 2, 4 core



https://csit.fd.io/report/#eNrtVstqwzAQ_Br3UrZlG7nupYek_o-iypvG1HHUlWJlv75KCKxNycGQNBcd9GJG7GiHAYW4Y3oP1L0W5aqoVgVWbZOmYrF8TAt3AY1WMHgPaJ7SjqkjGwiwh-AZtFKfhF7Ti1buG-ywhtYbeDYfoB1Q3BxPaQRnO0KlvoD7BvqGjzXw7VzjT0FBm30UNMmYlAOxgBN9QvObw4hzSbXwLZOVC0m6QJHCSMyFxwl9zXZLof0huZP6lrhLrRdlu2mhePAj9Nyxqj4x_ssnn32a6ZO_nU-Y8zTHJ7xXnjDn6Xo-3TBPJudpjk_mXnkyOU_X80nyVNYP_Y63p39fWf8CCTuxOg

SONiC - FD.io Integration:

Now Available!

https://github.com/sonic-net/sonic-platform-vpp

CSIT Resources

- **Technical Papers**
  - SPR 2Tbps IPsec (2023)
    - https://networkbuilders.intel.com/solutionslibrary/intel-avx-512-high-performance-ipsec-with-4th-gen-intel-xeon-scalable-processor-technology-guide
  - “Benchmarking Software Data Planes Intel® Xeon® Skylake vs. Broadwell” (2019)
    - https://www.lfnetworking.org/wp-content/uploads/sites/55/2019/03/benchmarking_sw_data_planes_skx_bdx_mar07_2019.pdf
  - “Benchmarking and Analysis of Software Data Planes” (2017)
    - https://fd.io/docs/whitepapers/performance_analysis_sw_data_planes_dec21_2017.pdf
- **Technology Demonstrator Video Clips**
  - “VPP: A Terabit Secure Network Data-plane” (Intel Xeon Icelake 07-APR-2021)
    - https://www.youtube.com/watch?v=ipQQmjzE_g0
  - “FD.io: A Universal Terabit Network Dataplane" (Intel Xeon Skylake, 11-JUL-2017)
    - https://www.youtube.com/watch?v=aLJ0XLeV3V4
- **FD.io Presentations**
  - https://wiki.fd.io/view/Presentations
- **Other FD.io Materials**
  - https://fd.io/
  - https://fd.io/news/whitepapers/

CSIT Resources

- **Project**
  - Wiki pages: https://wiki.fd.io/view/CSIT
  - Meetings: https://wiki.fd.io/view/CSIT/Meeting
  - Mailing list: csit-dev@lists.fd.io
- **CDash**
  - Dashboard: https://csit.fd.io
- **Source Code**
  - Git repo: https://git.fd.io/csit
  - Github mirror: https://github.com/FDio/csit
  - Gerrit reviews: https://gerrit.fd.io
- **Standalone libraries**
  - Speeding up binary search using shorter measurements: https://pypi.org/project/MLRsearch/
  - Locating changes in time series by grouping results: https://pypi.org/project/jumpavg/

PANTHEON.tech

SONiC ♥ FD.io: Better Together

January 22, 9 a.m. PT

Presenters:

Miroslav Mikluš
Chief Product Officer, PANTHEON.tech

Maciek Konstantynowicz
FD.io CSIT Technical Project Lead,
Distinguished Engineer, Cisco

intel partner
Gold