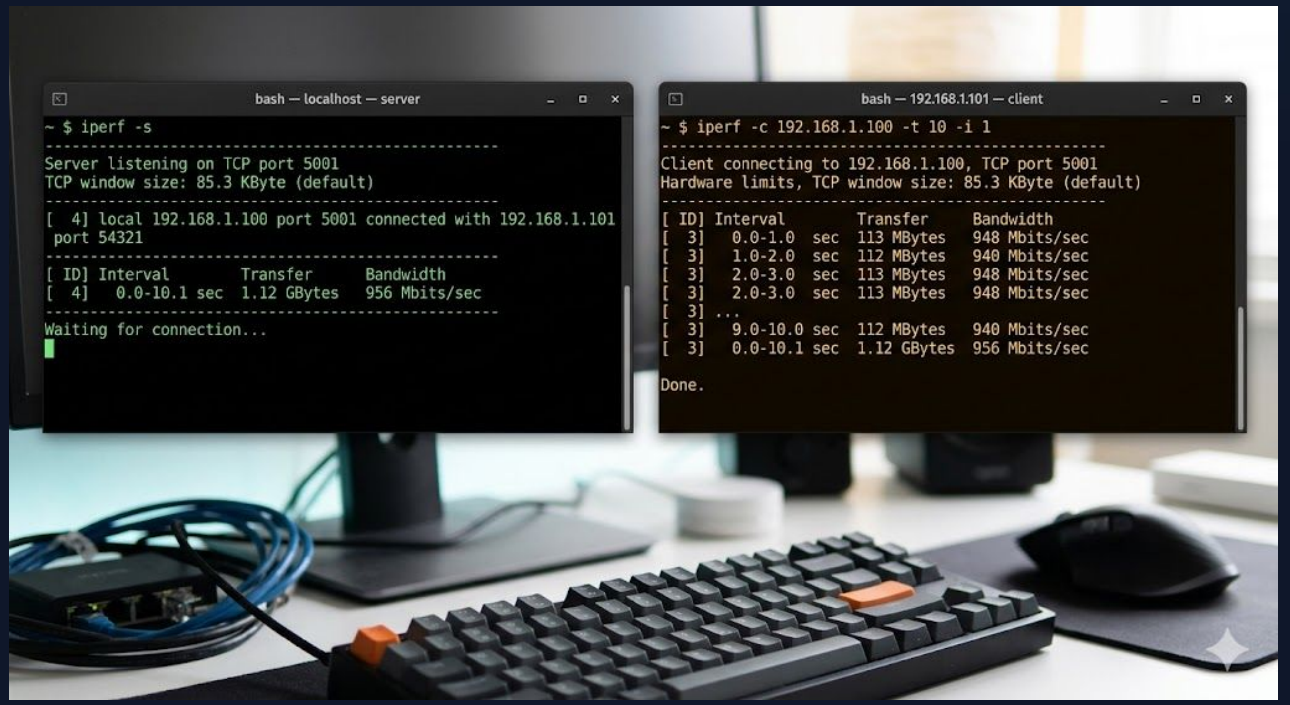
iperf, qperf, perftest comparison and integration in perfSONAR

iPerf: The Heritage of Testing

Originally developed by the National Laboratory for Applied Network Research (NLANR) under the DAST project in the early 2000s.

- **Evolution:** Evolved from the 'ttcp' tool for benchmarking.
- **Legacy:** Often referred to as iPerf2 to distinguish from iPerf3.



iPerf: Implementation Logic

C++ Core

Built using C++ providing object-oriented structure for managing various stream objects and timing mechanisms.

Multi-threaded

Uses a thread-per-stream model, allowing it to leverage multiple CPU cores efficiently for high-throughput tests.

Client-Server

Relies on a classic architecture where a server waits on a port and a client pushes traffic for duration or volume.

iPerf: Capabilities & Tuning

- **TCP Tuning:** Ability to set window sizes, MSS, and use Nagle's algorithm toggles.
- **UDP Benchmarking:** Measure jitter, packet loss, and achieve specific target bandwidth rates.
- **Multicast Support:** Testing group communication throughput and connectivity.
- **Bidirectional Testing:** Run simultaneous traffic in both directions to test full-duplex capacity.

iPerf3: The Modern Successor

Started in 2009 by ESnet and Lawrence Berkeley National Laboratory, iPerf3 was a complete "rewrite from scratch".

The goal was to create a smaller, simpler code base that is easier to maintain and extend for modern network needs.

iPerf3: Implementation Logic

Modular C

Written in pure C with a modular architecture centered around 'libiperf' for easy integration into other apps.

Single Process

Historically used a single-threaded event loop for control logic, reducing context switching overhead.

JSON Output

Native support for JSON formatting, making it the industry standard for automation and parsing results.

iPerf vs iPerf3

Feature	iPerf (v2)	iPerf3
Language	C++	C
Threading	Multi-threaded (per stream)	Single-threaded (*support for MT from v3.16)
API	Limited / Command-line only	libiperf available
Output	Human readable text	JSON & Text
Compatibility	Incompatible with iPerf3	Incompatible with iPerf2
Supported protocols	TCP / UDP	TCP / UDP / SCTP

qperf: the multi-protocol tool

qperf is a Linux utility used to measure bandwidth and latency over various transport protocols.

- **RDMA Support:** Unlike basic iPerf, it natively supports InfiniBand and RoCE protocols.
- **One-Stop Tool:** Can test TCP, UDP, SCTP, and RDMA with a single server instance.

http://network-switch.com/cdn/shop/articles/SEOon_understanding_infiniband_cable.jpg?v=1758696747

perftest: RDMA Specialist Toolset

Specific Binaries

A collection of tools: ib_write_bw, ib_read_lat, etc., each optimized for one test.

Verbs Layer

Bypasses the kernel stack by using native RDMA verbs for ultra-low latency testing.

Golden Standard

Considered the definitive benchmark for InfiniBand and high-speed RoCE fabrics.

Comparing RDMA Tools

qperf

Focus: Versatility and convenience.

- Single binary.
- Great for general socket testing.
- “stale” - last update 8 years ago

perftest

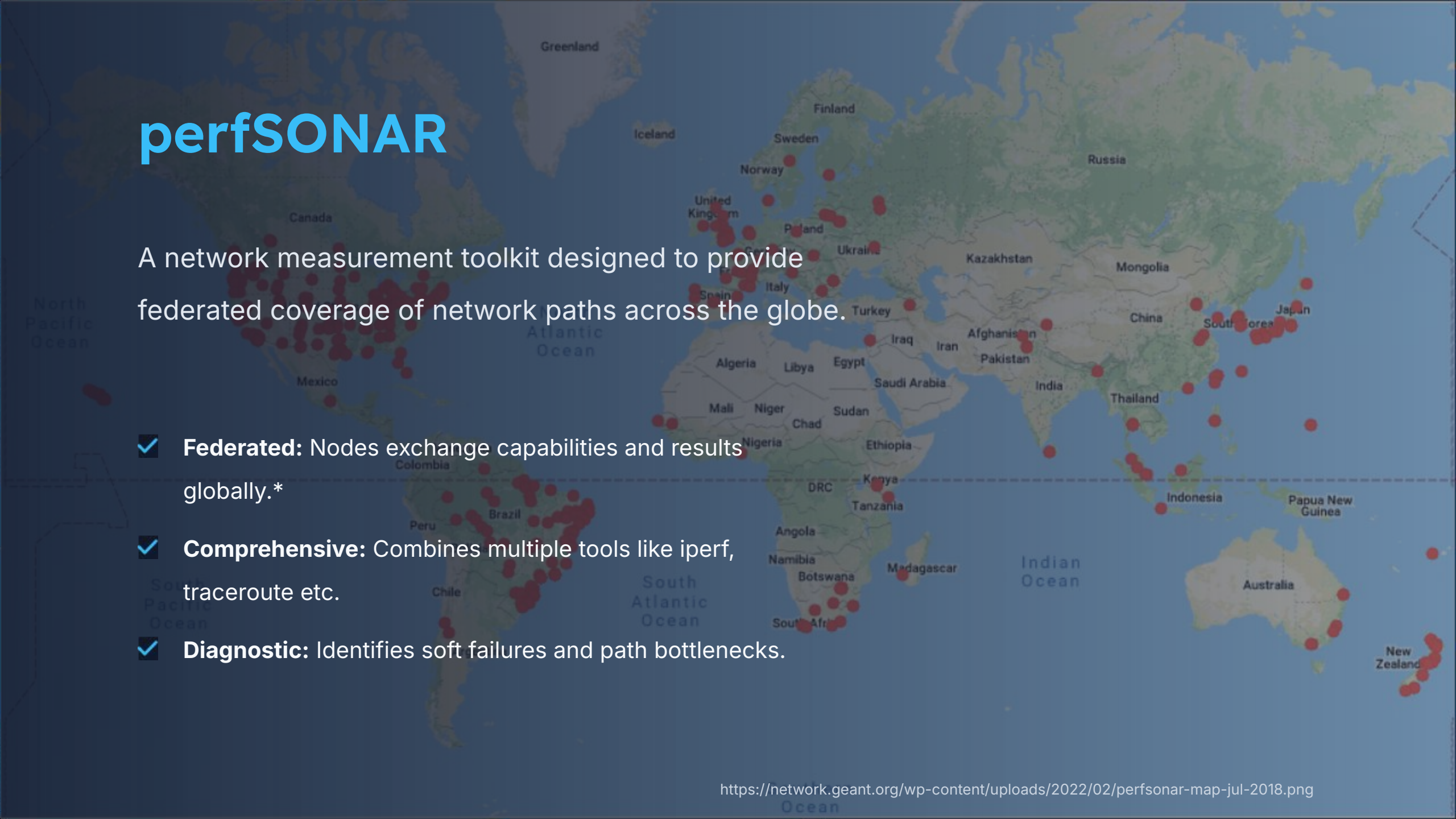
Focus: Raw performance and depth.

- More fine tuning options
- Supports modern features like GPUDirect.
- “fresh” - actively maintained

perfSONAR

A network measurement toolkit designed to provide federated coverage of network paths across the globe.

- ✅ **Federated:** Nodes exchange capabilities and results globally.*
- ✅ **Comprehensive:** Combines multiple tools like iperf, traceroute etc.
- ✅ **Diagnostic:** Identifies soft failures and path bottlenecks.



https://network.geant.org/wp-content/uploads/2022/02/perfsonar-map-jul-2018.png

perfSONAR: Core Architecture

pScheduler

The scheduler that handles resource reservation and executes tests to prevent interference.

Archiver

Stores data in a time-series format for historical trend analysis and long-term visualization.

Visualization

Supports result visualization “out of the box” via grafana

The bright side of tool integration...

Integration: iperf(3), qperf & perftest

- iperf & iperf3 already fully* integrated
- qperf partially* supported
  (should be available from 5.3)
- perftest partial* support still in development
  (should be done & available from 5.3)

Tool integration in general...

- ✓ Does the tool bring / offer something new that isn't already available?
- ✓ Do you need all the bells & whistles of the tool?
- ✓ Do you need "exotic" equipment for testing?

The documentation is far from perfect, but ... we can approach new tool integration "creatively":

"If we squint hard enough - the tools start to look very similar"

Questions?